THESIS FOR THE DEGREE OF LICENTIATE OF PHILOSOPHY

Model Adaptivity in Elasticity

DAVID HEINTZ

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2008

Model Adaptivity in Elasticity
DAVID HEINTZ

Thesis for Licentiate of Philosophy
NO 2008:37
ISSN 1652-9715

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31–772 1000

Prepared with LaTeX
Printed in Göteborg, Sweden 2008

Model Adaptivity in Elasticity

DAVID HEINTZ

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

We consider model adaptivity in elasticity for dimensionally reduced forms, and shall treat different conceptual approaches. The basic idea, however, is to adaptively refine, not only the computational mesh, but also the underlying mathematical formulation. The intention is that the algorithm, provided with an hierarchy of models, should have the local complexity tailor-made for each problem, and thus become more efficient.

Reduced forms of the 3D-elasticity theory are typically obtained via simplified deformation relations. A typical example is the Bernoulli and Timoshenko beam theories. We discuss Navier's equations of linear elasticity in a thin domain setting, and construct a model hierarchy based on increasingly higher polynomials approximations through the thickness of the domain, coupled with a Galerkin approach.

We suggest a finite element method for an extension of the Kirchhoff-Love plate equation, which includes the effects of membrane stresses. The stresses are obtained from the solution of a plane-stress problem, and plays the role of underlying model. Since the modeling error actually is a discretization error, it is not the same as the construction of a model hierarchy.

The aim has been to establish efficient solution procedures alongside accurate error control. To succeed in this ambition, *a posteriori* error estimate are derived, which separate the discretization and modeling errors. Frameworks for adaptive algorithms are suggested, and accompanied by numerical results to exemplify their behavior.

Keywords: *dimension reduction, model error, model adaptivity, a posteriori error*

List of Appended Papers

The licentiate thesis consists of an introductory text to subjects and methods, and new contributions based on the work in the following papers:

A D. Heintz, **Model Adaptivity for Elasticity on Thin Domains**, in Conference Proceedings, *WCCM8/ECCOMAS 2008, Venice, Italy, 30 June–4 July 2008*; Preprint 2008:34 (Blå serien), ISSN 1652-9715, Chalmers University of Technology and University of Gothenburg

B P. Hansbo, D. Heintz and M. G. Larson, **An Adaptive Finite Element Method for Second Order Plate Theory**, Preprint 2008:36 (Blå serien), ISSN 1652-9715, Chalmers University of Technology and University of Gothenburg

Both papers are available at `www.math.chalmers.se/Math/Research/Preprints/`.

Paper B was prepared in collaboration with co-authors. The author of this thesis planned for and carried out the numerical simulations, and wrote the corresponding parts of the article.

Acknowledgments

The work in this thesis was funded by the Swedish Research Council, as part of the ongoing project 621-2004-5147, by the name *Model Adaptivity in Computational Mechanics*.

Foremost, my gratitude goes to my supervisor, Professor Peter Hansbo: without his support and encouragement, this thesis would not rest in your palms today. I also wish to thank my co-author Mats G. Larson.

I am much grateful to Thomas Ericsson. His expertise has guided me on numerous occasions (tends to `Inf`), and he is always in the mood for friendly conversations (as long as you "Don't mention the element!").

Niklas Ericsson has given me appreciated feedback—he taught me when I was an undergraduate, and continues doing so today.

I would like to thank the members of the Computational Mathematics group for providing a pleasant working environment: especially Christoffer Cromvik, Karin Kraft and Göran Starius.

Last, but not least, I wish to address my colleagues at the former division of Computational Technology: Thank you.

David Heintz
Göteborg, November 2008

Contents

Part I

Introduction

Chapter 1

Model Adaptivity

We consider model adaptivity in computational mechanics, the focus being on hierarchical modeling within linear elasticity. However, let us begin by introducing the concept in general terms, and hence assume that we are studying a physical system, say, various potential fields (gravitational, electrostatic, ...), stationary heat flow, or the displacements of a loaded elastic body. In order to answer the variety of questions which may arise, we try to describe the reality in terms of mathematical models, often in the guise of differential equations.

Since few physical problems are amenable to analytical methods, we need other means to solve the mathematical model. In order to do so by using a computer, we must discretize the continuous problem, and we choose to employ the finite element method (FEM), which has been closely intertwined with the solid mechanics community since the 1950s. We shall return to a discusson of FEM, particularly in the linear elasticity setting, in Section 1.2.

The discretized version constitutes a simplified problem, whose solution—which in our case is obtained via FEM—converges towards the continuous one, or at least becomes accurate enough, should the computational mesh just be sufficiently resolved. The meaning of "accurate enough" depends on the application at hand, and has turned into an important research topic: solving the discretized problem is associated with a computational cost, in terms of time and memory, and hence doing so without careful consideration could prove intractable. This was the advent of adaptive procedures in FEM, which basically are techniques for optimizing the underlying mesh. We want to distribute the degrees of freedom in such a way as to get high accuracy with respect to the computational cost. The *a posteriori* error estimate is the cornerstone of such algorithms, providing local indicators to govern the design of the mesh, and, inherently, bringing error control.

However, the physical system may usually be described by several mathematical models of increasingly higher complexity (referring to how accurate they are)—a typical example being the Bernoulli and Timoshenko beam theories, the latter being a refined model, in the sense of including the effects of shear deformations. Thus their exact solutions will deviate, and we call the difference the *modeling error* e_{M}. If we use the Bernoulli equation to calculate the deflection of a beam, we define the total error e with respect to the Timoshenko model as

$$e = u - u^h = \underbrace{(u - \hat{u})}_{\textit{modeling error}} + \underbrace{(\hat{u} - u^h)}_{\textit{discretization error}} = e_{\mathrm{M}} + e_{\mathrm{D}},$$

where u^h is the (approximate) FE-solution of the Bernoulli model, whose exact solution then is denoted $\hat{u}$. The conventional a posteriori error estimate measures the *discretization error* e_{D} (caused by not completely resolving the mesh). A schematic represention of the different error sources is shown in Figure 1. Note that even though $e_{\mathrm{D}} \to 0$, implying $u^h \to \hat{u}$, there still remains a modeling error, which is not decreased by refining the mesh.

Suppose that we have a set of models to choose from, would it not be desirable also to adaptively refine the model? By starting at a simple model, and increase its complexity only where it is necessary, we ought to get more efficient algorithms. Developing such techniques, in various settings, has become a new subject of research, which has intensified during the last decade.

We have briefly introduced the concept of model error and model adaptivity, and will now continue to discuss it, but concentrate on its application to elasticity.

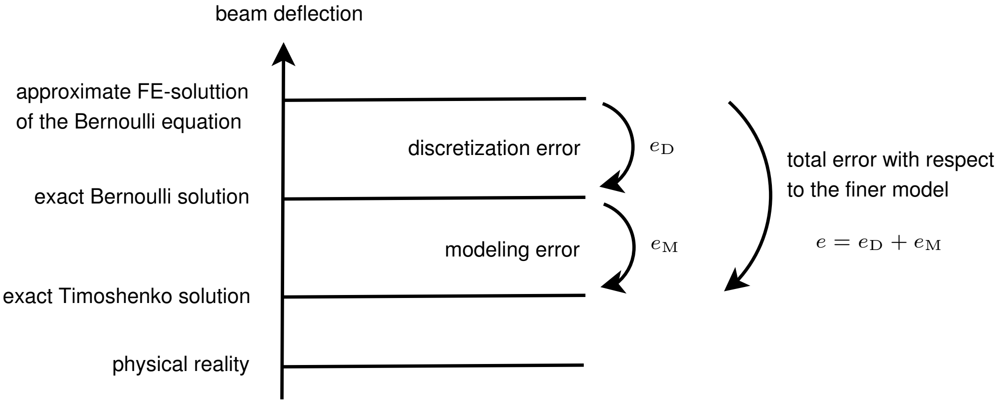

beam deflection
approximate FE-soluttion of the Bernoulli equation
discretization error
e_{D}
total error with respect to the finer model
exact Bernoulli solution
modeling error
e_{M}
$e = e_{\mathrm{D}} + e_{\mathrm{M}}$
exact Timoshenko solution
physical reality

FIGURE 1: *Illustrating different error sources*

1.1 Hierarchical Modeling in Elasticity

Boundary value problems (BVPs) encountered in engineering applications are often posed on *thin domains*, e.g., beams, plates or shells. The term *thin* then relates to the physical domain being much smaller in one direction. As an example consider the beam, which is dominated by its extension in the axial direction. This may justify making simplifying assumptions on the exact solution, effectively replacing the original problem with a lower-dimensional one. This is known as *dimension reduction*.

Lower-dimensional methods—as compared to the full elliptic 3D-BVPs—are more susceptible to analytical techniques: sometimes it is even possible to derive exact solutions by means of Fourier series. Following the evolution of computers, these methods have gained more momentum, and are commonly used in today's software. Some reasons for their popularity are, firstly, the possible computational savings, and secondly, the fact that h-methods in 3D FE-discretizations often require excessive mesh refinement, and could suffer from bad conditioning (especially on thin domains), or exhibit so-called *locking* phenomena.

Lower-dimensional methods are usually advocated by being *asymptotically exact*, i.e., the difference from the higher-dimensional model vanishes as the thickness (the extension in the thin direction) tends to zero. Nevertheless, in practice the thickness is prescribed, and hence we inevitably incur a corresponding modeling error. But what should we do if our model turns out not to be accurate enough?

In the late 1980s and early 1990s the idea arose to embed classical models into a hierarchy of lower order models, the term *hierarchic models* being introduced by Szabo and Sahrmann, and the method further developed by Actis, Babuška and co-workers (see Schwab [26] for references). These hierarchies are typically constructed by imposing restrictions on the displacements of the 3D-formulations, say, by prescribing a polynomial expansion in the transverse direction. Consecutive models are then readily obtained by increasing the degree of the approximation. Once we have an available hierarchy, it could be used to solve the original formulation adaptively. The underlying model would not be uniform over the domain: the optimal model is tailor-made for each particular problem (illustrated in Figure 2).

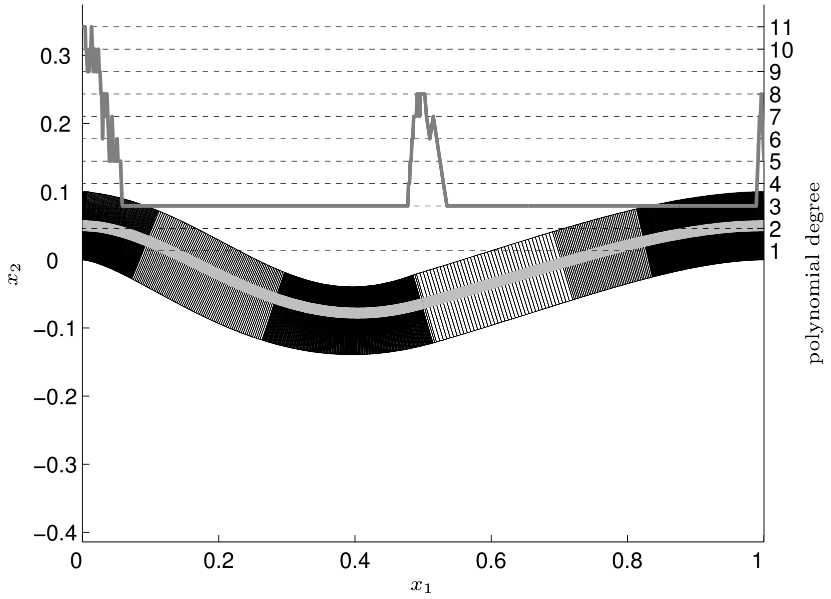


FIGURE 2: *FE-solution for fixed loaded beam by means of model adaptivity; the solution of the corresponding Bernoulli beam equation is seen in solid gray* (**Paper I**)

The goal of adaptive algorithms is to reduce the total error within a prescribed tolerance. The notion of the total error being divided into two distinct parts, namely the discretization and modeling errors, means that we, in each stage of our computation, need to determine whether to refine the model or enrich the finite dimensional subspace. The decision ought to bring forward an equidistribution of the total error, which in turn requires any a posteriori error analysis to include two estimates, one for each error type. Our primary concern thus becomes to decompose the error into these distinct parts.

The early approaches assumed the discretization error to be negligible, so that the modeling error, more or less, constituted the total error itself. The error estimates were refined to measure the error in global norms (energy norm), and in recent years, extended to encompass upper and lower bounds in linear functionals of the solution. We will return to discuss such techniques in Sections 1.2.2 and 1.2.2.

1.2 The Finite Element Method

FEM is a numerical technique for solving general partial differential equations (PDEs) over complex geometries. Instead of approximating the differential equation directly, which a traditional finite difference method (FDM) would do, it uses integrated forms, corresponding to alternative descriptions of the physical problem.

FEM is closely related to global balance laws, e.g., *minimization of the potential energy* and the balance of *virtual work*. To exemplify the latter, we consider the stationary 1D heat conduction in a bar, which is described by the following model problem (see [11, Section 6.2.1] for details)

$$-(ku')' = f, \quad 0 < x < 1, \tag{1}$$

where $k > 0$ is the heat conductivity, $-ku'$ is the heat flux, and f is an external heat source. Assuming homogeneous Dirichlet boundary conditions, $u(0) = u(1) = 0$, we multiply (1) by a function v, such that $v(0) = v(1) = 0$, and integrate over the domain

$$\int_0^1 -(ku')'v \,\mathrm{d}x = \int_0^1 fv \,\mathrm{d}x, \tag{2}$$

which, following integration by parts, leads to the *weak* form

$$\int_0^1 ku'v' \,\mathrm{d}x = \int_0^1 fv \,\mathrm{d}x. \tag{3}$$

We call the functions u and v the *trial* and *test functions*, and say that we have tested (1) with v. If one tests by a sufficiently large number of functions v, we expect the integrated form (2) to actually satisfy (1) pointwise (the virtual work principle becomes equivalent with the energy conservation law and Fourier's law underlying (1)). Note that (3) imposes fewer restrictions on u than a *classical* solution of (1) does. The weak solution, e.g, is not required to be twice differentiable, the integrals should just exist. This is an important point: it is easier to generate approximate solutions of less regularity (consequently FEM produces approximate solutions of (3) rather than (1)).

Galerkin's method is based on seeking an approximate solution in a finite-dimensional space, spanned by a set of basis functions, which are easy to differentiate and integrate. This could be piecewise linear continuous functions $\varphi_i = \varphi_i(x)$ with local support (the basic form of a FEM). If the FE-solution is written

$$u_h(x) = \sum_{i=1}^{n} u_i \varphi_i(x), \tag{4}$$

then Galerkin suggested (3) to hold for all test functions of the same form. To ensure this, the equation is tested against each φ_i separately (then it will be satisfied for an arbitrary linear combination of the basis functions). We get a linear system of equations to be solved for the coefficients u_i of (4) using a computer.

Galerkin's method can be described as a *projection method*, where the solution is projected onto a subspace, spanned by a set of basis functions. These functions are not necessarily linear, but can be polynomials of higher order, or even trigonometric functions (so-called *spectral methods*). We can make a discontinuous *ansatz* for the FE-solution as well. The method can also be formulated using different functions in the test and trial spaces, which is known as a *Petrov-Galerkin method*.

FEM has a solid mathematical foundation, see, e.g., [12] and [7], which is a strength, since it provides tools for deriving analytical error estimates, that, in turn, allow us to improve on our approximate solutions. FEM has typically been the natural choice for applications in solid mechanics. FDM, somewhat easier to implement than FEM (at least for simpler model problems on rectangular domains), is more common within the field of computational fluid dynamics. Here one otherwise tends to employ lower-order finite volume methods.

We now turn to the equations of linear elasticity, and introduce the corresponding strong, weak and FE-formulations. Solving Navier's equations—or, in some sense, avoiding it!—has been an integral part of this thesis. In **Paper I** we treat a reduced form on thin domains, whereas **Paper II** deals with second order effects for plate theory. The terminology of finite elements is closely intertwined with elasticity, e.g., the linear system of equations arising in FEM, $\mathbf{S}\mathbf{u} = \mathbf{f}$, usually calls $\mathbf{S}$ and $\mathbf{f}$ the *stiffness matrix* and *load vector*, respectively, regardless of the actual application. How to assemble FE-matrices in an efficient manner is discussed in Chapter 2.

1.2.1 Navier's Equations of Elasticity

Consider a convex polygonal domain $\Omega \subset \mathbb{R}^2$, representing a deformable medium subjected to external loads. These include body forces $\boldsymbol{f}$ and surface tractions $\boldsymbol{g}$, causing deformations of the material, which we describe by the following model problem: Find the displacement field $\boldsymbol{u} = (u_1, u_2)$ and the symmetric stress tensor $\boldsymbol{\sigma} = (\sigma_{ij})_{i,j=1}^2$, such that

$$\begin{aligned} \boldsymbol{\sigma}(\boldsymbol{u}) &= \lambda \operatorname{div}(\boldsymbol{u})\,\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) && \text{in } \Omega && (5) \\ -\operatorname{div}(\boldsymbol{\sigma}) &= \boldsymbol{f} && \text{in } \Omega && (6) \\ \boldsymbol{u} &= \boldsymbol{0} && \text{on } \partial\Omega_{\mathrm{D}} \\ \boldsymbol{\sigma}\cdot\boldsymbol{n} &= \boldsymbol{g} && \text{on } \partial\Omega_{\mathrm{N}} \end{aligned}$$

where $\partial\Omega = \partial\Omega_{\mathrm{D}} \cup \partial\Omega_{\mathrm{N}}$ is a partitioned boundary of Ω. Let the Lamé coefficients

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}, \tag{7}$$

with E and ν being Young's modulus and Poisson's ratio, respectively. Furthermore, $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{n}$ denotes the outward unit normal to $\partial\Omega_{\mathrm{N}}$, and the strain tensor is

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \tfrac{1}{2}\big(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^{\mathrm{T}}\big).$$

The vector-valued tensor divergence is

$$\operatorname{div}(\boldsymbol{\sigma}) = \left(\sum_{j=1}^{2} \frac{\partial\sigma_{ij}}{\partial x_j}\right)_{i=1}^{2},$$

representing the internal forces of the equilibrium equation (6). This formulation assumes a constitutive relation corresponding to linear isotropic elasticity (the material properties are the same in all directions), with stresses and strains related by

$$\boldsymbol{\sigma}_{\mathrm{v}} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{12} \end{bmatrix} = \boldsymbol{D}(\lambda,\mu)\boldsymbol{\varepsilon}_{\mathrm{v}},$$

referred to as *Hooke's generalized law*. Should the material be homogeneous, $\boldsymbol{D}$ becomes independent of position.

In order to pose a weak formulation we introduce the function space

$$V = \left\{ v : \ v \in H^2(\Omega), \ v|_{\partial\Omega_{\mathrm{D}}} = 0 \right\},$$

and state: Find $\boldsymbol{u} \in V \times V$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V \times V, \tag{8}$$

where the bilinear form

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{x} \tag{9}$$

is the integrated tensor contraction

$$\boldsymbol{\sigma} : \boldsymbol{\varepsilon} \overset{\mathrm{def}}{=} \sum_{i,j=1}^{2} \sigma_{ij} \varepsilon_{ij},$$

and the linear functional

$$L(\boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) + (\boldsymbol{g}, \boldsymbol{v})_{\partial\Omega_{\mathrm{N}}} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_{\mathrm{N}}} \boldsymbol{g} \cdot \boldsymbol{v} \, \mathrm{d}s. \tag{10}$$

We usually interpret (8) as a balance between the internal (9) and external (10) "virtual work" (with the test functions $\boldsymbol{v}$ being "virtual displacements").

For the numerical approximation of (8), we shall need a discrete counterpart, and as such establish a finite element method. To simplify its formulation we define the kinematic relation

$$\boldsymbol{\varepsilon}_{\mathrm{V}}(\boldsymbol{u}) = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \tilde{\boldsymbol{\nabla}} \boldsymbol{u},$$

and specify the constitutive matrix

$$\boldsymbol{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix},$$

for the purpose of rewriting the bilinear form as

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\varepsilon}_{\mathrm{V}}(\boldsymbol{u})^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\varepsilon}_{\mathrm{V}}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{x},$$

which facilitates implementation. We introduce a partition $\mathcal{T}_h$ of Ω, dividing the domain into N_{el} elements. More precisely, we let $\mathcal{T}_h = \{K\}$ be a set of triangles K, such that

$$\Omega = \bigcup_{K \in \mathcal{T}_h} K,$$

with the element vertices referred to as the nodes $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N_{\mathrm{no}}$, of the triangulation. The intersection of any two triangles is either empty, a node, or a common edge, and no node lies in the interior of an edge (there are no *hanging nodes*). The function

$$h_K = \mathrm{diam}(K) = \max_{\boldsymbol{y}_1, \boldsymbol{y}_2 \in K} (\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2), \quad \forall K \in \mathcal{T}_h,$$

represents the local mesh size. Moreover, $\mathcal{E}_h = \{E\}$ denotes the set of element edges, which we split into two disjoint subsets, $\mathcal{E}_h = \mathcal{E}_I \cup \mathcal{E}_B$, namely the sets of interior and boundary edges, respectively. The partition is associated with a function space

$$V_h = \{v \in \mathcal{C}(\Omega) : \ v \text{ is linear on } K \text{ for each } K \in \mathcal{T}_h, \ v|_{\partial\Omega_{\mathrm{D}}} = 0\}, \tag{11}$$

consisting of continuous, piecewise linear functions, that vanish on the Dirichlet boundary. A function $v \in V_h$ is uniquely determined by its values at $\boldsymbol{x}_i$, together with the set of shape functions

$$\{\varphi_j\}_{j=1}^{N_{\mathrm{no}}} \subset V_h, \quad \varphi_j(\boldsymbol{x}_i) := \delta_j(\boldsymbol{x}_i),$$

which constitute a nodal basis for (11). It then follows that any $v \in V_h$ can be expressed as a linear combination

$$v = \sum_{j=1}^{N_{\mathrm{no}}} v_j \varphi_j(\boldsymbol{x}), \tag{12}$$

where $v_j = v(\boldsymbol{x}_j)$ represents the j:th nodal value of v. We make an ansatz for a FE-solution of this type (12), and hence our FE-formulation of (8) becomes: Find $\boldsymbol{u}_h \in V_h \times V_h$ such that

$$a(\boldsymbol{u}_h, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_h \times V_h, \tag{13}$$

whose solution usually is written on the standard form

$$\boldsymbol{u}_h = \begin{bmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \ldots \\ 0 & \varphi_1 & 0 & \varphi_2 & \ldots \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ \vdots \end{bmatrix} = \boldsymbol{\varphi}\mathbf{u},$$

associating odd and even elements of $\mathbf{u}$ with displacements in x_1 and x_2, respectively. Since testing against all $\boldsymbol{v} \in V_h \times V_h$ reduces to testing against $\{\varphi_j\}_{j=1}^{N_{\mathrm{no}}}$, and $\boldsymbol{\varepsilon}_{\mathrm{v}}(\boldsymbol{u}_h) = \tilde{\boldsymbol{\nabla}}\boldsymbol{\varphi}\mathbf{u} = \boldsymbol{B}\mathbf{u}$, (13) corresponds to solving

$$\int_\Omega \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} \, \mathrm{d}\boldsymbol{x} \, \mathbf{u} = \int_\Omega \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{f} \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_{\mathrm{N}}} \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{g} \, \mathrm{d}s, \tag{14}$$

i.e., the matrix problem $\mathbf{Su} = \mathbf{f}$, making (14) a suitable starting point for FE-implementation.

1.2.2 Adaptivity

The goal in FE-analysis, from a practical point of view, is to utilize the available computing resources in an optimal way, usually adhering to either of two principles:

- obtain the prescribed accuracy TOL at minimal amount of work;
- obtain the best accuracy for a prescribed amount of work.

In order to achieve this aim, the traditional approach is by means of automatic mesh adaption, based on local error indicators. These are functions of the FE-solution, and presumably measure the local roughness of the continuous solution. The overall process involves some distinct steps:

i) a choice of norm in which the error is defined (different problems may call for different norms);

ii) a posteriori error estimates with respect to the chosen norm, in terms of known quantities, i.e., the data and the FE-solution (which provide information about the continuous problem);

iii) local error indicators extracted from the (global) a posteriori error estimates;

iv) a strategy for changing the mesh characteristics (the mesh size and/or the polynomial interpolation) to reduce the error in an (nearly) optimal way.

In the following we deal with questions i)-ii) in some detail. The discussion will not be comprehensive, but focuses on exemplifying the techniques employed in this thesis: 1) residual-based energy norm control (used in **Paper I**); and 2) goal-oriented adaptivity (**Paper II**).

For the sake of simplicity, we let Poisson's equation serve as model problem, representing the linear elliptic PDE. Let us adopt parts of the notation from Section 1.2.1, and hence pose the continuous problem: Find u such that

$$\begin{aligned} \mathcal{A}u := -\nabla \cdot (k\nabla u) &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega_{\mathrm{D}}, \\ n \cdot k\nabla u &= g, \quad \text{on } \partial\Omega_{\mathrm{N}}, \end{aligned} \tag{15}$$

where we assume the coefficient $k = k(\boldsymbol{x})$ to be smooth, whereas $f \in L_2(\Omega)$ and $g \in L_2(\partial\Omega_{\mathrm{N}})$ are data to the problem.

Residual-Based Error Estimates

The main idea—as opposed to solving local problems or using stress projections—is to substitute the FE-solution into the PDE: since u_h is an approximation, it does not satisfy (15) exactly, and this hopefully provides useful information about the error $e = u - u_h$.

