THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Adaptive finite element methods for plate bending problems

DAVID HEINTZ

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2011

Adaptive finite element methods for plate bending problems
DAVID HEINTZ
ISBN 978-91-7385-495-5

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 3176
ISSN 0346-718X

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 2011

Adaptive finite element methods for plate bending problems

DAVID HEINTZ

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

Abstract

The major theme of the thesis is the development of goal-oriented model adaptive continuous-discontinuous Galerkin (c/dG) finite element methods (FEM), for the numerical solution of the Kirchhoff and Mindlin-Reissner (MR) plate models. Hierarchical modeling for linear elasticity on thin domains (beam-like) in two spatial dimensions is also considered, as a natural extension of the Bernoulli and Timoshenko beam theories.

The basic idea behind *model adaptivity* is to refine, not only the computational mesh, but the underlying physical model as well. Consequently different mathematical formulations—usually partial differential equations—may be discretized on the element level. Our algorithms use duality-based *a posteriori* error estimates, which separate the discretization and modeling errors into an additive split (allows for independent reduction the error contributions). The error representation formulas are linear functionals of the error, which is often more relevant in engineering applications.

In standard FEM the continuity constraints can make it difficult to construct the approximating spaces on unstructured meshes. When solving the plate formulations, continuous quadratic polynomials are used for the lateral displacements, and first-order discontinuous polynomials for the rotation vector, whose inter-element continuity is imposed weakly by Nitsche's method. The bilinear form is coercive if a computable penalty parameter is large enough (and small enough to avoid locking).

The discretization of the MR model converges to the Kirchhoff model as the thickness of the plate tends to zero. This makes the introduced c/dG FEM particularly interesting in the context of model adaptivity, and as such it constitutes the main result of the thesis.

Keywords: *goal-oriented adaptivity, model adaptivity, discontinuous Galerkin, Nitsche's method, Kirchhoff plate, Mindlin-Reissner plate*

List of appended papers

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

A

P. HANSBO, D. HEINTZ AND M. G. LARSON, *An adaptive finite element method for second-order plate theory*, Int. J. Numer. Meth. Engng., 81 (2010), pp. 584-603. (The appended paper has been slightly revised from the published version.)

B

P. HANSBO, D. HEINTZ AND M. G. LARSON, *A finite element method with discontinuous rotations for the Mindlin-Reissner plate model*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 638-648. (The appended paper has been slightly revised from the published version.)

Presented at USNCCM 10, Columbus, Ohio, USA, 16-19 July 2009.

C

D. HEINTZ, *Model combination in plate bending problems*, Preprint 33, Chalmers University of Technology and Göteborg University, 2010. ISSN 1652-9715. Submitted for international publication.

Presented at ECCM 4, Paris, France, 16-21 May 2010.

D

D. HEINTZ, P. HANSBO, *A two-model adaptive finite element method for plates*, Preprint 45, Chalmers University of Technology and Göteborg University, 2010. ISSN 1652-9715. Submitted for international publication.

E

D. HEINTZ, *Hierarchical modeling for elasticity on thin domains*, Preprint 34, Chalmers University of Technology and Göteborg University, 2008. ISSN 1652-9715.

Presented at WCCM 8/ECCOMAS 5, Venice, Italy, 30 June-4 July 2008.

In the introduction of the thesis they will be referred to as Paper A-E.

Contributions to co-authored papers

A

Planned and carried out the numerical simulations. Wrote the corresponding parts of the paper (Sections 4-5).

B

Planned and carried out the numerical simulations. Wrote the corresponding parts of the paper (Sections 6-7).

D

Took part in the theoretical development. Planned and carried out the numerical simulations. Wrote the paper.

Acknowledgements

The work in this thesis was funded by the Swedish Research Council, as part of the projects 2004-5147 and 2007-4522, entitled *Model Adaptivity in Computational Mechanics*.

Foremost, my gratitude goes to my supervisor, Professor Peter Hansbo: his support and encouragement have been invaluable. I also wish to thank my co-author Professor Mats G. Larson.

I am much grateful to Dr. Thomas Ericsson. For his helpful professional advices, always happy to assist a PhD student in need, and for being the good friend that he is during my years at the Department of Mathematical Sciences.

I would like to thank former and present colleagues of the Computational Mathematics group for providing a pleasant working environment: Dr. Christoffer Cromvik, Dr. Karin Kraft, and Associate Professor Göran Starius, to name a few.

I also wish to express my thanks to Associate Professor Fredrik Larsson, who kindly reviewed my licentiate thesis, and supported me at conferences in Venice, Columbus, and Paris.

Last but not least, my words go missing for the never-ending love and patience from my family: my parents with their families, my brother and my sister-in-law, my nieces, and Helena. Thank you.

David Heintz
Göteborg, January 2011

Contents

Part I

Introduction

Chapter 1

Background

This chapter begins with a general introduction to adaptive modeling, and serves as a motivation behind the work presented in the thesis. It concludes by exemplifying the use of dimensional reduction and hierarchical models in elasticity, which will be two important concepts in the appended papers.

1.1 Motivation

The finite element method (FEM) is a numerical method for solving differential and integral equations approximatively using computers (relatively few such equations are amenable to analytical methods). FEM originated from the need of solving complicated problems in elasticity and structural mechanics in the middle and late 1950s. Today it is employed in a variety of engineering disciplines, including the fields of electromagnetics and fluid dynamics. In industrial applications, say deformation during car crashes, virtual prototyping by FE software enhances design and shortens production cycles.

For any problem there are usually several mathematical models, more or less accurate, which describe the same physical phenomena. Should a simple model be adequate for solving the problem this is preferable, since less computational resources—in terms of run time and memory—are required. Without careful consideration the computational complexity could easily prove intractable. We must then ask ourselves: How do we know if the numerical solution, i.e., an approximation, is accurate enough for our purposes? Assume that numerical errors (round-off etc.) and errors in data (measurements, etc.) are negligible. It is left to determine whether

- the mathematical model describes the physical reality sufficiently well (the discrepancy is referred to as the *modeling error*);

- the numerical solution of the mathematical model is sufficiently accurate (their difference is called the *discretization error*).

Answering these questions are important research topics, the latter being the advent of adaptive procedures in FEM, which, at least traditionally, are techniques for optimizing the underlying mesh[†]. The goal is to distribute the degrees of freedom to get high accuracy with respect to the computational cost. In practice the computer uses analytically derived *a posteriori* error estimates, based on the initial numerical solution, to determine the size and the locality of the error. If the error is within a prescribed tolerance, the numerical solution is accepted; otherwise new elements are introduced where the error was the largest. An improved numerical solution is computed, and the entire procedure is repeated, in an automated fashion, until convergence. This technique, known as *mesh adaptivity*, only reduces the discretization error—the choice of mathematical model still lies with the engineer. But if an hierarchy of models is available, the most accurate considered as the *master model*, it is equally possible to have the computer adaptively decide which one to use on the element level. The rationale is to balance the different error sources. Ideally, the discretization and modeling errors should be of the same size: there is no point in spending computational resources on solving an inadequate model, just as there is no point in solving a complex model using too few elements (the relatively larger error source deteriorates the accuracy of the numerical solution). The idea of locally changing the mathematical formulation of the problem is known as *model adaptivity*. This field is not as well developed, but the research has intensified. The ultimate goal is to combine mesh and model adaptivity—let the engineer decide how accurate the numerical solution must be, and then leave it to the computer to resolve mesh and model.

1.2 Hierarchical modeling in elasticity

Boundary value problems in elasticity are often posed on *thin domains*, e.g., beams, plates, or shells. Here the term *thin* relates to the physical domain being much smaller in one direction: a beam, e.g., is dominated by its extension in the axial direction. This may justify making certain simplifying assumptions on the master model (the full three-dimensional elasticity theory), effectively replacing the original problem with a lower-dimensional one, which consequently is known as *dimension reduction*.

[†]The mesh is a discretization of the continuous domain, e.g., a car subjected to a crash, into a set of subregions, referred to as *elements*. On each element the deformation of the car is approximated, and by introducing more elements, or degrees of freedom, the accuracy of the numerical solution improves. (In theory the discretization error vanishes as the number of elements tends to infinity.)

Lower-dimensional problems are more susceptible to analytical techniques: the simplest formulations can be integrated directly[†], and sometimes the exact solution can be derived by means of Fourier series. In the wake of the evolution of computers, lower-dimensional methods have also gained momentum to be commonly used in today's FE software. The reasons why include, firstly, computational savings (using less complex models typically reduces the number of degrees of freedom), and secondly, that excessive mesh refinement in three-dimensional problems could cause bad conditioning (especially bothersome for problems posed on thin domains).

Lower-dimensional models are usually advocated by being *asymptotically exact*, i.e., the difference in the numerical solution, as compared to the one using the higher-dimesional model, vanishes as the thickness (the extension in the thin direction) tends to zero. In practice, however, the thickness is non-zero, and hence a corresponding modeling error is inevitably incurred.

In the late 1980s and early 1990s the idea arose to embed classical models into hierarchies of lower-order models, the term *hierarchic models* was introduced by Szabo and Sahrmann, and the approach further developed by Actis, Babuška and co-workers (see Schwab [40] for references). Simplified models were constructed by imposing restrictions on the displacements of the three-dimensional formulations, say, by using a prescribed polynomial expansion in the transverse direction. Refined models are then readily obtained by increasing the order of the approximating polynomial. A set of consecutive models constitute a hierarchy, which is used for solving the problem adaptively. The final model will be tailor-made for each particular problem (illustrated in Figure 1.1).

