

Article

# Resolving Boundary Layers with Harmonic Extension Finite Elements

Harri Hakula 

Department of Mathematics and Systems Analysis, Aalto University, Otakaari 1, FI-00076 Aalto, Finland; Harri.Hakula@aalto.fi

**Abstract:** In recent years, the standard numerical methods for partial differential equations have been extended with variants that address the issue of domain discretisation in complicated domains. Sometimes similar requirements are induced by local parameter-dependent features of the solutions, for instance, boundary or internal layers. The adaptive reference elements are one way with which harmonic extension elements, an extension of the  $p$ -version of the finite element method, can be implemented. In combination with simple replacement rule-based mesh generation, the performance of the method is shown to be equivalent to that of the standard  $p$ -version in problems where the boundary layers dominate the solution. The performance over a parameter range is demonstrated in an application of computational asymptotic analysis, where known estimates are recovered via computational means only.

**Keywords:** finite element method;  $p$ -version; harmonic extensions



**Citation:** Hakula, H. Resolving Boundary Layers with Harmonic Extension Finite Elements. *Math. Comput. Appl.* **2022**, *27*, 57. <https://doi.org/10.3390/mca27040057>

Academic Editor: Gianluigi Rozza

Received: 27 May 2022

Accepted: 4 July 2022

Published: 8 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

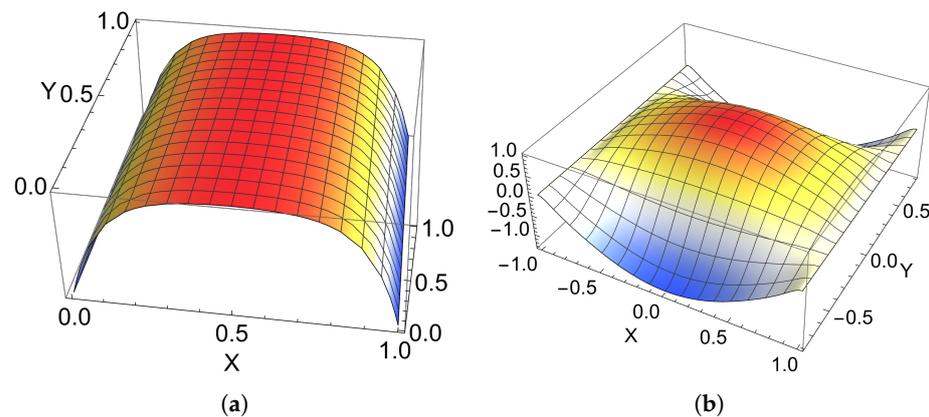
## 1. Introduction

In many practical engineering applications, either the computational domains or indeed the solutions have features that ultimately lead to more expensive finite element simulations than what at first would appear to be necessary. The numerical analysis community has responded to this challenge by introducing different ways for discretising the underlying partial differential equations (PDEs). The methods can either be continuous or discontinuous, even if the problem itself is continuous, or alternatively, the domain discretisations can either be simplicial or polygonal and not necessarily conforming. Notable examples of new developments are virtual elements [1] and the so-called hybrid high-order method [2].

In this paper, the focus is on problems with boundary layers and their resolution with the so-called adaptive reference elements, an  $hp$ -finite element method (FEM) (see, for example, [3,4]) where a conforming formulation is constructed on a non-conforming mesh by adapting the shape functions as harmonic extensions [5]. The theoretical foundations for the method can be directly adopted from Weißer's work on boundary element based FEM [6,7]. The work by Owall is another important reference [8,9].

Examples of solution profiles with boundary layers are shown in Figure 1. From the mesh generation point of view, the issue with standard conforming triangulation is that the effect which is essentially one-dimensional introduces a characteristic length scale to the mesh, which leads to superfluous elements unless the mesh is properly aligned and, simultaneously, the polynomial order is sufficiently high.

The process by which the discretisation is performed so that the parameter-dependent features are properly captured is referred to as the boundary layer resolution. In certain problems, such as shell structures, the solution can be interpreted as a linear combination of features, each with its own characteristic length scale. This includes the so-called smooth component, which spans the entire domain. In order to avoid numerical locking, it is advantageous to use high-order methods for capturing the smooth component accurately [10,11].



**Figure 1.** Solutions of model problems. In elasticity problems, the strongest boundary layers may occur in other vector field components. (a) Reaction–diffusion. (b) Pitkäranta Cylinder. Detail of transverse deflection (one quarter of the cylinder unfolded).

The novelty of this work is that the efficiency of the adaptive reference elements compared with the standard  $p$ -version is demonstrated also in the solution of problems with boundary layers. Furthermore, for problems with linear systems with identical parameter-dependent structures, it is possible to perform computational asymptotic analysis, that is, learn how the overall solution changes as the parameter, for instance, tends to zero. It is known that for thin cylindrical shells depending on the loading and kinematical constraints, it is possible to derive parameter-dependent asymptotic error amplification factors, indicating whether the discretisation has to be adjusted as the parameter changes. In the context of a thin cylindrical benchmark shell, two known error amplification factors are recovered within the numerical experiments below via computation only. This shows that the proposed method maintains its efficiency over the whole practical parameter range.

The rest of this paper is structured as follows: In Section 2, first the boundary layers are introduced formally, and next the basic construction of the harmonic extension shape functions is described before the model problems. The replacement rule-based mesh generation is discussed in more detail in Section 3. Computational asymptotic analysis is developed in Section 4, including the description of the framework for recovering quantities of interest. Numerical examples followed by conclusions are in the next two sections.

## 2. Preliminaries

In this section, the background concepts, namely boundary layers, construction of the 2D adaptive reference elements, and the model problems, are introduced. Familiarity with the basic concepts of the  $p$ - and  $hp$ -versions of the finite element method is assumed. The  $hp$ -solver is implemented with Mathematica [12]—for its design principles, see [13].

### 2.1. Boundary Layers

The theory of one-dimensional  $hp$ -approximation of boundary layers is due to Schwab [4]. Boundary layer functions are of the form

$$u(x) = \exp(-a x / \delta), \quad 0 < x < L, \quad (1)$$

where  $\delta \in (0, 1]$  is a small parameter,  $a > 0$  is a constant, and  $L$  is the characteristic length scale of the problem under consideration. Even though the classical  $p$ -method, see, for example, [3], is capable of asymptotic superexponential convergence, the judicious choice of a minimal number of elements using *a priori* knowledge of the boundary layers leads to far more efficient solution in the practical range of  $p$ . Moreover, in certain classes of problems, it is possible to choose a robust strategy leading to convergence uniformity in  $\delta$ . However, the distribution of the mesh nodes depends on  $p$ , and over a range of polynomial degrees  $p = 2, \dots, 8$ , say, the mesh is different for every  $p$ ! In 1D, this is relatively simple,

but in 2D, much more difficult since we must allow for the mesh topology to change over the range of polynomial degrees. This is the topic of Section 3 below.

In the absence of exact analysis for all of the problem classes discussed below, the central result is given in the form of a definition independent of the dimension of the problem.

**Definition 1 (Layer Element Width).** *For every boundary layer in the problem, optimal convergence can be guaranteed if an element of width  $O(p \delta)$  is aligned in the direction of the decay of the layer.*

Notice, that with  $c$  constant, if  $c p \delta \rightarrow L$  as  $p$  increases, the standard  $p$ -method can be interpreted as the limiting method.

Boundary layers can also occur within the domains, i.e., be internal layers, or emanate from a point. For the discussion here, it is useful to define the concept of boundary layer generators (see [14]).

