Finite Element Methods with Hierarchical WEB-splines

Von der Fakultät Mathematik und Physik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

vorgelegt von

Muhammad Mustahsan

geboren in Chakwal, Pakistan

Hauptberichter:	Prof. Dr. Klaus Höllig
Mitberichter:	Prof. Dr. Ulrich Reif
Tag der Einreichung:	19.01.2011
Tag der mündlichen Prüfung:	03.02.2011

Institut für Mathematische Methoden in den Ingenieurwissenschaften,
Numerik und geometrische Modellierung
Universität Stuttgart
2011

Dedicated to my late father
Ghulam Sarwar

Das Gleichnis derjenigen, die ihren Besitz auf Allahs Weg ausgeben, ist das eines Saatkorns, das sieben Ähren wachsen lässt, in jeder Ähre hundert Körner. Allah vervielfacht, wem Er will. Und Allah ist Allumfassend und Allwissend

(Koran 2:261)

Abstract

Piecewise polynomial approximations are fundamental to geometric modeling, computer graphics, and finite element methods. The classical finite element method uses low order piecewise polynomials defined on polygonal domains. The domains are discretized into simple polygons called the mesh. These polygons might be triangles, quadrilaterals, etc., for two-dimensional domains, and tetrahedra, hexahedra, etc., for three-dimensional domains. Meshing is often the most timeconsuming process in finite element methods. In classical two-dimensional finite element methods, the basis functions are usually hat functions defined on triangulations. Another possible selection of a finite element basis in two dimensions are tensor product b-splines.

Bivariate B-splines are piecewise polynomials of degree n with support having $(n+1)^2$ cells. The domain is discretized via a uniform grid. Relevant are those b-splines for which the support intersects the domain. To keep the support of a relevant B-spline within the domain, we multiply it by a weight function. The weight function is positive in the interior of the domain and vanishes on the boundary and outside of the domain. The resulting weighted B-splines conform to homogeneous boundary conditions. They satisfy the usual properties of a finite element basis.

The insertion of new knots into the grid is not a good adaptive strategy because of the global effect of knot insertion. Instead, hierarchical refinement is very effective for tensor product splines. It permits the change of control points and subsequent editing of fine details in some parts while keeping the other parts unaffected. For programming, a data structure is required that not only keeps track of the refinement but also stores the information about the discretization of the domain. Moreover, algorithms for assembling and solving the finite element system are needed. In this thesis, we have developed such adaptive schemes with weighted B-splines and implemented them in MATLAB with an appropriate data structure.

We proposed two different adaptive schemes for the selection of the sequence of subdomains characterizing the refinement. The first scheme uses a predefined and strongly nested domain sequence, appropriate, e.g., near a reentrant corner of the domain. For strongly nested domains, the distance between the boundary of the subdomain with grid width h and the subdomain with grid width $h/2$ is $\geq (2n+1)h$. For such a domain sequence, an error estimate can be obtained.

The second adaptive scheme is an automatic refinement process. The refinement is determined by comparing the B-spline coefficients of an approximation with those of an approximation obtained by refining all subdomains. The hierarchical refinement is then based on the regions where the difference between the coefficients exceeds a given tolerance. Both adaptive schemes yield convergence of the hierarchical approximations. The adaptive schemes are tested by solving Poisson's problem on domains with reentrant corners with refinement in the neighborhood of the geometric singularity.

Abstract

Stückweise polynomiale Approximationen spielen in der Geometrischen Modellierung, Computer graphik und bei Finite-Elemente-Methoden eine fundamentale Rolle. Klassische Finite-Elemente-Verfahren benutzen stückweise Polynome niedriger Ordnung auf polygonalen Gebieten. Die Gebiete werden in einfache Polygone, das so genannte Netz, zerlegt. Diese Polygone können für zweidimensionale Gebiete beispielsweise Dreiecke oder Vierecke sein, für dreidimensionale Gebiete Tetraeder oder Hexaeder. Die Netzgenerierung ist oft der zeitaufwändigste Prozess in der Finite-Elemente-Methode. In der klassischen Finite-Elemente-Methode sind die Basisfunktionen in zwei Dimensionen meist auf Triangulierungen definierte Hut-Funktionen. Eine andere mögliche Wahl einer bivariaten Finite-Elemente-Basis sind Tensor-Produkt-B-Splines.

Bivariate B-Splines sind stückweise Polynome vom Grad n mit einem Träger, bestehend aus $(n+1)^2$ Gitterzellen. Das Gebiet wird durch ein uniformes Gitter diskretisiert. Relevant sind diejenigen B-Splines, deren Träger das Gebiet schneidet. Um den Träger eines relevanten B-Splines auf das Gebiet einzuschränken, wird er mit einer Gewichtsfunktion multipliziert. Die Gewichtsfunktion ist im Innern des Gebietes positiv und verschwindet auf dem Rand und außerhalb des Gebietes. Die resultierenden gewichteten B-Splines erfüllen homogene Randbedingungen. Sie besitzen die üblichen Eigenschaften einer Finite-Elemente-Basis.

Das Einfügen neuer Knoten in das Gitter ist keine gute adaptive Strategie aufgrund der globalen Auswirkung des Knoteneinfügens. Stattdessen ist eine hierarchische Verfeinerung für Tensorprodukt-Splines sehr effektiv. Sie erlaubt eine lokale Änderung von Kontrollpunkten mit anschließender Modifikation kleiner Details in einigen Bereichen, ohne dabei andere Bereiche zu beeinflussen. Zur Programmierung ist eine Datenstruktur notwendig, die sowohl die Verfeinerung beschreibt als auch die Diskretisierung des Gebietes berücksichtigt. Des Weiteren werden Algorithmen zur Aufstellung und Lösung des Finite-Elemente-Systems benötigt. In dieser Dissertation

wurden solche adaptive Verfahren für gewichtete B-Splines entwickelt und mit einer entsprechenden Datenstruktur in MATLAB implementiert.