The sophisticated mesh and model adaptive algorithm should promote an equidistribution of the total error, meaning that the local errors on each element are of the same size in magnitude. Such *a posteriori* error representations would include both the discretization error and the modeling error. Early work on model adaptivity assumed the discretization error to be negligible, so that the modeling error summed up to the total error (see, e.g., Babuška *et al.* [7]). The error estimates were improved on to include both error sources, first by measuring the total error in global norms (typically the generic energy norm), and in more recent work, including papers by Babuška and Schwab [8], Ainsworth [1], Vemaganti [45], and Repin *et al.* [38], to give upper and lower bounds of the total error in terms of linear functionals.

[†]The exact solution of the Bernoulli beam equation, for a constant bending moment, can be computed by four successive integrations.

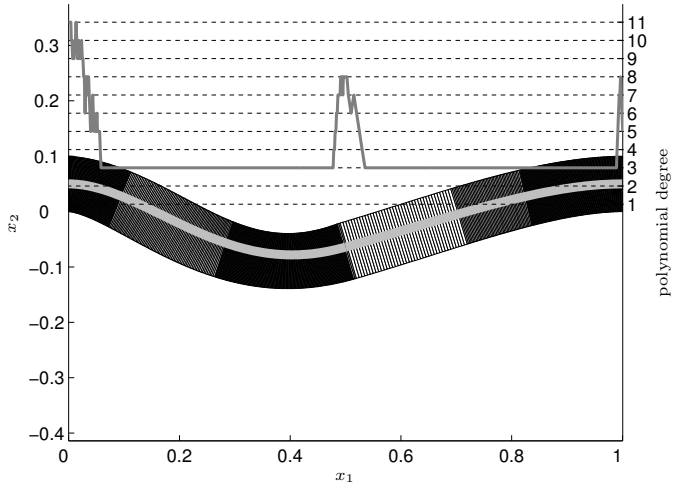

Figure 1.1: *Adapted numerical solution for a clamped loaded beam. The mesh consists of rectangular elements, where the polynomial approximation in the thin direction varies along the axial direction. The exact solution of the Bernoulli beam equation, a simpler model, is seen in solid gray. (For the problem formulation see Section 6.3.2 in Paper E.)*

Chapter 2

Theory

The ambition of this chapter is to present the theory and methods treated in the appended papers in a concise manner.

In the sequel—if not explicitly expressed—it is simply assumed that functions belong to admissible function spaces, in the sense of them being sufficiently regular for the formulations to make sense.

2.1 The finite element method

FEM is a technique for solving general partial differential equations (PDEs), and is well-suited for problems posed on complex geometries. Instead of approximating the differential equation directly, which a traditional finite difference method (FDM) does, FEM 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, consider the stationary one-dimensional heat conduction in a bar, which is described by the following model problem (see [22, 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, $u(0) = u(1) = 0$, (1) is multiplied by a function v, such that $v(0) = v(1) = 0$, and integrated 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

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

The functions u and v are called the *trial* and *test functions*, respectively, and it is said that (1) has been tested with v. Should one test by a sufficiently large number of functions v, the integrated form (2) is expected to actually satisfy (1) pointwise, given sufficient regularity of the weak solution (the virtual work principle then is 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 actually 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 the approximate solution in a finite-dimensional space, spanned by a set of shape functions, which are easy to differentiate and integrate. The new idea in FEM is the choice of approximating functions: they are chosen to be *piecewise polynomials*. In its basic form FEM uses piecewise linear continuous functions $\varphi_i = \varphi_i(x)$ with local support. If the FE *ansatz* is taken as

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

Galerkin suggested (3) to hold for all test functions of the same form. To ensure this property, the equation is tested against each φ_i separately (then it will be satisfied for an arbitrary linear combination of the shape functions). A linear system of equations is obtained to be solved for the unknown 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 certain subspace, spanned by a set of shape functions. These functions are not necessarily linear, but can be polynomials of higher degrees, or trigonometric functions (so-called *spectral methods*). A discontinuous *ansatz* for the FE solution is also possible, cf. Section 2.4. 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., the textbooks by Strang and Fix [44], Ern and Guermond [23], and Brenner and Scott [15], which is a strength, since it provides tools for deriving analytical error estimates, thereby allowing improvements of numerical methods. FEM has typically been the choice of discretization for applications in solid mechanics. FDM, on the other hand, is more straightforward to implement than FEM (at least for simpler model problems on rectangular domains), but the

method is more common within the field of computational fluid dynamics (CFD). Otherwise CFD applications tend to employ lower-order finite volume methods.

2.2 The equations of linear 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 is described 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

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

where $\partial\Omega = \partial\Omega_{\mathrm{D}} \cup \partial\Omega_{\mathrm{N}}$ is a partitioned boundary of Ω. 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 ((5) and (7) pertain to a state of plain strain). Moreover, $\boldsymbol{I}$ 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

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

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 the function space

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

is first introduced, and then it follows (the details of the derivation has been omitted): find $\boldsymbol{u} \in V \times V$ such that

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

where the bilinear form

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

is the integrated tensor contraction

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

and the linear functional

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

Here (8) may be interpreted as a balance between the internal (9) and external (10) "virtual work" (with the test functions $\boldsymbol{v}$ acting as "virtual displacements").

For the numerical approximation of (8) a FEM is established. To simplify its formulation 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 the constitutive matrix

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

will be used for the purpose of rewriting the bilinear form as

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

which facilitates implementation. Let $\mathfrak{T}_h$ be a partition of Ω, dividing the domain into N_{el} elements. To be more precise, have $\mathfrak{T}_h = \{K\}$ to be a set of triangles K, such that

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

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

$$h_K = \text{diam}(K) = \max_{\boldsymbol{y}_1, \boldsymbol{y}_2 \in K} (\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2), \quad \text{for all } K \in \mathfrak{T}_h,$$

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

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

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

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

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

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

where $v_j = v(\boldsymbol{x}_j)$ represents the j:th nodal value of v. An *ansatz* for a solution of the type (12) is made, and hence the FE formulation of (8) becomes: find $\boldsymbol{u}_h \in V_h \times V_h$ such that

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

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 and y, respectively, having a total of $2N_{\text{no}}$ degrees of freedom. Since testing against all $\boldsymbol{v} \in V_h \times V_h$ reduces to testing against $\{\varphi_j\}_{j=1}^{N_{\text{no}}}$ in each direction, i.e.,

$$\boldsymbol{v}_1 = (\varphi_1, 0),\, \boldsymbol{v}_2 = (0, \varphi_1), \ldots,\, \boldsymbol{v}_{2N_{\text{no}}-1} = (\varphi_{N_{\text{no}}}, 0),\, \boldsymbol{v}_{2N_{\text{no}}} = (0, \varphi_{N_{\text{no}}}),$$

and $\boldsymbol{\varepsilon}_{\mathrm{v}}(\boldsymbol{u}_h) = \tilde{\boldsymbol{\nabla}}\boldsymbol{\varphi}\mathbf{u} = \boldsymbol{B}\mathbf{u}$, (13) corresponds to solving

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

which is a linear system of equations $\mathbf{S}\mathbf{u} = \mathbf{f}$, making (14) a suitable starting point for FE implementation.

The equations of linear elasticity were solved as part of Papers A and E.

2.3 Plate theory

Plate theory is an engineering approximation, which reduces the full three-dimensional elasticity theory into a simpler two-dimensional problem (the thickness of the plate is regarded as the thin direction). In many applications, however, the plate theory provides accurate solutions.

When a plate is subjected to transverse (normal to its midsurface) loads, the lateral displacements, and the stress and strain distributions across the thickness will no longer be uniform—finding them is known as the problem of *plate bending*. A typical situation is that of a roof or a bridge carrying out lateral loads to the support.

In the context of adaptive modeling for thin to moderately thick structures a two-model hierarchy is given by the Kirchhoff and Mindlin-Reissner (MR) plate theories.

2.3.1 The Kirchhoff plate model

Kirchhoff described the plate bending problem in the middle 1850s [25], and did so mathematically by a fourth-order partial differential equation. For a clamped plate, represented by a polygonial domain $\Omega \in \mathbb{R}^2$, the standard Kirchhoff plate theory is formulated as follows: find the lateral displacements u such that

$$\begin{aligned} \nabla \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\nabla u) &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega, \\ \boldsymbol{n} \cdot \nabla u &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{15}$$

where (15) expresses equilibrium between internal and external forces. Here f is the transverse load, $\boldsymbol{n}$ is the exterior unit normal on the boundary, and $\boldsymbol{\sigma}$ represents the moment tensor,

$$\boldsymbol{\sigma}(\boldsymbol{\theta}) := \lambda \nabla \cdot \boldsymbol{\theta}\mathbf{1} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{\theta}),$$

for a linearly elastic material. $\mathbf{1}$ is the identity tensor, and $\boldsymbol{\varepsilon}$ is the curvature tensor,

$$\boldsymbol{\varepsilon}(\boldsymbol{\vartheta}) := \begin{bmatrix} \frac{\partial \boldsymbol{\vartheta}_1}{\partial x} & \frac{1}{2}\left(\frac{\partial \boldsymbol{\vartheta}_1}{\partial y} + \frac{\partial \boldsymbol{\vartheta}_2}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial \boldsymbol{\vartheta}_2}{\partial x} + \frac{\partial \boldsymbol{\vartheta}_1}{\partial y}\right) & \frac{\partial \boldsymbol{\vartheta}_2}{\partial y} \end{bmatrix}.$$