The weak form of the model problem is: Find $u \in V$ such that

$$a(u, v) = L(v), \quad \forall v \in V, \tag{16}$$

where the bilinear form $a(\cdot,\cdot)$ and the linear functional $L(\cdot)$ are

$$a(u, v) = \int_{\Omega} k\nabla u \cdot \nabla v \,\mathrm{d}\boldsymbol{x}, \quad L(v) = \int_{\Omega} fv \,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_{\mathrm{N}}} gv \,\mathrm{d}s.$$

The corresponding FE-formulation becomes: Find $u_h \in V_h$ such that

$$a(u_h, v) = L(v), \quad \forall v \in V_h, \tag{17}$$

and by subtracting (17) from (16), we recognize the Galerkin orthogonality

$$a(e,v) = 0, \quad \forall v \in V_h, \tag{18}$$

stating that the error is orthogonal to the subspace $V_h \subset V$.

When $a(\cdot,\cdot)$ is symmetric, coercive and bounded (with respect to the function space V), we may define a norm

$$\|v\|_a := a(v,v)^{1/2}, \quad \forall v \in V,$$

which is referred to as the *energy norm* (generic for the problem). The symmetry and positive-definiteness of $a(\cdot,\cdot)$ is typical for problems encountered in solid mechanics.

We now outline a means for a posteriori error estimation: First note that

$$\|e\|_a^2 = a(e,e) = a(u-u_h,e) = a(u,e) - a(u_h,e) = L(e) - a(u_h,e), \tag{19}$$

via (16) for $v = e \in V$. Take $\pi_h : V \to V_h$ to be the standard nodal interpolation operator (or the L_2-projection), and it follows from (17), using $\pi_h e \in V_h$,

$$\|e\|_a^2 = L(e - \pi_h e) - a(u_h, e - \pi_h e).$$

Then, by elementwise integration by parts of the second RHS-term,

$$\begin{aligned}\|e\|_a^2 = &\sum_{K\in\mathcal{T}_h} \int_K (f + \nabla\cdot k\nabla u_h)(e - \pi_h e)\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_\mathrm{N}} g(e-\pi_h e)\,\mathrm{d}s \\ &- \sum_{K\in\mathcal{T}_h} \int_{\partial K} \boldsymbol{n}_K \cdot k\nabla u_h (e - \pi_h e)\,\mathrm{d}s,\end{aligned}$$

where $\boldsymbol{n}_K$ denotes the outward unit normal to the boundary ∂K of element K. Since each $E \in \mathcal{E}_I$ is common to two elements, we can regroup terms, and thus get

$$\begin{aligned}\|e\|_a^2 = &\sum_{K\in\mathcal{T}_h} \int_K (f + \nabla\cdot k\nabla u_h)(e - \pi_h e)\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_\mathrm{N}} (g - \boldsymbol{n}\cdot k\nabla u_h)(e-\pi_h e)\,\mathrm{d}s \\ &+ \sum_{E\in\mathcal{E}_I} \int_E [\boldsymbol{n}_E \cdot k\nabla u_h](e - \pi_h e)\,\mathrm{d}s,\end{aligned} \tag{20}$$

where we define

$$[\boldsymbol{n}_E \cdot k\nabla u_h](\boldsymbol{x}) := \lim_{\epsilon\to 0^+} \big((\boldsymbol{n}_E\cdot k\nabla u_h)(\boldsymbol{x} + \epsilon\boldsymbol{n}_E) + (\boldsymbol{n}_E\cdot k\nabla u_h)(\boldsymbol{x} - \epsilon\boldsymbol{n}_E)\big)$$

to be the jump in $\boldsymbol{n}_E \cdot k\nabla u_h$ across the element edge E with unit normal $\boldsymbol{n}_E$. This, eventually, leads to a representation of the error of the form

$$\|e\|_a^2 \le \sum_{K\in\mathcal{T}_h} \omega_K \rho_K, \tag{21}$$

by applying Cauchy's inequality elementwise to (20), and using suitable interpolation error estimates. The weights ω_K relate to the interpolation error, whereas ρ_K represents residuals (with respect to either the interior or the boundary of the domain) of the FE-solution.

(21) is an a posteriori error estimate, i.e., an estimate in terms of the computed solution and data, which constitutes an upper bound of the error. However, being based on Cauchy's inequality, it may not be sharp. A viable alternative comes by solving an auxiliary problem on a refined mesh. The enhanced discrete solution $\tilde{u}$ is then interpolated onto the primal mesh, and the difference provides information about the error. This technique was used in **Paper I** for Navier's equations of elasticity, and it is also shown how local error indicators can be extracted.

Approximating u on a refined mesh clearly is expensive—hence we comment on the option of solving local problems (see Verfürth [32, Chapter 1.3] for details). It is usually cheaper to solve a set of smaller problems, but possibly less straightforward to implement. Dirichlet problems can be constructed by raising the degree of the polynomial approximation. Let us exemplify by going from linear to quadratic: The boundary values are then inherited from u_h, providing an extra degree of freedom on each element. Alas, to obtain reliable estimates for $\tilde{u}$, one often needs to consider patches of elements (say all elements including a node, to yield additional degrees of freedom), which requires extra work. For Neumann problems, on the other hand, we have to equilibrate an approximation of the flux $k\nabla u$, to keep the estimates from degrading (since there are no Dirichlet boundary conditions). The use of local Neumann problems for error estimation in hierarchical models have been pursued by Stein and Ohnimus in a series of papers [27, 28, 29].

Goal-Oriented Adaptivity

The traditional approach to adaptivity is to estimate the error in energy norm or the global L_2-norm. However, more often than not, we are rather interested in controlling the error of local physical quantities, like the maximum deflection of a plate subjected to external loads.

In order to assess this error we shall use duality techniques, which essentially means that we multiply the residuals by certain weights, namely the solution of a so-called *dual problem* (hence the approach is known as the *dual-weighted residual* method—or the *DWR*-method for short). To show this we begin by introducing the continuous dual (adjoint) problem:

$$\mathcal{A}^{\mathrm{T}} z = j, \tag{22}$$

where the dual operator $\mathcal{A}^{\mathrm{T}}$ is defined by

$$(\mathcal{A}^{\mathrm{T}} z, \psi) = (z, \mathcal{A}\psi). \tag{23}$$

We can show that

$$(e, j) \overset{(22)}{=} (e, \mathcal{A}^{\mathrm{T}} z) \overset{(23)}{=} (\mathcal{A}e, z) = (f - \mathcal{A}u_h, z) \overset{(18)}{=} (f - \mathcal{A}u_h, z - \pi_h z), \tag{24}$$

using the Galerkin orthogonality for $v = \pi_h z \in V_h$. Now, as compared to (19), we are free to choose the data j according to the quantity we wish to control adaptively—if j is taken as an approximate Dirac delta function, the LHS of (24) reduces to the error in the corresponding point (cf. **Paper II**).

Example. In order to concretize (24), we return to Poisson's equation, and test (15) against a function z:

$$\begin{aligned}\int_\Omega -\nabla\cdot(k\nabla u)z\,\mathrm{d}\boldsymbol{x} &= \int_\Omega k\nabla u\cdot\nabla z\,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega_\mathrm{N}} \boldsymbol{n}\cdot(k\nabla u)z\,\mathrm{d}s\\ &= \int_\Omega -\nabla\cdot(k\nabla z)u\,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega_\mathrm{N}} (\boldsymbol{n}\cdot(k\nabla u)z - \boldsymbol{n}\cdot(k\nabla z)u)\,\mathrm{d}s,\end{aligned}$$

using integration by parts twice. Taking the Neumann boundary conditions into account (u is prescribed on the Dirichlet boundary), this indicates that the dual problem should be

$$\begin{aligned}\mathcal{A}^\mathrm{T}z = -\nabla\cdot(k\nabla z) = j_1, &\quad \text{in } \Omega,\\ z = 0, &\quad \text{on } \partial\Omega_\mathrm{D},\\ \boldsymbol{n}\cdot k\nabla z = j_2, &\quad \text{on } \partial\Omega_\mathrm{N},\end{aligned}$$

i.e., $\mathcal{A} = \mathcal{A}^\mathrm{T}$ is a self-adjoint operator. We first note that

$$(e, \mathcal{A}^\mathrm{T}z) = \int_\Omega -\nabla\cdot(k\nabla z)e\,\mathrm{d}\boldsymbol{x} = \int_\Omega k\nabla z\cdot\nabla e\,\mathrm{d}\boldsymbol{x} - \int_{\partial\Omega_\mathrm{N}} j_2 e\,\mathrm{d}s,$$

and then, via (24) for data j_1 and j_2,

$$\begin{aligned}(e, j_1) + (e, j_2)_{\partial\Omega_\mathrm{N}} &= \int_\Omega k\nabla z\cdot\nabla e\,\mathrm{d}\boldsymbol{x} = a(e,z) \overset{(18)}{=} a(e,z) - a(e,\pi_h z)\\ &= a(u-u_h, z-\pi_h z) = a(u, z-\pi_h z) - a(u_h, z-\pi_h z).\end{aligned}$$

By expressing the former $a(\cdot,\cdot)$-functional in terms of data,

$$(e, j_1) + (e, j_2)_{\partial\Omega_\mathrm{N}} = (f, z-\pi_h z) + (g, z-\pi_h z)_{\partial\Omega_\mathrm{N}} - a(u_h, z-\pi_h z),$$

allowing us to control functionals of the error both inside the domain and on the Neumann boundary.

We emphasize that the dual problem cannot be solved exactly, at least not in general, and thus has to be approximated. So a tempting idea, from a practical point of view, would be to reuse the primal mesh: Find $z_h \in V_h$ such that

$$a(v, z_h) = (j, v) = J(v), \quad \forall v \in V_h, \tag{25}$$

but this does not work. The reason why is that the Galerkin orthogonality leads to the trivial error representation

$$J(e) \overset{(24)}{=} (f - \mathcal{A}u_h, z_h - \pi_h z) = 0,$$

since z_h and $\pi_h z_h$ will coincide. Hence we typically need the dual approximation to be more accurate than the FE-solution—the approach in **Paper II** was to solve the dual plate and plane stress problems with respect to enriched function spaces. This may be straightforward, but also computationally demanding. Focusing on evaluating the suggested FEM, we did not pursue any alternatives, e.g., by post-processing a solution of (25) to get an estimate of $z - \pi_h z$ (we refer to Larsson et al. [15] for working strategies). The procedure is merited only for illustrating the convergence of the error estimator for various meshes.

If the data of the dual problem is a function of the error itself, i.e., $j = j(e)$, we somehow have to approximate the error, besides estimating $z - \pi_h z$, which obviously becomes time-consuming. In **Paper II** this was not necessary for the dual plate problem, where $j = j(u_h)$ (a pointwise displacement), but the dual plane stress problem required an approximation $\tilde{u}$ (which was retrieved alongside $\tilde{z}$ at a small cost). If we want to control the error in norms other than the energy norm, or in a linear functional of the error, the information inevitably comes at a price.

This short review did not encompass the (interesting) non-linear variational formulation, as this was outside the scope of **Paper II**, which then calls for a linearization of the continuous dual problem. The interested reader is referred to Larsson [14, Section 2] and Bangerth and Rannacher [4].

Using duality arguments in a posteriori error estimation was introduced during the early 1990s, in works by Eriksson et al. [10] among others. It was later developed into the DWR-method by Becker and Rannacher [5]. Quantitative error control by computational means akin to the DWR-method, in the context of model adaptivity, has been seen in, e.g., Oden et al. [18, 20, 31] (solid/fluid mechanics applications and heterogeneous materials) and Braack and Ern [6] (Poisson's equation, convection-diffusion-reaction equations). Recent work in model and goal adaptivity, within the field of multiscale modeling, includes Oden et al. [19].

1.3 Summary of Appended Papers

In **Paper I**, *Model Adaptivity for Elasticity on Thin Domains*, we consider Navier's equations in two spatial dimensions, with the main idea of constructing a model hierarchy to facilitate the solution procedure. Being based on increasingly higher polynomial expansions through the thickness of the domain (coupled with a Galerkin approach), the suggested hierarchy seems like a natural extension of the Bernoulli and Timoshenko beam theories. Energy norm error estimates are outlined, which motivate uncoupling of the discretization and modeling errors, thereby providing local error indicators. We introduce an adaptive algorithm, concurrently refining mesh and model, and evaluate its behavior. The numerical results indicate sharp error control.

In **Paper II**, *An Adaptive Finite Element Method for Second Order Plate Theory*, the fourth-order Kirchhoff-Love model is supplemented by a second-order term to include the effects of membrane stresses. The plate is approximated by piecewise continuous second degree polynomials (having discontinuous derivatives), whereas the in-plane deformations (which are not used explicitly) are represented by a constant-strain element. We derive an a posteriori error estimate, separating the bending and membrane effects (the stresses thereby appear as a modeling error), for controlling a linear functional of the error. A goal-oriented adaptive algorithm is proposed, and evaluated with respect to the maximum plate deflection, under various loading conditions. Effectivity indices close to unity suggest sharp error control.

1.4 Conclusions and Future Work

We consider model adaptivity in linear elasticity for dimensionally reduced forms: **Paper I** discusses a thin domain setting, whereas **Paper II** treats an extension of the Kirchhoff-Love plate equation. The aim has been to establish efficient solution procedures—including accurate error control—by adapting, not the discretization of the computational domain alone, but also the underlying model, i.e., the mathematical formulation of the physical problem. To change the model we need an hierarchy (the approach in **Paper II** is somewhat different) to choose from, and a means to indicate when and where a substitution should take place. This means that the complexity of the model varies over the domain, just as the local mesh size does, to ensure that the total error is efficiently minimized.

In **Paper I** we constructed a model hierarchy, which was used to solve a small set of test problems, and it managed to do so accurately. The adaptive algorithm was not thoroughly tested, though, but the approach seems promising. For the cantilever beam (square domain) it was actually more efficient than the Zienkiewicz-Zhu method used in [2], where bilinear elements were applied in a standard setting (without a thin domain approach). We emphasize this to be in terms of degrees of freedom, as the computational costs were not compared.

Paper II has the more flexible approach of goal-adaptive error control. The error estimator was efficient throughout the iterative procedures, which is important for problems in engineering analysis (the mesh size does not tend to zero in practice).

The extension of the existing model hierarchy in **Paper I** is the goal for future work, i.e., to introduce the Bernoulli and Timoshenko beam theories as simpler models. The elements should be altered to adopt hierarchical basis functions for improved numerical stability. This could be done by following Vogelius and Babuška [33], and employ Legendre polynomials instead. The error estimates may be sharpened, e.g., to account for geometrical anisotropy.

However, another and more challenging problem, is to apply the thin domain approach in a higher dimension. The 3D-elasticity formulation could then be reduced by using increasingly higher polynomial expansions through the thickness. The idea is by no means novel, being the subject of study in numerous articles, e.g., in Babuška and Schwab [3], Ainsworth [1], and Repin et al. [25]. They derived residual-based estimates in energy norm, whereas Vemaganti [30] used duality-based techniques to control a continuous linear functional of the error. Our approach would deviate by relying on the method in **Paper II**, for the Kirchhoff-Love model, supplemented by the Reissner-Mindlin plate. The former is to be regarded as the basic model, and the latter is more complex, since it introduces additional rotational degrees of freedom. The FEM presented by Hansbo and Larson [13], based on discontinuous P^1-approximations for the rotations, and continuous piecewise quadratic polynomials for the transverse displacements, may serve as a starting point. Additional ideas for bridging the Reissner-Mindlin plate with a thin domain hierarchy stem from Nitsche's method [17]. The focus shall be on a goal-adaptive approach, where solving the adjoint problems in cost-efficient manner will be desirable.

Chapter 2

Implementation

We shall provide a few hints on how to implement FE-algorithms in an efficient manner. The intention is not to be comprehensive, as getting good performance out of a computer system is a complex task, but focuses on some common tools, which were used for writing the codes underlying the numerical simulations of this thesis.

When writing scientific software there are several things to bear in mind, and depending on its use, the order of priority will be different. However, the basic ubiquitous principles are:

- *correctness*,
- *numerical stability*,
- *flexibility*,
- *efficiency*.

The first point needs no further explanation—if the program is not correct, it will not do the work. If the algorithm is not stable, one cannot trust the results[1] (unless they are thoroughly tested). It must be flexible to be useful—making small changes should come easy. Assuming a correct implementation of a numerically stable algorithm, then we can turn to optimization in terms of speed and memory[2]. A nice textbook for implementing scientific software is [24].

Experience set aside—hard lessons have taught us that FE-implementations are not free from bugs!—we turn our attention to efficiency. In particular, we concentrate on single-core CPUs (although the techniques we exemplify could be parallelized), and MATLAB [22] becomes our choice for a programming environment. The reason why is closely related to flexibility: MATLAB is based on the abstraction of matrices, using a consistent set of internal data structures, allowing for relatively effortless FE-implementation. The builtin visualization capabilities are powerful enough for most purposes. Hence it suits the needs of research codes well, where the primary objective is the evaluation of algorithms, and hard time spent on optimization is less worthwile (the final code may only be executed a few times). An alternative to MATLAB, which has the benefit of being distributed under an open-source license,

[1] Thus means for estimating the error of approximations are essential, error control being an integral part of adaptive FE-algorithms, just as important, if not more, as efficiency is.

[2] Memory is crucial, since an algorithm requiring more memory than the machine has, simply will not run.

is Python. Production codes, that run on a daily basis, have different demands, and usually rely on compiled languages, say, Fortran or C/C++.

Even if programmer time is prioritized over computer time, a common complaint about MATLAB is the supposedly slow implementation of large sparse FE-problems. We shall aim at addressing this issue, and hopefully reach a compromise between flexibility and efficiency. In all tuning, the effort should be put with vital and time-consuming components of the code, according to the 90/10-rule ("90 % of the time is spent in 10 % of the code"). At the heart of FE-implementations is the assembly process. This is where the element stiffnesses and loads of (14)

$$\mathbf{S}_K = \int_K \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} \, \mathrm{d}\boldsymbol{x}, \quad \boldsymbol{f}_K = \int_K \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{f} \, \mathrm{d}\boldsymbol{x} + \int_{\partial K \cap \partial \Omega_{\mathrm{N}}} \boldsymbol{\varphi}^{\mathrm{T}} \boldsymbol{g} \, \mathrm{d}s,$$

are computed locally, element by element, and gathered into their corresponding positions of the global system. This is typically implemented by means of explicit for-loops, since the use of MATLAB's vectorization capabilities becomes cumbersome, if at all possible, for that level of complexity. In other situations, one is usually advised to avoid such constructs, due to MATLAB being an interpreted language, which produces slow codes if not used properly (the use of built-in functions consisting of compiled C-code is preferable). The introduction of the JIT-accelerator with MATLAB 6.5 [16], though, has improved matters, and thus we will point out another caveat: the generation of the sparse matrix.

Example. The support for handling sparse matrices is a convenient feature of MATLAB, but it should not be abused, which would lead to poor performance in applications. Assume that a local matrix $\mathbf{S}_K$ has already been computed—let us present four alternatives for updating the global (sparse) matrix $\mathbf{S}$:

1) update $\mathbf{S}$ elementwise (using nested for-loops)

```
% # elements/element nodes:
[nele, nenod] = size(nodes);

% # element DOFs:
nedof = 2 * nenod;

[...]    % ---> left-out code (initializations, ...)

for iel = 1:nele    % ---> outer element loop

  % Element nodes:
  enod = nodes(iel, :);

  % DOFs affected by element:
  edof([1:2:end, 2:2:end]) = [2 * enod - 1, 2 * enod];

  [...]   % ---> left-out code for computing Sk

  % Update global matrix:
  for j = 1:nedof
    for i = 1:nedof
      S(edof(i), edof(j)) = S(edof(i), edof(j)) + Sk(i, j);
    end
  end
```

```
end
```

2) update **S** “matrixwise” (vectorizing loops)

```
  % Update global matrix:
  S(edof, edof) = S(edof, edof) + Sk;
```

3) generate **S** using index vectors

```
% # DOFs in mesh:
ndof = 2 * length(xnod);

% # local matrix elements:
nn = (2 * nenod)^2;

% Index vectors:
row = zeros(nn * nele, 1);
col = row;
val = row;
up = 0;

[...]

for iel = 1:nele

  [...]

  % Update index vectors:
  XM = edof(:, ones(1, nedof));
  YM = XM';
  lo = up + 1;
  up = up + nn;
  row(lo:up) = XM(:);
  col(lo:up) = YM(:);
  val(lo:up) = Sk(:);

end

% Construct sparse stiffness matrix:
S = sparse(row, col, val, ndof, ndof);
```

4) use faster `sparse2` function

```
% Construct sparse stiffness matrix:
S = sparse2(row, col, val, ndof, ndof);
```

We have only included necessary parts of the assembly routine (dots within brackets means that code has been omitted): `nodes` is a topology matrix, where the i:th row holds the global node numbers on element K_i, and `xnod` stores the x_1-coordinates of all nodes. A benchmark test[3], solving the plain strain formulation of **Paper I** (cantilever beam with $t = 1$) using linear triangles, reveals significant differences in performance:

[3] Setup (hardware/OS/version): AMD Athlon™ 64 3200+ (single core CPU with 512 kB L2-cache) and 1 024 MB RAM; running RHEL 4.6 and MATLAB 2007b.

TABLE 2.1: *Benchmark test (runtimes in seconds)*

`ndof`	Method 1	Method 2	Method 3	Method 4
4 290	6.2	5.1	0.38	0.36
8 386	46.0	42.6	0.75	0.72
16 770	131.9	125.0	1.51	1.44
33 154	...	...	3.04	2.88
66 306	...	...	6.13	5.75
131 842	...	...	12.27	11.50

The timings are with respect to the assembly process, i.e., including calculation of local matrices, but excluding solving the linear system $\mathbf{Su} = \mathbf{f}$. So how can this be? In each iteration a local 6×6-matrix $\mathbf{S}_K$ is stored, where apparently vectorizing the code accounts only for minor improvements (cf. Methods 1-2). The explanation lies elsewhere, namely in how MATLAB stores sparse matrices, which, basically, is done by means of three vectors: `row`, `col`, and `val` (compressed column format). The two former represent element indices, whereas the latter is the element value. Changing an element $\mathbf{S}_{ij}$ means that the triplets must be updated, which takes time proportional to the number of non-zero elements of $\mathbf{S}$ [21]. The idea in Methods 3-4 is to create the list of triplets instead, and have MATLAB convert them into a sparse matrix all at one (if there are duplicates these are summed), which becomes much faster. The difference between the `sparse` and `sparse2` routines, is that the latter employs a quicker sorting (linear-time bucket sort). The speed-up, counting the time for generating $\mathbf{S}$, was about a factor 2 (in this particular model problem).

This was merely a small example of how to improve FE-codes, but it applies to other (sparse) matrix problems as well, and is a minor tweak for the gain in performance. An introduction to high-performace computing, with emphasis on compiled languages and Fortran90, is provided by Ericsson [9].

`sparse2` is part of CHOLMOD, included in the SuiteSparse package, downloadable at the MATLAB Central or [23]; the author has written comprehensive texts on solving sparse linear systems [8].

Bibliography

[1] M. Ainsworth, *A posteriori error estimation for fully discrete hierarchic models of elliptic boundary problems on thin domains*, Numerische Mathematik **80** (1998), 325–362.

[2] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz, *Analysis of the Zienkiewicz-Zhu a posteriori error estimator in the finite element method*, International Journal for Numerical Methods in Engineering **28** (1989), 2161–2174.

[3] I. Babuška and C. Schwab, *A posteriori error estimation for hierarchic models of elliptic boundary value problems on thin domains*, SIAM Journal on Numerical Analysis **33** (1996), no. 1, 221–246.

[4] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser Verlag, 2003.

[5] R. Becker and R. Rannacher, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica **10** (2001), 1–102.

[6] M. Braack and A. Ern, *A posteriori control of modeling errors and discretization errors*, SIAM: Multiscale Modeling and Simulation **1** (2003), no. 2, 221–238.

[7] S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 2:nd ed., Springer-Verlag, 2002.

[8] T.A. Davis, *Direct Methods for Sparse Linear Systems*, SIAM: Fundamentals of Algorithms, 2006.

[9] T. Ericsson, *Performance engineering*, `http://www.math.chalmers.se/~thomas/PDC/springer.pdf`.

[10] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to Adaptive Methods for Differential Equations*, Acta Numerica (1995), 105–158.

[11] ———, *Computational Differential Equations*, Studentlitteratur, 1996.

[12] A. Ern and J-L. Guermond, *Theory and Practice of Finite Elements*, Springer-Verlag, 2004.

[13] P. Hansbo and Larson M.G., *A P^2-continuous P^1-discontinuous finite element method for the Mindlin-Reissner plate model*, Preprint 2001:22 (Chalmers Finite Element Center), ISSN 1404-4382, Chalmers University of Technology (2001), no. 22.

[14] F. Larsson, *Goal-oriented adaptive finite element analysis in computational material mechanics*, Chalmers University of Technology, Ph.D. dissertation, 2003.

[15] F. Larsson, P. Hansbo, and K. Runesson, *Strategies for computing goal-oriented a posteriori error measures in non-linear elasticity*, International Journal for Numerical Methods in Engineering **55** (2002), 879–894.

[16] The MathWorks™, *Accelerating MATLAB: The MATLAB JIT-Accelerator*, Newsletters—MATLAB Digest **10** (2002), no. 5.

[17] J. Nitsche, *Über ein Variationsprincip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind*, Abh. Math. Sem. Univ. Hamburg **36** (1971), 9–15.

[18] J.T. Oden and S. Prudhomme, *Estimation of Modeling Error in Computational Mechanics*, Journal of Computational Physics **182** (2002), 496–515.

[19] J.T. Oden, S. Prudhomme, A. Romkes, and P. Bauman, *Multiscale modeling of physical phenomena: adaptive control of models*, SIAM: Journal on Scientific Computing **28** (2006), no. 6, 2359–2389.

[20] J.T. Oden and K. Vemaganti, *Estimation of Local Modeling Error and Goal-Oriented Adaptive Modeling of Heterogeneous Materials: Part I*, Journal of Computational Physics **164** (2000), 22–47.