Es wurden zwei verschiedene adaptive Schemata für die Auswahl der Folge der Teilgebiete, die die Verfeinerung beschreiben, vorgeschlagen. Das erste Schema benutzt eine vorab definierte stark geschachtelte Gebietsfolge, wie sie beispielsweise in der Umgebung einer einspringenden Ecke sinnvoll ist. Für eine stark geschachtelte Gebietsfolge ist der Abstand zwischen dem Rand des Gebiets mit Gitterweite h und dem Gebiet mit Gitterweite $h/2$ größer gleich $(2n+1)h$. Für eine solche Gebietsfolge kann eine Fehlerabschätzung gezeigt werden.

Das zweite adaptive Schema ist ein automatischer Unterteilungsprozess. Die Unterteilung wird durch Vergleich der B-Spline-Koeffizienten einer Approximation mit denen einer Approximation, die durch Unterteilung aller Teilgebiete entsteht, bestimmt. Die hierarchische Unterteilung basiert dann auf den Bereichen, bei denen die Differenz zwischen den Koeffizienten eine vorgegebene Toleranz überschreitet. Für beide adaptiven Schemata erhält man Konvergenz der hierarchischen Approximationen. Die adaptiven Schemata wurden für das Poisson-Problem auf Gebieten mit einspringenden Ecken mit Verfeinerung in der Umgebung der geometrischen Singularität getestet.

Contents

Notation and Symbols

General

$\mathbb{R}$ $\mathbb{N}$ $\mathbb{Z}$	Set of real numbers, natural numbers and integers
$\preceq, \succeq, \asymp$	Inequality up to constant
Δ	Laplace operator
dist	Distance function
supp	Support of a function

Finite elements

$a(\cdot, \cdot)$	Bilinear form
$\mathfrak{L}$	Linear differential operator
$L^2(D)$	Square integrable functions
$\mathfrak{D}(D)$	Space of infinitely differentiable functions on D
$H^k(D)$	Sobolev space
$\langle \cdot, \cdot \rangle_{H^k(D)}$	Scalar product
$\|\cdot\|_{H^k(D)}$	Sobolev norm
$\|\cdot\|_{H^k(D)}$	Seminorm
grad	Gradient
div	Divergence
$D \subset \mathbb{R}^m$	Bounded domain in $\mathbb{R}^m$
$\overline{D}$	Closure of D
∂D	Boundary of the domain D

B-splines

b_k	Linear tensor B-spline
$b_{k,h}$	Linear tensor product B-spline on a uniform grid with grid width h
$\operatorname{supp} b_{k,h}$	Support of $b_{k,h}$
$Q_{l,h} = lh + [0,1]^m h$	Grid cell
$\mathcal{P}^n$	Orthogonal projection onto polynomial of degree $\leq n$
$\mathcal{Q}$	Tensor product quasiinterpolant
$\mathbb{B}_h(D)$	Splines on bounded domain D
$w_h\mathbb{B}_h(D)$	Weighted splines on bounded domain D
$\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$	Hierarchical spline space on D
w	Weight function

List of Figures

Chapter 1

Introduction

The finite element method is used to solve complex structural analysis problems in civil and aeronautical engineering. It was first introduced in the work of Courant [Cou43], Argyris, Turner et. al. [TCMT56], Clough and Zienkiewics [KJ96] in the middle of the last century (1941 and 1942). Previously, Hrennikoff (1941) [Hre41] used a similar discretization technique. Turner, Martin and Topp [TCMT56] as well as Argyris [Arg60, Arg64] used variational approximation in particular for problems in civil engineering. The method provided a rigorous mathematical foundation in 1978 with the publication of Strang and Fix [Str73]. Since then, it has been generalized and used in many branches of applied mathematics for numerical modeling of physical systems.

The finite element method is particularly useful for problems that do not have a classical solution. Solutions are obtained in Sobolev spaces named after the Russian mathematician S. L. Sobolev [Sob38]. Using Galerkin's method, we find approximate solutions by solving the variational problem in a finite dimensional subspace.

Most commonly, the finite elements, the basis of the trial space, are defined on a mesh i.e., a partition of domain D into triangles, quadrilateral, tetrahedral or hexahedral cells. Triangles and tetrahedra are preferred because of their geometric flexibility. In particular, generating hexahedral meshes in three dimensions is rather difficult.

Splines play an important role in approximation and geometric modeling. They are widely used in data fitting, computer aided design (CAD), automated manufacturing (CAM), and computer graphics. B-splines were first defined by Schoenberg [Sch46] over uniform knots. This early work revealed that splines possess powerful approximation properties. As a result, many approx-

imation schemes have been proposed [ANW67]. These schemes have become popular, especially, after de Boor's work about splines [dB72]. Another contribution to spline theory is due to Bézier who introduced the modern techniques of CAD/CAM. He uses Bernstein polynomials [Ber13] for the geometric description of free form of curves and surfaces. Similar results were obtained by de Casteljau and soon generalized to splines.

The use of B-splines as finite element basis functions seems promising. But due to the uniform support, it seems to be infeasible for two reasons. The first one is the implementation of the Dirichlet boundary conditions which is not so easy. To implement the boundary condition the approximation must be zero on boundary and outside of the domain. To accomplish this, the coefficients of the B-splines having a nonempty intersection with boundary must be zero, which results in poor approximation order. The second reason is that the B-spline basis is not uniformly stable due to very small support of some B-splines near the boundary. This leads to a large condition number of the finite element system and can cause extremely slow convergence of iterative methods.

These two problems were resolved by Prof. Klaus Höllig, Prof. Ulrich Reif and Dr. Joachim Wipper [HÖ3, Wip05] by developing a new type of splines called Weighted Extended B-splines (WEB splines). In the WEB method, the relevant B-splines are classified into two disjoint sets. The inner B-splines have at least one complete grid cell of their support inside D. All other B-splines with small support in D are classified as outer B-splines.

The first problem mentioned above can be resolved by modeling the essential boundary condition through a weight function. This technique was already employed by Kantorowich and Krylow [KK56]. For example, solutions which vanishes on boundary ∂D are approximated with linear combinations of weighted B-splines:

$$wb_k, \quad k \in K,$$

where K denotes indices of the B-splines with some support in D and w is a smooth function defined on D such that $w_{|\partial D} = 0$. The construction of weight functions has been extensively studied by Rvachev et al.