The constitutive parameters, or Lamé constants, are given by

$$\lambda = \frac{E\nu}{12(1-\nu)^2} \quad \text{and} \quad \mu = \frac{E}{24(1+\nu)},$$

where $E > 0$ and $0 \leq \nu < 1/2$ are Young's modulus and Poisson's ratio, respectively. Moreover, the notation

$$\nabla \cdot \boldsymbol{\vartheta} = \frac{\partial \boldsymbol{\vartheta}_1}{\partial x} + \frac{\partial \boldsymbol{\vartheta}_2}{\partial y} \quad \text{and} \quad \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \begin{bmatrix} \frac{\partial \boldsymbol{\sigma}_{11}}{\partial x} + \frac{\partial \boldsymbol{\sigma}_{12}}{\partial y} \\ \frac{\partial \boldsymbol{\sigma}_{21}}{\partial x} + \frac{\partial \boldsymbol{\sigma}_{22}}{\partial y} \end{bmatrix}^{\mathrm{T}}$$

is used, so terms of the internal force are

$$\nabla \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \frac{\partial^2 \boldsymbol{\sigma}_{11}}{\partial x^2} + \frac{\partial^2 \boldsymbol{\sigma}_{12}}{\partial x \partial y} + \frac{\partial^2 \boldsymbol{\sigma}_{21}}{\partial y \partial x} + \frac{\partial^2 \boldsymbol{\sigma}_{22}}{\partial y^2}.$$

To pose a weak formulation of (15), the equilibrium equation is multiplied by a scalar test function v, followed by repeated integration by parts over the domain,

$$\begin{aligned} \int_\Omega \nabla \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} v \,\mathrm{d}x\mathrm{d}y &= \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} v \,\mathrm{d}s - \int_\Omega \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \cdot \nabla v \,\mathrm{d}x\mathrm{d}y \\ &= \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} v \,\mathrm{d}s - \int_{\partial\Omega} \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \nabla v \,\mathrm{d}s \\ &\quad + \int_\Omega \boldsymbol{\sigma}(\nabla u) : \boldsymbol{\varepsilon}(\nabla v) \,\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{16}$$

Now, if using the equality $\nabla v = \boldsymbol{n}(\boldsymbol{n} \cdot \nabla v) + \boldsymbol{t}(\boldsymbol{t} \cdot \nabla v)$, where $\boldsymbol{t}$ is the unit tangent vector,

$$\begin{aligned} \int_{\partial\Omega} \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \cdot \nabla v \,\mathrm{d}s &= \int_{\partial\Omega} (\boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}) \boldsymbol{n} \cdot \nabla v \,\mathrm{d}s + \int_{\partial\Omega} (\boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{t}) \boldsymbol{t} \cdot \nabla v \,\mathrm{d}s \\ &= \int_{\partial\Omega} M_{nn} \boldsymbol{n} \cdot \nabla v \,\mathrm{d}s + \int_{\partial\Omega} M_{nt} \boldsymbol{t} \cdot \nabla v \,\mathrm{d}s, \end{aligned}$$

by substituting the normal and twisting moments, $M_{nn} = \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}$ and $M_{nt} = \boldsymbol{n} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{t}$, respectively. Since the boundary consists of linear edges E, connected at the boundary vertices,

$$\begin{aligned} \int_{\partial\Omega} M_{nt} \boldsymbol{t} \cdot \nabla v \,\mathrm{d}s &= \sum_{E \subset \partial\Omega} \int_E M_{nt} \boldsymbol{t} \cdot \nabla v \,\mathrm{d}s \\ &= \sum_{E \subset \partial\Omega} [M_{nt} v]_{\partial E} - \int_{\partial\Omega} \boldsymbol{t} \cdot \nabla M_{nt} v \,\mathrm{d}s, \end{aligned}$$

using integration by parts (here $[w]_{\partial E}$ denotes the difference between the endpoint values of w on E). Finally, the transverse force can be identified

by $T = \boldsymbol{n} \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \boldsymbol{t} \cdot \nabla M_{nt}$, and hence (16) is the same as

$$\int_\Omega \nabla \cdot \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} v \,\mathrm{d}x\mathrm{d}y = \int_\Omega \boldsymbol{\sigma}(\nabla u) : \boldsymbol{\varepsilon}(\nabla v) \,\mathrm{d}x\mathrm{d}y - \int_{\partial\Omega} M_{nn} \boldsymbol{n} \cdot \nabla v \,\mathrm{d}s + \int_{\partial\Omega} T v \,\mathrm{d}s - \sum_{E \subset \partial\Omega} [M_{nt} v]_{\partial E} \,,$$

where the pointwise twisting moments at the corners vanish if the boundary is smooth. This being the case, alongside $v|_{\partial\Omega} = 0$ and $\boldsymbol{n} \cdot \nabla v|_{\partial\Omega} = 0$, gives the variational statement: find $u \in$ such that

$$a(\nabla u, \nabla v) = \int_\Omega f v \,\mathrm{d}x\mathrm{d}y, \tag{17}$$

for all $v \in V = \left\{ v \in H_0^2(\Omega) : \boldsymbol{n} \cdot \nabla v = 0 \text{ on } \partial\Omega \right\}$. The bilinear form, which corresponds to the bending energy, is defined in terms of the tensor contraction,

$$a(\boldsymbol{\theta}, \boldsymbol{\vartheta}) := \int_\Omega \boldsymbol{\sigma}(\boldsymbol{\theta}) : \boldsymbol{\varepsilon}(\boldsymbol{\vartheta}) \,\mathrm{d}x\mathrm{d}y. \tag{18}$$

A difficulty with the Kirchhoff plate theory (or other fourth-order differential equations) is that standard FEMs require $\mathcal{C}^1$ continuous approximating spaces, which, at least on general meshes, are cumbersome to construct: the $\mathcal{C}^1$ Argyris triangle, e.g., carries 21 degrees of freedom for a quintic approximation, see Argyris *et al.* [2], whereas macro-element techniques involve third-order polynomials, cf. Ciarlet [18]. Consequently nonconforming elements, like the well-known Morley triangle [29], are commonly used.

The Kirchhoff plate theory has been used in Papers A, C-D.

2.3.2 The Mindlin-Reissner plate model

The Kirchhoff theory (15) typically models thin plates whose lateral displacements are small. Should the plate be thicker, or shear effects otherwise be non-negligible, however, the accuracy of the Kirchhoff solution tends to decay. Then (15) is commonly replaced by the MR plate theory [36, 37, 28], which is a system of two second-order partial differential equations,

$$-\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{\theta}) - \kappa t^{-2}(\nabla u - \boldsymbol{\theta}) = \mathbf{0}, \tag{19}$$

$$-\kappa t^{-2} \nabla \cdot (\nabla u - \boldsymbol{\theta}) = f, \tag{20}$$

where $\boldsymbol{\theta}$ denotes the rotations of the midsurface of the plate, the material parameter $\kappa = Ek/\big(2(1+v)\big)$, with $k = 5/6$ being a shear correction factor, and t is the constant thickness of the plate. For clamped boundary conditions $u = 0$ and $\boldsymbol{\theta} = \mathbf{0}$ at $\partial\Omega$.

The variational form of the MR system of equations (19) and (20) can be derived by, firstly, multiplying (19) by a vector-valued test function $\boldsymbol{\vartheta}$, and then integrating by parts over the domain,

$$\begin{aligned} 0 &= -\int_\Omega \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{\theta})\cdot\boldsymbol{\vartheta}\,\mathrm{d}x\mathrm{d}y - \kappa t^{-2}\int_\Omega(\nabla u-\boldsymbol{\theta})\cdot\boldsymbol{\vartheta}\,\mathrm{d}x\mathrm{d}y \\ &= -\int_{\partial\Omega}\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{\theta})\cdot\boldsymbol{\vartheta}\,\mathrm{d}s + \int_\Omega\boldsymbol{\sigma}(\boldsymbol{\theta}):\boldsymbol{\varepsilon}(\boldsymbol{\vartheta})\,\mathrm{d}x\mathrm{d}y \\ &\quad -\kappa t^{-2}\int_\Omega(\nabla u-\boldsymbol{\theta})\cdot\boldsymbol{\vartheta}\,\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{21}$$

Secondly, now multiplying (20) by a scalar test function v, followed by integration by parts over the domain,

$$\begin{aligned} \int_\Omega fv\,\mathrm{d}x\mathrm{d}y &= -\kappa t^{-2}\int_\Omega\nabla\cdot(\nabla u-\boldsymbol{\theta})v\,\mathrm{d}x\mathrm{d}y \\ &= \kappa t^{-2}\int_\Omega(\nabla u-\boldsymbol{\theta})\cdot\nabla v\,\mathrm{d}x\mathrm{d}y \\ &\quad -\kappa t^{-2}\int_{\partial\Omega}\boldsymbol{n}\cdot(\nabla u-\boldsymbol{\theta})v\,\mathrm{d}s, \end{aligned} \tag{22}$$

and so by adding equations (21) and (22),

$$\begin{aligned} \int_\Omega fv\,\mathrm{d}x\mathrm{d}y = &\int_\Omega\boldsymbol{\sigma}(\boldsymbol{\theta}):\boldsymbol{\varepsilon}(\boldsymbol{\vartheta})\,\mathrm{d}x\mathrm{d}y + \kappa t^{-2}\int_\Omega(\nabla u-\boldsymbol{\theta})\cdot(\nabla v-\boldsymbol{\vartheta})\,\mathrm{d}x\mathrm{d}y \\ &-\int_{\partial\Omega}\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{\theta})\cdot\boldsymbol{\vartheta}\,\mathrm{d}s - \kappa t^{-2}\int_{\partial\Omega}\boldsymbol{n}\cdot(\nabla u-\boldsymbol{\theta})v\,\mathrm{d}s. \end{aligned}$$