[21] Blog of Loren Shure (guest-author: T.A. Davis; 1:st March 2007), *Creating Sparse Finite-Element Matrices in MATLAB*, `http://blogs.mathworks.com/loren/2007/03/01/creating-sparse-finite-element-matrices-in-matlab/`.

[22] Homepage of MathWorks™, `http://www.mathworks.com/`.

[23] Webpage of Tim Davis, `http://www.cise.ufl.edu/research/sparse/`.

[24] S. Oliveira and D. Stewart, *Writing Scientific Software*, Cambridge University Press, 2006.

[25] S. Repin, S. Sauter, and A. Smolianski, *A posteriori error estimation of dimension reduction errors for elliptic problems on thin domains*, SIAM Journal on Numerical Analysis **42** (2004), no. 4, 1435–1451.

[26] C. Schwab, *A posteriori error estimation for hierarchical plate models*, Numerische Mathematik **74** (1996), 221–259.

[27] E. Stein and S. Ohnimus, *Dimensional Adaptivity in Linear Elasticity with Hierarchical Test-Spaces for h- and p-Refinement Processes*, Engineering with Computers **12** (1996), 107–119.

[28] ______, *Coupled model- and solution-adaptivity in the finite element method*, Computer Methods in Applied Mechanics and Engineering **150** (1997), 327–350.

[29] ______, *Anisotropic discretization- and model-error estimation in solid mechanics by local Neumann problems*, Computer Methods in Applied Mechanics and Engineering **176** (1998), 363–385.

[30] K. Vemaganti, *Local error estimation for dimensionally reduced models of elliptic boundary value problems*, Computer Methods in Applied Mechanics and Engineering **192** (2003), 1–14.

[31] K. Vemaganti and J.T. Oden, *Estimation of local modeling error and goal-oriented adaptive modeling of heterogeneous materials: Part II*, Computer Methods in Applied Mechanics and Engineering **190** (2001), 6089–6124.

[32] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley and Teubner, 1996.

[33] M. Vogelius and I. Babuška, *On a Dimensional Reduction Method: I. The Optimal Selection of Basis Functions*, Mathematics of Computations **37** (1981), no. 155, 31–46.

Part II

Papers

Paper I

Model Adaptivity for Elasticity on Thin Domains

Model Adaptivity for Elasticity on Thin Domains

David Heintz

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg

Abstract

We consider the equations of linear elasticity on thin domains in two spatial dimensions. The main idea is the construction of a model hierarchy, that facilitates an efficient solution procedure. An energy norm *a posteriori* error estimate is outlined, which provides an upper bound on the total error. However, and more important, a preceding semi-discrete estimate motivates uncoupling of the discretization and model errors—thereby we obtain a means for extracting local error indicators. We introduce an adaptive algorithm, which concurrently refines mesh and model, aiming at a balance between different error contributions. Numerical results are presented to exemplify the behavior of the algorithm.

Keywords: *model adaptivity, model error, a posteriori error*

1 Introduction

Adaptive techniques based on a posteriori error estimates in the finite element method (FEM) are well-developed. The algorithms usually strive to efficiently reduce the discretization error, meaning the discrepancy between the continuous model—the exact solution of the differential equation at hand—and the corresponding FE-solution. The goal is to ascertain a user-specified tolerance on the error to a (nearly) minimal computational cost.

However, if the prescribed accuracy should be with respect to the total error, one has to consider the choice of model carefully. The total error e_{T} is

$$e_{\mathrm{T}} = e_{\mathrm{D}} + e_{\mathrm{M}},$$

including the model error e_{M}. Unfortunately, the most complex model (thus implying $e_{\mathrm{M}} \to 0$) could be inherently expensive to use, just as resolving a simpler one ($e_{\mathrm{D}} \to 0$) does not improve the accuracy, once the relatively large e_{M} dominates. Therefore we seek an adaptive strategy taking both error sources into account. Ideally, the local error contributions should be balanced, by refining the computational mesh and the model concurrently, which is known as *model adaptivity.*

In this paper we apply model adaptivity to the equations of linear elasticity in 2D on thin domains, where, given $\boldsymbol{x} = (x_1, x_2)$, x_2 is understood as the thin direction. This requires an available hierarchy of models, and such reduced models—as compared to the linear elasticity theory—are typically obtained using simplified deformation relations, e.g., the Bernoulli and Timoshenko beam theories. We shall instead follow Babuška, Lee and Schwab [2], and employ a model hierarchy based on increasingly higher polynomial expansions through the thickness of the domain, coupled with a Galerkin approach. However, we make no assumptions on the discretization error being negligible, and thus strive for simultaneous a posteriori estimation of both discretization and modeling errors.

For a certain polynomial expansion $\boldsymbol{q}$, we emphasize that the dimension of the problem could be reduced, if the x_2-dependence of the weak FE-formulation is integrated. The resulting boundary value problem, a system of ordinary differential equations (ODEs), for any $\boldsymbol{q}$, is said to correspond to a particular model. The kinematic assumptions would rely on a minimization principle, since Galerkin's method corresponds to minimizing the potential energy, together with a prescribed polynomial dependence of the displacements in the thin direction.

This viewpoint contrasts that of regarding the polynomial expansion as purely algorithmic, a certain simplified *hp*-refinement process (with separated *h*- and *p*-refinements in the x_1- and x_2-directions respectively), which instead attributes the model error to a discretization error. In the literature this kind of model adaptivity is known as *q*-adaptivity, which consequently becomes *hq*-adaptivity, when used in conjunction with *h*-adaptivity for the FE-discretization.

The reason for implementing the thin domain problem in a higher dimension, is to obtain a straightforward means for estimating e_{M}, information that is used for changing the underlying model locally.

The proposed model hierarchy will be a natural extension to another hierarchy, by bridging the abovementioned beam theories and the linear elasticity theory. This is shown by a simple example to conclude Section 3, once the relevant equations have been introduced.

We derive an energy norm a posteriori error estimate (31), based on orthogonality relations and interpolation theory, that is an upper bound of the total error.

A semi-discrete error estimate (30) justifies splitting the total error in two distinct parts, representing the effects of the discretization and model errors. It thus becomes the cornerstone for an adaptive algorithm (Algorithm 1), which strives to balance the local error contributions. Consecutive updates of mesh and model are governed by (42) and (43), local error indicators derived using a residual-based approach (with respect to the complete solution space).

In brief the paper consists of the following parts: in Section 2 we present the model problem and its corresponding weak and finite element formulations; next, in Section 3, follows a review of beam theory; in Section 4 the a posteriori error estimate is derived; and finally, in Section 5, we propose the framework of an adaptive algorithm and present some numerical results.

2 A Finite Element Method for Navier's Equations

Consider a thin rectangular domain $\Omega \subset \mathbb{R}^2$, representing a deformable medium subjected to external loads. These include body forces $\boldsymbol{f}$ and surface tractions $\boldsymbol{g}$, causing deformations of the material, which we describe by the following model problem: Find the displacement field $\boldsymbol{u} = (u_1, u_2)$ and the symmetric stress tensor $\boldsymbol{\sigma} = (\sigma_{ij})_{i,j=1}^2$, such that

$$\begin{aligned}
\boldsymbol{\sigma}(\boldsymbol{u}) &= \lambda \operatorname{div}(\boldsymbol{u})\, \boldsymbol{I} + 2\mu \boldsymbol{\varepsilon}(\boldsymbol{u}) && \text{in } \Omega && (1)\\
-\operatorname{div}(\boldsymbol{\sigma}) &= \boldsymbol{f} && \text{in } \Omega && (2)\\
\boldsymbol{u} &= \boldsymbol{0} && \text{on } \partial\Omega_{\mathrm{D}}\\
\boldsymbol{\sigma} \cdot \boldsymbol{n} &= \boldsymbol{g} && \text{on } \partial\Omega_{\mathrm{N}}
\end{aligned}$$

where $\partial\Omega = \partial\Omega_{\mathrm{D}} \cup \partial\Omega_{\mathrm{N}}$ is a partitioned boundary of Ω. Let the Lamé coefficients

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}, \tag{3}$$

with E and ν being Young's modulus and Poisson's ratio, respectively. Furthermore, $\boldsymbol{I}$ is the identity tensor, $\boldsymbol{n}$ denotes the outward unit normal to $\partial\Omega_\mathrm{N}$, and the strain tensor is

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \tfrac{1}{2}\big(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\mathrm{T}\big).$$

The vector-valued tensor divergence is

$$\mathrm{div}(\boldsymbol{\sigma}) = \left(\sum_{j=1}^{2} \frac{\partial \sigma_{ij}}{\partial x_j}\right)_{i=1}^{2},$$

representing the internal forces of the equilibrium equation. This formulation assumes, firstly, a constitutive relation corresponding to linear isotropic elasticity (the material properties are the same in all directions), with stresses and strains related by

$$\boldsymbol{\sigma}_\mathrm{v} = \begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{12}\end{bmatrix} = \begin{bmatrix}D_{11} & D_{12} & D_{13}\\ D_{21} & D_{22} & D_{23}\\ D_{31} & D_{32} & D_{33}\end{bmatrix} \begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{12}\end{bmatrix} = \boldsymbol{D}(\lambda,\, \mu)\boldsymbol{\varepsilon}_\mathrm{v},$$

referred to as *Hooke's generalized law.* If the material is homogeneous, $\boldsymbol{D}$ becomes independent of position. Secondly, a state of plain strain prevails, i.e., the only non-zero strain components are ϵ_{11}, ϵ_{22} and ϵ_{12}. This situation typically occurs for a long and thin body, loaded by forces invariant and perpendicular to the longitudinal axis, and restricted from movement along its length [11, Chapter 12.2.1]. Lastly, we make the assumption of $\boldsymbol{u}$ belonging to a tensor-product space

$$\boldsymbol{u} = \big(\phi_1(x_1)\psi_1(x_2), \phi_2(x_1)\psi_2(x_2)\big), \tag{4}$$

i.e., the solution components are products of two functions with separated spatial dependence. The tensor-product Lagrangian finite elements, which are introduced in Section 5.2, yield FE-solutions $\boldsymbol{u}^h$ on this form. The reason for considering such solutions, is for the straightforward construction of a model hierarchy, where the displacement field has a prescribed polynomial dependence in the thin direction.

Next, relating to (4), we introduce the function spaces

$$\begin{aligned} V_\phi \otimes V_\psi &= \left\{\boldsymbol{v} = (\phi_1\psi_1,\, \phi_2\psi_2) :\ \phi_i\psi_i \in V \cap H^2\right\}, \\ V &= \left\{w :\ w \in H^1,\ w|_{\partial\Omega_\mathrm{D}} = 0\right\}, \end{aligned}$$

where $\phi_i = \phi_i(x_1)$, $\psi_i = \psi_i(x_2)$, $H^k = H^k(\Omega)$ and $i,\, k = 1,\, 2$. The equilibrium equation (2) is multiplied by a test function $\boldsymbol{v} = (v_1,\, v_2) \in V_\phi \otimes V_\psi$, and the inner products are integrated (by parts) over the domain. Having reached thus far, we pose the following weak formulation: Find $\boldsymbol{u} \in V_\phi \otimes V_\psi$ such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_\phi \otimes V_\psi, \tag{5}$$

where the bilinear form

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, \mathrm{d}\boldsymbol{x} \tag{6}$$

is the integrated tensor contraction

$$\boldsymbol{\sigma} : \boldsymbol{\varepsilon} \overset{\mathrm{def}}{=} \sum_{i,j=1}^{2} \sigma_{ij}\varepsilon_{ij},$$

and the linear functional of the right-hand side is

$$L(\boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) + (\boldsymbol{g}, \boldsymbol{v})_{\partial\Omega_{\mathrm{N}}} = \int_{\Omega} \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_{\mathrm{N}}} \boldsymbol{g} \cdot \boldsymbol{v} \, \mathrm{d}s. \tag{7}$$

Remark. An equivalent formulation of (5), mainly due to the symmetry and positive definiteness of the bilinear form (we refer to [5] for more details), comes in the guise of a minimization problem: Find $\boldsymbol{u} \in V_{\phi} \otimes V_{\psi}$ such that

$$F(\boldsymbol{u}) \leq F(\boldsymbol{w}), \quad \forall \boldsymbol{w} \in V_{\phi} \otimes V_{\psi},$$

where

$$F(\boldsymbol{u}) = \tfrac{1}{2} a(\boldsymbol{u}, \boldsymbol{u}) - L(\boldsymbol{u}), \tag{8}$$

is recognized as the potential energy of $\boldsymbol{u}$.

For the numerical approximation of (5), we shall need a discrete counterpart, and as such establish a finite element method. To simplify its formulation we define the kinematic relation

$$\boldsymbol{\varepsilon}_{\mathrm{v}}(\boldsymbol{u}) = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \tilde{\boldsymbol{\nabla}} \boldsymbol{u},$$

and specify the constitutive matrix

$$\boldsymbol{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix},$$

for the purpose of rewriting the bilinear form as

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega} \boldsymbol{\varepsilon}_{\mathrm{v}}(\boldsymbol{u})^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\varepsilon}_{\mathrm{v}}(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{x},$$

which facilitates implementation. Then we introduce a partition $\mathcal{T}_h$ of Ω, dividing the domain into N_{el} quadrilateral—suitable for tensor-product approximations—elements K_i (thus having $N_{\mathrm{ed}} = N_{\mathrm{el}} + 1$ vertical edges), such that $\mathcal{T}_h = \{K_i\}_{i=1}^{N_{\mathrm{el}}}$, with nodes $\boldsymbol{x}_i$, $i = 1, 2, \ldots, N_{\mathrm{no}}$. The function

$$h_K = \mathrm{diam}(K) = \max_{\boldsymbol{y}_1, \boldsymbol{y}_2 \in K} (\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2), \quad \forall K \in \mathcal{T}_h,$$

represents the local mesh size, with $h = \max_{K \in \mathcal{T}_h} h_K$. Let $\mathcal{E}^h = \{E\}$ denote the set of element edges, which we split into two disjoint subsets, $\mathcal{E}^h = \mathcal{E}_I^h \cup \mathcal{E}_B^h$, namely the sets of interior and boundary edges, respectively.

The partition is associated with a function space

$$V_{\phi}^h \otimes V_{\psi}^h = \left\{ \boldsymbol{v} \in [\mathcal{C}(\Omega)]^2 : \, \boldsymbol{v}|_K \in \mathbb{Q}^2 \text{ for each } K \in \mathcal{T}_h, \, \boldsymbol{v}|_{\partial\Omega_{\mathrm{D}}} = \boldsymbol{0} \right\}, \tag{9}$$

where

$$\mathbb{Q} = \left\{ w : \, w = w_1(x_1) w_2(x_2), \, w_1 \in \mathbb{P}^1, \, w_2 \in \mathbb{P}^q \right\},$$

and $\mathbb{P}^q$ denotes the space of polynomials of degree $q \geq 1$ in one variable. A function in $V_\phi^h \otimes V_\psi^h$ is uniquely determined by its values at $\boldsymbol{x}_i$, together with the set of shape functions

$$\{\varphi_j\}_{j=1}^{N_{\text{no}}} \subset V_\phi^h \otimes V_\psi^h, \quad \varphi_j(\boldsymbol{x}_i) := \delta_j(\boldsymbol{x}_i),$$

which constitute a nodal basis for (9). It then follows that any $\boldsymbol{v} \in V_\phi^h \otimes V_\psi^h$ can be expressed as a linear combination

$$\boldsymbol{v} = \sum_{j=1}^{N_{\text{no}}} \boldsymbol{v}_j \varphi_j(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{v}_j = \boldsymbol{v}(\boldsymbol{x}_j)$ represent the nodal values of $\boldsymbol{v}$ (note that the number of degrees of freedom $N_{\text{d}} = 2N_{\text{no}}$, since the problem is vector-valued). We make an *ansatz* for a FE-solution of this type (10), and hence the FE-formulation of (5) becomes: Find $\boldsymbol{u}^h \in V_\phi^h \otimes V_\psi^h$ such that

$$a(\boldsymbol{u}^h, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_\phi^h \otimes V_\psi^h, \tag{11}$$

whose solution usually is written on the standard form

$$\boldsymbol{u}^h = \begin{bmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \dots \\ 0 & \varphi_1 & 0 & \varphi_2 & \dots \end{bmatrix} \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ \vdots \end{bmatrix} = \boldsymbol{\varphi}\mathbf{u},$$

associating odd and even elements of $\mathbf{u}$ with displacements in x_1 and x_2, respectively. Since testing against all $\boldsymbol{v} \in V_\phi^h \otimes V_\psi^h$ reduces to testing against $\{\varphi_j\}_{j=1}^{N_{\text{no}}}$, and $\boldsymbol{\varepsilon}_{\text{v}}(\boldsymbol{u}^h) = \tilde{\boldsymbol{\nabla}}\boldsymbol{\varphi}\mathbf{u} = \boldsymbol{B}\mathbf{u}$, (11) corresponds to solving

$$\int_\Omega \boldsymbol{B}^{\text{T}}\boldsymbol{D}\boldsymbol{B}\,\mathrm{d}\boldsymbol{x}\,\mathbf{u} = \int_\Omega \boldsymbol{\varphi}^{\text{T}}\boldsymbol{f}\,\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega_{\text{N}}} \boldsymbol{\varphi}^{\text{T}}\boldsymbol{g}\,\mathrm{d}s, \tag{12}$$

i.e., the matrix problem $\mathbf{S}\mathbf{u} = \mathbf{f}$, making (12) a suitable starting point for FE-implementation.

3 The Bernoulli and Timoshenko Beam Equations

The geometry of a problem sometimes allows for simplifications, although such formulations usually violate the field equations, i.e., the equilibrium balance or the kinematic and constitutive relations. Let us exemplify by considering the beam, which is dominated by its extension in the axial direction. Bernoulli stated how *"plane sections normal to the beam axis remain in that state during deformation"* (it follows that $\theta = \mathrm{d}u/\mathrm{d}x_1$, i.e., the slope of the deflection is the first order derivative). Further kinematic assumptions eventually lead to the only non-zero strain component being ϵ_{11}. Consequently, for an isotropic material with a linear elastic response, this would correspond to

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \end{bmatrix} = \frac{E\epsilon_{11}}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu \\ \nu \end{bmatrix}, \tag{13}$$

and in particular that $\sigma_{12} = 0$, so the effects of transverse shear deformations are neglected. The constitutive relation of the Bernoulli theory is actually less complex, assuming a uniaxial state of stress with $\sigma_{11} = E\epsilon_{11}$, suggesting that $\nu = 0$ in (13). The simplified formulation, as compared to (2), becomes a fourth order ODE (we refer to [11, Chapter 17.1] or [12, Chapter 5.9] for a detailed derivation):

$$\frac{\mathrm{d}^2}{\mathrm{d}x_1^2}\left(EI\frac{\mathrm{d}^2 u}{\mathrm{d}x_1^2}\right) = f, \tag{14}$$

where $u = u(x_1)$ and $f = f(x_1)$ represents a distributed load [`N/m`]. We restrict the discussion to prismatic beams, with rectangular cross-sections of size $A = wt$, which will have a constant flexural rigidity EI [`Nm`^{2}]. Here I [`m`^{4}] is the moment of inertia, and with respect to unit length (set the width $w = 1$), we now get

$$I = \int_A x_2^2 \,\mathrm{d}A = \int_{-t/2}^{t/2} x_2^2 \, dx_2 = \frac{t^3}{12}.$$

Timoshenko proposed a more accurate model, which accounts for deflections due to shear. Thus a plane section normal to the beam axis, although still plane, is not necessarily normal after deformation. The system of ODEs has the form (rewritten from [12, Chapter 5.12]):

$$\begin{cases} EI\dfrac{\mathrm{d}^3\theta}{\mathrm{d}x_1^3} = f, \\ \dfrac{\mathrm{d}u}{\mathrm{d}x_1} = \theta - \dfrac{EI}{A\kappa G}\dfrac{\mathrm{d}^2\theta}{\mathrm{d}x_1^2} \end{cases} \tag{15}$$

where κ represents the shear coefficient [`1`] (geometry dependent), and G is the shear modulus [`N/m`^{2}]. Should the last term of the second equation be omitted, (15) and (14) are equivalent.

The Bernoulli beam theory provides close approximations for long slender beams, typically when $L/t > 5$–10 [11, Chapter 17.1], since the shear strain σ_{12} then usually is small. Thicker beams are better modeled using the Timoshenko beam theory. For still higher beams, we now show the thin domain approach, as mentioned briefly in Section 1, to be a natural extension of the latter.

Linear polynomial dependence. Starting at (5), consider a completely fixed uniform beam of length L and thickness t, subjected to a constant volume load, $\boldsymbol{f} = [0, -a]$, $a > 0$ [`N/m`^{2}]. If we assume a linear polynomial dependence of the displacements (in the thin direction), this model (the simplest available in our hierarchy) has the semi-discrete solution

$$\boldsymbol{u}(\boldsymbol{x}) = \begin{bmatrix} u_1(\boldsymbol{x}) \\ u_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} u_1^{\mathrm{L}}(x_1)\left(1 - \frac{x_2}{t}\right) + u_1^{\mathrm{U}}(x_1)\frac{x_2}{t} \\ u_2^{\mathrm{L}}(x_1)\left(1 - \frac{x_2}{t}\right) + u_2^{\mathrm{U}}(x_1)\frac{x_2}{t} \end{bmatrix}, \quad \boldsymbol{u} \in V_\phi \otimes V_\psi^h, \tag{16}$$

where u_i^{L} and u_i^{U} denote displacements on the lower and upper sides, respectively. Moreover, when imposing the additional kinematic relations (according to the Bernoulli and Timoshenko theories)

$$u_1^{\mathrm{U}} = -u_1^{\mathrm{L}}, \quad u_2^{\mathrm{L}} = u_2^{\mathrm{U}}, \tag{17}$$

as shown in Figure 1, and assuming small deformations, so that $\theta \approx \tan(\theta) \approx u_1^{\mathrm{U}}/(t/2)$, (16) reduces to

$$\boldsymbol{u}(\boldsymbol{x}) = \begin{bmatrix} \left(x_2 - \frac{t}{2}\right)\theta \\ -u \end{bmatrix}, \tag{18}$$

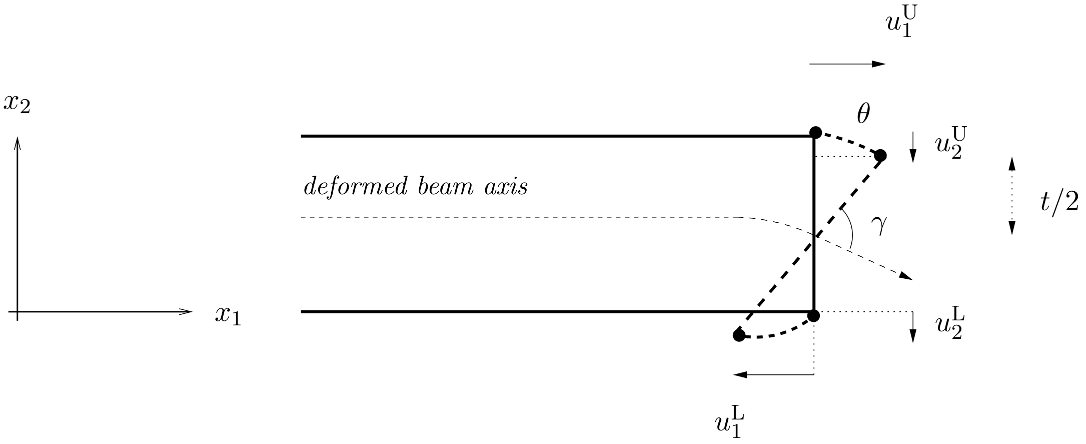


FIGURE 1: *Plane normal sections may not be normal after deformation* ($\gamma < \pi/2$)

writing $-u = u_2^{\mathrm{U}}$ for the transversal deflection. Let us substitute (18) into (5) for $\nu = 0$ (since there is no lateral contraction in the beam theory). It then follows by (3) that $\lambda = 0$, and the shear modulus

$$\mu = G = \frac{E}{2(1+\nu)} = \frac{E}{2},$$

whereas the stress tensor (1) simplifies to $\boldsymbol{\sigma}(\boldsymbol{u}) = E\boldsymbol{\varepsilon}(\boldsymbol{u})$. Next, in accordance with Galerkin's method, we test the weak form against

$$\boldsymbol{v}_1(\boldsymbol{x}) = \begin{bmatrix} v_1(x_1)\left(\frac{2}{t}x_2 - 1\right) \\ 0 \end{bmatrix}, \quad \boldsymbol{v}_2(\boldsymbol{x}) = \begin{bmatrix} 0 \\ v_2(x_1) \end{bmatrix}, \quad \boldsymbol{v}_i \in V_\phi \otimes V_\psi^h.$$

Note that the x_2-dependence of (6) can be integrated: if denoting, e.g., $\mathrm{d}u/\mathrm{d}x_1 = u'$ for short, that means

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}_1) &= E\int_0^L \int_0^t \Big(\big(x_2 - \tfrac{t}{2}\big)\big(\tfrac{2}{t}x_2 - 1\big)\theta' v_1' + \tfrac{1}{t}(\theta - u')v_1 \Big)\,\mathrm{d}x_2\mathrm{d}x_1 \\ &= E\int_0^L \Big(\tfrac{t^2}{6}\theta' v_1' + (\theta - u')v_1 \Big)\,\mathrm{d}x_1 \\ &= E\int_0^L \big((\theta - u') - \tfrac{t^2}{6}\theta''\big)v_1\,\mathrm{d}x_1, \end{aligned}$$

using integration by parts—assume u and θ to be sufficiently regular functions—in conjunction with the prescribed boundary conditions (we have neither translation nor rotation at the fixed ends $x_1 = 0$, $x_1 = L$). Analogously, for the second test function,

$$\begin{aligned} a(\boldsymbol{u}, \boldsymbol{v}_2) &= E\int_0^L \int_0^t \tfrac{1}{2}(\theta - u')v_2'\,\mathrm{d}x_2\mathrm{d}x_1 = E\int_0^L \tfrac{t}{2}(\theta - u')v_2'\,\mathrm{d}x_1 \\ &= E\int_0^L -\tfrac{t}{2}(\theta - u')'v_2\,\mathrm{d}x_1, \end{aligned}$$

whereas the linear functional (7) evaluates to

$$L(\boldsymbol{v}_1) = 0, \quad L(\boldsymbol{v}_2) = -ta\int_0^L v_2\,\mathrm{d}x_1,$$

since the inner products $\boldsymbol{f}\cdot\boldsymbol{v}_1 = 0$, $\boldsymbol{f}\cdot\boldsymbol{v}_2 = -av_2$. Now, by standard arguments (see, e.g., [5, Chapter 8.1.2]), we may expect the weighted averages

$$\int_0^L \big[E\big((\theta - u') - \tfrac{t^2}{6}\theta''\big)\big]v_1\,\mathrm{d}x_1 = 0,$$

$$\int_0^L \big[-\tfrac{Et}{2}(\theta - u')' + ta\big]v_2\,\mathrm{d}x_1 = 0,$$

to actually hold pointwise, and thereby we identify the strong forms

$$\frac{Et}{2}\frac{\mathrm{d}}{\mathrm{d}x_1}\left(\frac{\mathrm{d}u}{\mathrm{d}x_1} - \theta\right) = -ta, \tag{19}$$

$$\frac{Et^2}{6}\frac{\mathrm{d}^2\theta}{\mathrm{d}x_1^2} + E\left(\frac{\mathrm{d}u}{\mathrm{d}x_1} - \theta\right) = 0. \tag{20}$$

If substituting (20) into (19) we obtain the system of ODEs

$$\begin{cases} \dfrac{Et^3}{12}\dfrac{\mathrm{d}^3\theta}{\mathrm{d}x_1^3} = ta \\ \dfrac{\mathrm{d}u}{\mathrm{d}x_1} = \theta - \dfrac{t^2}{6}\dfrac{\mathrm{d}^2\theta}{\mathrm{d}x_1^2} \end{cases} \tag{21}$$

which relates closely to (15). To see this, set $I = t^3/12$, $f = ta$ in the first equation, and then for the second, observe that

$$\frac{EI}{A\kappa G} = \frac{t^2}{6\kappa},$$

by using $E/G = 2$, $A = wt = t$.