The stability problem caused by the outer B-splines can be resolved by forming appropriate linear combinations

$$b_i + \sum_{j \in J(i)} e_{i,j} b_j, \quad i \in I,$$

with inner B-splines as described in [HRW]. The sets $J(i) \subset K \setminus I$ are the sets of neighboring outer indices j. These extended B-splines inherit all the basic characteristics of standard B-splines b_i. In particular their linear span maintains full approximation order.

The combination of the above ideas lead us to the definition of weighted extended B-splines (WEB-splines)[HRW]. These new basis functions exhibit all the usual properties of standard finite elements.

For complex domains the mesh generation is the most difficult part of classical finite element methods. The WEB method requires no mesh generation. Utilizing a regular grid, the WEB method eliminates the mesh generating process. Furthermore it accelerates the solution algorithms. The uniform grid is ideally suited for parallelization and multigrid techniques. Moreover, the use of B-splines reduces the dimension of the trial space and hence the Galerkin system. Accurate approximations are possible with relatively low-dimensional subspaces. Hierarchical spline bases permit an adaptive refinement process which, however, has not yet been developed. In this thesis we study hierarchical finite element approximations with weighted B-splines in detail and develop an adaptive refinement strategy. For the sake of simplicity we use linear B-splines and consider Poisson's equation in two variables as a model problem. Extension of our ideas to a higher degree and general multidimensional problems are straightforward.

Local grid refinement is necessary to assure the rapid convergence of numerical solutions of boundary value problems. In particular, adaptive methods are well suited for domains with reentrant corners and local complexity in the solution. For classical finite element methods, adaptive refinement is studied in detail. In particular, the local refinement via Lagrange functions is well understood by Babuška.

Forsey and Bartels [FB88, FB95, FW98] have introduced hierarchical B-spline refinement. For curves, splines permit the change of control vertices and subsequent editing of fine details in some parts of the curve while keeping the other parts unaffected. For tensor product surfaces this editing and change of control points is not possible due to the global effect of knot insertion.

Instead, hierarchical refinement is very effective.

Kraft in his Ph.D. thesis has approximated functions and data [Kra98] by using hierarchical B-splines. He showed that by an iterative approximation algorithm, a good adaptive approximation with optimal local approximation order can be found.

Our approach builds on these techniques. Some ideas are similar. Key new features are the approximation results for the weighted B-spline bases in the finite element context and an adaptive refinement strategy based on B-spline subdivision.

1.1 Outline of the Thesis

After briefly introducing the history of splines as finite elements in chapter 1, we discuss elliptic problems, in particular our model problem, Poisson's problem in chapter 2. Moreover, we define Sobolev spaces. The weak form and variational formulation of Poisson's problem are also discussed in this chapter. Some of the properties of Galerkin systems along with Cea's approximation lemma are described.

Chapter 3 focusses on the construction of weighted hierarchical B-spline bases. In section 3.1 we define weight functions and discuss their construction for constructive solid geometry via the R-function method. In section 3.2 we define linear B-splines and their tensor product. The description of weighted linear B-splines and the construction of hierarchical B-spline bases are also part of this chapter. Finally, we introduce the concept of strongly nested domains and show that hierarchical B-splines have optimal local approximation order.

In chapter 4, we sketch the implementation of our adaptive method. We used MATLAB as programming language. In section 4.1, we explain the data structure to store different information about the discretization and the B-spline bases. We also explain the procedure for assembling the Ritz-Galerkin system. Numerical integration is used to generate the entries of the stiffness matrix. In section 4.5, we describe our refinement strategy. We conclude with numerical results taking an L-shaped region as a test case.

The last chapter consists of a summary and discussion of the main results and possible generalizations.

Chapter 2

Finite Element Approximation

In this section, we review some basic facts about finite element discretizations. First, we describe a standard elliptic boundary value problem. After introducing Sobolev spaces, we discuss the concept of weak solution. Then we show how this variational form of a boundary value problem leads to the Galerkin approximation.

2.1 Elliptic problems

The finite element method is particularly well suited for elliptic boundary value problems as described below.

Definition 2.1.1 (Linear differential equation of second order)**.** Let D be a bounded domain in $\mathbb{R}^2$. We consider the Dirichlet boundary value problem for the general second order elliptic partial differential equation

$$\begin{aligned} -\operatorname{div}(A\nabla u) &= f \quad \text{in } D \\ u &= 0 \quad \text{on } \partial D \end{aligned} \tag{1.1}$$

where $A : D \to \mathbb{R}^{2\times 2}$ is Lipschitz, symmetric and positive definite with smallest eigenvalue λ_1 and largest eigenvalue λ_2; in particular

$$\lambda_1(x)|\xi|^2 \le \xi^t A(x)\xi \le \lambda_2(x)|\xi|^2, \quad \forall \xi \in \mathbb{R}^2,\ x \in D. \tag{1.2}$$

The solution u of (1.1) is smooth as long as the boundary ∂D and f is smooth. Singularities can therefore only occur, when the boundary or some parts of the data are not smooth.

With the help of the linear differential operator of second order

$$\mathfrak{L} := -\operatorname{div}(A\nabla)$$

the equation (1.1) can be written as

$$\mathfrak{L}u = f \text{ in } D, \quad u = 0 \text{ on } \partial D. \tag{1.3}$$

The differential operator of the form $\mathfrak{L}$ has divergence form.

An important example is the Laplace operator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

The differential operator Δ is a linear elliptic operator $\mathfrak{L}$ with A the unit matrix. The following boundary value problem for Laplace's equation often serves as a model problem.

Definition 2.1.2 (Poisson Problem)**.** Let D be a bounded domain in $\mathbb{R}^2$ with boundary ∂D. The boundary value problem

$$-\Delta u = f \quad \text{in} \quad D \tag{1.4}$$

and

$$u = 0 \quad \text{on} \quad \partial D,$$

where f is a real valued function on D, is referred to as Poisson problem.

Poisson's equation is an elliptic, linear, non-homogeneous (if $f \neq 0$) second order partial differential equation. We call (1.4) the strong formulation of Poisson's equation.

Physically, u can represent the vertical displacement of an elastic membrane due to the application of a force f.