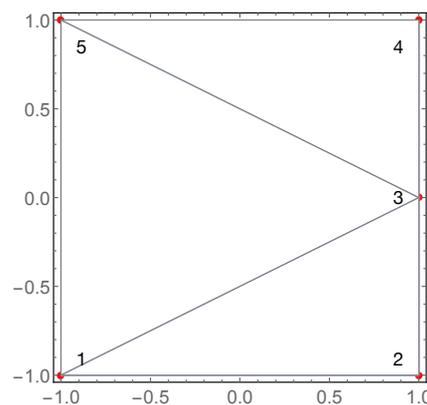
**Definition 2 (Layer Generator).** *The subset of the domain from which the boundary layer decays exponentially, is called the layer generator. Formally, the layer generator is of measure zero.*

The layer generators are independent of the length scale of the problem under consideration.

### 2.2. Adaptive Reference Elements

As usual, in finite elements, every element is an instance of some reference element. To fix terminology every *adaptive element* (AE) is said to be an instance of an adaptive reference element (ARE). The adaptive reference element is defined with a set of points or nodes as usual. What is different is that a subset of nodes is used to define the mapping of the element.

Let us consider the example in Figure 2, where an adaptive reference element is shown. The quadrilateral has *five* nodes, divided into four mapping nodes and one edge node. This choice is not unique, however. For instance, the adaptive reference element (ARE) corresponding to element *A* could be a triangle with mapping nodes  $\{1,2,4\}$  and two edge (hanging) nodes 3 and 5, where the edge  $\{1,5,4\}$  would be curved and passing through 5.



**Figure 2.** Adaptive reference element *A*: Quadrilateral with five nodes. Minimal implementation mesh with three triangles. The element mapping is defined using the four corner nodes only.

**Definition 3. (Planar adaptive reference element (ARE))**

*Given a set of points  $K$ ,  $|K| \geq 3$ , any partition of  $K$  into mapping and edge nodes is admissible, if the edge nodes lie on the boundary of some valid (univalent) mapping of the standard reference element defined by the mapping points. This partition defines the adaptive reference element. If the set of edge nodes is empty, the adaptive reference element is equivalent to a standard element.*

### 2.2.1. Shape Functions

The shape functions are computed as harmonic extensions of the restrictions of the standard FEM nodal and edge functions on the element boundary. In other words, for any standard nodal or edge shape function  $\phi(x, y)$  we compute its harmonic extension  $\varphi(x, y)$ :

$$\begin{cases} \Delta\varphi(x, y) = 0, & \text{in } K, \\ \varphi(x, y) = \phi(x, y)|_{\partial K}, & \text{on } \partial K, \end{cases} \tag{2}$$

where  $K$  is either a reference quadrilateral or triangle. Some examples of such shape functions are given in Figures 3 and 4. This construction guarantees a continuous formulation combining AEs and standard finite elements. Also notice that the nodal modes automatically form a partition of unity.

For instance, in the case of Figure 2, the nodal shape function associated with the node 3 is computed by setting the restriction  $\phi(1, y)|_{\partial K}$  over the edge at  $x = 1$  to be the standard linear hat function with  $\phi(1, 0)|_{\partial K} = 1$ .

The associated inner modes  $\hat{\varphi}(x, y)$  are the functions satisfying

$$\begin{cases} \Delta\hat{\varphi}(x, y) = q(x, y), & \text{in } K, \\ \hat{\varphi}(x, y) = 0, & \text{on } \partial K, \end{cases} \tag{3}$$

where  $q(x, y)$  is some polynomial. For instance,  $q(x, y) = 1$  (constant) induces a standard bubble function. The set of elemental inner modes  $\hat{\varphi}(x, y)_K$  is constructed with products of Legendre polynomials, that is, all  $q(x, y) \in q(x, y)_K$ , where  $q(x, y)_K = \{P_i(x)P_j(y), i = 0, \dots, p - 2, j = 0, \dots, p - 2\}$ . With this choice the number of inner modes is the same as with the standard  $p$ -version, although the approximation properties are not. There exists a family of polynomials that could be used instead of the computed ones (see [9]).

The computation of the shape functions is done with finite elements (naturally). Hence, the concept of the implementation mesh arises, or more precisely, implementation discretisation. In order to simplify the evaluation of the inner products between the shape functions, every shape function associated with a given element is computed using the same implementation discretisation. One consequence of this is that the same extension may be computed using many different implementation discretisations.

For quadrilaterals, the baseline implementation discretisation is a regular grid with two elements per segment and uniform  $p = 20$ , and for triangles, a minimal triangulation of the nodes with uniform  $p = 16$ .

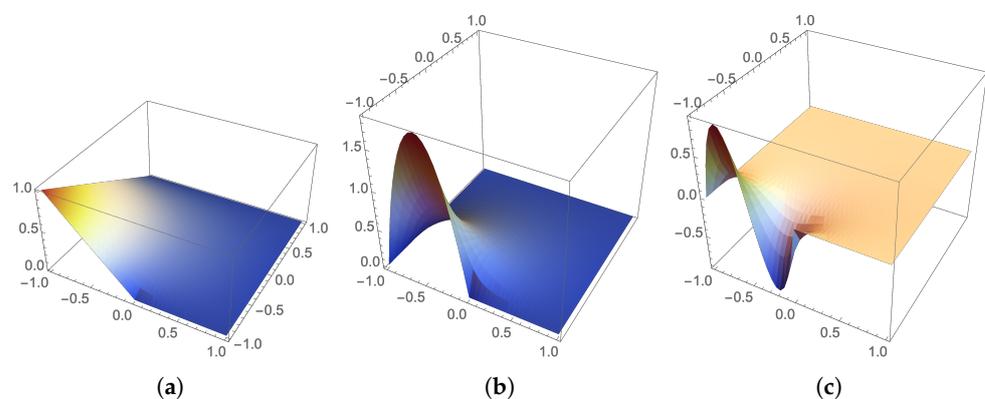
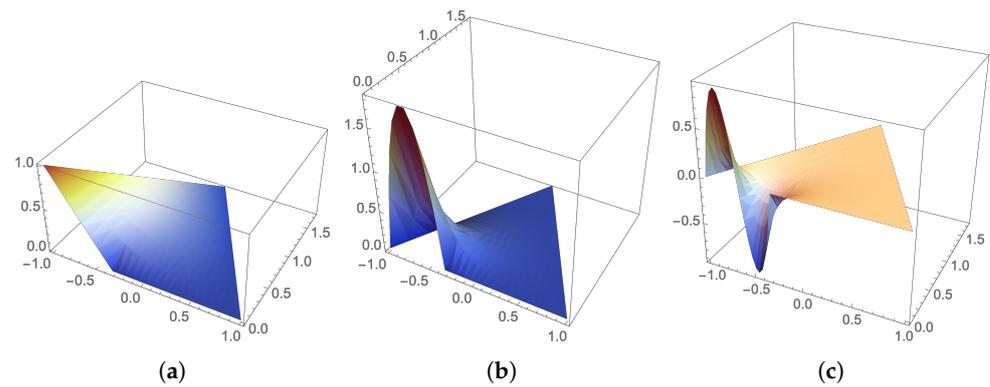


Figure 3. Quadrilateral. (a) Nodal mode. (b) Edge mode with  $p = 2$ . (c) Edge mode with  $p = 3$ .