In conclusion, making appropriate assumptions on the kinematic and constitutive relations in (5), allows for reducing the weak formulation to 1D, by integrating along the thickness of the beam. We retrieve the equations of the Timoshenko beam theory, apart from an absent shear coefficient κ, which compensates for the shear stress not being uniform over the cross-section R (it has a parabolic shape). Experimental data for a rectangular R suggests how

$$\kappa = \frac{5(1+\nu)}{6+5\nu} = \frac{5}{6}, \quad \text{if } \nu = 0,$$

according to [8]. Note that (21) approaches (14) as $t \to 0$, i.e., this model corresponds exactly to the Bernoulli beam theory in the limiting case (just as (15) does).

Remark. We emphasize that the additional kinematic relations (17) imposed on the solution, actually means that it does not belong to our model hierarchy, and consequently, neither does the Timoshenko beam. However, (21) then suggests the thin domain approach, in our setting, to be a natural extension of the beam theories, with less constraints on the solution.

4 A Posteriori Error Estimate

We pose two auxiliary problems: Find $\boldsymbol{u}^\psi \in V_\phi^h \otimes V_\psi$ and $\boldsymbol{u}^\phi \in V_\phi \otimes V_\psi^h$ such that

$$a(\boldsymbol{u}^\psi, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_\phi^h \otimes V_\psi, \tag{22}$$

$$a(\boldsymbol{u}^\phi, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_\phi \otimes V_\psi^h, \tag{23}$$

where the tensor-product solutions are semi-discrete (exact in one variable and approximate in the other). We shall outline estimates of the total error in energy norm

$$\|\boldsymbol{e}\|_a = \|\boldsymbol{u} - \boldsymbol{u}^h\|_a := a(\boldsymbol{u} - \boldsymbol{u}^h, \boldsymbol{u} - \boldsymbol{u}^h)^{1/2},$$

which uncouple terms representing the effects of the discretization and model errors. For this purpose, we consider (22) and (23) separately, first observing that

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a = \|\boldsymbol{u} - \boldsymbol{u}^\psi + \boldsymbol{u}^\psi - \boldsymbol{u}^h\|_a \leq \|\boldsymbol{u} - \boldsymbol{u}^\psi\|_a + \|\boldsymbol{u}^\psi - \boldsymbol{u}^h\|_a \tag{24}$$

by the triangle inequality. The terms of the right-hand side are bounded, which we motivate by studying $\boldsymbol{e}^\psi = \boldsymbol{u}^\psi - \boldsymbol{u}^h$. Let $\pi_h : V_\phi^h \otimes V_\psi \to V_\phi^h \otimes V_\psi^h$ be a standard nodal interpolation operator[1] (or the L_2-projection), and note that

$$\begin{aligned}\|\boldsymbol{u}^\psi - \boldsymbol{u}^h\|_a^2 = a(\boldsymbol{u}^\psi - \boldsymbol{u}^h, \boldsymbol{e}^\psi) &\overset{(25)}{=} a(\boldsymbol{u}^\psi - \boldsymbol{u}^h, \boldsymbol{e}^\psi - \pi_h \boldsymbol{u}^\psi)\\ &\overset{(11)}{=} L(\boldsymbol{e}^\psi - \pi_h \boldsymbol{u}^\psi) - a(\boldsymbol{u}^h, \boldsymbol{e}^\psi - \pi_h \boldsymbol{u}^\psi),\end{aligned}$$

using the energy orthogonality

$$a(\boldsymbol{u}^\psi - \boldsymbol{u}^h, \boldsymbol{v}) = 0, \quad \forall \boldsymbol{v} \in V_\phi^h \otimes V_\psi^h. \tag{25}$$

Elementwise integration by parts of the second term gives

$$\begin{aligned}\|\boldsymbol{u}^\psi - \boldsymbol{u}^h\|_a^2 = &\sum_{K \in \mathcal{T}_h} \int_K (\boldsymbol{f} + \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u}^h)) \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}\boldsymbol{x}\\ &+ \int_{\partial\Omega_\mathrm{N}} \boldsymbol{g} \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}s\\ &- \sum_{K \in \mathcal{T}_h} \int_{\partial K} \boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n}_K \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}s,\end{aligned}$$

for $\boldsymbol{n}_K$ being the outward unit normal of the element boundary. Since each $E \in \mathcal{E}_I^h$ is common to two elements, we may regroup terms as

$$\begin{aligned}\|\boldsymbol{u}^\psi - \boldsymbol{u}^h\|_a^2 = &\sum_{K \in \mathcal{T}_h} \int_K (\boldsymbol{f} + \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u}^h)) \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}\boldsymbol{x}\\ &+ \int_{\partial\Omega_\mathrm{N}} (\boldsymbol{g} - \boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n}) \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}s\\ &+ \sum_{E \in \mathcal{E}_I^h} \int_E [\boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n}_E] \cdot (\boldsymbol{e}^\psi - \pi_h \boldsymbol{e}^\psi) \, \mathrm{d}s,\end{aligned}$$

where we define

$$[\boldsymbol{\sigma}(\boldsymbol{u}^h) \cdot \boldsymbol{n}_E](\boldsymbol{x}) := \lim_{\epsilon \to 0^+} \big((\boldsymbol{\sigma} \cdot \boldsymbol{n}_E)(\boldsymbol{x} + \epsilon \boldsymbol{n}_E) - (\boldsymbol{\sigma} \cdot \boldsymbol{n}_E)(\boldsymbol{x} - \epsilon \boldsymbol{n}_E)\big), \quad \boldsymbol{x} \in E,$$

[1] The existence of such an interpolant is guaranteed, since $\boldsymbol{v} \in V_\phi \otimes V_\psi \subset H^2$ by assumption, and thus has pointwise values (see [9, Chapter 5.3]).

to be the jump in traction across the element edge E with unit normal $\boldsymbol{n}_E$. Then, by means of Cauchy's inequality and suitable estimates of the interpolation error $\boldsymbol{e}^{\psi} - \pi_h \boldsymbol{e}^{\psi}$, following Johnson and Hansbo [7, Theorem 2.1], we eventually arrive at

$$\|\boldsymbol{u}^{\psi} - \boldsymbol{u}^h\|_a \leq C_1\big(\|hR_1(\boldsymbol{u}^h)\|_{L_2(\Omega)} + \|hR_2(\boldsymbol{u}^h)\|_{L_2(\Omega)}\big), \tag{26}$$

where

$$R_1(\boldsymbol{u}^h) = |\boldsymbol{R}_1(\boldsymbol{u}^h)|, \quad R_2(\boldsymbol{u}^h) = h^{1/2}\frac{\|\boldsymbol{R}_2(\boldsymbol{u}^h)\|_{L_2(\partial\Omega)}}{V(K)},$$

with $V(K)$ as the volume of K, and

$$\begin{aligned} \boldsymbol{R}_1(\boldsymbol{u}^h) &= \boldsymbol{f} + \operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u}^h)), && \text{on } K,\ K \in \mathcal{T}_h, \\ \boldsymbol{R}_2(\boldsymbol{u}^h) &= \begin{cases} \frac{1}{2}[\boldsymbol{\sigma}(\boldsymbol{u}^h)\cdot \boldsymbol{n}_E]/h_K, & \text{on } E,\ E \in \mathcal{E}_I^h, \\ (\boldsymbol{g} - \boldsymbol{\sigma}(\boldsymbol{u}^h)\cdot\boldsymbol{n})/h_K, & \text{on } E,\ E \in \mathcal{E}_B^h. \end{cases} \end{aligned}$$

$\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ represent the residuals related to the interior and the boundary of each element, respectively, whereas C_1 is a bounded interpolation constant, typically computable by a finite dimensional eigenvalue problem, see, e.g., [7, Equation 2.9, Section 2.3]. In the same manner, with $\pi_h : V_\phi \otimes V_\psi \to V_\phi^h \otimes V_\psi$, using the orthogonality relation

$$a(\boldsymbol{u} - \boldsymbol{u}^{\psi}, \boldsymbol{v}) = 0, \quad \forall \boldsymbol{v} \in V_\phi^h \otimes V_\psi,$$

we obtain

$$\|\boldsymbol{u} - \boldsymbol{u}^{\psi}\|_a \leq C_2\big(\|hR_1(\boldsymbol{u}^{\psi})\|_{L_2(\Omega)} + \|hR_2(\boldsymbol{u}^{\psi})\|_{L_2(\Omega)}\big), \tag{27}$$

Then, by adding and subtracting $\boldsymbol{u}^{\phi}$ in (24), analogous arguments eventually lead to

$$\|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_a \leq C_3\big(\|hR_1(\boldsymbol{u}^h)\|_{L_2(\Omega)} + \|hR_2(\boldsymbol{u}^h)\|_{L_2(\Omega)}\big), \tag{28}$$

$$\|\boldsymbol{u} - \boldsymbol{u}^{\phi}\|_a \leq C_4\big(\|hR_1(\boldsymbol{u}^{\phi})\|_{L_2(\Omega)} + \|hR_2(\boldsymbol{u}^{\phi})\|_{L_2(\Omega)}\big). \tag{29}$$

We assume the residuals (27) and (29), from the semi-discrete spaces, to be smaller than their discrete counterparts (26) and (28), i.e.,

$$\|\boldsymbol{u} - \boldsymbol{u}^{\psi}\|_a = (1-\alpha)\|\boldsymbol{u}^{\psi} - \boldsymbol{u}^h\|_a, \quad \|\boldsymbol{u} - \boldsymbol{u}^{\phi}\|_a = (1-\beta)\|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_a,$$

for some $0 \leq \alpha,\ \beta \leq 1$. This, given $\alpha = \beta = 0$, implies

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a \leq 2\|\boldsymbol{u}^{\psi} - \boldsymbol{u}^h\|_a, \quad \|\boldsymbol{u} - \boldsymbol{u}^h\|_a \leq 2\|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_a,$$

as upper bounds of the total error, in terms of the model and discretization errors, respectively. It follows directly

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a \leq \|\boldsymbol{u}^{\psi} - \boldsymbol{u}^h\|_a + \|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_a, \tag{30}$$

or, with $C = C_1 + C_3$,

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a \leq C\big(\|hR_1(\boldsymbol{u}^h)\|_{L_2(\Omega)} + \|hR_2(\boldsymbol{u}^h)\|_{L_2(\Omega)}\big), \tag{31}$$

which is an (completely discretized) a posteriori error estimate.

Remark. The computational mesh is subjected to geometrical anisotropy—the elements have different dimension in different directions (one element spans the thickness of the domain, so $h(\boldsymbol{x}) \to t$, as more elements are introduced; see Section 5.2 for details). The a posteriori error estimate (31) does not take this into account, but doing so may lead to sharper error bounds. An example indicating how to get improved estimates is discussed in [7, Section 2.4]. We did not pursue this here.

5 Implementation

5.1 Fundamental concepts

Using adaptivity requires some tools, e.g., a suitable norm in which the error $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u}^h$ is measured. In Section 4 the focus was on the energy norm $\|\cdot\|_a = a(\cdot,\, \cdot)^{1/2}$, seeing $\boldsymbol{u}^h$ as the minimizer to $\|\boldsymbol{u} - \boldsymbol{v}\|_a$ over $V_\psi^h \otimes V_\psi^h$. Note how (8) states that $\boldsymbol{u}^h$, as compared to $\boldsymbol{u}$, has a larger potential energy. Hence, since $F(\boldsymbol{u})$ may be expressed in terms of the energy norm,

$$F(\boldsymbol{u}) = \tfrac{1}{2}a(\boldsymbol{u},\boldsymbol{u}) - L(\boldsymbol{u}) = \tfrac{1}{2}a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u},\boldsymbol{u}) = -\tfrac{1}{2}\|\boldsymbol{u}\|_a^2,$$

the relation $\|\boldsymbol{u}^h\|_a \leq \|\boldsymbol{u}\|_a$ holds, so the computed strains $\boldsymbol{\varepsilon}(\boldsymbol{u}^h)$ are underestimated, and the numerical problem gets too stiff. In Section 4 we used the well-known energy orthogonality

$$a(\boldsymbol{e},\boldsymbol{v}) = 0, \quad \forall \boldsymbol{v} \in V_\psi^h \otimes V_\psi^h, \tag{32}$$

stating how the error $\boldsymbol{e}$ is orthogonal to the subspace $V_\psi^h \otimes V_\psi^h$. Important relations involving the energy norm can be derived from (32), e.g., the best approximation property

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a = \inf_{\boldsymbol{v}} \|\boldsymbol{u} - \boldsymbol{v}\|_a, \quad \boldsymbol{v} \in V_\psi^h \otimes V_\psi^h,$$

which implies any refined FE-solution $\boldsymbol{u}^i$ to have larger energy norm, i.e.,

$$\|\boldsymbol{u}^i\|_a \geq \|\boldsymbol{u}^{i-1}\|_a, \quad i = 1, 2, \ldots, \tag{33}$$

since we are solving a minimization problem with respect to a larger function space. Another relation is the equality

$$\begin{aligned} \|\boldsymbol{e}\|_a^2 &= a(\boldsymbol{u} - \boldsymbol{u}^h, \boldsymbol{u} - \boldsymbol{u}^h) = a(\boldsymbol{u}, \boldsymbol{u} - \boldsymbol{u}^h) - a(\boldsymbol{u}^h, \boldsymbol{u} - \boldsymbol{u}^h) \overset{(32)}{=} a(\boldsymbol{u}, \boldsymbol{u} - \boldsymbol{u}^h) \\ &= a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u},\boldsymbol{u}^h) \overset{(32)}{=} a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u},\boldsymbol{u}^h) - a(\boldsymbol{u}^h - \boldsymbol{u}, \boldsymbol{u}^h) \\ &= a(\boldsymbol{u},\boldsymbol{u}) - a(\boldsymbol{u}^h,\boldsymbol{u}^h) = \|\boldsymbol{u}\|_a^2 - \|\boldsymbol{u}^h\|_a^2, \end{aligned} \tag{34}$$

which holds only in energy norm.

5.2 The element

In Section 2 we mentioned the nodal basis, and to elaborate, tensor-product Lagrangian finite elements were implemented. The basis functions are constructed by means of one-dimensional Lagrange polynomials

$$l_i^{n-1} = \frac{(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}, \quad i = 1, 2, \ldots, n, \tag{35}$$

and are complete up to the highest order term (including additional mixed terms).

In order to treat larger classes of element geometries, one usually considers (isoparametric) mappings from a reference element $\hat{K}$ to the physical elements K_i. Let $\hat{K}$ be the quadrilateral with local coordinates $-1 \leq \xi, \eta \leq 1$, for which basis functions can be written

$$\varphi_j(\xi,\eta) = \varphi_{IJ}(\xi,\eta) = l_I^{q_{x_1}}(\xi)\, l_J^{q_{x_2}}(\eta),$$

identifying each node j with an index pair (I, J), where $1 \leq I \leq q_{x_1}{+}1$ and $1 \leq J \leq q_{x_2}{+}1$. q_{x*} corresponds to the polynomial degree of the approximation, and we set $q_{x_1} = 1$, $1 \leq q_{x_2} \leq 12$, meaning, e.g., that $\boldsymbol{u}^h$ is linear in x_1. The approximation in x_2, through the thickness of the domain, vary edgewise (restricted to 12:th order polynomials for practical reasons), suggesting that q_{x_2} globally is represented by a N_{ed}-vector $\boldsymbol{q}$, with elements q_i, $i = 1, \ldots, N_{\text{ed}}$. Note that each mesh $\mathcal{T}_h$ becomes associated with a particular model $\boldsymbol{q}_h$ (in the sequel, to ease notation, this shall be implicitly assumed).

Example. The polynomial approximation $\boldsymbol{q} = (1, 2)$, as shown in Figure 2, gives rise to the local basis functions

$$\begin{aligned}
\varphi_1(\xi,\eta) &= l_1^1(\xi)\, l_1^1(\eta) = \tfrac{1}{4}(1-\xi)(1-\eta)\\
\varphi_2(\xi,\eta) &= l_1^1(\xi)\, l_2^1(\eta) = \tfrac{1}{4}(1-\xi)(1+\eta)\\
\varphi_3(\xi,\eta) &= l_2^1(\xi)\, l_1^2(\eta) = \tfrac{1}{4}(1+\xi)\eta(\eta-1)\\
\varphi_4(\xi,\eta) &= l_2^1(\xi)\, l_2^2(\eta) = \tfrac{1}{2}(1+\xi)(1+\eta)(1-\eta)\\
\varphi_5(\xi,\eta) &= l_2^1(\xi)\, l_3^2(\eta) = \tfrac{1}{4}(1+\xi)\eta(1+\eta)
\end{aligned}$$

readily derived via (35).

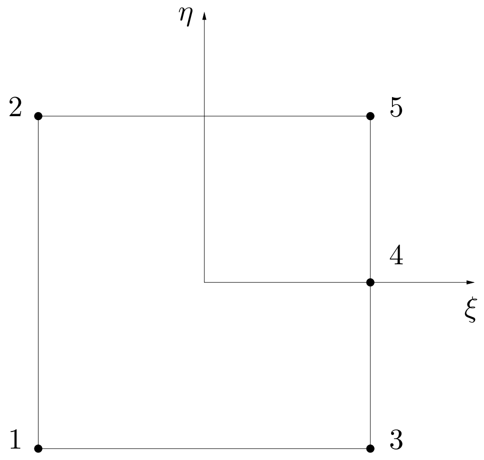
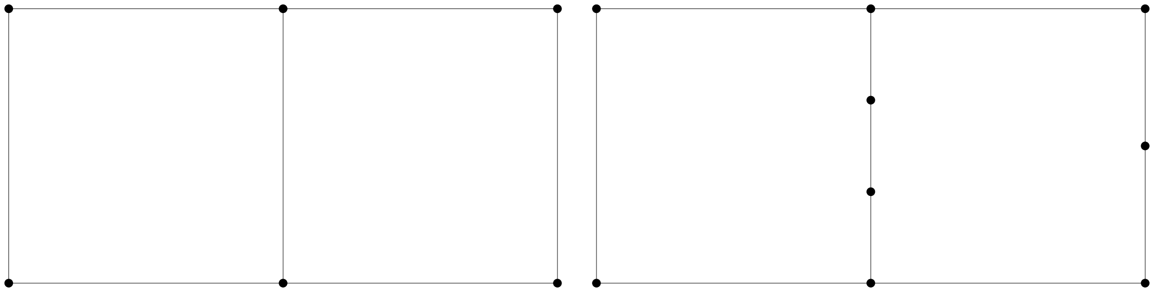


FIGURE 2: *Local node numbering for the 5-node quadrilateral* $\boldsymbol{q} = (1, 2)$

The domain Ω is partitioned into a conforming mesh, where the thickness t is spanned by a single element. Adjacent elements will overlap, leaving no hanging nodes. The discretization error e_{D}, related to the x_1-direction, is reduced when introducing more elements to the mesh. However, that does not resolve the model error e_{M} in x_2, which requires other means—instead



(A) *Simplest model using bilinear elements*

(B) *Added complexity for $\boldsymbol{q} = (1, 3, 2)$*

Figure 3: *The model hierarchy is based on increasingly higher polynomial expansions*

convergence is obtained by increasing the polynomial degree of the interpolation along vertical edges. Examples of different model complexities for $N_{\mathrm{el}} = 2$ is given in Figure 3.

Working with p-type FE-methods could impose restrictions on the choice of polynomial basis. Lagrangian finite elements have a potential caveat, as they tend to yield relatively dense stiffness matrices, subjected to bad conditioning. The better alternative would be to employ a well-conditioned modal hierarchical basis, represented by Legendre or Jacobi polynomials; we refer to [6, Chapter 1.1.5] for an introduction and further references.

Should $\mathrm{cond}(\mathbf{S})$ grow large, that indicates how the problem should be treated differently, e.g., by resorting to (full) elasticity theory. The numerical simulations in Section 5.4 managed without evident loss of accuracy[2].

5.3 Adaptive strategy

We opted for a residual-based approach, based on the idea that $\boldsymbol{u}^h$ does not satisfy (5) exactly, hoping to yield information about the error by exploiting this fact. Other common strategies—for error control in energy norm—include solving local Dirichlet or Neumann problems, and smoothening of discontinuous stresses by projection (known as the Zienkiewicz-Zhu method).

The adaptive procedure hinges on local error indicators identified via (30). We have

$$\begin{aligned} \|\boldsymbol{u}^\phi - \boldsymbol{u}^h\|_a^2 &= a(\boldsymbol{u}^\phi - \boldsymbol{u}^h, \boldsymbol{u}^\phi - \boldsymbol{u}^h) = a(\boldsymbol{u}^\phi, \boldsymbol{u}^\phi - \boldsymbol{u}^h) - a(\boldsymbol{u}^h, \boldsymbol{u}^\phi - \boldsymbol{u}^h) \\ &\overset{(23)}{=} L(\boldsymbol{u}^\phi - \boldsymbol{u}^h) - a(\boldsymbol{u}^h, \boldsymbol{u}^\phi - \boldsymbol{u}^h). \end{aligned} \tag{36}$$

Then (11), using $\boldsymbol{v} = \pi_n(\boldsymbol{u}^\phi - \boldsymbol{u}^h) = \pi_n \boldsymbol{u}^\phi - \boldsymbol{u}^h$, where $\pi_n : V_\phi \otimes V_\psi \to V_\phi^h \otimes V_\psi^h$ is a nodal interpolation operator, gives

$$L(\pi_n \boldsymbol{u}^\phi - \boldsymbol{u}^h) - a(\boldsymbol{u}^h, \pi_n \boldsymbol{u}^\phi - \boldsymbol{u}^h) = 0. \tag{37}$$

Adding (37) to (36) leads to

$$\|\boldsymbol{u}^\phi - \boldsymbol{u}^h\|_a^2 = L(\boldsymbol{u}^\phi - \pi_n \boldsymbol{u}^\phi) - a(\boldsymbol{u}^h, \boldsymbol{u}^\phi - \pi_n \boldsymbol{u}^\phi), \tag{38}$$

[2] The implemented Lagrangian finite element was not feasible for calculating reference solutions (cf. (45))

and analogously

$$\|\boldsymbol{u}^{\psi} - \boldsymbol{u}^h\|_a^2 = L(\boldsymbol{u}^{\psi} - \pi_n \boldsymbol{u}^{\psi}) - a(\boldsymbol{u}^h, \boldsymbol{u}^{\psi} - \pi_n \boldsymbol{u}^{\psi}), \tag{39}$$

representing the effects of the discretization and model errors. (The rationale behind (38) and (39), in the present context, are on the subtle side, since (36) gives equal estimates. Including the interpolant provides standard error estimates for further analysis.)