A classical solution u of (1.4) belongs to $C^2(D) \cap C^0(\overline{D})$. Such a solution exists, e.g., if f and the boundary ∂D are smooth. For a domain with corners or discontinuous f, the classical formulation is inadequate. We have to use a weak formulation of Poisson's problem as described in the following sections.

2.2 Sobolev space

We denote by

$$L^2(D) = \left\{ f : D \mapsto \mathbb{R} \ s.t. \int_D (f(x))^2 \ dD < +\infty \right\}$$

the space of square integrable functions on D. More precisely, $L^2(D)$ is a space of equivalence classes of measurable functions, i.e., v is equivalent to w if and only if v and w are equal almost everywhere.

The space $L^2(D)$ is a Hilbert space with the scalar product

$$\langle f, g \rangle_{L^2(D)} = \int_D f(x)g(x) \ dD$$

and the induced norm

$$\|f\|_{L^2(D)} = \sqrt{\langle f, f \rangle_{L^2(D)}}.$$

For a domain $D \subset \mathbb{R}^2$, $\mathfrak{D}(D)$ is the space of infinitely differentiable functions with compact support in D, i.e.,

$$\mathfrak{D}(D) = \{f \in C^\infty(D) : \exists K \subset D, \text{compact} : \operatorname{supp} f \subset K\}.$$

With the aid of the space of test functions $\mathfrak{D}(D)$, we now introduce the concept of weak differentiation.

Let $\alpha = (\alpha_1, \alpha_2)$ be a tuple of non-negative integers and let $f : D \to \mathbb{R}$ be a function defined on D. We define

$$D^\alpha f(x) = \frac{\partial^{|\alpha|} f(x)}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2}}$$

with $|\alpha| = \alpha_1 + \alpha_2$ the order of the partial derivative of f.

We say that $g \in L^2(D)$ is the weak derivative $D^\alpha f$ of a function $f \in L^2(D)$ if

$$\langle g, \phi \rangle_{L^2(D)} = (-1)^{|\alpha|} \langle f, D^\alpha \phi \rangle_{L^2(D)} \quad \forall \phi \in \mathfrak{D}(D). \tag{2.5}$$

By the formula for integrating by parts, this definition extends the classical notion of partial derivatives. The Sobolev space of order $k \in \mathbb{N}_0$ on a domain D consists of all functions in $L^2(D)$ for which all their derivatives up to order k belong to $L^2(D)$:

$$H^k(D) = \left\{f \in L^2(D) : D^\alpha f \in L^2(D)\ \forall \alpha : |\alpha| \le k\right\}.$$

The Sobolev spaces $H^k(D)$ are Hilbert spaces with respect to the scalar product

$$\langle f, g \rangle_{H^k(D)} := \sum_{|\alpha| \le k} \int_D (D^\alpha f)(D^\alpha g)\, dD$$

and the associated norm

$$\|f\|_{H^k(D)} = \sqrt{\langle f, f \rangle_{L^k(D)}} = \sqrt{\sum_{|\alpha| \le k} \int_D (D^\alpha f)^2\, dD}. \tag{2.6}$$

We frequently also use the semi-norm

$$|f|_{H^k(D)} = \sqrt{\sum_{|\alpha| = k} \int_D (D^\alpha f)^2\, dD}.$$

With the aid of this definition, we can rewrite the Sobolev norm (2.6) as

$$\|f\|_{H^k(D)} = \sqrt{\sum_{m=0}^{k} |f|^2_{H^m(D)}}.$$

Finally, we want to investigate the continuity properties of functions in the Sobolev space $H^k(D)$.

Theorem 2.2.1 (Sobolev Embedding Theorem)**.** *If D is a bounded open subset of $\mathbb{R}^2$ with sufficiently regular boundary, then*

$$H^k(D) \subset C^m(\overline{D}) \quad \text{if } k > m+1.$$

In particular, functions in $H^2(D)$ are continuous. The subspace $H_0^k(D) \subset H^k(D)$ consists of all functions which vanish on the boundary ∂D. More precisely, the space $H_0^k(D)$ is the closure of all the smooth functions with compact support in D with respect to the norm (2.6).

2.3 Variational form

In this section we generalize the classical concept of solution and describe a weak formulation which is better suited for finite element approximation.

Definition 2.3.1 (Weak formulation and weak solution)**.** Let D be a bounded domain with boundary ∂D, $\mathfrak{L}$ a linear elliptic operator of second order and $f \in L^2(D)$. We define a symmetric bilinear form

$$a : H^1(D) \times H^1(D) \to \mathbb{R}$$

and a linear functional $F : L^2(D) \to \mathbb{R}$ by

$$a(u,v) := \int_D [\nabla u]^T \mathbf{A} [\nabla v] \, dx$$

$$F(v) := \int_D fv \, dx.$$

A function $u \in H_0^1(D)$ satisfying the equation

$$a(u,v) = F(v) \quad \forall v \in H_0^1(D) \tag{3.7}$$

is known as a weak solution of the boundary value problem

$$\begin{cases} \mathfrak{L}u = f & \text{in} \quad D \\ u = 0 & \text{on} \quad \partial D. \end{cases} \tag{3.8}$$

The equation (3.7) is called the weak form of the elliptic problem (3.8).

In particular, for the Poisson's equation $-\triangle u = f$, we have

$$a(u,v) = \int_D \nabla u \cdot \nabla v \, dD.$$

Definition 2.3.2. (Elliptic bilinear form). The bilinear form $a : V \times V \mapsto \mathbb{R}$ over a Hilbert space V is said to be bounded if there exists a constant $\alpha_b > 0$ such that

$$|a(u,v)| \leq \alpha_b \|u\|_V \|v\|_V \quad \forall u, v \in V \tag{3.9}$$

and elliptic if, in addition, there exists a constant $\alpha_e > 0$ such that

$$a(v, v) \geq \alpha_e \|v\|_V^2. \tag{3.10}$$

In particular, this implies that a is positive definite since $a(v, v) \geq 0$ for all $v \in V$ and $a(v, v) = 0$ iff $v = 0$.