**Figure 4.** Triangle. (a) Nodal mode. (b) Edge mode with  $p = 2$ . (c) Edge mode with  $p = 3$ .

### 2.2.2. Type of Reference Element

In order to minimise the computational work, it is necessary to introduce a way to maintain bookkeeping for the evaluated shape functions and AREs. Since compatibility with the standard  $p$ -version is required (or desired), on split edges, the shape functions must have correct parities.

Legendre polynomials of degree  $n$  can be defined using a recursion formula

$$(n + 1)P_{n+1}(x) - (2n + 1)xP_n(x) + nP_{n-1}(x) = 0, \quad P_0(x) = 1. \tag{4}$$

For our purposes, the central polynomials are the integrated Legendre polynomials for  $\xi \in [-1, 1]$

$$\phi_n(\xi) = \sqrt{\frac{2n - 1}{2}} \int_{-1}^{\xi} P_{n-1}(t) dt, \quad n = 2, 3, \dots \tag{5}$$

which can be rewritten as linear combinations of Legendre polynomials

$$\phi_n(\xi) = \frac{1}{\sqrt{2(2n - 1)}} (P_n(\xi) - P_{n-2}(\xi)), \quad n = 2, 3, \dots \tag{6}$$

The normalising coefficients are chosen so that

$$\int_{-1}^1 \frac{d\phi_i(\xi)}{d\xi} \frac{d\phi_j(\xi)}{d\xi} d\xi = \delta_{ij}, \quad i, j \geq 2. \tag{7}$$

Therefore, the  $\phi_n(\xi)$  inherit the property of the Legendre polynomials,

$$\phi_n(\xi) = (-1)^n \phi_n(-\xi). \tag{8}$$

This means that every edge has to be oriented in such a way that the shape function has a consistent sign or parity on both elements sharing it.

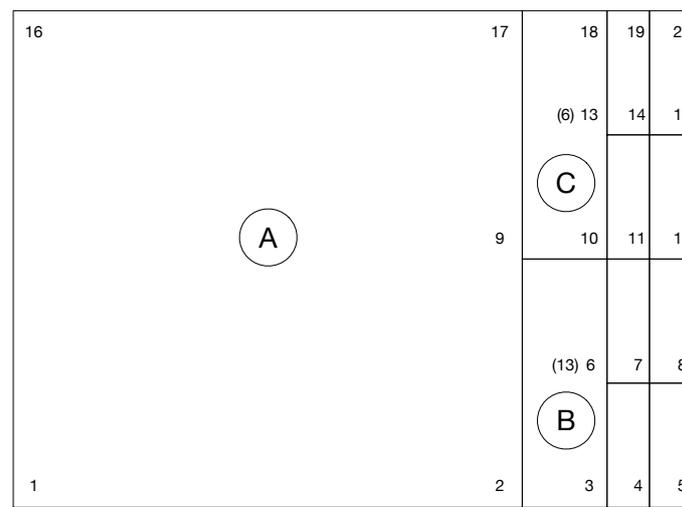
For every element a *type* is assigned in the following way: first a mapping node with the smallest identifier is chosen and the simplex is rotated so that the selected node is in a fixed position (normalisation); next for each edge, the parameter range of its support on the reference element is derived; finally each edge segment is assigned a parity by comparing the identifiers of the end points. Thus, every edge, split or not, has its contribution to the type of the ARE in form of a tuple  $(s, [a, b])$ , where  $s = \pm 1$ , and  $-1 \leq a < b \leq 1$ . For instance, the ARE of Figure 2 has the type  $S_Q$ — assuming that the nodes are identified as in the picture—

$$S_Q = ((1, [-1, 1]), ((1, [-1, 0]), (1, [0, 1])), (1, [-1, 1]), (-1, [-1, 1])). \tag{9}$$

The convention is that the positive direction is from the node with the smallest identifier. For standard  $p$ -version quadrilaterals, there are four types, and for triangles, two types, in both cases assuming that the inner nodes need not have an orientation. The inner nodes are always assumed to be oriented in the same way, that is, they do not affect the type.

### 2.2.3. Mesh Generation and Refinement

Since the proposed method is an extension of the  $p$ -version, mesh generation is identical to that of  $p$ -version. The power of the method lies in its ability to adapt to (seemingly) non-conforming mesh refinements. In Figure 5, two examples of strategies for refining the mesh to a corner singularity are illustrated.



(a)

Tag	$e_1$	$e_2$	$e_3$	$e_4$	Type	Tag	$e_1$	$e_2$	$e_3$	$e_4$	Type
A	(1,2)	(2,9,17)	(17,16)	(16,1)	1	A	(1,2)	(2,9,17)	(17,16)	(16,1)	1
B	(2,3)	(3,6,10)	(10,9)	(9,2)	1	B	(2,3)	(3,13,10)	(10,9)	(9,2)	2
C	(9,10)	(10,13,18)	(18,17)	(17,9)	1	C	(9,10)	(10,6,18)	(18,17)	(17,9)	3

(b)

(c)

**Figure 5.** Boundary layer mesh example. Four levels of elements toward the right hand boundary. Normalised AEs are given as node lists, every split edge has the same parametrisation on the reference element. Types are labelled in the order of occurrence. (a) Boundary layer mesh. (b) Uniform: List of AEs and AREs (types). The node identifiers 6 and 13 have been swapped and hence all three labelled elements have different types. (c) Non-uniform: (6) ↔ (13): List of AE and ARE (types).

The nodes are given identifiers in the order of creation. Thus, in both instances there are four AEs, but only two different types; hence, only two AREs have to be computed. Moreover, if one wanted to solve some problem using both meshes one at the time, the representative reference elements of each type would have to be computed only once. In Figure 5a, the AEs are given using nodes on edges.

### 2.3. Model Problems

Two classes of model problems are considered: a standard reaction–diffusion problem, and a thin cylindrical shell. Both of these problems are classical and have been discussed by many authors. It is of particular interest that the resulting linear systems have an identical parameter-dependent structure that can be exploited in the solution for multiple parameter values with a given right-hand-side.

### 2.3.1. Reaction-Diffusion Problem

The standard model problem is the reaction–diffusion problem. In the case of constant diffusion, it has the form

$$\begin{aligned} -\epsilon\Delta u(x, y) + u(x, y) &= f(x, y), \text{ in } \Omega, \\ u(x, y) &= 0, \text{ on } \partial\Omega. \end{aligned} \tag{10}$$

For the full 2D problem on a unit square, one can derive an exact solution by restricting the diffusion to some specific direction, for instance, along the  $x$ -axis only. In the numerical experiments, such a case is considered (see Figure 1a above). The expected characteristic length scale of the solution is  $\sqrt{\epsilon}$ .

### 2.3.2. Cylindrical Shells

Shells are thin three-dimensional structures that are very expensive to simulate, unless one performs dimension reduction in the model, that is, the original problem is written as lower-dimensional one, where the reduced dimension is represented by a parameter [15]. Here the resulting two-dimensional shell model has a vector field with five components  $\mathbf{u} = (u, v, w, \theta, \psi)$ , where the first three are the displacements and the latter two are the rotations in the axial and angular directions, respectively. The convention is that the computational domain  $D$  is given by the surface parametrisation and the axial/angular coordinates are denoted by  $x$  and  $y$ .

The total energy is given by a quadratic functional

$$\mathcal{F}(\mathbf{u}) = \frac{1}{2}Y\mathbf{A}(\mathbf{u}, \mathbf{u}) - \mathbf{Q}(\mathbf{u}), \tag{11}$$

where  $\mathbf{A}$  represents strain energy and  $\mathbf{Q}$  is the load potential. The constant factor  $Y = E/(12(1 - \nu^2))$ , where  $E$  is the Young modulus of the material, and  $\nu$  is the Poisson ratio.