The functions $\boldsymbol{u}^{\phi}$ and $\boldsymbol{u}^{\psi}$ have to be discretized, and therefore we pose the problems: Find $\tilde{\boldsymbol{u}}^{\phi} \in V_{\phi}^* \otimes V_{\psi}^h$ and $\tilde{\boldsymbol{u}}^{\psi} \in V_{\phi}^h \otimes V_{\psi}^*$ such that

$$a(\tilde{\boldsymbol{u}}^{\phi}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_{\phi}^* \otimes V_{\psi}^h, \tag{40}$$

$$a(\tilde{\boldsymbol{u}}^{\psi}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V_{\phi}^h \otimes V_{\psi}^*, \tag{41}$$

seeking enhanced FE-solutions as approximations. The function spaces $V_{\phi}^* \otimes V_{\psi}^h$ and $V_{\phi}^h \otimes V_{\psi}^*$ are defined with respect to $\mathcal{T}_h$: (40) is solved on a bisected mesh $\mathcal{T}_{\phi}$, having twice the number of elements as $\mathcal{T}_h$, with added vertical edges inheriting $\max\{q_i,\, q_{i+1}\}$ from the parent element K_i, exemplified by $\boldsymbol{q} = (1,3,2) \rightarrow (1,3,3,3,2)$ (the children retain the highest order term of the parental approximation). (41) is solved on a mesh $\mathcal{T}_{\psi}$, with the same number of elements as $\mathcal{T}_h$, but $q_i \rightarrow q_i + 1$ (all polynomial degrees along vertical edges are raised by 1), so that $\boldsymbol{q} = (1,3,2) \rightarrow (2,4,3)$. Since

$$\|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_a^2 = \sum_{i=1}^{N_{\mathrm{el}}} \|\boldsymbol{u}^{\phi} - \boldsymbol{u}^h\|_{a,\,K_i}^2,$$

(38) and (39) suggest the computable local error indicators

$$e_{\mathrm{D},\,K} = L(\tilde{\boldsymbol{u}}^{\phi} - \pi_n \tilde{\boldsymbol{u}}^{\phi})_K - a(\boldsymbol{u}^h, \tilde{\boldsymbol{u}}^{\phi} - \pi_n \tilde{\boldsymbol{u}}^{\phi})_K, \tag{42}$$

$$e_{\mathrm{M},\,K} = L(\tilde{\boldsymbol{u}}^{\psi} - \pi_n \tilde{\boldsymbol{u}}^{\psi})_K - a(\boldsymbol{u}^h, \tilde{\boldsymbol{u}}^{\psi} - \pi_n \tilde{\boldsymbol{u}}^{\psi})_K, \tag{43}$$

where assessing (43) typically requires less degrees of freedom, but also yields a denser stiffness matrix $\mathbf{S}$, due to more connections between neighbors (cf. *computational cost* below). In each iteration a fixed ratio r of the local error indicators—the largest absolute values regardless of error source—have their corresponding elements marked for refinement (neighbors subjected to model refinement have q_i raised by 1 on common edges). Notice that this direct comparison of local indicators is justified by (30).

The approach is straightforward, benefits from controlling the sizes[3] of the refined mesh and model, but becomes insensitive to presence of local singularities (then tends to overrefine). Any initial mesh/model $\mathcal{T}_0$ should not be too coarse (typically we want the error to fall within the asymptotic convergence rate of the error produced by pure h-refinement).

The reliability of the local error indicators is validated by the effectivity index

$$\theta = \frac{\left(\sum e_{\mathrm{D},\,K}\right)^{1/2} + \left(\sum e_{\mathrm{M},\,K}\right)^{1/2}}{\|\boldsymbol{u} - \boldsymbol{u}^h\|_a} = \frac{e_{\mathrm{est}}}{\|\boldsymbol{e}\|_a}, \tag{44}$$

which ideally remains constant during the adaptive refinement, thus indicating the algorithm to converge with the same order as the underlying FE-method. If unknown, the exact solution

[3] The more sophisticated strategy would be to predict the actual mesh size and model required to achieve the prescribed accuracy.

$\boldsymbol{u}$ was approximated by a reference solution $\tilde{\boldsymbol{u}}$, which substitutes into the denominator of (44) as $\tilde{\boldsymbol{e}} = \|\tilde{\boldsymbol{u}} - \boldsymbol{u}^h\|_a$. For that purpose we state the problem: Find $\tilde{\boldsymbol{u}} \in V^h \times V^h$ such that

$$a(\tilde{\boldsymbol{u}}, \boldsymbol{v}) = L(\boldsymbol{v}), \quad \forall \boldsymbol{v} \in V^h \times V^h, \tag{45}$$

where V^h is the space of piecewise continuous quadratic functions (which vanishes on $\partial\Omega_{\mathrm{D}}$). The reference solution was resolved on a dense uniform triangulation $\mathcal{T}_r$ ($N_{\mathrm{d}}(\mathcal{T}_r) = 5\,254\,146$).

Remark. We emphasize that (30) foremost is a motivation behind (42) and (43), i.e., selecting elements for refinement, rather than providing accurate error control. When approximating the semi-discrete solutions by (40) and (41), it follows from (33) and (34) that

$$\|\boldsymbol{u}^\phi - \boldsymbol{u}^h\|_a - \|\tilde{\boldsymbol{u}}^\phi - \boldsymbol{u}^h\|_a = \|\boldsymbol{u}^\phi\|_a^2 - \|\tilde{\boldsymbol{u}}^\phi\|_a^2 \geq 0 \implies \|\boldsymbol{u}^\phi\|_a \geq \|\tilde{\boldsymbol{u}}^\phi\|_a,$$

which means we may lose the upper bound quality of (30). (This property is ensured by (31) for an upper bound of the interpolation constant, but we get no information on how to update mesh and model. Also, since (31) employs Cauchy's inequality on each element, it could yield considerable overestimations.)

A crude—but implementation independent—estimate of the computational cost for solving the primal problem is

$$N_{\mathrm{d}}(\mathcal{T}_h) = 2\sum_{i=1}^{N_{\mathrm{ed}}} q_i + 2N_{\mathrm{ed}},$$

in terms of the number of degrees of freedom N_{d} (mesh dependent). The adaptive procedure involves solving two refined problems in each iteration, to the separate costs

$$\begin{aligned} N_{\mathrm{d}}(\mathcal{T}_\phi) &= N_{\mathrm{d}}(\mathcal{T}_h) + 2\sum_{i=1}^{N_{\mathrm{el}}} \max\{q_i, q_{i+1}\} + 2N_{\mathrm{el}}, \\ N_{\mathrm{d}}(\mathcal{T}_\psi) &= N_{\mathrm{d}}(\mathcal{T}_h) + 2N_{\mathrm{ed}}, \end{aligned}$$

where $N_{\mathrm{ed}} = N_{\mathrm{ed}}(\mathcal{T}_h)$, $N_{\mathrm{el}} = N_{\mathrm{el}}(\mathcal{T}_h)$. If assuming $|q_i - q_{i+1}| \leq 1$ we get

$$2\sum_{i=1}^{N_{\mathrm{el}}} \max\{q_i, q_{i+1}\} \leq \sum_{i=1}^{N_{\mathrm{el}}} (q_i + q_{i+1} + 1) < 2\sum_{i=1}^{N_{\mathrm{ed}}} q_i + N_{\mathrm{el}} = N_{\mathrm{d}}(\mathcal{T}_h) + N_{\mathrm{el}} - 2N_{\mathrm{ed}},$$

and thus the total cost becomes

$$N_{\mathrm{d}}(\mathcal{T}_\phi) + N_{\mathrm{d}}(\mathcal{T}_\psi) < 3\big(N_{\mathrm{d}}(\mathcal{T}_h) + N_{\mathrm{el}}\big), \tag{46}$$

about three times the primal cost of solving (11). Moreover, (46) does not include calculating the local error indicators, and hence the overall procedure is more expensive, even though the cost of solving (40) and (41) is larger.

As for stopping criterion, the adaptive algorithm, detailed in Algorithm 1, halted once the finest model was locally introduced, i.e., if any $q_i = q_{\max} = 11$ (the last polynomial degree is reserved for $\tilde{\boldsymbol{u}}^\psi$). This was done for purpose of evaluation—a user-specified tolerance TOL is used in practice.

The implementation utilized a direct sparse Cholesky solver [3, 4] for solving the matrix problem (12).

Algorithm 1: Adaptive scheme

```
Data: initial mesh $\mathcal{T}_0$
Result: FE-solution $\boldsymbol{u}^h$, internal energy $\|\boldsymbol{u}^h\|_a^2$, estimated total error $e_{\text{est}}$

Let $*$ denote either sub- and superscripts $\phi$ and $\psi$

for $j = 0, 1, \dots$ do
    solve primal problem (11) for $\boldsymbol{u}^h$ on $\mathcal{T}_j$
    calculate internal energy $\|\boldsymbol{u}^h\|_a^2$
    solve refined problems (40) and (41) for $\tilde{\boldsymbol{u}}^*$ on $\mathcal{T}_*$
    for $i = 1, \dots, N_{\text{el}}$ do
        calculate local error indicators $e_{\text{D}, K_i}$ and $e_{\text{M}, K_i}$ via (42) and (43)
    end
    estimate total error $e_{\text{est}}$
    mark elements for refinement ($h, q$ or $hq$)
    if any $q_i = q_{\max}$ then
        break
    else
        refine mesh and model: $\mathcal{T}_j \to \mathcal{T}_{j+1}$
    end
end
```

5.4 Numerical results

Cantilever beam. Let Ω be the unit square $0 \leq x_1, x_2 \leq 1$, which is kept fixed at $x_1 = 0$, and subjected to a surface traction $\boldsymbol{g} = (0, -1)$ at $x_2 = 1$. The material parameters are $E = 1$ and $\nu = 0.3$. The domain is not typically thin, but the solution, according to [1, p. 2170], has the exact internal energy $\|\boldsymbol{u}\|_a^2 = 1.903697$, and makes a suitable reference.

The problem was solved adaptively for an initial configuration $N_{\text{el}} = 10$, $q_i = 3$ and $r = 0.15$. Consecutive updates of mesh and model, $\mathcal{T}_j \to \mathcal{T}_{j+1}$, $j = 0, 1, 2, \dots$, were dominated by the model error. The finest local model was introduced after 9 iterations, when the overall model complexity corresponded to 58 % of the finest global model:

$$\%(\text{model}) = 100 \cdot \frac{N_{\text{d}} - \min(N_{\text{d}})}{\max(N_{\text{d}}) - \min(N_{\text{d}})},$$

where $N_{\text{d}} = N_{\text{d}}(\mathcal{T}_j)$. Extreme values relate to the present refinement level j, such that

$$\min(N_{\text{d}}) = N_{\text{d}}(\mathcal{T}_j;\, q_i = 1),\ \max(N_{\text{d}}) = N_{\text{d}}(\mathcal{T}_j;\, q_i = q_{\max}).$$

The refinements were concentrated at the clamped end $x_1 = 0$, coinciding with large stresses and strains, close to the points $\boldsymbol{x}_1 = (0, 1)$, $\boldsymbol{x}_2 = (0, 0)$. The accuracy of $\|\boldsymbol{u}_9^h\|_a$ came within 0.2 %, not possible for the bilinear approximation (the simplest model at hand), for which $\|\boldsymbol{u}^h\|_a^2 \lessapprox 1.559$. Figure 4 further advocates the benefits of model adaptivity: the successive FE-solutions were more accurate than those obtained by uniform triangular P^2-approximations (according to (45) for subsequent mesh sizes $h = \sqrt{5}/4$, $1/2$, $\sqrt{5}/8$, $1/4$, $\sqrt{5}/16$). We normally expect singularities to be difficult to resolve using higher-order polynomial interpolations (the exact solution would not be differentiable). Therefore the small error is somewhat surprising—$\boldsymbol{u}^h$ still manages to capture the local behavior of $\boldsymbol{u}$ in the vicinity of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. In [1] the refinement strategy was another (based on the Zienkiewicz-Zhu method, trying to predict the mesh size), employing a bilinear approximation (not in a thin domain setting); the energy was $\|\boldsymbol{u}^h\|_a^2 = 1.89289$ over 718 degrees of freedom. This is less accurate than our FE-solution, which, however, was superseded at 964 degrees of freedom. Figure 6(a) shows the local error

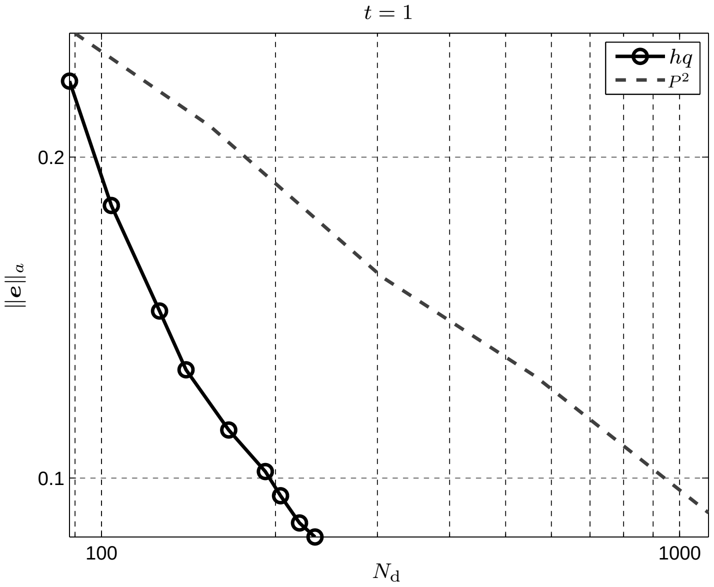


FIGURE 4: *Comparing model adaptivity (hq) and uniform triangular P^2-approximations*

contributions (40) and (41) to be of similar order, apart from a larger model error at $x_1 = 0$. The effectivity index θ, defined in (44), remained fairly constant, and judging by the data enclosed in Table 1, it also provides accurate error control. The triangle inequality leading to (24) approximately overestimated the total error by a factor $C \approx$ 1.3-1.4

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a = C\big(\|\boldsymbol{u} - \tilde{\boldsymbol{u}}^\phi\|_a + \|\tilde{\boldsymbol{u}}^\phi - \boldsymbol{u}^h\|_a\big),$$
$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_a = C\big(\|\boldsymbol{u} - \tilde{\boldsymbol{u}}^\psi\|_a + \|\tilde{\boldsymbol{u}}^\psi - \boldsymbol{u}^h\|_a\big),$$

during the iterative procedure; alongside the FE-solutions $\tilde{\boldsymbol{u}}^\phi$ and $\tilde{\boldsymbol{u}}^\phi$ having smaller energy norms (as compared to their semi-discrete counterparts), that may, at least in some extent, account for $\theta \approx 1$.

Reducing the domain thickness ($t \rightarrow 1/10$) induced a shift towards h-refinement, suggesting how higher polynomial expansions are less important, when nature of the problem turns one-dimensional. The accuracy of $\|\boldsymbol{u}_{14}^h\|_a$—with respect to a reference solution $\|\tilde{\boldsymbol{u}}\|_a^2 = 1.903687$ (the surface traction was scaled, $\boldsymbol{g} = (0, -0.058582)$, to preserve the internal energy)—was less than 0.02 %, implying a linear improvement with respect to t. In Table 2 the data shows that 26.8 % of the finest model was used at the last refinement level, the first instance of q-refinement occurring at the 7:th iteration. Figure 6(b) indicates large variations in the model error at $x_1 = 0$, which suggests how the singularities cause problems after all (at least by being

TABLE 1: *Cantilever beam* ($t = 1$)

#iter	N_d	N_el	%(model)	cond($\mathbf{S}$)	$\|\boldsymbol{u}^h\|_a^2$	$\|\boldsymbol{e}\|_a$	θ
1	88	10	20.0	$2.02 \cdot 10^4$	1.848	$2.356 \cdot 10^{-1}$	0.941
2	104	11	23.3	$4.23 \cdot 10^4$	1.871	$1.802 \cdot 10^{-1}$	1.026
3	126	12	28.4	$1.05 \cdot 10^5$	1.883	$1.435 \cdot 10^{-1}$	1.013
4	140	12	33.9	$1.65 \cdot 10^5$	1.888	$1.264 \cdot 10^{-1}$	1.070
5	166	13	39.3	$2.92 \cdot 10^5$	1.891	$1.110 \cdot 10^{-1}$	1.061
6	192	14	44.0	$1.20 \cdot 10^6$	1.893	$1.015 \cdot 10^{-1}$	1.114
7	204	14	48.0	$2.56 \cdot 10^6$	1.894	$9.635 \cdot 10^{-2}$	1.128
8	220	14	53.3	$6.58 \cdot 10^6$	1.895	$9.086 \cdot 10^{-2}$	1.141
9	234	14	58.0	$1.64 \cdot 10^7$	1.896	$8.815 \cdot 10^{-2}$	1.146

difficult to resolve beyond a certain level). The discretization error is close to equidistributed, whereas the model error drops considerably away from the fixed end, which may be consistent with no local model refinements. The effectivity index was stable and accurate, the triangle inequality overestimating e_T roughly by a factor $C \approx 1.0$-1.4.

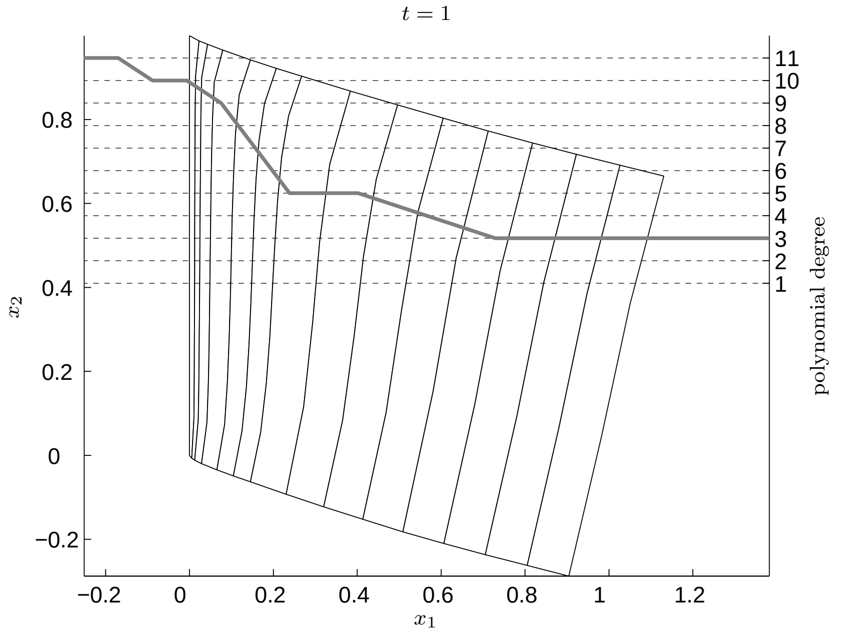
Figures 5(a) and 5(b) shows the FE-solutions for scaled displacements (due to large material deformations). Note the effects of shear deformations in the former.

TABLE 2: *Cantilever beam* ($t = 1/10$)

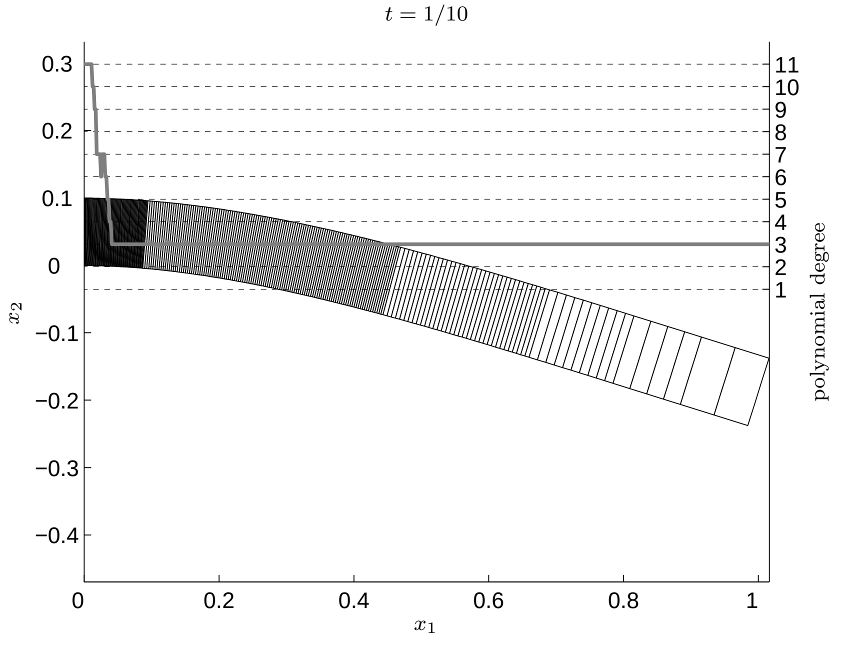
#iter	N_d	N_el	%(model)	cond($\mathbf{S}$)	$\|\boldsymbol{u}^h\|_a^2$	$\|\tilde{\boldsymbol{e}}\|_a$	θ
1	88	10	20.0	$2.13 \cdot 10^6$	1.395	$7.133 \cdot 10^{-1}$	0.826
2	112	13	20.0	$2.32 \cdot 10^6$	1.680	$4.725 \cdot 10^{-1}$	0.862
3	144	17	20.0	$3.31 \cdot 10^6$	1.780	$3.516 \cdot 10^{-1}$	0.901
4	192	23	20.0	$7.72 \cdot 10^6$	1.841	$2.515 \cdot 10^{-1}$	0.970
5	248	30	20.0	$9.42 \cdot 10^6$	1.865	$1.973 \cdot 10^{-1}$	0.990
6	320	39	20.0	$2.13 \cdot 10^7$	1.881	$1.524 \cdot 10^{-1}$	1.053
7	412	50	20.4	$3.01 \cdot 10^7$	1.890	$1.192 \cdot 10^{-1}$	1.023
8	526	63	21.1	$8.07 \cdot 10^7$	1.895	$9.532 \cdot 10^{-2}$	1.014
9	656	78	21.5	$1.52 \cdot 10^8$	1.898	$7.610 \cdot 10^{-2}$	1.002
10	842	97	23.0	$6.68 \cdot 10^8$	1.900	$6.125 \cdot 10^{-2}$	0.991
11	1062	122	23.2	$1.57 \cdot 10^9$	1.901	$5.019 \cdot 10^{-2}$	1.003
12	1352	153	23.9	$8.09 \cdot 10^9$	1.902	$4.000 \cdot 10^{-2}$	1.007
13	1708	189	25.0	$2.69 \cdot 10^{10}$	1.903	$3.281 \cdot 10^{-2}$	1.001
14	2154	229	26.8	$1.51 \cdot 10^{11}$	1.903	$2.785 \cdot 10^{-2}$	1.002

Varying Young's modulus. The domain is defined by $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 0.1$, completely fixed at both ends, and subjected to a surface traction $\boldsymbol{g} = (0, -1)$ at $x_2 = 0.1$. The material parameters are

$$E = \begin{cases} E_0 & x_1 \leq 0.5, \\ \alpha E_0 & \text{otherwise,} \end{cases}, \quad \text{for } E_0 = 100 \text{ and } \alpha = 10,$$

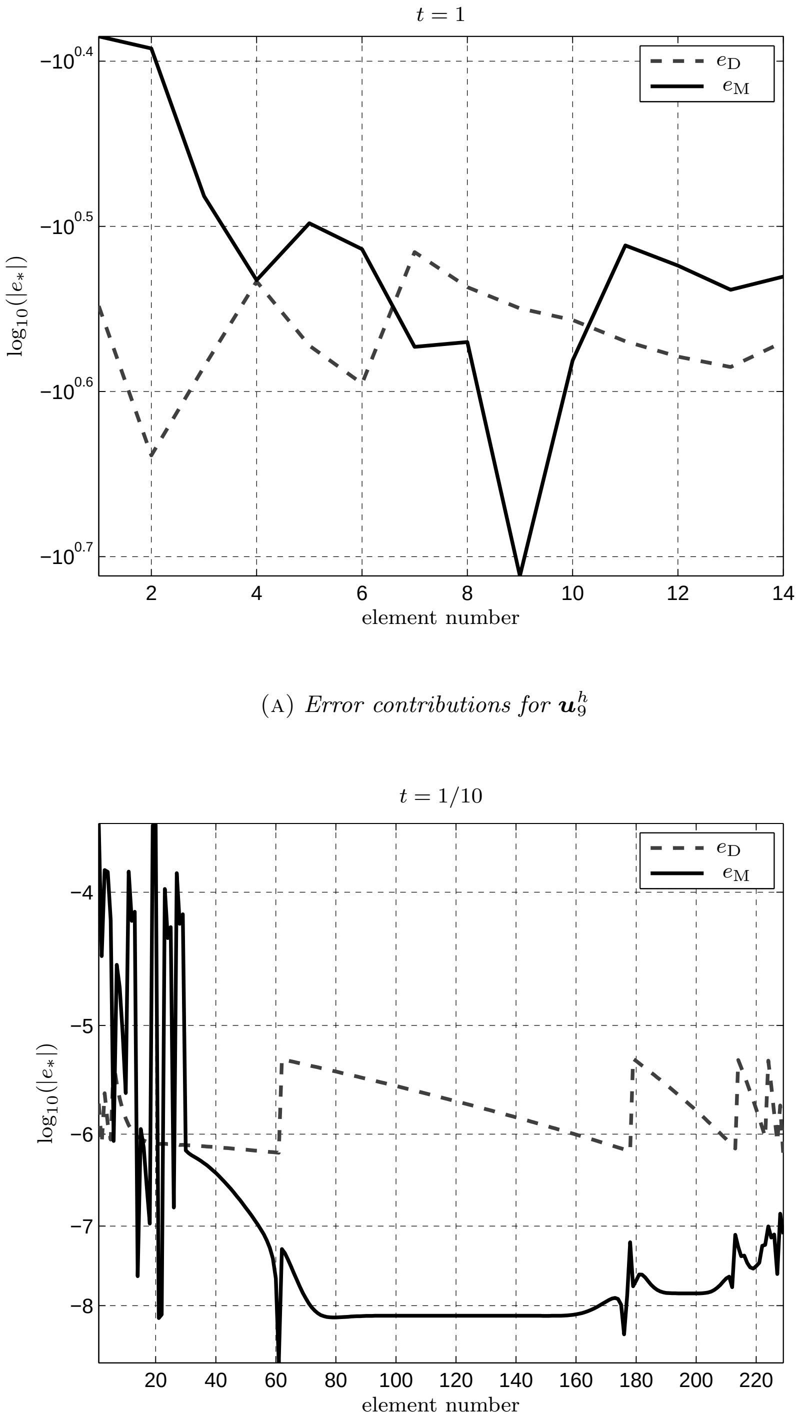


(A) *FE-solution* $\boldsymbol{u}_9^h$



(B) *FE-solution* $\boldsymbol{u}_{14}^h$

FIGURE 5: *FE-solutions for cantilever beams of varying thicknesses*



(A) *Error contributions for* $\boldsymbol{u}_9^h$

(B) *Error contributions for* $\boldsymbol{u}_{14}^h$

Figure 6: *Local error contributions at the last refinement levels*

whereas $\nu = 0$ (no lateral contraction), which then reduces $(\sigma_{11}, \sigma_{22}, \sigma_{12}) = E(\epsilon_{11}, \epsilon_{22}, \frac{1}{2}\epsilon_{12})$, without any coupling (the constitutive matrix $\boldsymbol{D}$ becomes diagonal). The internal energy was estimated $\|\tilde{\boldsymbol{u}}\|_a^2 \approx 0.069682$ with respect to $\mathcal{T}_r$.