The variational formulation thus becomes: find $(u,\boldsymbol{\theta})\in H_0^1(\Omega)\times\left[H_0^1(\Omega)\right]^2$ such that

$$a(\boldsymbol{\theta},\boldsymbol{\vartheta}) + b(u,\boldsymbol{\theta};v,\boldsymbol{\vartheta}) = \int_\Omega fv\,\mathrm{d}x\mathrm{d}y, \tag{23}$$

for all $(v,\boldsymbol{\vartheta})\in H_0^1(\Omega)\times\left[H_0^1(\Omega)\right]^2$, with the shear energy functional

$$b(u,\boldsymbol{\theta};\,v,\boldsymbol{\vartheta}) := \kappa t^{-2}\int_\Omega(\nabla u-\boldsymbol{\theta})\cdot(\nabla v-\boldsymbol{\vartheta})\,\mathrm{d}x\mathrm{d}y. \tag{24}$$

Note that the Kirchhoff formulation (17) requires higher regularity of the lateral displacements than (23) does, owing to the second-order derivatives present in the bilinear form (18). Hence, from a computational point-of-view, the MR formulation (19)–(20) actually seems simpler—it consists of a system of two second-order partial differential equations, and requires only a C^0 approximation. The difficulty, however, is another: the MR plate model has an independent approximation of the rotations of the midsurface (for the Kirchhoff plate model the rotation vector equals the lateral

displacement gradient). Besides being more expensive to solve, for a FEM to work asymptotically as the thickness of the plate tends to zero, the discrete MR solution must satisfy the Kirchhoff theory (since the continuous MR solution converges towards the continuous Kirchhoff solution). If too few trial functions fulfils the condition the numerical solution is degraded by locking (should the difference $\nabla u^h - \boldsymbol{\theta}^h$ not vanish the shear energy (24) increases without bound). The situation is particularly difficult for low-order approximations, cf. Section 2.4.2. One useful approach has been to use projections in the shear energy term and consider modified energy functionals (the FE solution minimizes the sum of the bending energy, the shear energy, and the potential of the surface load) of the type

$$\mathfrak{F}_h(u,\boldsymbol{\theta}) := \tfrac{1}{2}a(\boldsymbol{\theta},\boldsymbol{\theta}) + \tfrac{1}{2}b(u,\boldsymbol{R}_h\boldsymbol{\theta};u,\boldsymbol{R}_h\boldsymbol{\theta}) - \int_\Omega fu\,\mathrm{d}x\mathrm{d}y,$$

where $\boldsymbol{R}_h$ is some interpolation or projection operator. This idea underpins, e.g., the MITC element family of Bathe and co-workers, first introduced in [11], and has been used extensively in the mathematical literature to prove convergence, see, e.g., [4, 16, 20, 35]. It should be noted that if the approximation corresponding to $\boldsymbol{R}_h\boldsymbol{\theta}$ was to be used also for the bending energy, the element would be non-conforming and potentially unstable (in effect one has to construct and match three different FE spaces, which indeed is how the approach was originally conceived—as a mixed method with an auxiliary set of unknowns (the shear stresses), cf. [11]).

The MR plate theory has been used in Papers B-D.

2.4 Nitsche's method

Nitsche introduced his method 1971 [30], and then suggested its use for imposing Dirichlet boundary conditions weakly, standing as an alternative to classical methods: the *Lagrange multiplier method* [5] (which has the drawback of introducing additional unknowns, the Lagrange multipliers, in the discrete problem), and the *penalty method* [6]. It was later on, in papers by, e.g., Baker [9], Wheeler [48], and Arnold [3], that piecewise discontinuous approximations and penalty were used to enforce inter-element continuity. On this form Nitsche's method is better known as the *discontinuous Galerkin method*. In [24] Hansbo reviews Nitsche's method, and emphasizes how the real strength of the method is the applicability to interface problems. Exemplified by Poisson's equation Nitsche's method is shown to be consistent (the penalty method, in contrast, is not consistent). Furthermore, the stiffness matrix is positive definite, and optimal order error estimates can be derived with preserved conditioning on quasi-uniform meshes. Recent textbooks on discontinuous Galerkin (dG) methods include Rivière [39].

Nitsche's method has been used in the FE formulations of Papers A-D.

2.4.1 Model problem

Let $\mathfrak{K}_h = \{K\}$ partition $\Omega \subset \mathbb{R}^2$ into a geometrically conforming rectangular mesh, whose edges $\mathfrak{E} = \{E\}$ can be divided into two disjoint subsets, $\mathfrak{E} = \mathfrak{E}_I \cup \mathfrak{E}_B$, where $\mathfrak{E}_I = \mathfrak{E} \setminus \partial\Omega$ and $\mathfrak{E}_B$ are the sets of interior and boundary edges, respectively. Each edge is associated with a fixed unit normal $\boldsymbol{n}$ with direction chosen so that $\boldsymbol{n}$ is the exterior unit normal on the boundary.

Two quantities for functions on edges are introduced: the jump $[\![\cdot]\!]$ and the average $\langle\cdot\rangle$. To this end, let K_1 and K_2 be two neighboring elements, sharing the interior edge E. For a scalar function $v \in V_h$ define

$$
\begin{aligned}
[\![v]\!] &:= v^- - v^+, \quad \text{for } E \in \mathfrak{E}_I, & [\![v]\!] &:= v^-, \quad \text{for } E \in \mathfrak{E}_B,\\
\langle v\rangle &:= \frac{v^- + v^+}{2}, \quad \text{for } E \in \mathfrak{E}_I, & \langle v\rangle &:= v^-, \quad \text{for } E \in \mathfrak{E}_B,
\end{aligned}
$$

where

$$
v^- = \lim_{\epsilon\to 0^+} v(\boldsymbol{x} - \epsilon\boldsymbol{n}), \quad v^+ = \lim_{\epsilon\to 0^+} v(\boldsymbol{x} + \epsilon\boldsymbol{n}), \quad \text{for } \boldsymbol{x} = (x,\, y) \in E.
$$

Let

$$
\mathcal{D}_1 := \{v : v|_K \in \mathcal{P}_1(K) \text{ for all } K \in \mathfrak{K}_h\}
$$

be the space of piecewise linear discontinuous polynomials.

Now consider the following Poisson problem: find seek $u \in \mathcal{C}^2(\Omega)$ such that

$$
\begin{aligned}
-\Delta u &= f, \quad \text{in } \Omega,\\
u &= 0, \quad \text{on } \partial\Omega.
\end{aligned}
\tag{25}
$$

To motivate the formulation of Nitsche's method a Green's formula is derived on an element,

$$
\int_K fv \,\mathrm{d}x\mathrm{d}y = -\int_K \Delta u \,\mathrm{d}x\mathrm{d}y = \int_K \nabla u \cdot \nabla v \,\mathrm{d}x\mathrm{d}y - \int_{\partial K} \boldsymbol{n} \cdot \nabla u v \,\mathrm{d}s,
$$

where $v \in \mathcal{D}_1$. By adding all element equations,

$$
\int_\Omega fv \,\mathrm{d}x\mathrm{d}y = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x\mathrm{d}y - \sum_{E\in\mathfrak{E}} \int_{\partial K} [\![\boldsymbol{n} \cdot \nabla u v]\!] \,\mathrm{d}s,
$$

and since

$$
[\![\boldsymbol{n} \cdot \nabla u v]\!] = \langle \boldsymbol{n} \cdot \nabla u\rangle [\![v]\!] + \underbrace{[\![\boldsymbol{n} \cdot \nabla u\,]\!]}_{=0 \text{ by cont.}} \langle v\rangle = \langle \boldsymbol{n} \cdot \nabla u\rangle [\![v]\!],
$$

it follows that

$$
\int_\Omega fv \,\mathrm{d}x\mathrm{d}y = \int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x\mathrm{d}y - \sum_{E\in\mathfrak{E}} \int_{\partial K} \langle \boldsymbol{n} \cdot \nabla u\rangle [\![v]\!] \,\mathrm{d}s.
$$

Due to continuity the jump in the exact solution vanishes too, so if u solves (25), it also satisfies

$$\begin{aligned}\int_\Omega fv\,\mathrm{d}x\mathrm{d}y = \int_\Omega \nabla u\cdot\nabla v\,\mathrm{d}x\mathrm{d}y \\ -\sum_{E\in\mathfrak{E}}\int_{\partial K}\langle \boldsymbol{n}\cdot\nabla u\rangle[\![v]\!]\,\mathrm{d}s - \sum_{E\in\mathfrak{E}}\underbrace{\int_{\partial K}\langle \boldsymbol{n}\cdot\nabla v\rangle[\![u]\!]\,\mathrm{d}s}_{I_{\text{sym}}} \\ +\gamma\sum_{E\in\mathfrak{E}}\underbrace{\int_{\partial K}h^{-1}[\![u]\!][\![v]\!]\,\mathrm{d}s}_{I_{\text{pen}}} = a(u,v),\end{aligned} \tag{26}$$

where, with $|\cdot|$ denoting the area of K or the length of E,

$$h := \begin{cases}\frac{|K_1|+|K_2|}{2|E|}, & \text{for } E\in\mathfrak{E}_I,\\ |K|/|E|, & \text{for } E\in\mathfrak{E}_B.\end{cases}$$

The corresponding FE problem of (25) is that of finding $u^h\in\mathcal{D}_1$ such that

$$a(u^h,v) = \int_\Omega fv\,\mathrm{d}x\mathrm{d}y, \tag{27}$$

for all $v\in\mathcal{D}_1$ (the variational form is given by replacing the FE space $\mathcal{D}_1$ with $H^1(\Omega)$). The addition of the term I_{sym} serves the purpose of making the method symmetric (this is known as a SIPG method, *symmetric interior penalty Galerkin*), which is a desirable property when solving linear systems of equations, whereas I_{pen} penalizes jumps in the rotation vector. The (computable) penalty parameter γ must be chosen judiciously: it must be large enough for the bilinear form $a(\cdot,\cdot)$ to be coercive, cf. [24], but γ should not be too large either. This could lead to over-constrained problems suffering from locking (the continuity constraint is enforced more severely than the approximation can allow).