Deformation energy  $\mathcal{A}(\mathbf{u}, \mathbf{u})$  is divided into bending, membrane, and shear energies, denoted by subscripts  $B, M$ , and  $S$ , respectively:

$$\mathcal{A}(\mathbf{u}, \mathbf{u}) = t^2\mathcal{A}_B(\mathbf{u}, \mathbf{u}) + \mathcal{A}_M(\mathbf{u}, \mathbf{u}) + \mathcal{A}_S(\mathbf{u}, \mathbf{u}), \tag{12}$$

where  $t$  is the dimensionless thickness,  $t = d/L$ , where  $d$  is the actual thickness and  $L$  is some characteristic length scale, for instance, the diameter of the domain.

Let us consider a cylindrical shell with a midsurface  $\omega$  generated by a function  $f_1(x) = R$ ,  $x \in [-x_0, x_0], x_0 > 0$ . In this case, the product of the Lamé parameters (metric),  $A_1(x)A_2(x) = R$ , and the reciprocal curvature radii are  $1/R_1(x) = 0$  and  $1/R_2(x) = 1/R$ , since

$$A_1(x) = \sqrt{1 + [f_1'(x)]^2}, \quad A_2(x) = f_1(x), \tag{13}$$

and

$$R_1(x) = -\frac{A_1(x)^3}{f_1''(x)}, \quad R_2(x) = A_1(x)A_2(x). \tag{14}$$

Bending, membrane, and shear energies are given as

$$\begin{aligned} t^2\mathcal{A}_B(\mathbf{u}, \mathbf{u}) &= t^2 \int_{\omega} [\nu(\kappa_{11}(\mathbf{u}) + \kappa_{22}(\mathbf{u}))^2 \\ &\quad + (1 - \nu) \sum_{i,j=1}^2 \kappa_{ij}(\mathbf{u})^2] A_1(x, y)A_2(x, y) \, dx \, dy, \end{aligned} \tag{15}$$

$$\begin{aligned} \mathcal{A}_M(\mathbf{u}, \mathbf{u}) &= 12 \int_{\omega} [\nu(\beta_{11}(\mathbf{u}) + \beta_{22}(\mathbf{u}))^2 \\ &\quad + (1 - \nu) \sum_{i,j=1}^2 \beta_{ij}(\mathbf{u})^2] A_1(x, y) A_2(x, y) dx dy, \end{aligned} \tag{16}$$

$$\begin{aligned} \mathcal{A}_S(\mathbf{u}, \mathbf{u}) &= 6(1 - \nu) \int_{\omega} [(\rho_1(\mathbf{u})^2 + \rho_2(\mathbf{u})^2)] \times \\ &\quad A_1(x, y) A_2(x, y) dx dy. \end{aligned} \tag{17}$$

We have omitted the scaling  $Y$ .

In the special case of constant radius, using the identities above, the bending, membrane, and shear strains [16],  $\kappa_{ij}$ ,  $\beta_{ij}$ , and  $\rho_i$ , respectively, can be written as

$$\begin{aligned} \kappa_{11} &= \frac{\partial \theta}{\partial x}, \quad \kappa_{22} = \frac{1}{R} \frac{\partial \psi}{\partial y}, \quad \kappa_{12} = \frac{1}{2} \left( \frac{\partial \psi}{\partial x} + \frac{1}{R} \frac{\partial \theta}{\partial y} - \frac{1}{R} \frac{\partial v}{\partial x} \right), \\ \beta_{11} &= \frac{\partial u}{\partial x}, \quad \beta_{22} = \frac{1}{R} \frac{\partial v}{\partial x} + \frac{w}{R}, \quad \beta_{12} = \frac{1}{2} \left( \frac{1}{R} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \\ \rho_1 &= \frac{\partial w}{\partial x} - \theta, \quad \rho_2 = \frac{1}{R} \frac{\partial w}{\partial y} - \frac{v}{R} - \psi. \end{aligned} \tag{18}$$

By minimising the the total energy Equation (11) the variational problem becomes the following: Find  $\mathbf{u} \in \mathcal{U} \subset [H^1(\omega)]^5$  such that

$$\mathbf{A}(\mathbf{u}, \mathbf{v}) = \mathbf{Q}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{U}. \tag{19}$$

and the corresponding finite element problem: Find  $\mathbf{u}_h \in \mathcal{U}_h$  such that

$$\mathbf{A}(\mathbf{u}_h, \mathbf{v}) = \mathbf{Q}(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{U}_h. \tag{20}$$

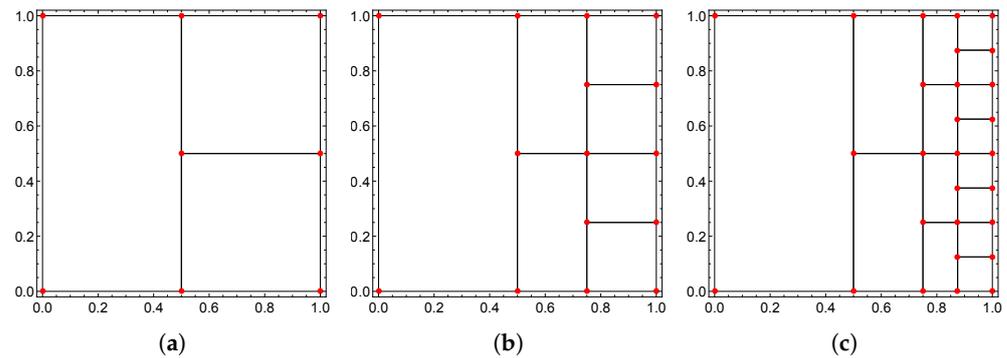
The load potential has the form

$$\mathbf{Q}(\mathbf{v}) = \int_{\omega} \mathbf{f}(x, y) \mathbf{v} dx dy. \tag{21}$$

If the load acts in the transverse direction of the shell surface, it has the form  $\mathbf{f}(x, y) = [0, 0, f_w(x, y), 0, 0]^T$ . It can be shown that if for the load  $\mathbf{f} \in [L^2(\omega)]^5$  holds, the problem Equation (19) has a unique weak solution  $\mathbf{u} \in [H^1(\omega)]^5$ . The corresponding result holds in the finite dimensional case, when the finite element method is employed.

### 3. Boundary Layer Resolution

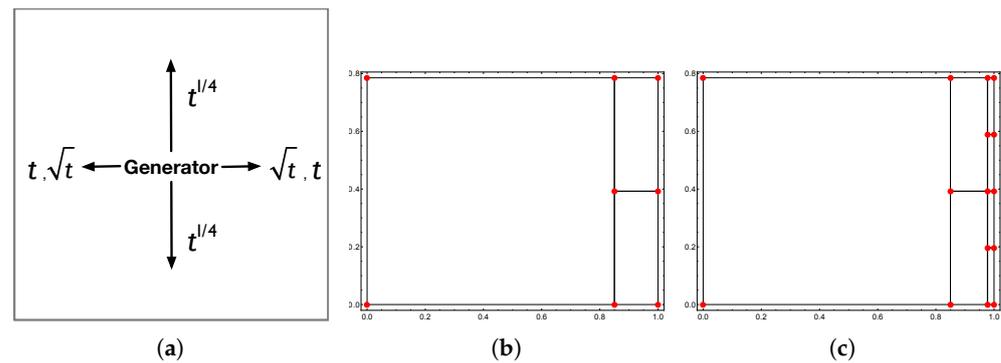
What makes adaptive reference elements particularly appealing in boundary layer dominated problems is the possibility to use replacement rules as the basic refinement strategy. This basic idea has been discussed earlier (see [5]), but the fact that any predicate can be used to select the elements or cells for refinement has not been addressed before. In Figure 6a, a sequence of meshes generated by applying the same rule three times is shown. The rule simply first verifies that the element has an edge on the boundary before the three new elements are generated, one ARE and two ordinary. In this simple case, the process is greedy in the sense that the elements not participating in the refinement step are not affected. In a general case, the replacement rules could be specified on the edge level with additional bookkeeping required to maintain the data structures for the elements themselves.