The problem was solved for the initial configuration $N_{\mathrm{el}} = 100$, $q_i = 3$ and $r = 0.15$, reaching an accuracy within 0.01 % at the 9:th refinement level, much improved over $\|\boldsymbol{u}^h\|_a^2 \lessapprox 0.068573$, obtained by using bilinear elements. Figure 7 shows a comparison between different refinement strategies (for hq, h and q), where solid lines indicate adaptive updates, and dashed ones mark uniform refinement (legend with subscript u). At the outset the problem is dominated by the discretization error (the q-methods hardly reduce the error), but subsequent introduction of additional elements makes the model error significant. The adaptive hq-strategy is the most accurate, converging with the same order as the corresponding uniform method. The model refinements were concentrated at large variations in $\boldsymbol{\sigma} : \boldsymbol{\varepsilon}$, located about the center and at the clamped ends, visualized in Figures 9(a) and 9(b), using the norm

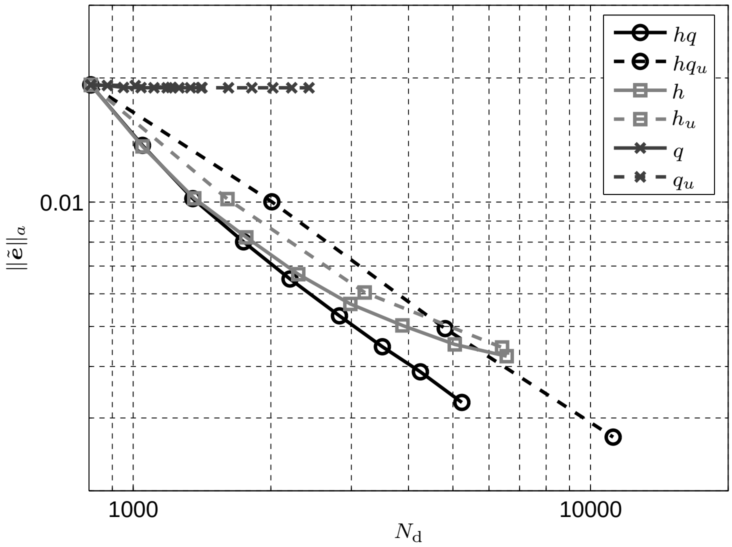


FIGURE 7: *Comparing refinement strategies*

$$\|\boldsymbol{A}\| = (\boldsymbol{A} : \boldsymbol{A})^{1/2},$$

for the stress and strain tensors. In Figure 8(b) we note that large local model errors typically follow the same pattern, just as the error is small in regions without model refinements. The adaptive algorithm manages, more or less, to have local discretization errors of the same order. The effectivity index remained both stable and accurate (according to data in Table 3); the triangle inequality approximately overestimated e_{T} by a factor $C \approx 1.1$-1.4.

Figure 8(a) includes the graph of the analytical solution to the Bernoulli beam equation (centered in light gray)—alongside (2) the Bernoulli beam theory makes a simplified deformation relation, cf. Section 3. As the thickness of the domain decrease, in the current setting ($\nu = 0$), the potential energy (8) approaches that of the beam

$$F(u) = \frac{I}{2} \int_0^1 E(x) \left(u''(x)\right)^2 \, \mathrm{d}x - f \int_0^1 u(x) \, \mathrm{d}x,$$

Table 3: *Completely fixed console*

#iter	N_d	N_el	%(model)	cond($\mathbf{S}$)	$\Vert\boldsymbol{u}^h\Vert_a^2$	$\Vert\tilde{\boldsymbol{e}}\Vert_a$	θ
1	808	100	20.0	$5.33 \cdot 10^6$	0.069312	$1.642 \cdot 10^{-2}$	0.940
2	1048	129	20.3	$1.24 \cdot 10^7$	0.069493	$1.176 \cdot 10^{-2}$	1.024
3	1350	164	20.9	$3.25 \cdot 10^7$	0.069578	$8.663 \cdot 10^{-3}$	0.997
4	1744	210	21.3	$3.60 \cdot 10^7$	0.069617	$6.772 \cdot 10^{-3}$	1.017
5	2202	262	21.9	$5.33 \cdot 10^7$	0.069639	$5.460 \cdot 10^{-3}$	1.018
6	2826	323	23.6	$1.56 \cdot 10^8$	0.069654	$4.458 \cdot 10^{-3}$	1.064
7	3506	389	24.5	$2.65 \cdot 10^8$	0.069662	$3.708 \cdot 10^{-3}$	1.049
8	4246	466	25.5	$1.20 \cdot 10^9$	0.069667	$3.225 \cdot 10^{-3}$	1.077
9	5232	559	26.7	$3.98 \cdot 10^9$	0.069671	$2.704 \cdot 10^{-3}$	1.094

given by [10, Equation 15-74]. The exact potential is

$$F(u) = -\frac{(1+\alpha)\big(1+\alpha(254+\alpha)\big)}{3840E_0\alpha\big(1+\alpha(14+\alpha)\big)} \cdot \frac{f^2}{t^3} \approx -3.139155 \cdot 10^{-2},$$

and by a comparison, listed in Table 4, we conclude

$$\lim_{t\to 0^+} F(\boldsymbol{u}) = F(u),$$

implying that the suggested model hierarchy, in this sense, converges towards the Bernoulli beam (or rather the other way around). In Section 3 we showed equivalence for the limiting case with respect to a restriction of the simplest model.

Remark. The potential energy $F(\boldsymbol{u})$ was preserved by scaling of $\boldsymbol{g} = (0,\ f)$ (note that F is proportional to f^2/t^3); $F(\boldsymbol{u})$ was then approximated over adaptively refined dense meshes.

Table 4: *Comparing potentials*

t	$F(\boldsymbol{u})$	$F(\boldsymbol{u})/F(u)$
1/10	−0.0348408	1.1099
1/20	−0.0322466	1.0272
1/40	−0.0316050	1.0068
1/80	−0.0314454	1.0017
1/160	−0.0314038	1.0004
1/320	−0.0313959	1.0001

6 Conclusions

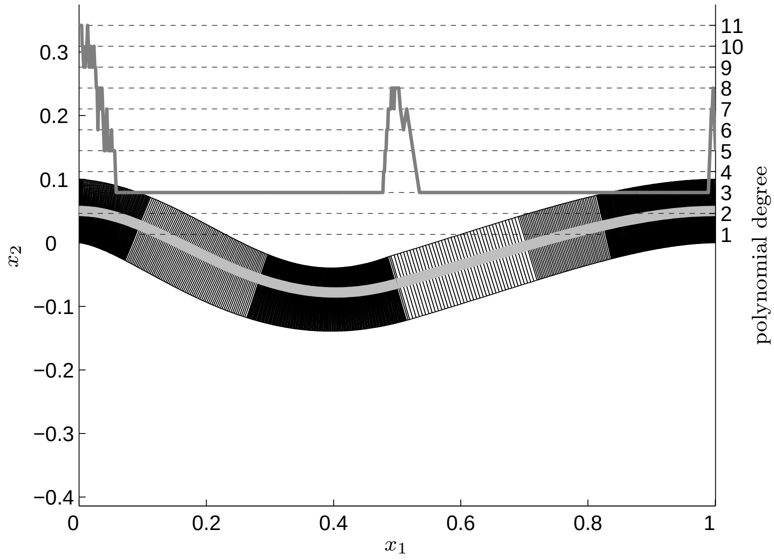
We summarize the results of the paper in the following list:

- The semi-discrete energy error estimate (30) provides a basis for an adaptive procedure, which, in the light of the examples in Section 5.4, seems not only reliable, but capable of sharp error control (which we have to admit is not obvious).

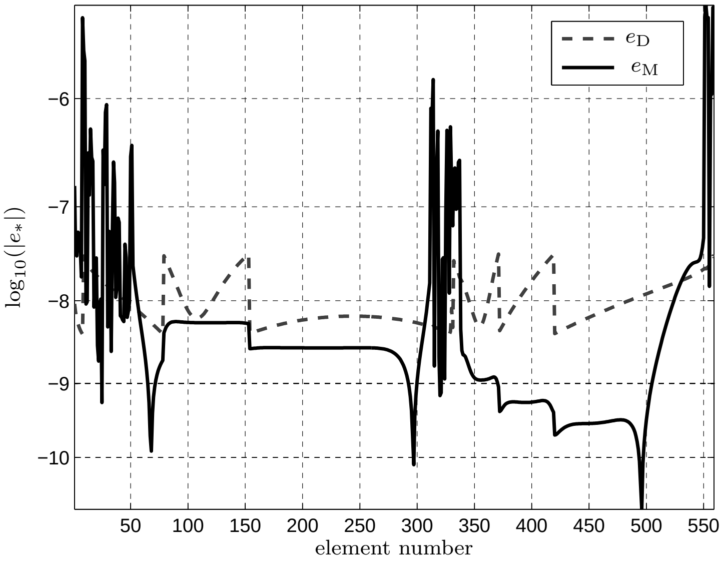
- The local error indicators (40) and (41) concentrated the updates of mesh and model to large variations in stresses and strains, i.e., at locations where we expect a predominant error.

- The accuracy of the FE-solutions were generally high, even when the physical domain had $L/t = 1$ (unit square), but we stress that this is not the typical thin domain setting. Refining the model adaptively was necessary to maintain efficiency; otherwise simpler models degenerate as $e_{\mathrm{D}} \rightarrow 0$.

- The suggested model hierarchy would be a natural extension to a larger hierarchy, by bridging the Bernoulli and Timoshenko beam theories with the (full) elasticity theory.

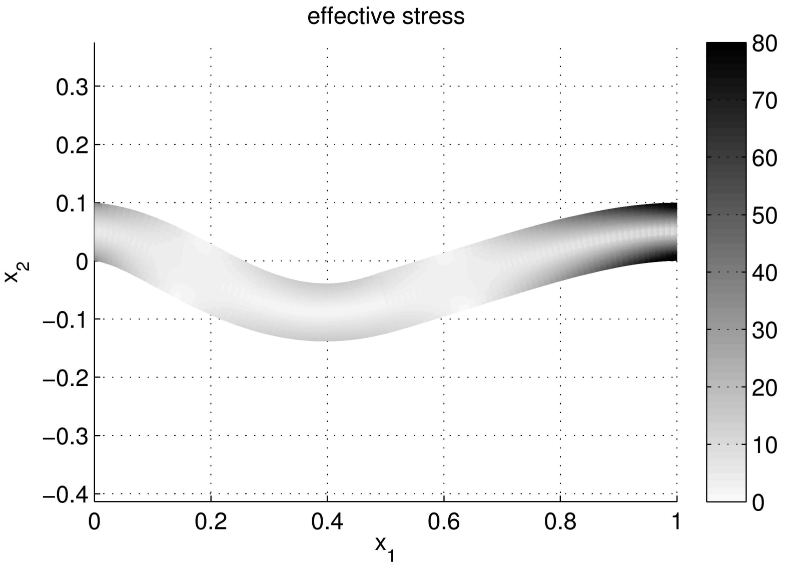


(A) *FE-solution* $\boldsymbol{u}_9^h$

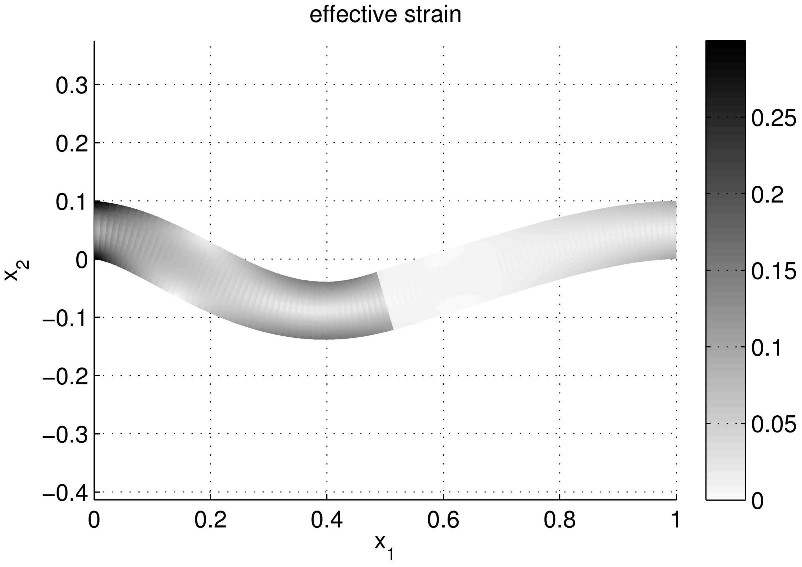


(B) *Local error contributions at last refinement level* $j = 9$

FIGURE 8: *Completely fixed console*



(A) *Large stresses located about $x_1 = 1$*



(B) *Large strains for $0 \leq x_1 \leq 1/2$ (where E is smaller)*

FIGURE 9: *Stresses and strains in the material*

References

[1] M. Ainsworth, J. Z. Zhu, A. W. Craig, and O. C. Zienkiewicz, *Analysis of the Zienkiewicz-Zhu a posteriori error estimator in the finite element method*, International Journal for Numerical Methods in Engineering **28** (1989), 2161–2174.

[2] I. Babuška, I. Lee, and C. Schwab, *On the a posteriori estimation on the modeling error for the heat conduction in a plate and its use for adaptive hierarchial modeling*, Applied Numerical Mathematics **14** (1994), 5–21.

[3] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam, *Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate*, ACM Transactions on Mathematical Software **35** (2008), no. 3.

[4] Timothy A. Davis and William W. Hager, *Dynamic Supernodes in Sparse Cholesky Update/Downdate and Triangular Solvers*, ACM Transactions on Mathematical Software **35** (2008), no. 4.

[5] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational differential equations*, Studentlitteratur, 1996.

[6] A. Ern and J-L. Guermond, *Theory and practice of finite elements*, Springer-Verlag, 2004.

[7] C. Johnson and P. Hansbo, *Adaptive finite element methods in computational mechanics*, Computer Methods in Applied Mechanics and Engineering **101** (1992), 143–181.

[8] T. Kaneko, *On Timoshenko's correction for shear in vibrating beams*, Journal of Physics D: Applied Physics **8** (1975), 1927–1936.

[9] S. Larsson and V. Thomeé, *Partial Differential Equations with Numerical Methods*, Springer-Verlag, 2003.

[10] H. Lundh, *Grundläggande hållfasthetslära*, 3:rd ed., Kungliga tekniska högskolan, 2000.

[11] N. S. Ottosen and H. Petersson, *Introduction to the finite element method*, Prentice Hall Europe, 1992.

[12] W. Weaver (JR), S. P. Timoshenko, and D.H. Young, *Vibration problems in engineering*, 5:th ed., John Wiley and Sons, 1990.

Paper II

An Adaptive Finite Element Method for Second Order Plate Theory

An adaptive finite element method for second order plate theory

Peter Hansbo[1], David Heintz[1] and Mats G. Larson[2]

[1] *Department of Mathematics, Chalmers University of Technology and University of Gothenburg, SE–412 96 Göteborg, Sweden*

[2] *Department of Mathematics and Mathematical Statistics, Umeå University, SE–901 87 Umeå, Sweden*

Abstract

We present a discontinuous finite element method for the Kirchhoff plate model with membrane stresses. The method is based on P^2-approximations on simplices for the out-of-plane deformations, using C^0-continuous approximations. We derive *a posteriori* error estimates for linear functionals of the error and give some numerical examples.

1 INTRODUCTION

Kirchhoff plate deformation analysis is often carried out without taking into account the effect of in-plane stresses. However, if these are significant or if a stability analysis is required, the fourth order plate equations must be supplemented with a second order term. In such a case, standard finite element methods for plates may not be well suited for the problem; an example of this is the well known Morley plate element which is known not to converge for the Laplace equation, cf. [7]. In this paper we instead use the continuous/discontinuous Galerkin method first proposed by Engel et al. [2], and proposed (using a different but related formulation) for second order theory by Wells and Dung [8]. In this paper, we focus on using continuous piecewise second degree polynomials (with discontinuous derivatives) for the plate and constant–strain elements for the in-plane deformations. We present an *a posteriori* error analysis including the modeling error in the stresses. The total error is thus composed of two parts: the plate discretization error and the error emanating from the modeling of the stresses. In our case the stress error is also a discretization error, which allows for *a posteriori* control of the total error. However, the separation of the errors allows for the use, e.g., of measured stresses, in which case some other control of the error in stresses must be used if a reliable estimate is to be obtained.

2 THE PLATE MODEL

Consider a domain $\Omega \subset \mathbb{R}^2$, for simplicity assumed to be a convex polygon, whose boundary $\partial\Omega$ has an outward pointing normal $\boldsymbol{n}$. The plate thickness in the out-of-plane x_3-direction is denoted by t, and the deflection in the x_3-direction by u. The differential equations describing the Kirchhoff plate model with membrane stresses can be derived from minimization of the sum of the bending energy and the potential of the surface and in-plane stress load,

$$\begin{aligned}\mathcal{U}_z(u) = \frac{D}{2}\int_\Omega \left(\left(\frac{\partial^2 u}{\partial x_1^2}+\frac{\partial^2 u}{\partial x_2^2}\right)^2 - 2\,(1-\nu)\left(\frac{\partial^2 u}{\partial x_1^2}\frac{\partial^2 u}{\partial x_2^2} - \left(\frac{\partial^2 u}{\partial x_1 \partial x_2}\right)^2\right)\right) d\Omega \\ + \frac{t}{2}\int_\Omega \left(\sigma_{11}\left(\frac{\partial u}{\partial x_1}\right)^2 + \sigma_{22}\left(\frac{\partial u}{\partial x_2}\right)^2 + 2\sigma_{12}\frac{\partial u}{\partial x_1}\frac{\partial u}{\partial x_2}\right) d\Omega - \int_\Omega g\,u\,d\Omega\end{aligned} \tag{1}$$

where g is the transverse surface load,

$$\boldsymbol{\sigma} = \begin{bmatrix}\sigma_{11} & \sigma_{12}\\ \sigma_{12} & \sigma_{22}\end{bmatrix}$$

is the in-plane stress tensor, given by the solution to the plane stress problem

$$\begin{aligned} -\nabla\cdot\boldsymbol{\sigma} = \boldsymbol{f} \quad \text{in } \Omega,\\ \boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{u}) := 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) + \lambda\,\nabla\cdot\boldsymbol{u}\,\mathbf{1},\end{aligned}$$

where $\lambda = E\nu/(1-\nu^2)$ and $\mu = E/(2\,(1+\nu))$; $D = Et^3/(12(1-\nu^2))$ is the flexural rigidity of the plate, and E and ν are Young's modulus and Poisson's ratio, respectively.

In the following, we shall assume that if the stresses are compressive, they are small enough to ensure coercivity of the full problem; the case of large compressive stresses occurs in the analysis of plate buckling, where the potential energy functional switches from being convex to being concave, resulting in instability.

By introducing the curvature tensor

$$\boldsymbol{\kappa}(u) := \begin{bmatrix} \dfrac{\partial^2 u}{\partial x_1^2} & \dfrac{\partial^2 u}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 u}{\partial x_1 \partial x_2} & \dfrac{\partial^2 u}{\partial x_2^2}\end{bmatrix}$$

the Lamé–type parameters $\mu_{\text{P}} := D(1-\nu)/2$, and $\lambda_{\text{P}} := D\nu$, and the moment tensor

$$\boldsymbol{M}(u) := 2\mu_{\text{P}}\boldsymbol{\kappa}(u) + \lambda_{\text{P}}\Delta u\,\mathbf{1}$$

where $\mathbf{1}$ is the identity tensor, we may rewrite (1) as

$$\mathcal{U}(u) = \frac{1}{2}\int_\Omega \left(\boldsymbol{M}(u) : \boldsymbol{\kappa}(u) + t\,(\boldsymbol{\sigma}\nabla u)\cdot\nabla u - 2gu\right)\,d\Omega \tag{2}$$

where $\boldsymbol{M} : \boldsymbol{\kappa} = \sum_{ij} M_{ij}\kappa_{ij}$ denotes the contraction of two tensors. Minimization of (2) yields the plate equation with second order effects

$$\sum_{ij}\left(\frac{\partial^2 M_{ij}(u)}{\partial x_i \partial x_j} - \frac{\partial}{\partial x_i}\left(t\sigma_{ij}\frac{\partial u}{\partial x_j}\right)\right) = g \quad \text{in } \Omega, \tag{3}$$

supplied with the appropriate boundary conditions.

For simplicity, we shall restrict the presentation to the case of clamped boundary conditions: $\boldsymbol{n} \cdot \nabla u = 0$ and $u = 0$ on $\partial\Omega$. We shall also assume that the constitutive parameters and the thickness are constant.

3 THE FINITE ELEMENT METHOD

3.1 Variational formulation

The virtual work equation related to (3) is the following variational statement: find

$$u \in V := \{v : \ v \in H^2(\Omega), v = 0 \text{ on } \partial\Omega, \ \boldsymbol{n} \cdot \nabla v = 0 \text{ on } \partial\Omega\}$$

such that

$$a_{\mathrm{P}}(u, v) + b(\boldsymbol{\sigma}; u, v) = (g, v)$$

for all $v \in V$, where

$$a_{\mathrm{P}}(u, v) := \int_\Omega \boldsymbol{M}(u) : \boldsymbol{\kappa}(v)\, d\Omega,$$

$$b(\boldsymbol{\sigma}; u, v) := t \int_\Omega (\boldsymbol{\sigma} \nabla u) \cdot \nabla v \, d\Omega.$$

To obtain $\boldsymbol{\sigma}$, we also have to solve the plane stress problem of finding $\boldsymbol{u} \in \boldsymbol{W} := [H_0^1(\Omega)]^2$ such that

$$(\boldsymbol{\sigma}(\boldsymbol{u}), \boldsymbol{\varepsilon}(\boldsymbol{v})) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{W}, \tag{4}$$

where

$$(\boldsymbol{\sigma}(\boldsymbol{u}), \boldsymbol{\varepsilon}(\boldsymbol{v})) := \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, d\Omega$$

In the following, we shall also use the L_2 scalar product notation

$$(u, v)_E := \int_E u\, v\, ds, \quad (\boldsymbol{M}, \boldsymbol{\tau})_T := \int_T \boldsymbol{M} : \boldsymbol{\tau}\, dxdy, \quad \text{etc.}$$

If the subscript is omitted, the product is taken over Ω.

3.2 Finite element method

We consider a subdivision $\mathcal{T} = \{T\}$ of Ω into a geometrically conforming, quasiuniform, simplicial finite element mesh. Denote by h_T the diameter of element T and by $h = \max_{T \in \mathcal{T}} h_T$ the global mesh size parameter. We shall use continuous, piecewise polynomial, approximations of the transverse displacement:

$$V_h := \{v \in H_0^1(\Omega) \cap C^0(\Omega) : \ v|_T \in P^2(T), \forall T \in \mathcal{T}\}.$$

To define our methods we introduce the set of edges in the mesh, $\mathcal{E} = \{E\}$, and we split $\mathcal{E}$ into two disjoint subsets

$$\mathcal{E} = \mathcal{E}_I \cup \mathcal{E}_B,$$

where $\mathcal{E}_I$ is the set of edges in the interior of Ω and $\mathcal{E}_B$ is the set of edges on the boundary. Further, with each edge we associate a fixed unit normal $\boldsymbol{n}$ such that for edges on the boundary

$\boldsymbol{n}$ is the exterior unit normal. We denote the jump of a function $\boldsymbol{v} \in \boldsymbol{\Gamma}_h$ at an edge E by $[\boldsymbol{v}] = \boldsymbol{v}^+ - \boldsymbol{v}^-$ for $E \in \mathcal{E}_I$ and $[\boldsymbol{v}] = \boldsymbol{v}^+$ for $E \in \mathcal{E}_B$, and the average $\langle \boldsymbol{v} \rangle = (\boldsymbol{v}^+ + \boldsymbol{v}^-)/2$ for $E \in \mathcal{E}_I$ and $\langle \boldsymbol{v} \rangle = \boldsymbol{v}^+$ for $E \in \mathcal{E}_B$, where $\boldsymbol{v}^\pm = \lim_{\epsilon \downarrow 0} \boldsymbol{v}(\boldsymbol{x} \mp \epsilon\, \boldsymbol{n})$ with $\boldsymbol{x} \in E$.

Our method can now be formulated as follows: find $u_h \in V_h$ such that

$$a_{\mathrm{P}}^h(u_h, v) + b(\boldsymbol{\sigma}; u_h, v) = (g, v) \tag{5}$$

for all $v \in V_h$. In (5) the bilinear form $a_{\mathrm{P}}^h(\cdot,\cdot)$ is defined on V_h as follows

$$\begin{aligned} a_{\mathrm{P}}^h(u, v) = & \sum_{T \in \mathcal{T}} (\boldsymbol{M}(u), \boldsymbol{\kappa}(v))_T \\ & - \sum_{E \in \mathcal{E}_I \cup \mathcal{E}_B} (\langle \boldsymbol{n} \cdot \boldsymbol{M}(u) \cdot \boldsymbol{n} \rangle, [\boldsymbol{n} \cdot \nabla v])_E \\ & - \sum_{E \in \mathcal{E}_I \cup \mathcal{E}_B} (\langle \boldsymbol{n} \cdot \boldsymbol{M}(v) \cdot \boldsymbol{n} \rangle, [\boldsymbol{n} \cdot \nabla u])_E \\ & + (2\mu + 2\lambda)\, \gamma_0 \sum_{E \in \mathcal{E}_I \cup \mathcal{E}_B} h_E^{-1} ([\boldsymbol{n} \cdot \nabla u], [\boldsymbol{n} \cdot \nabla v])_E \end{aligned}$$

Here γ_0 is a constant whose role is to enforce coercivity of the form $a_{\mathrm{P}}^h(\cdot,\cdot)$, and h_E is defined by

$$h_E = \left(|T^+| + |T^-| \right) / (2\,|E|) \quad \text{for } E = \partial T^+ \cap \partial T^-, \tag{6}$$

with $|T|$ the area of T, on each edge. Precise values for the choice of γ_0 are given in [5].