2.4.2 An example on numerical locking

Consider the Poisson problem (25) on the unit square with data

$$f(x,y) = 2(x(1-x)+y(1-y)),$$

corresponding to the exact solution

$$u(x,y) = x(1-x)y(1-y),$$

whose mean value is $1/36$. The problem is solved numerically using the dG FEM (27) with the set $\{1,(x-x_0)/\Delta x,(y-y_0)/\Delta y\}$ for local shape

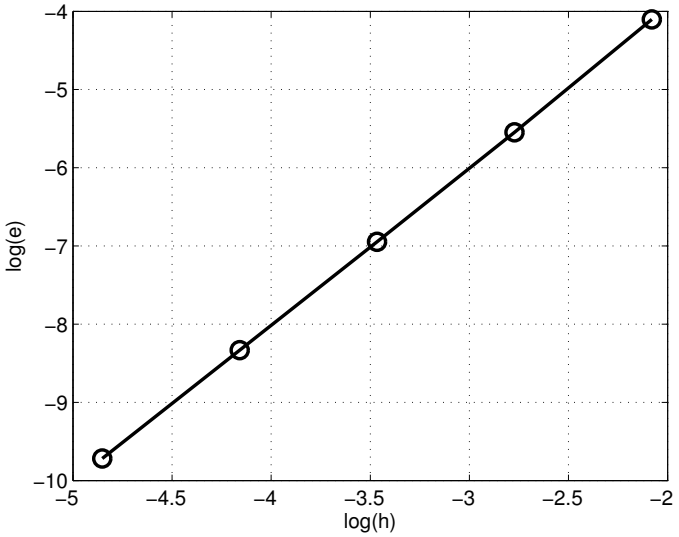

Figure 2.1: *Convergence study using full integration during the assembly and penalty parameter* $\gamma = 10$*. The mean value error* e *decreases quadratically in the mesh size* h *(*$\log(e) = p\log(h) + C$*, where* $p = 2.0219$*).*

functions (on each element it is assumed that $x_0 \leq x \leq x_1$, $\Delta x = x_1 - x_0$, and $y_0 \leq y \leq y_1$, $\Delta y = y_1 - y_0$). Clearly the basis is not nodal-based, and thus the homogeneous Dirichlet boundary conditions are imposed weakly. The implementation of the dG FEM is exemplified in Appendix A.

A convergence analysis, performed on a set of consecutive uniform meshes choosing $\gamma = 10$, indicates an optimal quadratic rate of convergence for the mean displacement, which is in agreement with the order of the linear polynomial approximation, see Figure 2.1. The relative error $e = 6.02 \cdot 10^{-5}$ for the smallest mesh size.

The robustness of the method with respect to the penalty parameter and the mesh size is considered. Typically, on a fixed mesh, γ cannot be chosen too large. Should $\gamma \to \infty$ the approximation is forced to be continuous—all degrees of freedom on the boundary elements are then prescribed, and in fact all interior degrees of freedom will be as well, meaning that the discrete solution locks, cf. Figure 2.2 (where the tendency is particularly evident on coarse meshes). The reason why is that the approximating space contains no $\mathcal{C}^0$ solution (besides the trivial one). On the other hand, had a bilinear element been chosen instead, the method would have been locking-free. The same difficulty underlies locking in the context of plate theory—the difference is that the approximating space then needs to be $\mathcal{C}^1$ continuous, cf. Paper B.

A trick that stabilizes the presented dG FEM is to use *under-integration*, i.e., to employ a numerical quadrature scheme that is not exact (with too

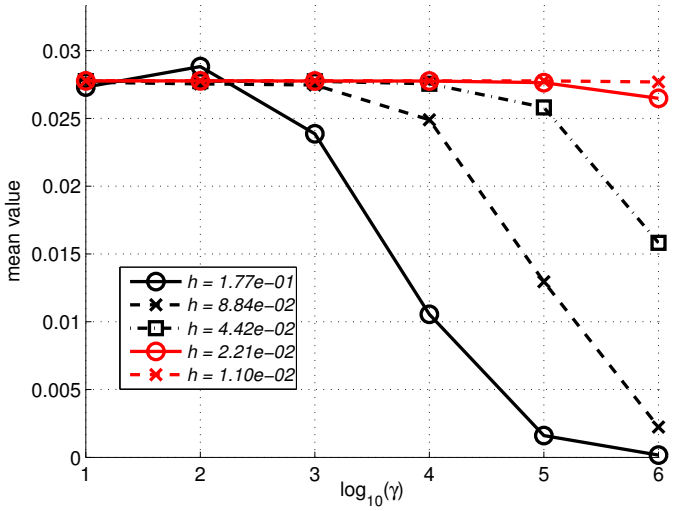

Figure 2.2: *The mean value of u^h for a set of different mesh sizes h as a function of the penalty parameter γ. (The edge terms are assembled using full integration.)*

few abscissas). Relaxing the continuity constraint by enforcing it only in the edge midpoints—corresponds to integrating the penalty term I_{pen} using midpoint quadrature—maintains the optimal rate of convergence, see Park and Sheen [34]. The result, which is independent of mesh size, is confirmed in Figure 2.3.

2.4.3 A comparison to continuous Galerkin methods

A comparison between continuous Galerkin (cG) and dG FEMs is made in [39, Section 2.12]. Practical issues are pointed-out—a few of them are listed below. Firstly, however, it should be mentioned how cG FEMs are some 60 years old, and the theory is well-developed. dG FEMs, on the other hand, are less mature, why much work still lies ahead.

- **Computational cost:** In dG FEMs the number of degrees of freedom DOF_{dG} is proportional to the number of elements (the constant of proportionality depends on the order d of the polynomial approximation), whereas DOF_{cG} depends on the number of elements *and* vertices. As d increases, on a quadrilateral mesh, the ratio $\text{DOF}_{cG}/\text{DOF}_{dG}$ typically increases to be larger than 1 (starting at quartic approximations), i.e., higher-order dG FEMs can be less expensive than the corresponding continuous method.

- **Hanging nodes:** In dG FEMs, since inter-element continuity is not a constraint, elements may carry (multiple) hanging nodes[†]. For a con-

[†]A *hanging node* lies on the common edge E between two neighboring elements, but

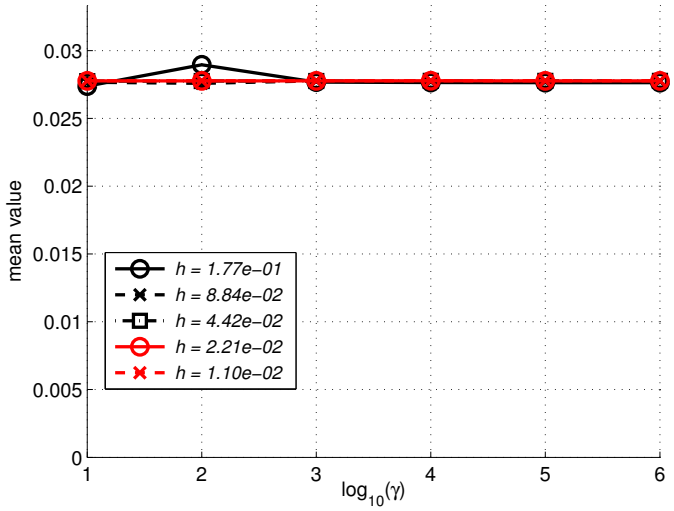

Figure 2.3: *The mean value of u^h for a set of different mesh sizes h as a function of the penalty parameter γ. (The edge terms are under-integrated by imposing the penalty only in their midpoints.)*

tinuous method, however, care must be taken. A possible approach is to use Lagrange multipliers to prescribe nodal values.

- **Order of polynomial approximation:** It is relatively straightforward to change the order of the polynomial approximation in dG FE software without major modifications of the source code being required. Keeping track of the degrees of freedom is more difficult in cG FEMs.

2.5 *A posteriori* error estimation

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.

To achieve this aim, the traditional approach is by means of automatic mesh adaption, using local error indicators. These would be functions of the FE solution, and presumably measure the "local roughness" of the continuous solution, on which the mesh refinement then is based. The overall process involves some distinct steps:

only carries degrees of freedom related to one of them. This situation typically occurs during mesh adaptivity, when the one neighbor is split into four children, making the newly introduced midpoint node on E a hanging node.

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 the two first steps are dealt with in some detail. The discussion will not be comprehensive, but focuses on exemplifying the techniques used in this thesis: 1) solving local problems (done in Paper D); 2) residual-based energy norm control (used in Paper E); and 3) goal-oriented adaptivity (employed in Papers A, B, and D). For a textbook on adaptive FEMs we refer to Bangerth and Rannacher [10].

2.5.1 Solving local problems

Finding improved solutions to be used for, say, estimating *a priori* terms in error representation formulas, can be done by solving a set of small *local* problems. The motivation is for the computational cost to be less than that of solving the refined global problem. For the bilinear approximation, e.g., an elementwise problem could be stated by using a quadratic approximation instead. The boundary values are prescribed by the primal solution, but an additional (unknown) degree of freedom remains in the centroid. This approach is known as solving local *Dirichlet* problems, which was done in Paper D, where patches of elements were considered to obtain a more reliable error estimate. More information on solving local Dirichlet problems can be found in the textbook by Verfürth [47]. (Another variant is to solve local *Neumann* problems—it use for error estimation in hierarchical modeling has been pursued by Stein and Ohnimus [41, 42, 43].)