**Figure 6.** Sequence of refined meshes. Same replacement rule applied three times with a unit square as the initial configuration. The underlying finite element discretisation is continuous, despite the apparent hanging nodes. (a) Rule applied once. (b) Rule applied twice. (c) After three applications of the rule.

In Figure 6, the rule is kept constant for illustration purposes. The refinement rule first divides the selected elements along the  $x$ -axis, and then the elements touching the boundaries aligned with the  $y$ -axis. The axial edge length ratio is 1:1. Of course, nothing prevents one from modifying the rules from one application or layer to another. The crucial requirement is that the set of rules at one step has to be consistent, that is, every topological component has to be refined in the same way, regardless of the element containing it. For example, in Figure 6a, there exists an edge connecting nodes  $(1/2, 1/2)$  and  $(1, 1/2)$ . In Figure 6b, that edge is split into two segments and node  $(3/4, 1/2)$  is added. During the process, both elements touching the boundary at  $x = 1$  have created the same node. The final step of the refinement algorithm verifies that all destructive changes are consistent and removes possible duplicates.

The shell problems have a rich boundary layer structure, including internal layers. In the axial direction, the characteristic length scales are  $\sqrt{t}$  and  $t$ , where the latter is the short scale often omitted from the models. In the angular direction, there exists a relatively long layer of  $\sqrt[4]{t}$ , which plays an important role in the free vibration of shells of revolution. This is not an exhaustive list; for a detailed discussion on boundary layer structures of cylindrical shells and their resolution in standard  $hp$ -FEM setting, see [17]. In Figure 7, both a schematic of the axial layer dependencies and a sample sequence of meshes adapted to a single layer generator represented by the entire boundary at  $x = 1$ . The axial edge length ratio is approximately 6:1. Without any loss of generality in the numerical examples, only problems with simple boundary segment layer generators are considered. It has to be emphasised that, strictly speaking, the smooth component spanning the entire domain can be interpreted as a boundary layer, and it is precisely this feature that necessitates the use of higher order methods, i.e., the  $p$ -version, unless special so-called shell elements are used.



**Figure 7.** Cylindrical shell: Layer structure. Examples of minimal meshes for a case with a generator at  $x = 1$  with  $t = 1/100$ . (a) Layer structure for the axial directions, there also exists a possible smooth component spanning the entire domain. (b) Mesh adapted for  $\sqrt{t}$ . (c) Mesh adapted for  $\sqrt{t}$  and  $t$ .

#### 4. Computational Asymptotic Analysis

Asymptotic analysis has been a standard tool for understanding the parameter-dependence of, for instance, mechanical systems. The goal is to understand what happens when the critical parameter tends to some limit, in the context of this paper, for instance, when the dimensionless thickness of a shell tends to zero, i.e.,  $t \rightarrow 0$ . Rather than utilising some a priori understanding of the behaviour of the systems, computational asymptotic analysis relies on accurate solution of individual realisations of the problem and the subsequent recovery of the quantities of interest from the computed solutions.

##### 4.1. Solving Parameter-Dependent Sequences of Linear Systems

It is necessary to solve systems in terms of  $\mathbf{A}$  and  $\mathbf{B}$  (for shells bending, and membrane and shear summed together, respectively)

$$\mathbf{S}(\sigma)v = (\mathbf{B} + \sigma\mathbf{A})v = b, \tag{22}$$

for every  $\sigma > 0$ . In order to avoid the factorisation of  $\mathbf{S}(\sigma)$  at every  $\sigma$ , it would be natural to consider iterative methods. Unfortunately, parameter-independent preconditioning of singularly perturbed systems is an open problem. However, if one considers a *sequence of problems*, it is possible to transform the problem as follows: Let  $\mathbf{B} = \mathbf{L}\mathbf{L}^T$  (Cholesky decomposition); then,

$$\mathbf{L}(\mathbf{I} + \sigma\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T})\mathbf{L}^T v = b, \tag{23}$$

where the subspace defined by  $\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}$  is invariant over all parameters  $\sigma$ . The inner systems  $(\mathbf{I} + \sigma\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T})\hat{v} = \hat{b}$ , have their spectra bounded by 1 from below, making it sufficient to collect the largest eigenvectors into a subspace  $W$ , say. Once the subspace  $W$  is constructed, the deflated conjugate gradient method can be applied [18,19]. Here, collecting Lanczos vectors is also sufficient since it is assumed that loading remains constant and independent of  $\sigma$ .

**Remark 1.** *Since the system can be singularly perturbed, i.e.,  $\mathbf{B}$  is not necessarily invertible, it is possible to use the stabilised version*

$$((\mathbf{B} + \epsilon_\sigma\mathbf{A}) + (\sigma - \epsilon_\sigma)\mathbf{A})v = b, \tag{24}$$

where  $\epsilon_\sigma \in [0, \sigma]$ . The choice of optimal shift  $\epsilon_\sigma$  is problem-dependent.

##### 4.2. Recovering Quantities of Interest

Once a sequence of problems is solved, in particular, in the case of elasticity, it is interesting to recover the parameter-dependence of the energy components, e.g., bending and membrane energies for shells. In short, rather than trying to a priori analytically

predict the behaviour, one simply derives the quantities of interest by numerical estimation a posteriori.

For instance, in the context of shells, given a sequence of solutions, the procedure is as follows: (i) first the displacements are normalised so that  $w \sim 1$ , (ii) then strains are evaluated with normalised displacements, (iii) rates of change for strains are derived through regression, and (iv) energy dependence is discovered by evaluating the energy expressions using the estimated rates for strains.

In the case of the cylindrical shell benchmark below, both known theoretical asymptotic dependencies on the dimensionless thickness are recovered using this procedure.

### 5. Numerical Experiments

The numerical experiments were designed to cover two types of PDEs with boundary layers. The goal here is to establish the convergence properties of the proposed method in comparison with the standard  $p$ -version. The idea is to extend the  $p$ -version, but of course with the approximative shape functions, there are non-standard sources of error that are very difficult to analyse, except through computational studies, such as these. The reference values of the quantities of interest are listed in Table 1 (see, for example, [20]). All convergence graphs presented in this section are log–log plots.

**Table 1.** Reference values used in numerical experiments. Reaction–diffusion with  $\epsilon = 1/100$  and Pitkäranta cylinder with  $t = 1/100$ .