Using Green's formula, we readily establish the following Lemma.

Lemma 1 *The method* (5) *is consistent in the sense that*

$$a_{\mathrm{P}}^h(u - u_h, v) + b(\boldsymbol{\sigma}; u - u_h, v) = 0 \tag{7}$$

for all $v \in V_h$.

Now, the $\boldsymbol{\sigma}$ in (5) is the exact stress solution to the problem (4) which we cannot solve exactly in general. Instead we introduce the discrete problem of finding $\boldsymbol{u}_h \in \boldsymbol{W}_h$, where

$$\boldsymbol{W}_h := \{\boldsymbol{v} \in \boldsymbol{W} \cap [C^0(\Omega)]^2 : \ \boldsymbol{v}|_T \in [P^1(T)]^2, \forall T \in \mathcal{T}\},$$

such that

$$(\boldsymbol{\sigma}(\boldsymbol{u}_h), \boldsymbol{\varepsilon}(\boldsymbol{v})) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{W}_h. \tag{8}$$

Using the notation $\boldsymbol{\sigma}_h := \boldsymbol{\sigma}(\boldsymbol{u}_h)$, our discrete plate problem becomes: find $u_h \in V_h$ such that

$$a_{\mathrm{P}}^h(u_h, v) + b(\boldsymbol{\sigma}_h; u_h, v) = (g, v) \tag{9}$$

for all $v \in V_h$, and the orthogonality relation (7) is weakened to

$$a_{\mathrm{P}}^h(u - u_h, v) + b(\boldsymbol{\sigma}; u, v) - b(\boldsymbol{\sigma}_h; u_h, v) = 0, \quad \forall v \in V_h. \tag{10}$$

3.3 An approach to *a posteriori* error estimation

Since the error in the discrete plate problem (9) does not couple to the plane stress problem (8), we shall here follow Larson and Bengzon [3] and view the fact that we must replace $\boldsymbol{\sigma}$ by $\boldsymbol{\sigma}_h$ as a *modeling error.*

Following Becker and Rannacher [1], we are interested in controlling a linear functional $L(u - u_h)$ of the error (where L is to be specified), and to this end we introduce the dual problem of finding z such that

$$\sum_{ij} \left(\frac{\partial^2 M_{ij}(z)}{\partial x_i \partial x_j} - \frac{\partial}{\partial x_i} \left(t\sigma_{ij}^h \frac{\partial z}{\partial x_j} \right) \right) = J \quad \text{in } \Omega, \tag{11}$$

$z = 0$ and $\boldsymbol{n} \cdot \nabla z = 0$ on $\partial\Omega$. Here J is the Riesz representer of L, $L(v) = (J, v)$. Then, since $[\boldsymbol{n} \cdot \nabla z] = 0$ on the edges,

$$\begin{aligned} (J, e) = L(u - u_h) &= a_{\mathrm{P}}^h(z, u - u_h) + b(\boldsymbol{\sigma}_h; z, u - u_h) \\ &= a_{\mathrm{P}}^h(u - u_h, z) + b(\boldsymbol{\sigma}_h; u - u_h, z) + b(\boldsymbol{\sigma}; u, z) - b(\boldsymbol{\sigma}; u, z) \\ &= (g, z - \pi_h z) - a_{\mathrm{P}}^h(u_h, z - \pi_h z) - b(\boldsymbol{\sigma}_h; u_h, z - \pi_h z) \\ &\quad + b(\boldsymbol{\sigma}_h - \boldsymbol{\sigma}; u, z), \end{aligned} \tag{12}$$

where we used (10) in the last step.

Since $\boldsymbol{\sigma} - \boldsymbol{\sigma}_h = \boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_h)$, we can now introduce a second functional, related to the modelling error.

$$L_{\mathrm{mod}}(\boldsymbol{u} - \boldsymbol{u}_h) := -b(\boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_h); u, z)$$

and the dual problem of finding $\boldsymbol{z} \in \boldsymbol{W}$ such that

$$(\boldsymbol{\sigma}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\boldsymbol{z})) = L_{\mathrm{mod}}(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \boldsymbol{W}.$$

Thus, the functional L_{mod} of the error can be evaluated as

$$\begin{aligned} L_{\mathrm{mod}}(\boldsymbol{u} - \boldsymbol{u}_h) &= (\boldsymbol{\sigma}(\boldsymbol{u} - \boldsymbol{u}_h), \boldsymbol{\varepsilon}(\boldsymbol{z})) \\ &= (\boldsymbol{f}, \boldsymbol{z} - \pi_h \boldsymbol{z}) - (\boldsymbol{\sigma}_h, \boldsymbol{\varepsilon}(\boldsymbol{z} - \pi_h \boldsymbol{z})), \end{aligned}$$

where we used that

$$(\boldsymbol{\sigma}(\boldsymbol{u}) - \boldsymbol{\sigma}(\boldsymbol{u}_h), \boldsymbol{\varepsilon}(\boldsymbol{v})) = 0, \quad \forall \boldsymbol{v} \in \boldsymbol{W}_h.$$

In conclusion, we have that

$$\begin{aligned} L(u - u_h) = {} & (g, z - \pi_h z) - a_{\mathrm{P}}^h(u_h, z - \pi_h z) - b(\boldsymbol{\sigma}_h; u_h, z - \pi_h z) \\ & + (\boldsymbol{f}, \boldsymbol{z} - \pi_h \boldsymbol{z}) - (\boldsymbol{\sigma}_h, \boldsymbol{\varepsilon}(\boldsymbol{z} - \pi_h \boldsymbol{z})). \end{aligned} \tag{13}$$

In (13), the error contribution has thus been split between discretization errors for the plate (the terms involving $z - \pi_h z$) and the model error (in this case discretization error) in the stresses (the terms involving $\boldsymbol{z} - \pi_h \boldsymbol{z}$). The continuous stresses do not enter directly into the discretization error for the plate; the effect of using approximate stresses has been separated out. In case the stresses have been obtained in some other way, e.g., by measurement, (12) can instead be used together with some direct estimate of the stress error.

4 IMPLEMENTATION

We seek to establish a practical adaptive scheme, such that a linear functional $L(\cdot)$ of the error, $e = u - u_h$, is bounded. To that end, duality-based techniques are employed, which include solving enriched problems—the corresponding dual problems—to distribute degrees of freedom in an efficient manner. In our numerical examples, the datum of the dual plate problem (11), i.e., the target quantity, is chosen as a Dirac delta function

$$J := \delta(x_1 - \bar{x}_1, x_2 - \bar{x}_2), \tag{14}$$

where $(\bar{x}_1, \bar{x}_2)$ is the node of maximum plate deflection (whose location is expected to converge rather quickly as the mesh size $h \to 0$). As opposed to estimating the error with respect to global energy norms or global L_2-norms, (14) exemplifies a typical local quantity, which often is more appropriate in applications.

The linear functional (13) was written as a sum, $L(e) = \eta_\mathrm{P} + \eta_\mathrm{M}$, where

$$\eta_\mathrm{P} = (g, z - \pi_h z) - a^h_\mathrm{P}(u_h, z - \pi_h z) - b(\boldsymbol{\sigma}_h; u_h, z - \pi_h z), \tag{15}$$

$$\eta_\mathrm{M} = (\boldsymbol{f}, \boldsymbol{z} - \pi_h \boldsymbol{z}) - (\boldsymbol{\sigma}_h, \boldsymbol{\varepsilon}(\boldsymbol{z} - \pi_h \boldsymbol{z})), \tag{16}$$

represent the effects of the discretization and model errors. This provides alternatives for the discretization of the domain Ω:

(i) use a common partition $\mathcal{T}$ (quadratic and linear triangles for the plate and plane stress problems coincide geometrically);

(ii) use disparate triangulations $\mathcal{T}_\mathrm{P}$ and $\mathcal{T}_\mathrm{M}$.

The first option associates each element $T_i \in \mathcal{T}$ with an error indicator

$$\eta^i = |\eta^i_\mathrm{P}| + |\eta^i_\mathrm{M}|, \quad i = 1, 2, \ldots, N, \tag{17}$$

where N is the number of elements of $\mathcal{T}$. This choice makes for an easier implementation, but the information from (15) and (16) is not thoroughly exploited, since we resort to sums of local errors contributions. The more flexible approach—in keeping with model adaptivity—would be a direct comparison, made possible by disparate triangulations. However, this necessitates making searches when solving the *point-in-triangle* problem (see, e.g., [9, Algorithm 4]), usually preceded by constructing a suitable search structure, a so-called *binary tree* or *quad tree*; we refer to [10] for further details. In the present context, the need for such searches typically arise when basis functions are evaluated in abscissas (query points), situated on different elements (targets) between $\mathcal{T}_\mathrm{P}$ and $\mathcal{T}_\mathrm{M}$.

Remark 1 *Preprocessing Ω into a search structure comes at a cost, and although intended to ease the effort, it still takes time to perform searches (the procedure ought to be parallelized since individual queries are independent). The winnings are increased flexibility in distributing the degrees of freedom, thereby yielding smaller matrix problems achieving the same level of accuracy.*

In the sequel the discussion focuses on using disparate meshes (the other case being a reduced analog), where $\mathcal{T}_\mathrm{P}$ and $\mathcal{T}_\mathrm{M}$ are assumed to consist of N_P and N_M elements, respectively. Hence, instead of (17), consider the concatenated $(N_\mathrm{P} + N_\mathrm{M})$-tuple

$$\eta = \left(\eta^1_\mathrm{P}, \ldots, \eta^{N_\mathrm{P}}_\mathrm{P}, \eta^1_\mathrm{M}, \ldots, \eta^{N_\mathrm{M}}_\mathrm{M}\right), \tag{18}$$

from which we select a fixed ratio r of the largest absolute local error indicators, regardless of the error source, and mark the corresponding elements for refinement. The primal meshes are updated via a Bänsch-algorithm (longest edge bisection) leaving no hanging nodes.

In order to determine (18) we need to approximate exact solutions. Therefore we introduce two auxiliary problems: Find $\tilde{u} \in V_h^*$ and $\tilde{z} \in V_h^*$ such that

$$\begin{aligned} a_{\mathrm{P}}^h(\tilde{u}, v) + b(\boldsymbol{\sigma}_h; \tilde{u}, v) &= (g, v), \quad \forall v \in V_h^*, \\ a_{\mathrm{P}}^h(v, \tilde{z}) + b(\boldsymbol{\sigma}_h; v, \tilde{z}) &= L(v), \quad \forall v \in V_h^*, \end{aligned}$$

which assemble the same stiffness matrix, only deviating by the RHS-data (thus passed together as arguments to a direct sparse Cholesky solver). The solutions are then used as input to the discretized dual plane stress problem: Find $\tilde{\boldsymbol{z}} \in \boldsymbol{W}_h^*$ such that

$$(\boldsymbol{\sigma}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\tilde{\boldsymbol{z}})) = -b(\boldsymbol{\sigma}(\boldsymbol{v}); \tilde{u}, \tilde{z}), \quad \forall \boldsymbol{v} \in \boldsymbol{W}_h^*.$$

The enriched function spaces $V_h^* \supset V_h$ and $\boldsymbol{W}_h^* \supset \boldsymbol{W}_h$ are constructed by subjecting the primal partitions to regular subdivision, leading to the refined triangulations $\tilde{\mathcal{T}}_{\mathrm{P}}$ and $\tilde{\mathcal{T}}_{\mathrm{M}}$, respectively. When calculating local error indicators by (15) and (16), $\pi_h v : V_h^* \to V_h$ and $\pi_h \boldsymbol{v} : \boldsymbol{W}_h^* \to \boldsymbol{W}_h$ were taken, for simplicity, to be nodal interpolation operators.

The stopping criterion of the adaptive algorithm is imposed on the relative error

$$e_{\mathrm{rel}} := \left| \frac{L(\tilde{e})}{\tilde{u}(\bar{x}_1, \bar{x}_2)} \right| \leq \mathrm{TOL},$$

where

$$L(\tilde{e}) = L_{\mathrm{P}} + L_{\mathrm{M}}, \quad L_{\mathrm{P}} = \sum_{j=1}^{N_{\mathrm{P}}} \eta_{\mathrm{P}}^j, \; L_{\mathrm{M}} = \sum_{k=1}^{N_{\mathrm{M}}} \eta_{\mathrm{M}}^k,$$

and TOL is a prescribed tolerance. Furthermore, let

$$|L_{\mathrm{P}}| = \sum_{j=1}^{N_{\mathrm{P}}} |\eta_{\mathrm{P}}^j|, \quad |L_{\mathrm{M}}| = \sum_{k=1}^{N_{\mathrm{M}}} |\eta_{\mathrm{M}}^k|,$$

denote the sums of the absolute local error indicators. The adaptive scheme is summarized in Algorithm 1.

Remark 2 *The bilinear form a_P^h contains a penalty term, controlling the normal derivative, which is scaled by a parameter γ_0. From [5] a lower bound of $\gamma_0 > 3C_I$, where C_I is a constant that depends on the order k of the approximation, ensures its coercivity. For this purpose we set $\gamma_0 = 3/2$ in all test problems, using the asymptotic value of the constant.*

The a posteriori error estimator is evaluated in terms of the effectivity index

$$\theta := \left| \frac{L(\tilde{e})}{L(\hat{e})} \right| = \left| \frac{L(\tilde{e})}{\hat{u}(\bar{x}_1, \bar{x}_2) - u_h(\bar{x}_1, \bar{x}_2)} \right|,$$

where $\hat{u}(\bar{x}_1, \bar{x}_2)$ is a reference solution replacing the unknown exact maximum plate deflection. $\hat{u}$ was obtained by solving the primal problems adaptively on a common dense mesh $\hat{\mathcal{T}}$.

The adaptive procedure is rather time-consuming—in every iteration two dual problems are solved (including the enhanced discrete plate problem). An effort to reduce the computational cost would be to solve the problems with respect to their primal FE-spaces, followed by post-processing to obtain solutions in enriched function spaces, see, e.g., Larsson et al. [6].

Algorithm 1: Adaptive scheme

```
Data: initial mesh $\mathcal{T}^0$, user-specified tolerance TOL
Result: FE-solution $u_h$, total error $L(\tilde{e})$

for $i = 0, 1, \ldots$ do
    solve primal plane stress problem for $\boldsymbol{\sigma}_h$ on $\mathcal{T}_{\mathrm{M}}^i$
    solve primal plate problem for $u_h$ on $\mathcal{T}_{\mathrm{P}}^i$
    construct dual meshes $\tilde{\mathcal{T}}_{\mathrm{P}}^i$ and $\tilde{\mathcal{T}}_{\mathrm{M}}^i$
    solve primal and dual plate problems for $\tilde{u}$ and $\tilde{z}$ on $\tilde{\mathcal{T}}_{\mathrm{P}}^i$
    solve dual plane stress problem for $\tilde{\boldsymbol{z}}$ on $\tilde{\mathcal{T}}_{\mathrm{M}}^i$
    compute local error indicators $\eta_{\mathrm{P}}^j$ and $\eta_{\mathrm{M}}^k$
    if TOL $> e_{\mathrm{rel}}$ then
        refine primal meshes: $\mathcal{T}_{\mathrm{P}}^i \to \mathcal{T}_{\mathrm{P}}^{i+1}$, $\mathcal{T}_{\mathrm{M}}^i \to \mathcal{T}_{\mathrm{M}}^{i+1}$
    else
        break
    end
end
```

5 NUMERICAL EXAMPLES

Algorithm 1 is applied to a set of test problems, which is not meant to be comprehensive, but to exemplify the behavior of the a posteriori error estimator. We shall compare two adaptive strategies: 1) refine a common mesh $\mathcal{T}^i$ (referred to as the CM-method); and 2) refine disparate meshes $\mathcal{T}_{\mathrm{P}}^i$ and $\mathcal{T}_{\mathrm{M}}^i$ (DM-method). Note that

- the transversal and in-plane boundary conditions are different in each setup;
- the load—either $\boldsymbol{f}$ or $\boldsymbol{g}$—causing the stress field $\boldsymbol{\sigma}$ is restricted to maintain the positive-definiteness of the stiffness matrix (typically the loading must not be too large);
- the refinement ratio is set to $r = 0.2$.

Remark 3 *The computational cost—per degree of freedom—associated with the plate problem is larger (than for the plane stress problem), partly due to the symmetry and penalty terms of the a_P^h-functional, alongside the additional b-functional. Consequently, having a relatively small plate problem could be beneficial, and hence relevant results are presented with respect to: 1) the total number of degrees of freedom $D_T = D_P + D_M$; and 2) the number of degrees of freedom for the plate problem D_P.*

5.1 Point Load on Unit Square

Consider the plate $\Omega = [0,1] \times [0,1]$, simply supported (so that $u = 0$ at $\partial\Omega$) and fixed on the left side ($\boldsymbol{u}|_{x_1=0} = \boldsymbol{0}$). The material parameters are $\nu = 0.3$, $E = 100$, and the thickness $t = 0.1$. The plate carries a point load at $\boldsymbol{x}_l = (1/4,\, 3/4)$, whereas in-plane stresses are induced by a surface traction $\boldsymbol{g} = (0,\, -1)$ at $x_2 = 1$.

The plane stress FE-formulation now becomes: Find $\boldsymbol{u}_h \in \boldsymbol{W}_h$ such that

$$(\boldsymbol{\sigma}(\boldsymbol{u}_h), \boldsymbol{v}) = (\boldsymbol{g}, \boldsymbol{v})_{\partial\Omega_{\mathrm{N}}}, \quad \forall \boldsymbol{v} \in \boldsymbol{W}_h,$$

where $\partial\Omega_{\mathrm{N}}$ denotes the Neumann boundary $x_2 = 1$, and the local model error indicator (16) changes accordingly

$$\eta_{\mathrm{M}} = (\boldsymbol{g}, \boldsymbol{z} - \pi_h \boldsymbol{z})_{\partial\Omega_{\mathrm{N}}} - (\boldsymbol{\sigma}_h, \boldsymbol{\varepsilon}(\boldsymbol{z} - \pi_h \boldsymbol{z})).$$

The point load is applied to a single node, and thus the discrete load vector $\mathbf{f}_{\mathrm{P}} = \delta_{ij}$, so that (15) reduces to

$$\eta_{\mathrm{P}} = -a^h_{\mathrm{P}}(u_h, z - \pi_h z) - b(\boldsymbol{\sigma}_h; u_h, z - \pi_h z).$$

The FE-solutions of the plate and plane stress problems are shown in Figures 1 and 2 (the latter uses non-deformed coordinates): the maximum plate deflection (cross-marked) is situated close to $\boldsymbol{x}_i$ (circle-marked), whereas large effective stresses

$$\sigma_{\mathrm{e}} := (\boldsymbol{\sigma} : \boldsymbol{\sigma})^{1/2}$$

appear in the vicinity of the vertices $(0, 0)$, $(0, 1)$.

Consecutive updates of $\mathcal{T}$ were presumably dominated by the discretization error, as implied by Table 1, where the larger and strictly decreasing L_{P} (almost) sums up to the total error $L(\hat{e})$ (disregarding cancellation effects). Refinements are concentrated around $\boldsymbol{x}_l$, as seen in Figure 6, and hence one anticipates $N_{\mathrm{P}} > N_{\mathrm{M}}$, should disparate meshes be used. This is also the case, being confirmed by Table 2 and Figure 14(a), showing the shift towards $\mathcal{T}_{\mathrm{P}}$-refinement as most prominent during the initial iterations. When comparing the two meshes in Figures 7 and 8, note the difference at the large-stress vertices, where consequently $\mathcal{T}_{\mathrm{M}}$ is refined while $\mathcal{T}_{\mathrm{P}}$ is not, following the almost symmetric (along the line $x_2 = 1 - x_1$) refinement pattern of $\mathcal{T}$ instead (further underlining the importance of the discretization error).

The adaptive algorithm is superior to uniform refinement, according to Figures 15(a) and 15(b), where the latter indicates how the DM-method avoids unnecessary model complexity, i.e., results in a coarser mesh $\mathcal{T}_{\mathrm{M}}$, and still satisfies the prescribed relative error $\mathrm{TOL} = 5 \cdot 10^{-3}$.

The effectivity indices are stable—we obtain convergence rates of the same order as the underlying FE-method—and the accuracy is high ($\theta \to 1^+$ in Figure 13(a)).

5.2 An L-shaped Membrane

Let Ω be the polygon with vertices $(0, 0), (c, 0), (c, c), (1, c), (1, 1)$ and $(0, 1)$ for $c = 1/2$, having material parameters $\nu = 0.3$, $E = 100$, and the thickness $t = 0.1$. The plate is subjected to a uniform load $g = 1$, and carries an in-plane point load at the tip $\boldsymbol{x}_l = (1, 1)$, such that $\boldsymbol{f}(\boldsymbol{x}_l) = (0, -4)$. It is simply supported ($u = 0$ at $\partial\Omega$) and fixed on the left side ($\boldsymbol{u}|_{x_1=0} = \mathbf{0}$).

In this setting, with the point load applied in a single node, the discretized load vector $\mathbf{f}_{\mathrm{M}} = \delta_{ij}$, and (16) reduces to

$$\eta_{\mathrm{M}} = -(\boldsymbol{\sigma}_h, \boldsymbol{\varepsilon}(\boldsymbol{z} - \pi_h \boldsymbol{z})),$$

otherwise (15) and the FE-formulations remain unchanged.

The maximum plate deflection (cross-marked in Figure 6) would occur along the symmetry line $x_2 = 1 - x_1$ (dashed), had it not been for large planar stresses at $\boldsymbol{x}_l$, and particularly around the non-convex corner singularity; thereby $\max u_h$ deviates to $(\bar{x}_1, \bar{x}_2) \approx (0.26, 0.51)$.

The influence of the large stresses implies how the model error now becomes more prominent. Tables 3 and 4 concurs that L_{P} and L_{M} are of the same order of magnitude (note that the cancellation effects are greater for the model error)—the DM-method recognizes this and concentrates refinements on $\mathcal{T}_{\mathrm{M}}$ (emphasized by Figure 14(b)). During the adaptive procedure additional elements are foremostly introduced at the interior corner $(1/2, 1/2)$, but $\mathcal{T}_{\mathrm{P}}$ is also resolved around $(\bar{x}_1, \bar{x}_2)$.

Using an adaptive algorithm is favorable, judging by Figure 16, and the solution improves as the complexity of the model increases. The tolerance was $\mathrm{TOL} = 2 \cdot 10^{-2}$.

The total error is slightly underestimated, but the effectivity indices remain fairly stable, and grow slowly (Figure 13(b) shows that $\theta \to 1^-$).

6 CONCLUDING REMARKS

We have presented a C^0 finite element element method for second-order analysis of thin plates. *A posteriori* error estimates for linear functionals of the error have been derived. Our numerical examples, focusing on controlling the error in maximum deflection, show that the estimates give effectivity indices close to unity.

References

[1] Becker R, Rannacher R. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West Journal of Numerical Mathematics* 1996; **4**:237–264.

[2] Engel G, Garikipati K, Hughes TJR, Larson MG, Mazzei L, Taylor RL. Continuous/discontinuous finite element approximations of fourth-order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:3669–3750.

[3] Larson MG, Bengzon F. Adaptive finite element approximation of multiphysics problems. *Communications in Numerical Methods in Engineering* 2008; **24**:505–521.

[4] Hansbo P, M.G. Larson MG. A discontinuous Galerkin method for the plate problem. *Calcolo* 2002; **39**:41–59.

[5] Hansbo P, Larson MG. A posteriori error estimates for continuous/discontinuous Galerkin approximations of the Kirchhoff-Love plate. Preprint (2008).

[6] Larsson F, Hansbo P, Runesson K. Strategies for computing goal-oriented *a posteriori error* measures in non-linear elasticity. *International Journal for Numerical Methods in Engineering* 2002; **55**:879–894.

[7] Nilssen TK, Tai XC, Winther R. A robust nonconforming H^2-element. *Mathematics of Computation* 2001; **70**:489–505.

[8] Wells GN, Dung, NT. A C^0 discontinuous Galerkin formulation for Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:3370–3380.

[9] Svensson ED. Optimal search in finite element triangulations using binary trees. Preprint (Department of Mathematical Sciences, Division of Mathematics, Chalmers University of Technology and Göteborg University), 2006:21, 1652-9715.

[10] O'Rourke J. Computational geometry in C. 2:nd ed., Cambridge University Press, 1998.