2.5.2 Residual-based error estimates

The main idea—as opposed to solving local problems—is to substitute the FE solution into the PDE: since u_h is an approximation, it does not satisfy (28) exactly, and this hopefully provides useful information about the error $e = u - u_h$.

For simplicity, let Poisson's equation serve as the model problem, representing a linear elliptic PDE. The continuous problem is that of finding u

such that

$$\begin{aligned} \mathcal{A}u := -\nabla \cdot (k\nabla u) &= f, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega_{\mathrm{D}}, \\ n \cdot k\nabla u &= g, \quad \text{on } \partial\Omega_{\mathrm{N}}, \end{aligned} \tag{28}$$

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

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

$$a(u, v) = L(v), \quad \text{for all } v \in V, \tag{29}$$

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

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

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

$$a(u_h, v) = L(v), \quad \text{for all } v \in V_h, \tag{30}$$

and by subtracting (30) from (29), the Galerkin orthogonality relation

$$a(e, v) = 0, \quad \text{for all } v \in V_h, \tag{31}$$

follows, 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), a norm

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

may be defined, 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.

A means for a posteriori error estimation is outlined. First note that

$$\|e\|_a^2 = a(e, e) = a(u - u_h, e) = a(u, e) - a(u_h, e) = L(e) - a(u_h, e), \tag{32}$$

via (29) 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 (30), 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,

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

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, terms may be reordered, and so

$$\begin{aligned}\|e\|_a^2 = &\sum_{K\in\mathcal{T}_h}\int_K (f+\nabla\cdot k\nabla u_h)(e-\pi_h e)\,\mathrm{d}x\mathrm{d}y\\ &+\int_{\partial\Omega_\mathrm{N}}(g-\boldsymbol{n}\cdot k\nabla u_h)(e-\pi_h e)\,\mathrm{d}s\\ &+\sum_{E\in\mathcal{E}_I}\int_E [\![\boldsymbol{n}_E\cdot k\nabla u_h]\!](e-\pi_h e)\,\mathrm{d}s,\end{aligned} \tag{33}$$

where

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

is 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 on the form

$$\|e\|_a^2 \le \sum_{K\in\mathcal{T}_h}\omega_K\rho_K, \tag{34}$$

by applying Cauchy's inequality elementwise to (33), 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.

(34) is an *a posteriori* error estimate, i.e., an estimate in terms of the computed solution and the problem data, constituting an upper bound of the error. However, since the estimate is based on Cauchy's inequality it may not be sharp. An alternative is to solve an auxiliary problem on a refined mesh, interpolate the enhanced discrete solution onto the primal mesh, and let the difference provide information about the error, cf. Paper E.

2.5.3 Goal-oriented adaptivity

The traditional approach to adaptivity was to measure the error in global norms (like the energy or L_2 norms). However, often it can be relevant to control the error in local physical quantities, say, the maximum deflection of a plate in a point or along line, or the largest stress value.

In order to assess this error one may use duality techniques, which essentially means multiplying 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 consider the continuous dual problem,

$$\mathcal{A}^\mathrm{T} z = j, \tag{35}$$

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

$$(\mathcal{A}^{\mathrm{T}} z, \psi) = (z, \mathcal{A}\psi). \tag{36}$$

If follows that

$$(e, j) \overset{(35)}{=} (e, \mathcal{A}^{\mathrm{T}} z) \overset{(36)}{=} (\mathcal{A}e, z) = (f - \mathcal{A}u_h, z) \overset{(31)}{=} (f - \mathcal{A}u_h, z - \pi_h z), \tag{37}$$

using the Galerkin orthogonality for $v = \pi_h z \in V_h$. Here, as compared to (32), one is free to choose the data j according to the quantity to be controlled adaptively. Should j is taken as an approximate Dirac delta function, e.g., the LHS of (37) reduces to the error in the corresponding point.

Example. In order to concretize (37) Poisson's equation is tested against a function z:

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

by using integration by parts twice. Taking the Neumann boundary conditions into account (u is prescribed on the Dirichlet boundary so no errors occur there), 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. Note that

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

and then, via (37) for data j_1 and j_2,

$$\begin{aligned}(e, j_1) + (e, j_2)_{\partial\Omega_{\mathrm{N}}} &= \int_\Omega k\nabla z \cdot \nabla e \,\mathrm{d}x\mathrm{d}y = a(e, z) \overset{(31)}{=} a(e, z) - a(e, \pi_h z) \\ &= a(u - u_h, z - \pi_h z) = a(u, z - \pi_h z) - a(u_h, z - \pi_h z).\end{aligned}$$

By expressing the first bilinear form $a(\cdot, \cdot)$ 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 for control of functionals of the error both inside the domain and on the Neumann boundary.

(37) is an exact error representation formula, but it is emphasized that the dual problem cannot be solved exactly, at least not in general, and consequently has to be approximated. A tempting idea, from a practical point-of-view, would be to simply reuse the primal mesh: find $z_h \in V_h$ such that

$$a(v, z_h) = (j, v) = J(v), \quad \text{for all } v \in V_h, \tag{38}$$

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

$$J(e) \overset{(37)}{=} (f - \mathcal{A}u_h, z_h - \pi_h z) = 0,$$

since z_h and $\pi_h z_h$ will coincide. Hence the approximation of the dual solution typically has to belong to an enhanced approximating space V_h^*. In Papers A and B this space was constructed by subjecting the primal mesh to regular subdivison. Then the dual problems were solved globally with respect to V_h^*. This may be straightforward, but is also computationally demanding. Another approach is to solve local problems, cf. Section 2.5.1 and Paper D. An analysis of different goal-oriented *a posteriori* error measures in the context of non-linear elasticity is given by Larsson *et al.* [26].

Using duality arguments for *a posteriori* error estimation was introduced during the early 1990s, e.g., in work by Eriksson *et al.* [21]. The ideas later developed into the DWR method by Becker and Rannacher [13, 12]. 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.* [31, 33, 46] (solid/fluid mechanics applications and heterogeneous materials) and Braack and Ern [14] (Poisson's equation, convection-diffusion-reaction equations). More recent work in model and goal adaptivity, within the field of multiscale modeling, includes Oden *et al.* [32].

Chapter 3

Summary

3.1 Main results

Paper A In *An adaptive finite element method for second-order plate theory*, the Kirchhoff plate model is supplemented by a second-order term to account for the effects of membrane stresses. The method is based on C^0 continuous $\mathcal{P}_2$ approximations on simplices for the out-of-plane deformations, whereas the in-plane deformations are approximated using a constant-strain element. An *a posteriori* error estimate is derived, which separates the bending and membrane effects, for control of a linear functional of the error. It is evaluated with respect to maximum plate deflection under various loading conditions. Effectivity indices close to unity suggest accurate error control.

Paper B In *A finite element method with discontinuous rotations for the Mindlin-Reissner plate model*, a c/dG FEM for the MR plate model is presented, based on continuous polynomials of degree $d \geq 2$ for the transverse displacements, and discontinuous polynomials of degree $d-1$ for the rotations. *A priori* convergence estimates, uniformly in the thickness of the plate, are proved and thus show that locking is avoided. Moreover, *a posteriori* error estimates are derived based on duality, together with corresponding adaptive procedures for controlling linear functionals of the error. Numerical results are presented.

Paper C In *Model combination in plate bending problems*, a FEM which combines the Kirchhoff and Mindlin-Reissner plate models is introduced. Hence, in different subregions of a physical plate, elements of either Kirchhoff or Mindlin-Reissner type can be used. The lateral displacements are approximated by continuous quadratic polynomials, and the rotations of the midsurface are approximated by discontinuous linear polynomials. To detail the suggested approach a matrix formu-

lation of the discrete problem, suitable for implementation, is given. The strength of the method lies in the ease at which the model is refined on the element level—by introducing additional rotational degrees of freedom, with weakly enforced inter-element continuity using Nitsche's method. The rationale is that of providing a means for model adaptivity.

Paper D In *A two-model adaptive finite element method for plates*, a goal-oriented adaptive c/dG FEM, which combines the Kirchhoff and MR plate models, is introduced. The lateral displacements are approximated by quadratic continuous polynomials, and the rotations of the mid-surface are approximated by linear discontinuous polynomials, cf. Paper C. A duality-based *a posteriori* error representation separates the discretization and modeling errors, and in this sense local mesh and model refinement are independent. The target quantity of interest is an arbitrary linear functional of the displacement and/or rotation errors.

Paper E In *Hierarchical modeling for elasticity on thin domains*, the two-dimensional linear elasticity problem is posed on thin domains. A model hierarchy is constructed based on increasingly higher polynomial expansions through the thickness of the domain, and it appears to be a natural extension of the Bernoulli and Timoshenko beam theory. An adaptive FEM is presented which uses standard tensor-product Lagrangian elements. An energy norm *a posteriori* error estimate is outlined, providing local error indicators to govern refinements of mesh and model. Some numerical results are presented, which indicates accurate error control, but this is not obvious—the error is only bounded upwards by a constant (the upped bound itself is approximated).

3.2 Future work

Paper A An interesting engineering application would be to apply the suggested method to the plate buckling problem.

Paper B Following this work, the long-term goal was to combine the Kirchhoff and MR plate models based on the c/dG FEM introduced in Papers A and B, which also became the subject of Papers C and D.

Paper C It remained to develop an *a posteriori* error representation, which separates the discretization and modeling errors. This was analyzed in Paper D.