Case	Norm	Value Used	Exact
Reaction-Diffusion	$L^2$	0.83673056017854	$\frac{\sqrt{\frac{1}{10}(13+40e^{10}+7e^{20})}}{1+e^{10}}$
Cylinder (Clamped)	Squared energy	2.6882879572571783	-
Cylinder (Free)	Squared energy	7043.3120530934690	-

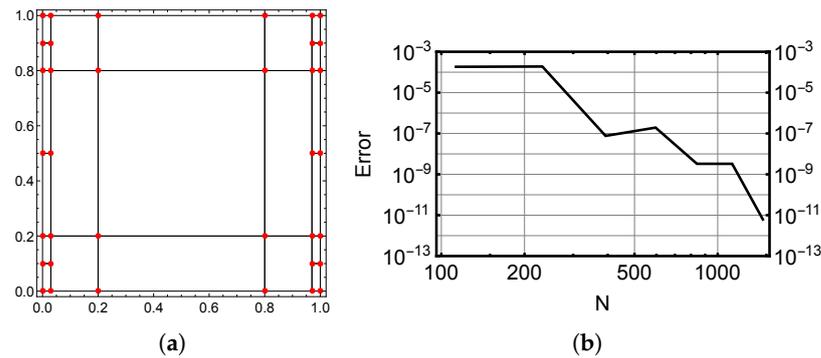
#### 5.1. Reaction–Diffusion

The first example is the standard model problem for simple boundary layers, the reaction–diffusion problem (10). The computational domain is the unit square and homogeneous Dirichlet boundary conditions are set on all boundaries. The typical shape of the potential function is shown above in Figure 1a. Here, the diffusion is restricted to  $x$  variable only with unit loading. The exact solution when setting  $\epsilon = 1/100$  is

$$u(x, y) = \frac{-e^{10-10x} - e^{10x} + 1 + e^{10}}{1 + e^{10}}. \tag{25}$$

The mesh and the observed convergence in the  $L^2$ -norm is given in Figure 8. The results are very good indeed. In particular, the convergence pattern has staircasing, which in fact is to be expected in the context of  $p$ -version since the exact solution is symmetric (that is, even locally), and as  $p = 2, \dots, 8$  one should not expect significant improvement as  $p$  changes from even to odd.

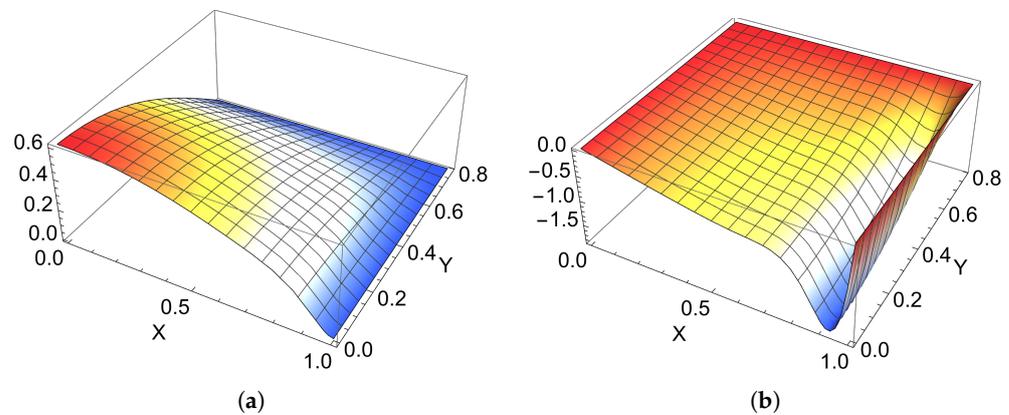
No attempt to find an optimal distribution of degrees of freedom has been made. This question is beyond the scope of this study. The mesh of Figure 8a has 36 nodes, 56 edges, and 21 quadrilateral (15 regular, six AREs with five nodes). Therefore at  $p = 8$  the number of degrees of freedom is  $36 + 56(p - 1) + 21(p - 1)(p - 1) = 1457$ .



**Figure 8.** Reaction–diffusion with  $\epsilon = 1/100$ . Convergence graph has a characteristic staircasing behaviour, where the even nature of the exact solution over the domain results in no convergence as the polynomial order changes from even to odd,  $p = 2, \dots, 8$ . (a) Symmetric mesh. (b)  $L^2$ -convergence (absolute error).

### 5.2. Pitkäranta Cylinder

Pitkäranta cylinder (see [20]) is a well-known and widely used benchmark problem. The computational domain is reduced to one sixteenth,  $\Omega = [0, 1] \times [0, \pi/4]$ , through clever application of symmetry–antisymmetry boundary conditions for a given Fourier mode type loading  $f(x, y) = \cos(2y)$ , which is constant in the axial direction. In the clamped case, the boundary conditions are  $y = 0 : v = \psi = 0, x = 1$  clamped,  $y = \pi/4 : u = w = \theta = 0$ , and  $x = 0 : u = \theta = 0$ , and in the free case there are no conditions on  $x = 1$ . Sample displacement fields are shown in Figure 9.

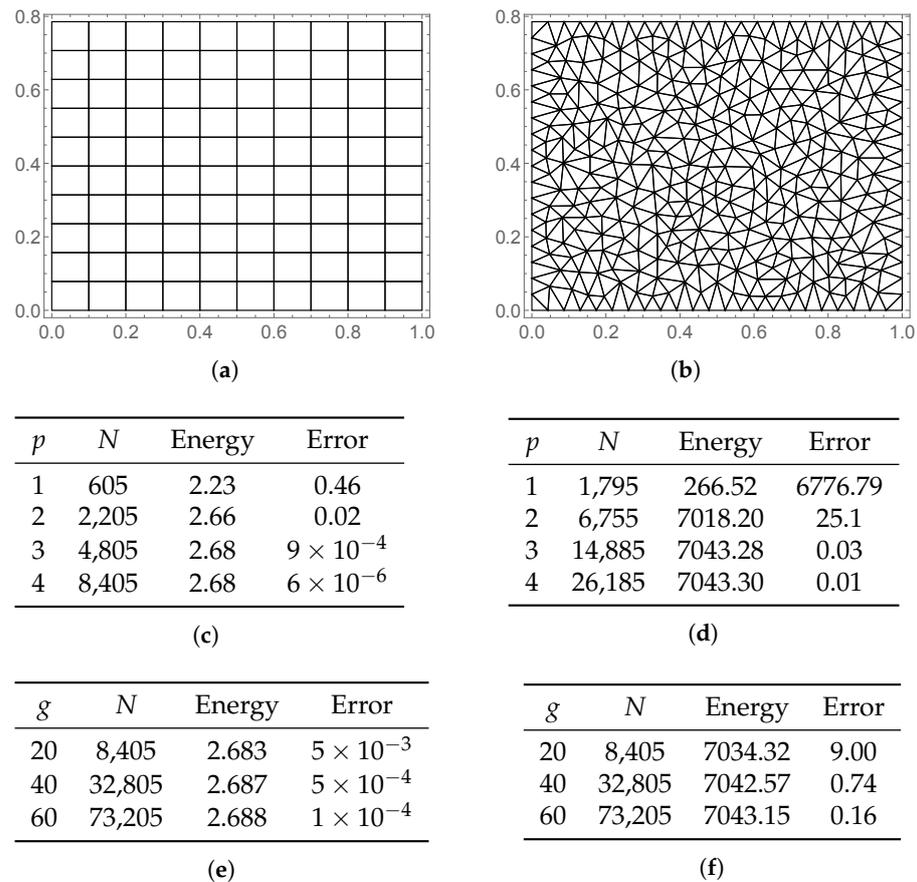


**Figure 9.** Pitkäranta cylinder: clamped case. Detail of the domain with symmetry/antisymmetry boundary conditions applied. The boundary layer in the rotation component with the characteristic length scale of  $\sqrt{t} = 1/10$  is clearly visible. (a) Transverse deflection. (b) Rotation  $\theta$ .