Table 1: *Common mesh $\mathcal{T}$*

#	D_{P}	D_{M}	L_{P}	L_{M}	$L(\hat{e})$	θ
1	545	290	$2.81 \cdot 10^{-2}$	$1.26 \cdot 10^{-4}$	$3.15 \cdot 10^{-2}$	0.90
2	668	356	$2.57 \cdot 10^{-2}$	$9.90 \cdot 10^{-5}$	$2.16 \cdot 10^{-2}$	1.20
3	839	442	$1.22 \cdot 10^{-2}$	$6.33 \cdot 10^{-5}$	$1.32 \cdot 10^{-2}$	0.93
4	1071	562	$9.72 \cdot 10^{-3}$	$1.53 \cdot 10^{-5}$	$9.53 \cdot 10^{-3}$	1.02
5	1412	736	$7.70 \cdot 10^{-3}$	$1.59 \cdot 10^{-5}$	$7.34 \cdot 10^{-3}$	1.05
6	1861	962	$5.64 \cdot 10^{-3}$	$1.54 \cdot 10^{-5}$	$5.45 \cdot 10^{-3}$	1.04
7	2453	1262	$4.35 \cdot 10^{-3}$	$1.74 \cdot 10^{-5}$	$4.23 \cdot 10^{-3}$	1.03
8	3230	1660	$3.37 \cdot 10^{-3}$	$1.24 \cdot 10^{-5}$	$3.28 \cdot 10^{-3}$	1.03
9	4328	2214	$2.63 \cdot 10^{-3}$	$3.57 \cdot 10^{-6}$	$2.52 \cdot 10^{-3}$	1.04
10	5780	2946	$1.97 \cdot 10^{-3}$	$2.34 \cdot 10^{-6}$	$1.91 \cdot 10^{-3}$	1.03
11	7659	3890	$1.46 \cdot 10^{-3}$	$4.80 \cdot 10^{-6}$	$1.43 \cdot 10^{-3}$	1.02
12	10185	5168	$1.11 \cdot 10^{-3}$	$3.16 \cdot 10^{-6}$	$1.10 \cdot 10^{-3}$	1.01

Table 2: *Disparate meshes $\mathcal{T}_P$ and $\mathcal{T}_M$*

#	D_{P}	D_{M}	L_{P}	L_{M}	$L(\hat{e})$	θ
1	545	290	$2.81 \cdot 10^{-2}$	$1.27 \cdot 10^{-4}$	$3.15 \cdot 10^{-2}$	0.90
2	739	308	$2.35 \cdot 10^{-2}$	$9.37 \cdot 10^{-5}$	$1.98 \cdot 10^{-2}$	1.19
3	1001	336	$9.06 \cdot 10^{-3}$	$7.26 \cdot 10^{-5}$	$1.05 \cdot 10^{-2}$	0.87
4	1409	382	$7.36 \cdot 10^{-3}$	$1.95 \cdot 10^{-5}$	$7.21 \cdot 10^{-3}$	1.02
5	1933	438	$5.24 \cdot 10^{-3}$	$1.41 \cdot 10^{-5}$	$5.16 \cdot 10^{-3}$	1.02
6	2662	526	$4.07 \cdot 10^{-3}$	$1.81 \cdot 10^{-5}$	$3.94 \cdot 10^{-3}$	1.04
7	3615	650	$2.90 \cdot 10^{-3}$	$1.01 \cdot 10^{-5}$	$2.84 \cdot 10^{-3}$	1.03
8	4909	848	$2.32 \cdot 10^{-3}$	$5.47 \cdot 10^{-6}$	$2.25 \cdot 10^{-3}$	1.03
9	6558	1128	$1.71 \cdot 10^{-3}$	$9.98 \cdot 10^{-6}$	$1.69 \cdot 10^{-3}$	1.02
10	8638	1508	$1.32 \cdot 10^{-3}$	$7.14 \cdot 10^{-6}$	$1.31 \cdot 10^{-3}$	1.01
11	11483	1990	$1.02 \cdot 10^{-3}$	$4.19 \cdot 10^{-6}$	$1.01 \cdot 10^{-3}$	1.01

Iterations during adaptive procedure (unit square)

TABLE 3: *Common mesh $\mathcal{T}$*

#	D_{P}	D_{M}	L_{P}	L_{M}	$\lvert L_{\mathrm{P}}\rvert$	$\lvert L_{\mathrm{M}}\rvert$	$L(\hat{e})$	θ
1	833	450	$9.53 \cdot 10^{-3}$	$2.31 \cdot 10^{-3}$	$1.16 \cdot 10^{-2}$	$1.26 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$	0.81
2	1029	548	$7.25 \cdot 10^{-3}$	$2.32 \cdot 10^{-3}$	$8.66 \cdot 10^{-3}$	$1.11 \cdot 10^{-2}$	$1.17 \cdot 10^{-2}$	0.82
3	1304	688	$5.53 \cdot 10^{-3}$	$1.96 \cdot 10^{-3}$	$7.59 \cdot 10^{-3}$	$1.00 \cdot 10^{-2}$	$8.83 \cdot 10^{-3}$	0.85
4	1662	868	$4.29 \cdot 10^{-3}$	$1.53 \cdot 10^{-3}$	$6.18 \cdot 10^{-3}$	$8.46 \cdot 10^{-3}$	$6.81 \cdot 10^{-3}$	0.85
5	2147	1114	$3.30 \cdot 10^{-3}$	$1.13 \cdot 10^{-3}$	$5.07 \cdot 10^{-3}$	$7.71 \cdot 10^{-3}$	$5.05 \cdot 10^{-3}$	0.88
6	2798	1446	$2.57 \cdot 10^{-3}$	$1.01 \cdot 10^{-3}$	$4.06 \cdot 10^{-3}$	$6.92 \cdot 10^{-3}$	$4.11 \cdot 10^{-3}$	0.87
7	3635	1870	$2.06 \cdot 10^{-3}$	$7.67 \cdot 10^{-4}$	$3.62 \cdot 10^{-3}$	$6.20 \cdot 10^{-3}$	$3.17 \cdot 10^{-3}$	0.89
8	4774	2442	$1.58 \cdot 10^{-3}$	$6.14 \cdot 10^{-4}$	$3.17 \cdot 10^{-3}$	$5.44 \cdot 10^{-3}$	$2.50 \cdot 10^{-3}$	0.88
9	6268	3196	$1.25 \cdot 10^{-3}$	$4.42 \cdot 10^{-4}$	$2.62 \cdot 10^{-3}$	$4.60 \cdot 10^{-3}$	$1.87 \cdot 10^{-3}$	0.91
10	8164	4152	$9.68 \cdot 10^{-4}$	$3.32 \cdot 10^{-4}$	$2.25 \cdot 10^{-3}$	$4.08 \cdot 10^{-3}$	$1.42 \cdot 10^{-3}$	0.91
11	10600	5384	$7.76 \cdot 10^{-4}$	$2.61 \cdot 10^{-4}$	$1.84 \cdot 10^{-3}$	$3.61 \cdot 10^{-3}$	$1.11 \cdot 10^{-3}$	0.93

TABLE 4: *Disparate meshes $\mathcal{T}_P$ and $\mathcal{T}_M$*

#	D_{P}	D_{M}	L_{P}	L_{M}	$\lvert L_{\mathrm{P}}\rvert$	$\lvert L_{\mathrm{M}}\rvert$	$L(\hat{e})$	θ
1	833	450	$9.53 \cdot 10^{-3}$	$2.31 \cdot 10^{-3}$	$1.16 \cdot 10^{-2}$	$1.26 \cdot 10^{-2}$	$1.46 \cdot 10^{-2}$	0.81
2	1029	550	$7.31 \cdot 10^{-3}$	$2.16 \cdot 10^{-3}$	$8.65 \cdot 10^{-3}$	$1.06 \cdot 10^{-2}$	$1.17 \cdot 10^{-2}$	0.81
3	1296	702	$5.59 \cdot 10^{-3}$	$1.68 \cdot 10^{-3}$	$7.63 \cdot 10^{-3}$	$8.85 \cdot 10^{-3}$	$8.32 \cdot 10^{-3}$	0.87
4	1706	902	$4.00 \cdot 10^{-3}$	$1.36 \cdot 10^{-3}$	$5.59 \cdot 10^{-3}$	$7.82 \cdot 10^{-3}$	$6.38 \cdot 10^{-3}$	0.84
5	2133	1252	$3.07 \cdot 10^{-3}$	$1.06 \cdot 10^{-3}$	$4.46 \cdot 10^{-3}$	$7.13 \cdot 10^{-3}$	$4.71 \cdot 10^{-3}$	0.87
6	2711	1670	$2.52 \cdot 10^{-3}$	$8.57 \cdot 10^{-4}$	$3.78 \cdot 10^{-3}$	$6.16 \cdot 10^{-3}$	$3.93 \cdot 10^{-3}$	0.86
7	3339	2344	$2.07 \cdot 10^{-3}$	$6.11 \cdot 10^{-4}$	$3.31 \cdot 10^{-3}$	$5.05 \cdot 10^{-3}$	$3.13 \cdot 10^{-3}$	0.86
8	4469	3090	$1.60 \cdot 10^{-3}$	$4.71 \cdot 10^{-4}$	$2.88 \cdot 10^{-3}$	$4.41 \cdot 10^{-3}$	$2.35 \cdot 10^{-3}$	0.88
9	5961	4090	$1.20 \cdot 10^{-3}$	$3.47 \cdot 10^{-4}$	$2.52 \cdot 10^{-3}$	$3.84 \cdot 10^{-3}$	$1.70 \cdot 10^{-3}$	0.91
10	7787	5436	$9.45 \cdot 10^{-4}$	$2.73 \cdot 10^{-4}$	$1.96 \cdot 10^{-3}$	$3.41 \cdot 10^{-3}$	$1.34 \cdot 10^{-3}$	0.91
11	9845	7304	$7.49 \cdot 10^{-4}$	$2.07 \cdot 10^{-4}$	$1.67 \cdot 10^{-3}$	$2.95 \cdot 10^{-3}$	$1.07 \cdot 10^{-3}$	0.90

Iterations during adaptive procedure (L-shaped membrane)

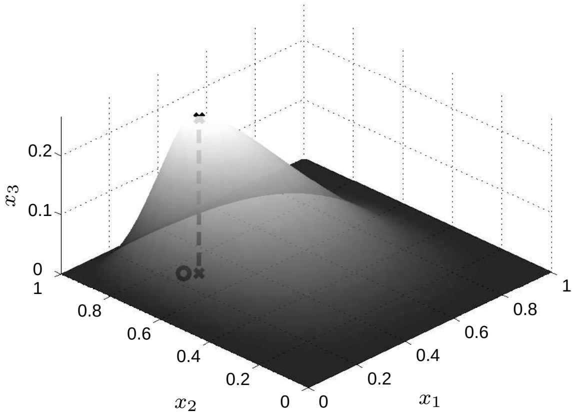


FIGURE 1: *Transversal plate deflection u_h*

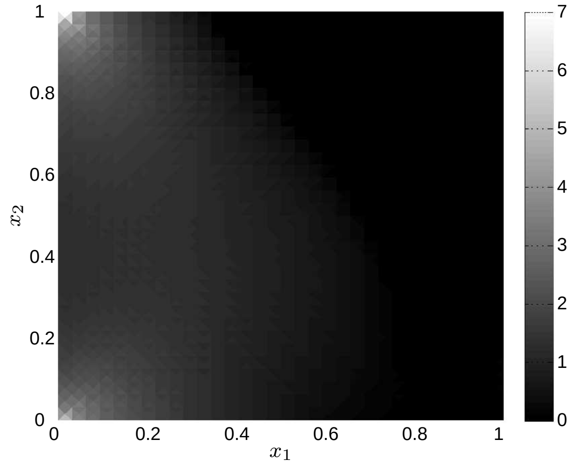


FIGURE 2: *Effective stress $\boldsymbol{\sigma}_\mathrm{e}$*

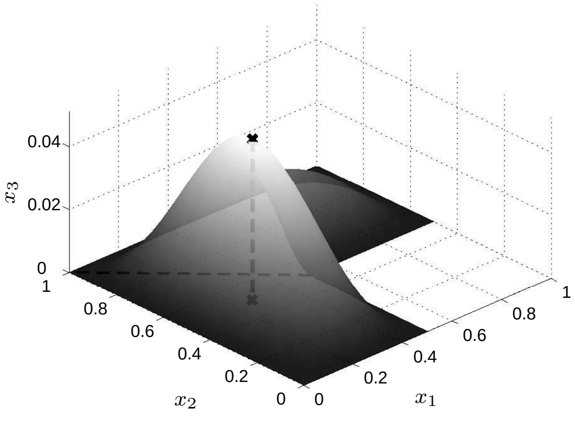


FIGURE 3: *Transversal plate deflection u_h*

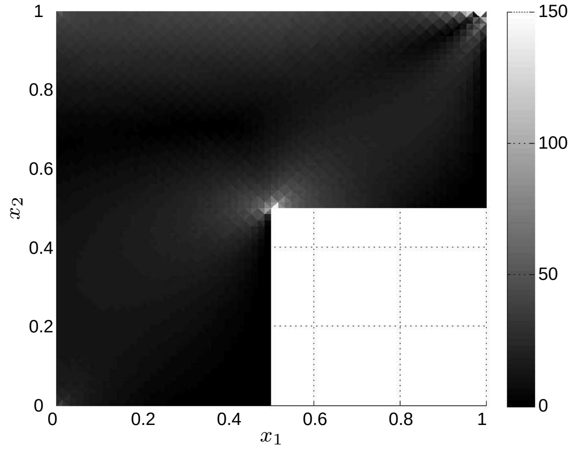


FIGURE 4: *Effective stress $\boldsymbol{\sigma}_\mathrm{e}$*

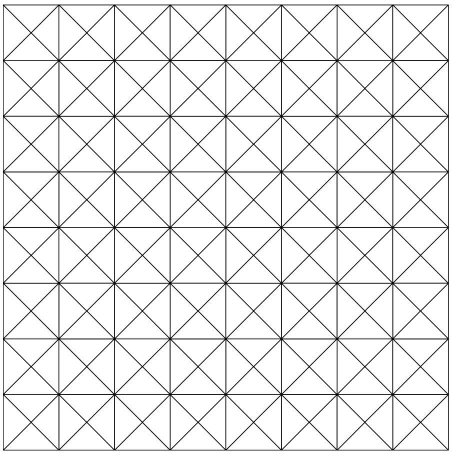


Figure 5: *Initial mesh $\mathcal{T}^0$ (always same)*

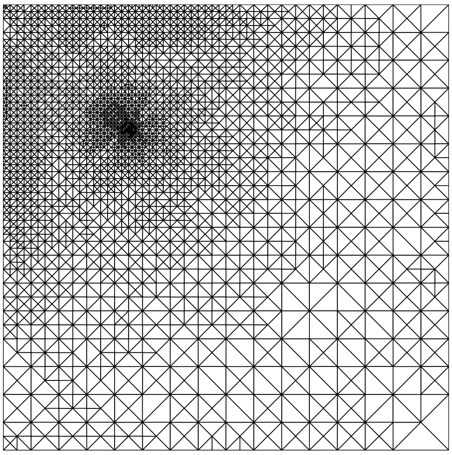


Figure 6: *Common mesh $\mathcal{T}^{12}$*

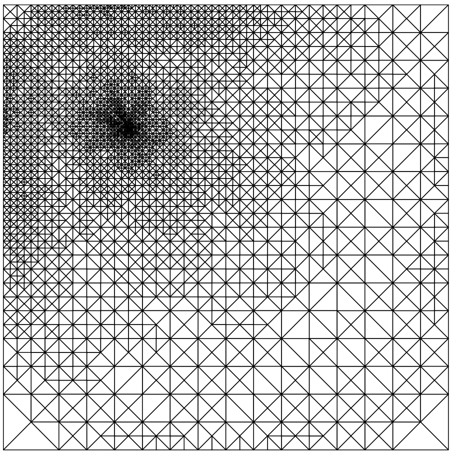


Figure 7: *Plate mesh $\mathcal{T}_P^{11}$*

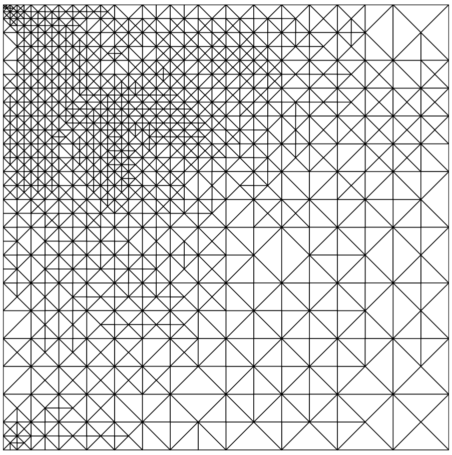


Figure 8: *Plane stress mesh $\mathcal{T}_M^{11}$*

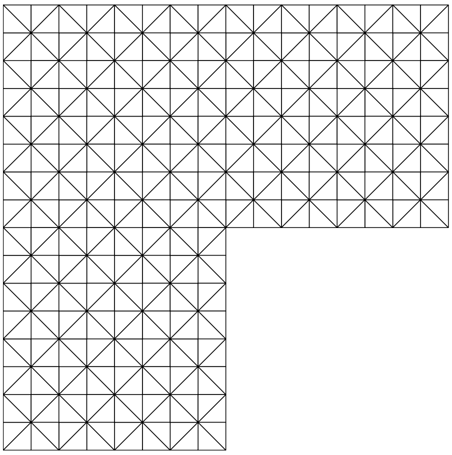


Figure 9: *Initial mesh $\mathcal{T}^0$ (always same,*

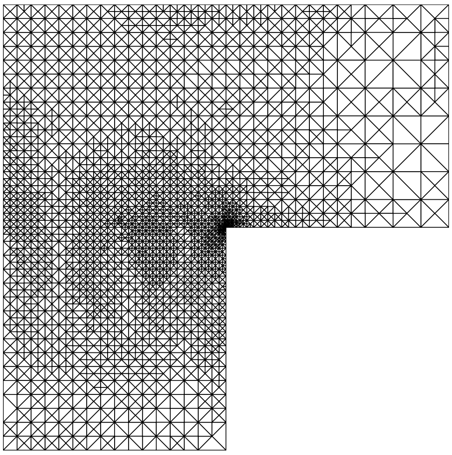


Figure 10: *Common mesh $\mathcal{T}^{11}$*

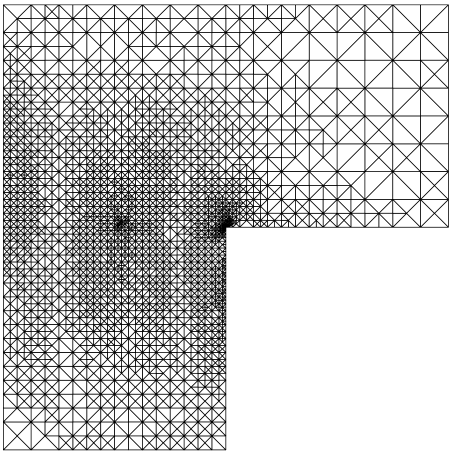


Figure 11: *Plate mesh $\mathcal{T}_P^{11}$*

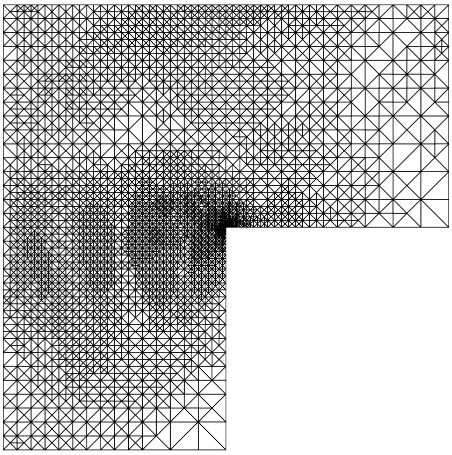
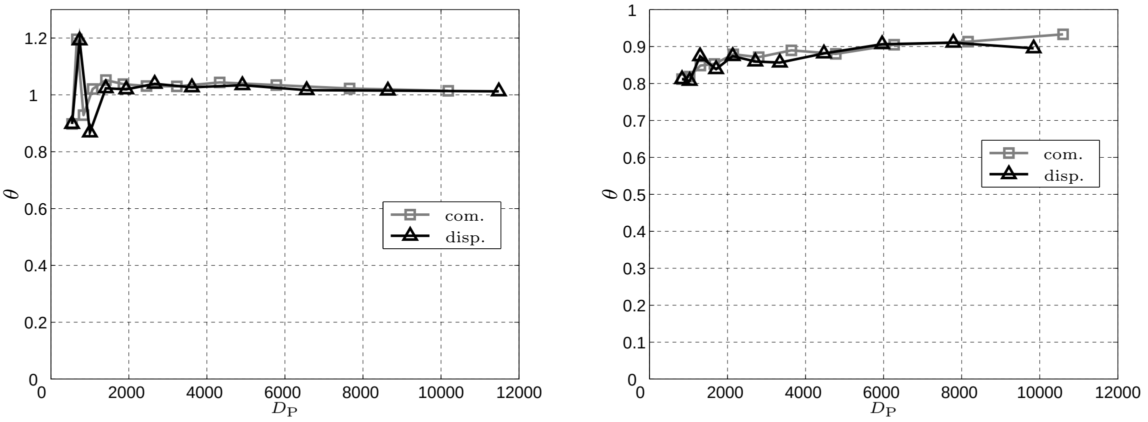


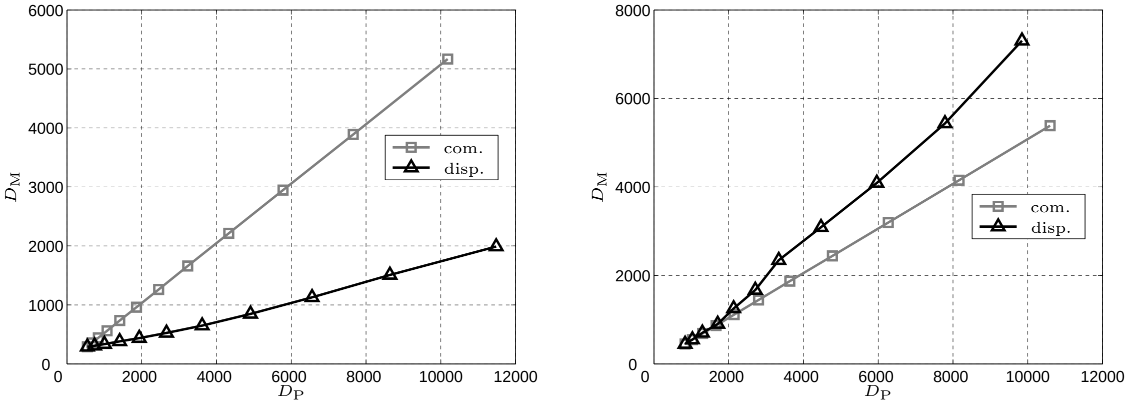
Figure 12: *Plane stress mesh $\mathcal{T}_M^{11}$*



(A) *Unit square*

(B) *L-shaped membrane*

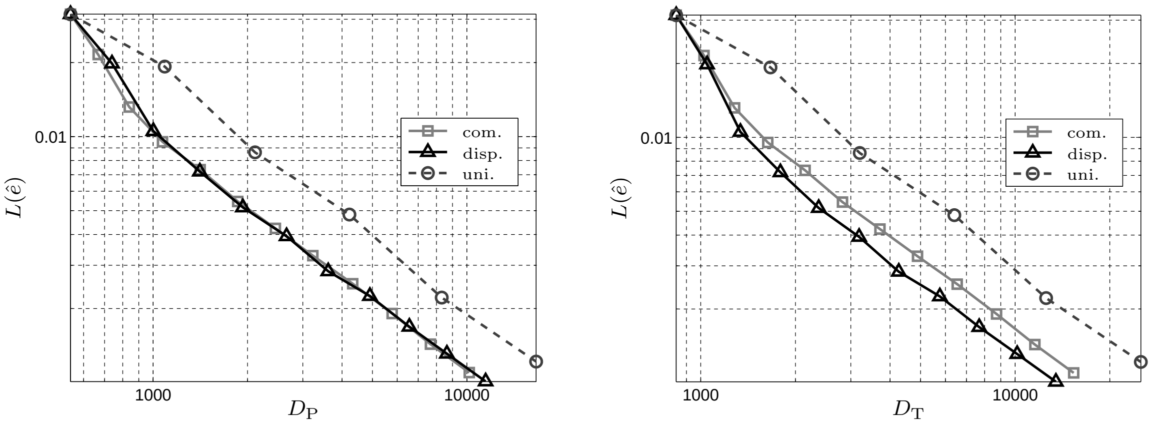
FIGURE 13: *Effectivity indices (common mesh:* `com.`*; disparate meshes:* `disp.`*)*



(A) *Unit square*

(B) *L-shaped membrane*

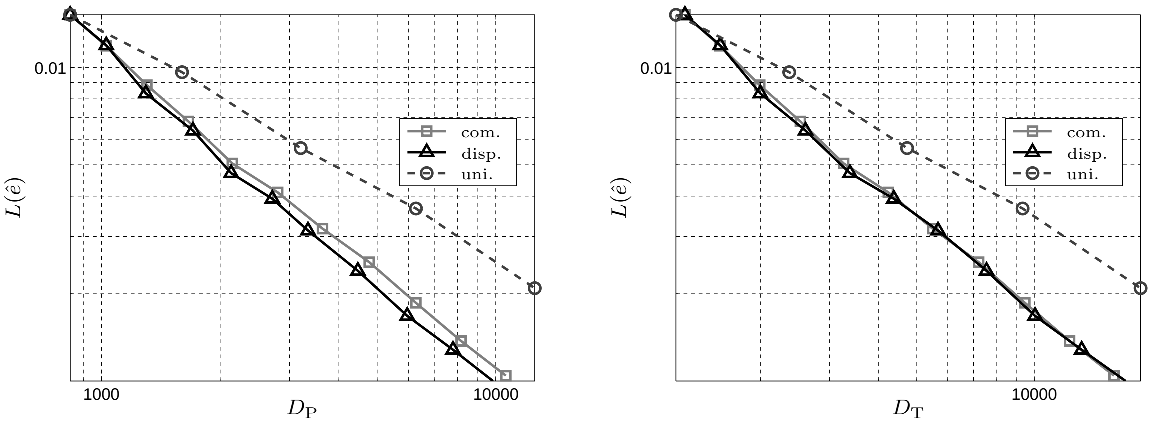
FIGURE 14: *Distribution of degrees of freedom*



(A) *CM and DM are about equally accurate*

(B) *DM uses a simpler model*

Figure 15: *Unit square (uniform mesh:* **uni.***)*



(A) *DM is the most accurate*

(B) *DM benefits from a more complex model*

Figure 16: *L-shaped membrane*