Remark

If an elliptic bilinear form $a(\cdot, \cdot)$ over a Hilbert space V is symmetric, then the following statements are equivalent:

1. $u \in V$ satisfies the variational equation

$$a(u, v) = F(v), \quad \forall v \in V, \tag{3.11}$$

2. u minimizes the quadratic functional

$$J : V \to \mathbb{R}, \quad J(v) := \frac{1}{2} a(v, v) - F(v),$$

i.e.,

$$J(u) = \min_{v \in V} J(v).$$

Ellipticity implies existence and uniqueness of a solution as asserted in the following theorem.

Theorem 2.3.1 (Lax-Milgram)**.** *Let V be a closed subspace of a Hilbert space H and $a(\cdot, \cdot)$ be an elliptic, symmetric bilinear form over V. Then the minimization problem*

$$J(u) = \min_{v \in V} J(v), \quad J(v) = \frac{1}{2} a(v, v) - F(v)$$

has a unique solution $u \in V$ for each $F \in H'$.

Combining the above statements, we conclude that the boundary value problem (3.8) has a unique weak solution for $V = H = H_0^1$.

2.4 Ritz-Galerkin system

As we have seen in the last section, the weak formulation (3.7) of an elliptic problem in the domain D can be written as

$$\text{find } u \in V : \ a(u, v) = F(v) \quad \forall v \in V \tag{4.12}$$

where V is the appropriate Hilbert space, $a(\cdot, \cdot)$ is an elliptic bilinear form from $V \times V \to \mathbb{R}$, and $F(\cdot)$ is a continuous linear functional from $V \to \mathbb{R}$. Under such conditions, the Lax-Milgram

theorem ensures the existence and uniqueness of the solution.

Let V_h be a family of spaces that depends on a positive parameter h, which is usually the grid width of the domain discretization, such that

$$V_h \subset V, \quad dim\ V_h = N_h < \infty \quad \forall h > 0.$$

The approximate problem takes the form

$$find\ u \in V_h : \ a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h. \tag{4.13}$$

Denoting the basis of V_h by $\{b_j : j = 1, 2, \cdots, N_h\}$, it is obvious that (4.13) holds for each linear combination of the basis function b_j, particularly for b_j itself. Then

$$a(u_h, b_j) = F(b_j) \quad j = 1, 2, \cdots, N_h. \tag{4.14}$$

Expressing $u_h \in V_h$ in terms of the basis functions,

$$u_h(x) = \sum_{i=1}^{N_h} u_i b_i(x),$$

equation (4.14) becomes

$$\sum_{i=1}^{N_h} u_i a(b_i, b_j) = F(b_j), \quad j = 1, 2, \cdots, N_h. \tag{4.15}$$

We denote by G the matrix (called stiffness matrix) with elements

$$g_{ji} = a(b_i, b_j) \tag{4.16}$$

and by $\mathbf{f}$ the vector with components $f_j = F(b_j)$. If we denote the vector by $\mathbf{u}$ having the unknown components u_i, then the linear equations (4.15) can be written as

$$G\mathbf{u} = \mathbf{f}.$$

We point out some of the characteristics of the stiffness matrix that are independent of the basis functions for V_h, but exclusively depend on the properties of the weak problem.

Lemma 2.4.1. *The matrix G associated with the discretization of the elliptic problem (4.12) via the Galerkin method is positive definite.*

Proof. The matrix G is said to be positive definite if

$$X^t G X \geq 0, \quad \forall X \in \mathbb{R}^{N_h} \quad \text{and also } X^t G X = 0 \Leftrightarrow X = 0.$$

The correspondence

$$X = (x_i) \in \mathbb{R}^{N_h} \leftrightarrow v_h(X) = \sum_{j=1}^{N_h} x_j b_j \in V_h$$

defines a bijective mapping between the spaces $\mathbb{R}^{N_h}$ and V_h. By definition of the Galerkin matrix we have

$$\begin{aligned} X^t G X &= \sum_{j=1}^{N_h}\sum_{i=1}^{N_h} x_j a_{j,i} x_i = \sum_{j=1}^{N_h}\sum_{i=1}^{N_h} x_i a(b_i, b_j) x_j \\ &= \sum_{j=1}^{N_h}\sum_{i=1}^{N_h} a(x_i b_i, x_j b_j) = a\left(\sum_{i=1}^{N_h} x_i b_i, \sum_{j=1}^{N_h} x_j b_j\right) \\ &= a(v_h, v_h) \geq \alpha_e \|v_h\|_V^2 \geq 0 \end{aligned}$$

Moreover, if $X^t G X = 0$, then the above inequality implies $\|v_h\|_V^2 = 0$, i.e., $X = 0$, which proves the statement. □

The lemma guarantees the unique solvability of the Galerkin system. Alternatively, we can invoke the Lax-Milgram theorem. It holds for any closed subspace of a Hilbert space, hence in particular for the finite dimensional space V_h.

The Galerkin method is stable uniformly with respect to h, as the solution satisfies the following upper bound

$$\|u_h\|_V \leq \frac{1}{\alpha_e} \|F\|_{V'}.$$

In particular, if u_h and w_h are two numerical solutions corresponding to different data F and G, then $\|u_h - w_h\|_V \leq \frac{1}{\alpha_e}\|F - G\|_{V'}$.

Now we want to prove that the solution of the Galerkin system is convergent with respect to h as h tends to zero. Consequently, we can approximate the exact solution u as accurate as desired by the Galerkin solution u_h.