Paper D An interesting development would be to consider an extension of the model hierarchy, where, say, the master model is replaced by the three-dimensional linear elasticity theory. As in Paper E such refined models could be constructed by increasing the order of the polynomial approximation through the thickness of the domain. In [24] the similar problem is studied, where Nitsche's method is used to couple the Bernoulli beam and the Kirchhoff plate models with elasticity.

Paper E Implementing goal-oriented adaptivity based on duality techniques would be relevant in the context of engineering applications. The model hierarchy should be extended to include, e.g., the Bernoulli and Timoshenko beam theory.

Appendix A

Implementation

In this chapter the source code of the dG FEM, as specified in Sections 2.4.1 and 2.4.2, is presented in detail. The code written in MATLAB [27] for flexibility: the interpreted language is based on the abstraction of matrices, using a consistent set of internal data structures, allowing for relatively effortless FE implementations. Debugging tools facilitates the code development. The built-in visualization capabilities are powerful enough for most purposes. Hence the programming environment suits the needs of research codes well, where the primary objective is evaluation of algorithms. Hard time spent on optimization is less worthwhile, as the final code may only be executed a few times. A viable alternative, however, with the benefits of being distributed under an open-source license, is to use Python. Production codes on the other hand, that run on a daily basis, have different demands. Usually they rely on compiled languages, say, Fortran or C/C++. (A good compromise is to employ MATLAB/Python as a steering language, which calls compiled routines for doing computationally demanding work.)

Two routines are not included: `rmesh` and `gauss1D`. The former constructs a rectangular mesh (basically using `meshgrid`), with elements and nodes numbered columnwise starting at the lower left (element nodes are numbered counter-clockwise). The latter simply returns the abscissas and weights for one-dimensional Gaussian quadrature. `sparse2`, the routine used to construct the sparse stiffness matrix, is part of CHOLMOD [17], a set of ANSI C routines for sparse Cholesky factorization. It is included in the SuiteSparse package, downloadable at the MATLAB Central or [19].

Besides exemplifying how dG FEMs are implemented in general, focus is on showing how the assembly process can be readily parallelized using `parfor` loops, thereby taking advantage of hardware with multiple cores. A small benchmark, tabulated in Table A.1, indicates that assembling the stiffness matrix and load vector *in parallel* scales well once the size of the problem increases. The most time-consuming part of the problem is then

to solve the resulting linear system of equations. The test system includes 4 Intel Xeon X5650@2.67GHz CPUs, 48 GB RAM, and runs RHEL 5.5.

Table A.1: *A benchmark for solving the Poisson problem* (25)*, specified in Sections 2.4.1 and 2.4.2, using the dG FEM with either serial or parallel assembly. All times are measured in seconds. (*N *is the number of degrees of freedom;* `sparse2` *is the time for constructing the sparse stiffness matrix using the same routine;* `mldivide` *is the time for solving the linear system of equations using the same routine;* assembly *is the time for the* `PARFOR` *loop;* speed-up *is the scaling factor when parallelizing the assembly process, i.e., the* `PARFOR` *loop, using a pool of 8 local workers.)*

N	`sparse2`	`mldivide`	*serial* assembly	*parallel* assembly	speed-up
3 702	0.01	0.02	0.44	0.13	3.51
12 288	0.03	0.14	1.65	0.28	5.90
49 152	0.12	1.03	6.55	0.97	6.72
196 608	0.47	11.27	26.23	3.81	6.88
786 432	1.92	109.57	104.31	15.01	6.95

```
function [umean, err] = poissonDG(xint, yint, gamma)

%
% Input data:
%
%   xint:  x-coordinates (e.g., xint = linspace(0, 1, n);),
%   yint:  y-coordinates (e.g., yint = linspace(0, 1, m);),
%   gamma: penalty parameter.
%

%
% Construct rectangular mesh:
%
%   nodes: element node numbers (stored columnwise),
%   xnod:  nodal x-coordinates,
%   ynod:  nodal y-coordinates,
%   nei:   element neighbors (stored columnwise; 0 is boundary),
%   nele:  number of elements.
%

[nodes, xnod, ynod, nei, nele] = rmesh(xint, yint);

A = zeros(nele, 1);    % ---> element areas
nedges = 4;            % ---> number of element edges
edof = 3;              % ---> number of element DOFs
ndof = edof * nele;    % ---> number of DOFs


% ------------------------------------------------------ Assembly process:

%
% Quadrature data:
%

% Polynomial degree of transverse load in x- and y-directions:
xdeg = 2;
ydeg = 2;

% Polynomial degree of penalty term (pdeg = 1 implies under-integration):
pdeg = 2;

% Abscissas and weights in x-direction...
[x0, w1] = gauss1D(xdeg, 0, 1);
w1 = w1(ones(1, ydeg), :)';

% ...and in y-direction (for domain assembly):
if xdeg ~= ydeg
  [y0, w2] = gauss1D(ydeg, 0, 1);
  w2 = w2(ones(1, xdeg), :);
else
  y0 = x0;
  w2 = w1;
end

% 2D weights:
wei = w1(:) .* w2(:);

% Abscissas and weights (for edge assembly):
[gs, gw] = gauss1D(pdeg, 0, 1);

% Number of abscissas:
ngp = length(gw);
```

```
% Diagonal matrix for "vectorized" quadratue:
diagGW = diag(gw);

%
% Coordinate triplets:
%

% Laplacian:
row = zeros(edof^2, nele);    % ---> SPARSE doesn't support integer types
col = row;
val = row;

% Load:
rowLoad = zeros(edof, nele);
valLoad = zeros(edof, nele);

% Edge terms:
rowEdge = zeros(nedges * (2 * edof)^2, nele);
colEdge = rowEdge;
valEdge = rowEdge;

parfor iel = 1 : nele    % ---> loop w.r.t. all elements

  %
  % Local element data:
  %

  % Degrees of freedom:
  dof = (edof * iel - (edof - 1) : edof * iel)';

  % Nodal coordinates (x, y):
  nod = nodes(:, iel);
  x = xnod(nod);
  y = ynod(nod);

  % Side lengths:
  dx = x(2) - x(1);
  dy = y(3) - y(2);

  % Area:
  A(iel) = dx * dy;

  % Neighbors:
  locNei = nei(:, iel);

  %
  % Quadrature data:
  %

  % Transform abscissas (onto I1):
  x1 = (1 - x0) * x(1) + x0 * x(2);    % ---> I1 = [x(1), x(2)]
  y1 = (1 - y0) * y(2) + y0 * y(3);    % ---> I1 = [y(2), y(3)]
  xrep = x1(:, ones(1, ydeg));
  gx = xrep(:);
  yrep = y1(:, ones(1, xdeg))';
  gy = yrep(:);

  %
  % Volume integrals (load, laplace):
  %
```

```
% Gradients of shape functions:
grad = [0, 1 / dx, 0; 0, 0, 1 / dy];    % ---> const. for linear approx.

% Shape functions:
phi = [ones(length(gx), 1), (gx - x(1)) / dx, (gy - y(1)) / dy];

% Transverse load:
f = 2 * (gx .* (1 - gx) + gy .* (1 - gy));

load = phi' * (f .* wei) * A(iel);
laplace = grad' * grad * A(iel);

% Store coordinate triplets:
r1 = dof(:, ones(length(dof), 1));    % ---> Laplacian
row(:, iel) = r1(:);
c1 = r1';
col(:, iel) = c1(:);
val(:, iel) = laplace(:);
rowLoad(:, iel) = dof;                % ---> load
valLoad(:, iel) = load(:);

%
% Edge assembly:
%
% Use local triplets (irow, icol, and ival) to allow slicing of PARFOR-
% variables.
%

irow = ones((2 * edof)^2, nedges);    % ---> zero indices not allowed
icol = irow;
ival = zeros((2 * edof)^2, nedges);

% Local edge nodes:
enodes = [1 2; 2 3; 3 4; 4 1]';    % ---> recreate on each worker

for iedge = 1 : nedges    % ---> loop w.r.t. local edges

  %
  % Local edge data:
  %

  % Vertex coordinates (vx, vy):
  enod = enodes(:, iedge);
  vx = x(enod);
  vy = y(enod);

  % Side length (ds):
  xs = vx(2) - vx(1);
  ys = vy(2) - vy(1);
  ds = sqrt(xs^2 + ys^2);    % ---> max(abs([sx, sy]))

  % Unit normal:
  nvec = [ys, -xs] / ds;

  % Transform abscissas (onto I1):
  xe = (1 - gs) * vx(1) + gs * vx(2);    % ---> I1 = [vx(1), vx(2)]
  ye = (1 - gs) * vy(1) + gs * vy(2);    % ---> I1 = [vy(1), vy(2)]

  % Shape functions:
  phi = [ones(ngp, 1), (xe - x(1)) / dx, (ye - y(1)) / dy]';

  % Neighbor:
```

```
edgeNei = locNei(iedge);

if edgeNei   % ---> interior edge

  %
  % Local element data (on neighbor):
  %

  % Degrees of freedom:
  dofn = (edof * edgeNei - (edof - 1) : edof * edgeNei)';

  % Nodal coordinates:
  nodn = nodes(:, edgeNei);
  xn = xnod(nodn);
  yn = ynod(nodn);

  % Side lengths:
  dxn = xn(2) - xn(1);
  dyn = yn(3) - yn(2);

  % Area:
  An = dxn * dyn;

  %
  % Edge integrals (sympen):
  %

  % Shape functions:
  phin = [ones(ngp, 1), (xe - xn(1)) / dxn, (ye - yn(1)) / dyn]';

  % Gradients of shape functions:
  gradn = [0, 1 / dxn, 0; 0, 0, 1 / dyn];    % ---> constant

  % Mean value of normal derivatives:
  meanND = 0.5 * nvec * [grad, gradn];    % ---> per definition

  % Jump in shape functions:
  JV = [phi; -phin];

  % h-parameter:
  h = 2 * ds / (A(iel) + An);    % ---> by definition

  JM = sum(JV * diagGW, 2) * meanND;
  sympen = 0.5 * ds * (gamma / h * (JV * diagGW * JV') - (JM + JM'));

  % Store coordinate triplets (on present edge):
  idof = [dof; dofn];
  r2 = idof(:, ones(length(idof), 1));
  irow(:, iedge) = r2(:);
  c2 = r2';
  icol(:, iedge) = c2(:);
  ival(:, iedge) = sympen(:);

else                        % ---> boundary edge
  meanND = nvec * grad;
  h = ds / A(iel);
  JM = sum(phi * diagGW, 2) * meanND;
  sympen = ds * (gamma / h * (phi * diagGW * phi') - (JM + JM'));
  r2 = dof(:, ones(edof, 1));
  irow(1 : edof^2, iedge) = r2(:);
  c2 = r2';
  icol(1 : edof^2, iedge) = c2(:);
```

```
      ival(1 : edof^2, iedge) = sympen(:);
    end

    % Store coordinate triplets (on all edges):
    rowEdge(:, iel) = irow(:);
    colEdge(:, iel) = icol(:);
    valEdge(:, iel) = ival(:);

  end
end


% ------------------------------------------------- Solve linear system:

% Construct stiffness matrix:
S = sparse2([row(:); rowEdge(:)], [col(:); colEdge(:)], ...
            [val(:); valEdge(:)], ndof, ndof);

% Construct load vector:
f = sparse2(rowLoad(:), 1, valLoad(:), ndof, 1);

% Discrete solution:
u = S \ f;

% Discrete solution in vertices (unod):
nodval = [1 0 0; 1 1 0; 1 1 1; 1 0 1];     % ---> CCW order
unod = nodval * reshape(u, edof, nele);

% Error in relative mean displacement (err):
umean = sum(unod) * A / 4;
uexact = 1 / 36;
err = (umean - uexact) / uexact;
```