#### 5.2.1. On Numerical Locking

It has been already discussed above that the  $p$ -version can be used to alleviate problems associated with numerical locking within the standard FEM framework. Of course, there exist many special shell elements that try to resolve the issue by modifying the underlying variational formulation [15].

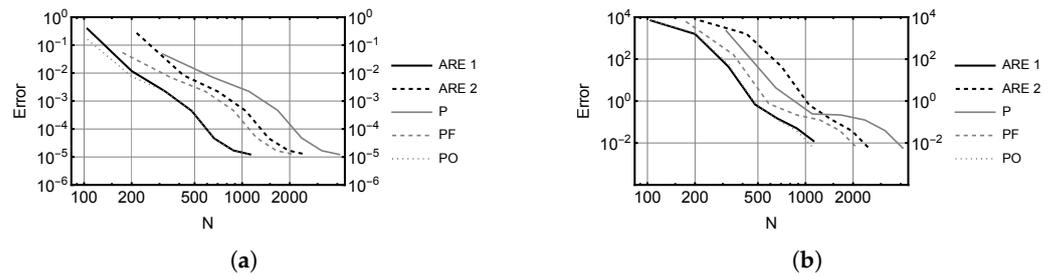
In Figure 10, two sets of convergence data is presented. In the first set, the mesh is fixed, but the polynomial order  $p$  varies,  $p = 1, 2, 3, 4$ . In the second set the polynomial order is fixed at  $p = 2$  over a set of  $g \times g$ -grids,  $g = 20, 40, 60$ . The numerical locking is evident. For both cases, clamped and free, there is a clear difference in performance as the polynomial order is increased from  $p = 2$  to  $p = 3$ . Figure 10f indicates that the standard  $h$ -version converges very slowly indeed, even at  $p = 2$ .



**Figure 10.** Pitkäranta cylinder,  $t = 1/100$ . (a) Clamped case on  $10 \times 10$ -grid: Mesh. (b) Free boundary: Mesh. (c) Clamped case on  $10 \times 10$ -grid: Errors. Squared energy norm convergence (absolute error) in  $p = 1, \dots, 4$  using different discretisations. (d) Free case: Errors. Squared energy norm convergence (absolute error) in  $p = 1, \dots, 4$  using different discretisations. (e) Clamped case on  $g \times g$ -grid: Errors. Squared energy norm convergence (absolute error) at  $p = 2$  over a series of uniform discretisations. (f) Free case on  $g \times g$ -grid: Errors. Squared energy norm convergence (absolute error) at  $p = 2$  over a series of uniform discretisations.  $N$  is the number of degrees of freedom.

### 5.2.2. Convergence Results

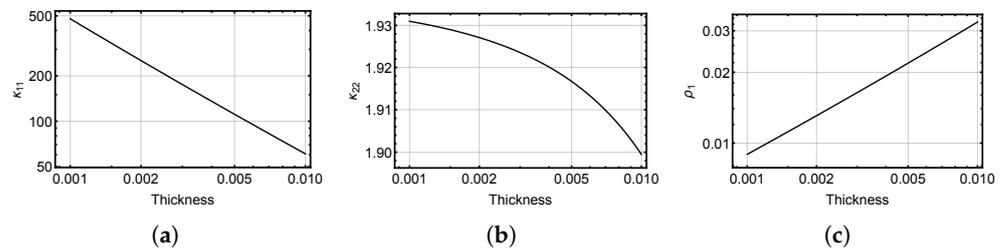
As in the reaction–diffusion problem, the convergence results are very good. In Figure 11, the proposed method practically matches that of the comparable  $p$ -version solution. As expected the number of degrees of freedom in the best observed case is significantly smaller than those given in Figure 10. One word of caution though, since the special form of loading actually allows for a 1D solution, for the  $p$ -version, there exists a mesh with exactly two elements which does not have a corresponding ARE configuration, and the proposed method degenerates to standard  $p$ -version. If one takes the number of significant digits into consideration or uses relative error, the convergence rate in the free case is in fact slightly better since the bulk of the energy is carried by the smooth component, which is resolved using a high polynomial order.



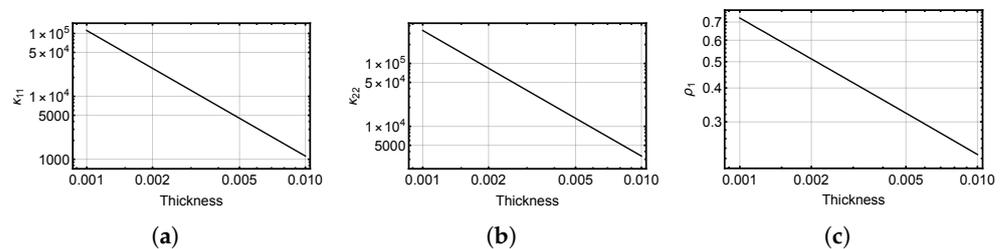
**Figure 11.** ARE convergence of Pitkäranta cylinder,  $t = 1/100$ . Squared energy norm convergence (absolute error) in  $p = 2, \dots, 8$  using different discretisations: ARE 1: one layer, ARE 2: two layers, P: full tensor product grid with two layers, PF: full tensor product grid with one layer, PO: minimal full tensor product grid conforming to ARE 1. The observed convergence of the proposed method in both test cases agrees with the standard  $p$ -version.  $N$  is the number of degrees of freedom. (a) Clamped case. (b) Free boundary.

### 5.2.3. Energy Dependence

The final and perhaps the most interesting experiment is the attempt to recover the energy dependencies via computational asymptotic analysis. Two sets of results are shown in Figures 12 and 13. Some of the strains diverge as  $t \rightarrow 0$ . This in itself is not a cause of worry if the overall energy remains constrained. However, it is possible that there remains a parameter-dependent error amplification factor  $C(t)$ , and indeed, for the free case, the expected  $C(t) \sim 1/t$  is recovered. This is the primary source of numerical locking, defined as an unavoidable loss of convergence rate.



**Figure 12.** Clamped case. Practically perfect agreement with the a priori predictions. Notice, that  $\kappa_{22}$  is essentially constant. Additionally,  $\rho_1 \rightarrow 0$  indicates that the Kirchhoff–Love condition is satisfied without imposing it within the model explicitly. (a)  $\kappa_{11} \sim 1/t$ . (b)  $\kappa_{22} \sim 1$ . (c)  $\rho_1 \sim t$ .



**Figure 13.** Free boundary. Notice, that  $\rho_1$  is diverging, however, with a very small constant. This indicates that the boundary layer resolution is not accurate enough to conform to Kirchhoff–Love condition. The aggregate parameter-dependence is correct, the bending terms dominate the energy completely. (a)  $\kappa_{11}(\mathbf{u}) \sim 1/t^2$ . (b)  $\kappa_{22}(\mathbf{u}) \sim 1/t^2$ . (c)  $\rho_1(\mathbf{u}) \sim 1/\sqrt{t}$ .

The strains are computed using the procedure outlined in Section 4.2 above. Notice that in the clamped case  $\rho_1 \rightarrow 0$ , but not in the free case. This convergence implies the so-called Kirchhoff–Love condition. In the free case, the divergence is mild and completely dominated by the strongly increasing bending terms. Nevertheless, this is a point where the traditional analysis would simply assume such convergence also in the free case. Of course,

this is ultimately a question of validity of the chosen model and cannot be resolved here. More pragmatically, one can interpret the result as an indication that the discretisation does not capture the short  $t$ -layer in the free case.