Lemma 2.4.2 (Céa Lemma)**.** *Consider the variational problem*

$$a(u, v) = F(v), \quad \forall v \in V. \tag{4.17}$$

For the weak solution $u \in V$ of (4.12) and the finite element solution $u_h \in V_h$ of (4.13), we have

$$\|u - u_h\|_V \leq \frac{\alpha_b}{\alpha_e} \inf_{w_h \in V_h} \|u - w_h\|_V. \tag{4.18}$$

Proof. Let us consider the bilinear form with the same arguments $u - u_h$:

$$a(u - u_h, u - u_h) = a(u - u_h, u - v_h) + a(u - u_h, v_h - u_h).$$

Since $v_h - u_h \in V_h$, the last term is zero. Moreover,

$$|a(u - u_h, u - v_h)| \leq \alpha_b \|u - u_h\|_V \|u - v_h\|_V.$$

On the other hand, by the coercivity of $a(\cdot, \cdot)$, it follows that

$$a(u - u_h, u - u_h) \geq \alpha_e \|u - u_h\|_V^2.$$

Hence we have

$$\|u - u_h\|_V \leq \frac{\alpha_b}{\alpha_e} \|u - v_h\|_V \quad \forall v_h \in V_h.$$

Since the inequality holds for all functions $v_h \in V_h$, we can conclude

$$\|u - u_h\|_V \leq \frac{\alpha_b}{\alpha_e} \inf_{w_h \in V_h} \|u - w_h\|_V.$$

□

It is then evident that, in order for the method to be convergent, h must be sufficiently small so that the space V_h tends to fill the entire space V. In this case

$$\lim_{h \to 0} \|u - u_h\|_V = 0.$$

Chapter 3

Weighted Hierarchical Bases

To get accurate approximations and efficient solution procedures, we often use adaptive refinement of the grid. For piecewise linear basis functions, we already know many techniques, e.g., as described for hierarchical grids by Yserentant in [Yse85] and for sparse grids by Zenger in [Zen90] and J. Bungartz in [BG04]. The sparse grid technique is based on a higher-dimensional multiscale basis, which is derived from a one-dimensional multiscale basis by a tensor product construction. Comparing the hierarchical and the sparse grid methods, it is important to consider both the accuracy of the approximation and the number of grid points. This has been studied in great detail for piecewise linear approximations by Bungartz in [Bun92].

It is expected that discretizations using piecewise quadratic basis functions and, generally, piecewise polynomials of higher degree will be more accurate than discretizations using piecewise linear basis functions. However, for the sake of simplicity, we present our new technique for piecewise linear splines only. The generalization to arbitrary degree is straightforward.

3.1 Weight functions

As we remarked before, weight functions are used to represent simulation domains. This means that a domain D is described in implicit form as the set where a weight function ω is positive. The boundary ∂D corresponds to the zero set of ω. Some mild requirements on the behavior of weight functions are needed, which are made precise in the following definition.

Definition 3.1.1 (Weight function)**.** For a domain $D \subset \mathbb{R}^2$, a weight function ω of order $\gamma \in N_0$ is continuous on $\overline{D}$ and satisfies

$$\omega(x) \asymp \operatorname{dist}(x, \Gamma)^\gamma \ , \ x \in D,$$

for a subset Γ of ∂D. We assume that Γ has positive one-dimensional measure and is sufficiently regular, so that the distance function has a bounded gradient. If ω is smooth and vanishes linearly on the entire boundary ($\gamma = 1$), then it is called a standard weight function.

The majority of the boundary conditions requires that solutions vanish to first order on a portion Γ of ∂D. Accordingly,

$$\omega(x) \asymp \operatorname{dist}(x, \Gamma)$$

holds for most applications. For such cases, a smoothed distance function can be used as weight function. There are some other types of weight functions. We discuss some of them briefly.

Many simple domains permit adhoc definitions of weight functions. For example, the weight function ω for the domain in figure 3.1 is constructed by the elementwise product of the equations of five circles,

$$\begin{array}{ll} \text{big circle :} & C(x,y) = x^2 + y^2 - 16 \\ \text{small circles :} & c_i(x,y) = (x \pm 2)^2 + (y \pm 2)^2 - 1, \\ \text{weight function :} & \omega(x,y) = C(x,y) \cdot \prod_{i=1}^{4} c_i(x,y) \end{array}$$

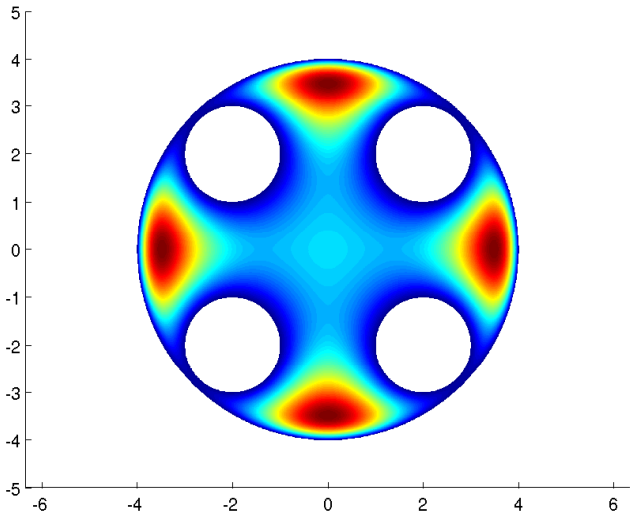

Figure 3.1: Analytic weight function.

Such a construction is possible for some special domains. If the boundary of the domain is smooth, then the distance function can be used to construct a weight function. The weight

function is defined by

$$\omega(x,y) = 1 - \left(\frac{max(\delta - \text{dist}((x,y),\partial D),0)}{\delta} \right)^{\gamma} \tag{1.1}$$

where

$\text{dist}((x,y),\partial D)$:	distance of a point (x,y) from the boundary ∂D
δ:	width of a boundary strip with increasing $\omega(x,y)$
γ:	smoothing parameter.

In an interior region of the domain, the weight function is blended with a plateau of height 1. The width of the boundary strip depends on the curvature of the boundary curve.

For domains arising in constructive solid geometry, Rvachev ([RS95]) introduced a convenient technique for constructing weight functions. It is based on signed weight functions and his R-function calculus. A signed weight function is a continuous function which is positive on D and negative on the complement of $\overline{D}$. Such weight functions can be constructed with R-functions r corresponding to Boolean set operations. More precisely, if ω_ν are signed weight functions for D_ν, then

$$\omega = r_c(\omega_1),\ \omega = r_{\bigcap}(\omega_1,\omega_2),\ \cdots,$$

are signed weight functions for $D_1^c, D_1 \bigcap D_2, \cdots$.

The R-function method of Rvachev (cf., e.g.,[RS95, RSST00]) provides a mechanism for constructing suitable functions $r_c, r_\cap, \cdots$.