Bibliography

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

[2] J. H. ARGYRIS, I. FRIED, AND D. W. SCHARPF, *The TUBA family of plate elements for the matrix displacement method*, Aero. J. Roy. Aero. Soc., 72 (1968), pp. 701–709.

[3] D. N. ARNOLD, *An interior penalty finite element method with discontinuous elements*, SIAM J. Num. Anal., 19 (1982), pp. 742–760.

[4] D. N. ARNOLD AND R. S. FALK, *A uniformly accurate finite element method for the Reissner-Mindlin plate*, SIAM J. Num. Anal., 26 (1989), pp. 1276–1290.

[5] I. BABUŠKA, *The finite element method with Lagrangian multipliers*, Numer. Math., 20 (1973), pp. 179–192.

[6] ——, *The finite element method with penalty*, Math. Comp., 27 (1973), pp. 221–228.

[7] 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 hierarchical modeling*, Applied Numerical Mathematics, 14 (1994), pp. 5–21.

[8] I. BABUŠKA AND C. SCHWAB, A posteriori *error estimation for hierarchic models of elliptic boundary value problems on thin domains*, SIAM J. Num. Anal., 33 (1996), pp. 221–246.

[9] G. A. BAKER, *Finite element methods for elliptic equations using nonconforming elements*, Math. Comp., 31 (1977), pp. 45–59.

[10] W. BANGERTH AND R. RANNACHER, *Adaptive finite element methods for differential equations*, Birkhäuser Verlag, 2003.

[11] K. J. BATHE AND E. N. DVORKIN, *A four-node plate bending element based on Mindlin/Reissner plate theory and a mixed interpolation*, Int. J. Numer. Meth. Engng., 21 (1985), pp. 367–383.

[12] R. BECKER AND R. RANNACHER, *A feed-back approach to error control in finite element methods: basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.

[13] ——, *An optimal control approach to a posteriori error estimation in finite element methods*, Acta Numerica, 10 (2001), pp. 1–102.

[14] M. BRAACK AND A. ERN, *A posteriori control of modeling errors and discretization errors*, SIAM: Multiscale Modeling and Simulation, 1 (2003), pp. 221–238.

[15] S. C. BRENNER AND L. R. SCOTT, *The mathematical theory of finite element methods*, Springer-Verlag, 2:nd ed., 2002.

[16] G. BREZZI, M. FORTIN, AND R. STENBERG, *Error analysis of mixed-interpolated elements for Reissner-Mindlin plates*, Math. Models Meth. Appl. Sci., 1 (1991), pp. 125–151.

[17] Y. CHEN, T. A. DAVIS, W. W. HAGER, AND S. RAJAMANICKAM, *Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate*, ACM Transactions on Mathematical Software, 35 (2008).

[18] P. G. CIARLET, *The finite element method for elliptic problems*, vol. 4 of Studies in mathematics and its applications, North-Holland, 1:st ed., 1978.

[19] T. A. DAVIS. `http://www.cise.ufl.edu/research/sparse/`.

[20] R. DURAN AND E. LIBERMAN, *On mixed finite-element methods for the Reissner-Mindlin plate model*, Math. Comp., 58 (1992), pp. 561–537.

[21] K. ERIKSSON, D. ESTEP, P. HANSBO, AND C. JOHNSON, *Introduction to adaptive methods for differential equations*, Acta Numerica, (1995), pp. 105–158.

[22] ——, *Computational differential equations*, Studentlitteratur, 1996.

[23] A. ERN AND J.-L. GUERMOND, *Theory and practice of finite elements*, Springer-Verlag, 2004.

[24] P. HANSBO, *Nitsche's method for interface problems in computational mechanics*, GAMM-Mitt., 28 (2005), pp. 183–206.

[25] G. KIRCHHOFF, *Über das Gleichgewicht und die Bewegung einer elastichen Scheibe*, Journal für die reine und angewandte Mathematik, 40 (1850), pp. 51–88.

[26] F. LARSSON, P. HANSBO, AND K. RUNESSON, *Strategies for computing goal-oriented a posteriori error measures in non-linear elasticity*, Int. J. Numer. Meth. Engng., 55 (2002), pp. 879–894.

[27] MATHWORKS™. `http://www.mathworks.com/`.

[28] R. D. MINDLIN, *Influence of rotary inertia and shear on flexural motion of isotropic elastic plates*, J. Appl. Mech., 18 (1951), pp. 31–38.

[29] L. S. D. MORLEY, *The triangular equilibrium element in the solution of plate bending problems*, Aero. Quart., 19 (1968), pp. 149–169.

[30] 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), pp. 9–15.

[31] J. T. ODEN AND S. PRUDHOMME, *Estimation of modeling error in computational mechanics*, J. Comput. Phys., 182 (2002), pp. 496–515.

[32] J. T. ODEN, S. PRUDHOMME, A. ROMKES, AND P. BAUMAN, *Multiscale modeling of physical phenomena: adaptive control of models*, SIAM J. Sci. Comput., 28 (2006), pp. 2359–2389.

[33] J. T. ODEN AND K. VEMAGANTI, *Estimation of local modeling error and goal-oriented adaptive modeling of heterogeneous materials: part I*, J. Comput. Phys., 164 (2000), pp. 22–47.

[34] C. PARK AND D. SHEEN, *$\mathcal{P}_1$-nonconforming quadrilateral finite element methods for second-order elliptic problems*, SIAM J. Numer. Anal., 41 (2003), pp. 624–640.

[35] J. PITKÄRANTA, *Analysis of some low-order finite-element schemes for Mindlin-Reissner and Kirchhoff plates*, Numer. Math., 53 (1988), pp. 237–254.

[36] E. REISSNER, *The effects of transverse shear deformation on the bending of elastic plates*, J. Appl. Mech., 12 (1945), pp. 69–77.

[37] ——, *On bending of elastic plates*, Quart. Appl. Math., 5 (1947), pp. 55–68.

[38] S. REPIN, S. SAUTER, AND A. SMOLIANSKI, A posteriori *error estimation of dimension reduction errors for elliptic problems on thin domains*, SIAM J. on Num. Anal., 42 (2004), pp. 1435–1451.

[39] B. RIVIÈRE, *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*, Frontiers in applied mathematics, SIAM, 2008.

[40] C. SCHWAB, *A posteriori error estimation for hierarchical plate models*, Numer. Math., 74 (1996), pp. 221–259.

[41] 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), pp. 107–119.

[42] ——, *Coupled model- and solution-adaptivity in the finite element method*, Comput. Methods Appl. Mech. Engrg., 150 (1997), pp. 327–350.

[43] ——, *Anisotropic discretization- and model-error estimation in solid mechanics by local Neumann problems*, Comput. Methods Appl. Mech. Engrg., 176 (1998), pp. 363–385.

[44] G. STRANG AND G. FIX, *An analysis of the finite element method*, Wellesley-Cambridge Press, 2:nd ed., 2008.

[45] K. VEMAGANTI, *Local error estimation for dimensionally reduced models of elliptic boundary value problems*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 1–14.

[46] K. VEMAGANTI AND J. T. ODEN, *Estimation of local modeling error and goal-oriented adaptive modeling of heterogeneous materials: Part II*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 6089–6124.

[47] R. VERFÜRTH, *A review of* a posteriori *error estimation and adaptive mesh-refinement techniques*, Wiley and Teubner, 1996.

[48] M. F. WHEELER, *An elliptic collocation-finite element method with interior penalties*, SIAM J. Numer. Anal., 15 (1978), pp. 152–161.