Once the strains are recovered, the error amplification factor  $C(t)$  can be deduced by inserting the rates into the energy formulation and finding the most significant power of  $t$  in the resulting expression. More precisely, simply evaluating Equations (15) and (16), and adding the resulting expressions, one gets an approximate dependence for the square of the energy norm. For the clamped case, the recovered strains (ignoring the shear terms) are

$$\begin{aligned}\beta_{11}(\mathbf{u}) &\sim 1, \beta_{22}(\mathbf{u}) \sim 1, \beta_{12}(\mathbf{u}) \sim 1, \\ \kappa_{11}(\mathbf{u}) &\sim 1/t, \kappa_{22}(\mathbf{u}) \sim 1, \kappa_{12}(\mathbf{u}) \sim 1/\sqrt{t}, \\ C(t) &\sim 1,\end{aligned}\tag{26}$$

and for the free case, equivalently

$$\begin{aligned}\beta_{11}(\mathbf{u}) &\sim 1/t, \beta_{22}(\mathbf{u}) \sim 1/t, \beta_{12}(\mathbf{u}) \sim 1\sqrt{t}, \\ \kappa_{11}(\mathbf{u}) &\sim 1/t^2, \kappa_{22}(\mathbf{u}) \sim 1/t^2, \kappa_{12}(\mathbf{u}) \sim 1/t^{3/2}, \\ C(t) &\sim 1/t,\end{aligned}\tag{27}$$

where the  $C(t)$  are square roots conforming to the definition of the energy norm (squared energy is used in experiments). The  $C(t) \sim 1/t$  is exactly the predicted worst-case error amplification factor. For more details, see [11]. The parameter-dependent error amplification simply means that as the parameter changes, the discretisation may have to be adjusted in order to maintain the same level of accuracy.

## 6. Conclusions

Harmonic extension finite elements provide an extension to the standard  $p$ -version. Indeed, if the discretisation is such that no adapted shape functions are needed, the method is the  $p$ -version. Adaptive reference elements are one way to implement the concept, and currently, the names can be used interchangeably. Until now, the method has been shown to perform well on problems with strong singularities. Here, the efficacy also on problems with boundary layers is established. For cylindrical shells, the numerical experiments indicate that the method converges in  $p$  as expected, and furthermore, the performance is maintained, even over a range of parameters. Via computational asymptotic analysis, the expected error amplification factors can be recovered also with the proposed method.

There are many open questions still when one considers implementation of the method. All numerical experiments considered here had constant coefficients. This simplifies numerical integration greatly and allows for the reuse of integrated reference elements. Similarly, the implementation meshes for the computed or adapted shape functions is the same for all shape functions in order to simplify the overall implementation and evaluation of the inner products. There are many opportunities for optimisation in this regard. Finally, even though adaptation of  $p$ -version specific error estimators should be straightforward, this work has not even started yet.

Discretisation of problems on complicated domains is a challenge that will not disappear any time soon. There are many options for non-standard approaches and many of them show great promise. Harmonic extension finite elements are a conservative, non-intrusive approach aimed for retaining as much as possible of the standard implementations and investments in the existing solvers. They also fit in within the standard a posteriori error analysis frameworks; however, rigorous proofs are still missing.

**Funding:** This research received no external funding.

**Acknowledgments:** We acknowledge the computational resources provided by the Aalto Science-IT project.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Beirão da Veiga, L.; Brezzi, F.; Cangiani, A.; Manzini, G.; Marini, L.D.; Russo, A. Basic principles of virtual element methods. *Math. Model. Methods Appl. Sci.* **2013**, *23*, 199–214. [[CrossRef](#)]
2. Di Pietro, D.A.; Droniou, J. *The Hybrid High-Order Method for Polytopal Meshes*; Springer: Berlin/Heidelberg, Germany, 2020.
3. Szabo, B.; Babuska, I. *Finite Element Analysis*; Wiley: Hoboken, NJ, USA, 1991.
4. Schwab, C. *p- and hp-Finite Element Methods*; Oxford University Press: Oxford, UK, 1998.
5. Hakula, H. Adaptive reference elements via harmonic extensions and associated inner modes. *Comput. Math. Appl.* **2020**, *80*, 2272–2288. [[CrossRef](#)]
6. Hofreither, C.; Langer, U.; Weißer, S. Convection-adapted BEM-based FEM. *ZAMM—J. Appl. Math. Mech./Z. Angew. Math. Mech.* **2016**, *96*, 1467–1481. [[CrossRef](#)]
7. Weißer, S. Arbitrary order Trefftz-like basis functions on polygonal meshes and realization in BEM-based FEM. *Comput. Math. Appl.* **2014**, *67*, 1390–1406. [[CrossRef](#)]
8. Owall, J.S.; Reynolds, S.E. A High-Order Method for Evaluating Derivatives of Harmonic Functions in Planar Domains. *SIAM J. Sci. Comput.* **2018**, *40*, A1915–A1935. [[CrossRef](#)]
9. Anand, A.; Owall, J.S.; Reynolds, S.E.; Weißer, S. Trefftz Finite Elements on Curvilinear Polygons. *SIAM J. Sci. Comput.* **2020**, *42*, A1289–A1316. [[CrossRef](#)]
10. Pitkäranta, J. The problem of membrane locking in finite element analysis of cylindrical shells. *Numer. Math.* **1992**, *61*, 523–542. [[CrossRef](#)]
11. Hakula, H.; Leino, Y.; Pitkäranta, J. Scale resolution, locking, and high-order finite element modelling of shells. *Comput. Methods Appl. Mech. Eng.* **1996**, *133*, 157–182. [[CrossRef](#)]
12. Wolfram Research, Inc. *Mathematica*, version 13.0.1; Wolfram Research, Inc.: Champaign, IL, USA, 2021.
13. Hakula, H.; Tuominen, T. Mathematica implementation of the high order finite element method applied to eigenproblems. *Computing* **2013**, *95*, 277–301. [[CrossRef](#)]
14. Pitkäranta, J.; Matache, A.M.; Schwab, C. Fourier mode analysis of layers in shallow shell deformations. *Comput. Methods Appl. Mech. Eng.* **2001**, *190*, 2943–2975. [[CrossRef](#)]
15. Chapelle, D.; Bathe, K.J. *The Finite Element Analysis of Shells*; Springer: Berlin/Heidelberg, Germany, 2003.
16. Malinen, M. On the classical shell model underlying bilinear degenerated shell finite elements: General shell geometry. *Int. J. Numer. Methods Eng.* **2002**, *55*, 629–652. [[CrossRef](#)]
17. Hakula, H. hp-boundary layer mesh sequences with applications to shell problems. *Comput. Math. Appl.* **2014**, *67*, 899–917. [[CrossRef](#)]
18. Saad, Y.; Yeung, M.; Erhel, J.; Guyomarch, F. A deflated version of the conjugate gradient algorithm. *SIAM J. Sci. Comput.* **2000**, *21*, 1909–1926. [[CrossRef](#)]
19. Meerbergen, K. The solution of parametrized symmetric linear systems. *SIAM J. Matrix Anal. Appl.* **2003**, *24*, 1038–1059. [[CrossRef](#)]
20. Pitkäranta, J.; Leino, Y.; Ovaskainen, O.; Piila, J. Shell Deformation states and the Finite Element Method: A Benchmark Study of Cylindrical Shells. *Comput. Methods Appl. Mech. Eng.* **1995**, *128*, 81–121. [[CrossRef](#)]