An R-function is a real valued function whose sign is completely determined by the sign of its arguments. For example, the function xyz can be negative only when the number of arguments is odd. A similar property holds for the functions $x+y+\sqrt{xy+x^2+y^2}$ and $xy+z+|z-yx|$, and so on. Such functions are called R-functions or Boolean Logic functions. As in constructive solid geometry models, "primitive" weight functions can be combined according to specified set operations.

Set Operation	Corresponding R-function
Complement: D^c	$r_c(\omega) = -\omega$
Intersection: $D_1 \cap D_2$	$r_\cap(\omega_1,\omega_2) = \omega_1+\omega_2-\sqrt{\omega_1^2+\omega_2^2}$
Union: $D_1 \cup D_2$	$r_\cup(\omega_1,\omega_2) = \omega_1+\omega_2+\sqrt{\omega_1^2+\omega_2^2}$

The theory of R-functions combines logical and set operations in geometric modeling. For example, the domain in figure 3.2 can be defined as a Boolean set combination of three primitives,

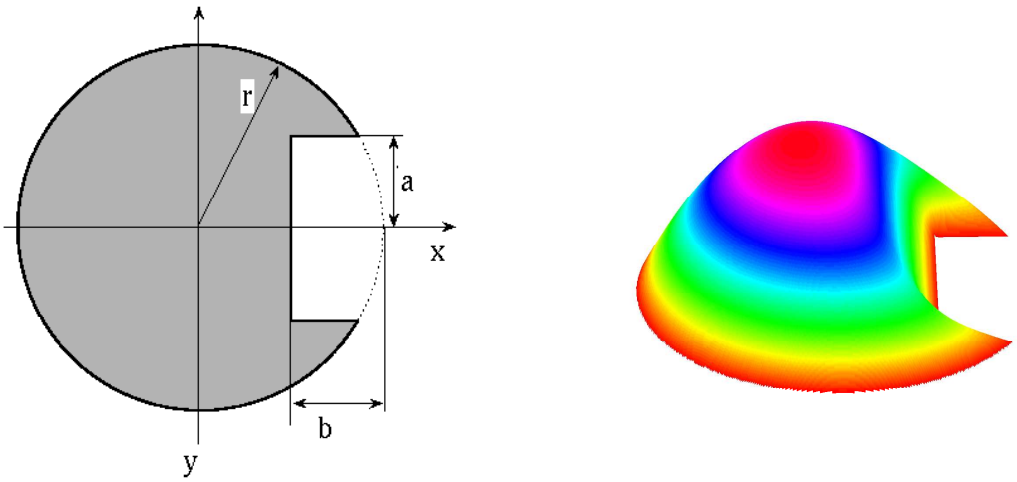

Figure 3.2: a,b. Two-dimensional domain with R-function vanishing on the boundary.

$D = D_1 \cap (D_2 \cap D_3)^c$. The weight functions $\omega_1,\ \omega_2,\ \omega_3$ corresponding to the domains $D_1,\ D_2,\ D_3$ are defined by

$$\begin{aligned} \omega_1 &= \frac{1}{2r}(r^2 - x^2 - y^2) \\ \omega_2 &= x - r + b \\ \omega_3 &= \frac{a^2 - y^2}{2a} \end{aligned}$$

The R-functions for $D_4 = D_2 \cap D_3$, $D_5 = D_4^c$ and $D = D_1 \cap D_5 = D_1 \cap D_4^c = D_1 \cap (D_2 \cap D_3)^c$ are

$$\begin{aligned} \omega_4 &= r_\cap(\omega_2, \omega_3) \\ \omega_5 &= r_c(\omega_4) \\ \omega &= r_\cap(\omega_1, \omega_5) \end{aligned}$$

where the operators $r_\cap,\ r_\cup$ and r_c are given in the above table. The final weight function w is parameterized by r, b, and a. This function is analytic everywhere except for the corner points and has normal derivative 1 on all regular points of the boundary.

3.2 Linear B-splines

Linear B-splines are the usual hat functions. In the one-dimensional case, they correspond to a uniform partition of the real line.

Definition 3.2.1 (Univariate B-splines)**.** The linear B-splines $b_{k,h}$ for a uniform grid

$$h\mathbb{Z} : \cdots, -2h, -h, 0, h, 2h, \cdots$$

with grid width h are defined by

$$b_{k,h} = \begin{cases} \frac{x}{h} - k & \text{if} \quad x \in [k, k+1]h \\ -\frac{x}{h} + k + 2 & \text{if} \quad x \in [k+1, k+2]h \\ 0 & \text{otherwise.} \end{cases}$$

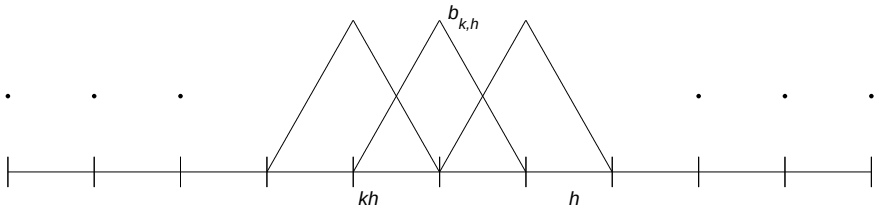

Figure 3.3: Univariate linear B-splines $b_{k,h}$, $k \in \mathbb{Z}$, with grid width h.

The B-spline $b_{k,h}$ has the value 1 at $x = (k+1)h$ and vanishes at all other grid points lh, $l \neq k+1$, as shown in figure 3.3. The support of $b_{k,h}$ is $kh + [0, 2]h$. The linear B-splines $b_{k,h}$, $k \in \mathbb{Z}$, span the space of linear splines with uniform knots.

The definition of univariate linear B-splines can be extended to two variables. There are several methods to construct bivariate linear B-splines which differ in the structure of the partition for polynomial segments. The simplest way is to form the tensor product of univariate B-splines.

Definition 3.2.2 (Tensor product linear B-splines)**.** The bivariate B-spline $b_{k,h}$ with index $k = (k_1, k_2)$ and grid width h is defined by

$$b_{k,h}(x) = b_{k_1,h}(x_1) b_{k_2,h}(x_2)$$

where $b_{k_\nu,h}$, $\nu = 1, 2$, are linear univariate B-splines along the x_1 and x_2 direction, respectively.

Figure 3.4 shows the bilinear B-spline $b_{k,h}$. The support of $b_{k,h}$ is

$$\operatorname{supp} b_{k,h} = [k_1, k_1 + 2]h \times [k_2, k_2 + 2]h \tag{2.2}$$

and the values along the grid lines correspond to multiples of the univariate B-splines. Moreover, we have marked the lower left corner kh of the support, which can be used to identify B-splines on the tensor product grid.

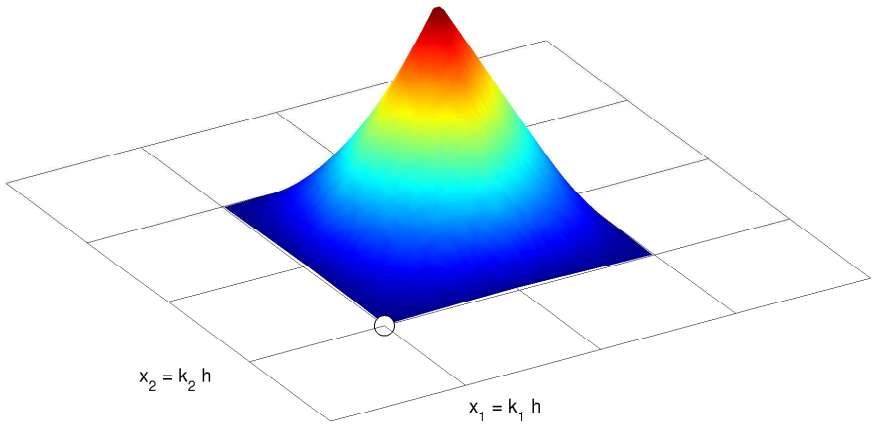

Figure 3.4: Bilinear B-spline.

The bivariate linear B-splines, like univariate linear B-splines, are positive on $kh + (0,2)^2 h$ and vanish outside this square. On each grid cell

$$D_{\ell,h} = lh + Q_* h, \quad Q_* = [0,1]^2, \quad l = (l_1, l_2) \in \mathbb{Z}^2$$

$b_{k,h}$ is a linear polynomial in each variable.

Linear combinations of bivariate B-splines span the space of linear bivariate splines with uniform grid. For a bounded domain D covered by a uniform grid with grid width h, we can define linear splines as follows:

Definition 3.2.3 (Spline functions on a bounded domain D)**.** A spline function is a linear combination of bivariate B-splines

$$\sum_{k \in K} c_k b_{k,h}.$$

The set K of relevant indices contains all k for which the B-spline $b_{k,h}(x) \neq 0$ for some $x \in D$. The space consisting of all such splines is denoted by $\mathbb{B}_h(D)$.

In figure 3.5, we illustrate the definition (3.2.3). The relevant B-splines $b_{k,h}$, $k \in K$, are marked at the lower left corner of their support $kh + [0,2]^2 h$. Depending on the shape of the

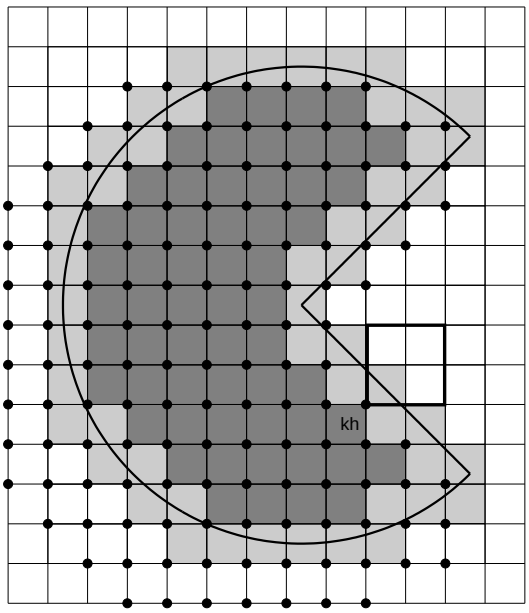

Figure 3.5: Relevant bilinear B-splines $b_{k,h}$, $k \in K$.

domain, the set of relevant indices can be irregular. Moreover, the intersection $\operatorname{supp} b_{k,h} \cap \overline{D}$ can be very small which can lead to instabilities in numerical computations.

3.3 Approximation with linear splines

Error estimates for splines are based on local polynomial approximations. These multivariate estimates play a fundamental role in finite element error analysis. Bramble-Hilbert's lemma ([BH70]) helps in finding error estimation for polynomial basis functions.

Lemma 3.3.1 (Bramble-Hilbert Lemma)**.** *Let $D \subset \mathbb{R}^2$ be a starshaped Lipschitz domain with diameter ϱ and $u \in H^n(D)$. Let $L^n u$ be the orthogonal projection onto polynomials $\mathbb{P}_{n-1}(D)$ of total degree $< n$ on D. Then*

$$\|u - L^n u\|_{H^\ell(D)} \leq const(D,n)\varrho^{k-\ell}\|u\|_{H^k(D)} \qquad \text{for } 0 \leq \ell < k \leq n.$$

The Bramble-Hilbert lemma is the prototype for error bounds for polynomial approximation in Sobolev spaces. An analogous estimate holds for linear splines.

Theorem 3.3.2. *The approximation error of a spline function* $u_h = \sum_{k\in K} c_k b_{k,h}$ *to* $u \in H^2(D)$ *satisfies*

$$\|u - \sum_{k\in K} c_k b_{k,h}\|_{H^\ell(D)} \preceq h^{2-\ell}\|u\|_{H^2(D)} \quad for \quad 0 \le \ell < 2.$$

Proof. The estimate is proved by analyzing the error separately on each grid cell $D_{\ell,h}$. Furthermore we use a quasi-interpolant

$$u_h = \mathcal{Q}u = \sum_{k\in K} \underbrace{(\mathcal{Q}_k \widetilde{u})}_{c_k} b_{k,h}$$

as convenient approximation to u, where $\widetilde{u}$ is a smooth extension of $u \in H^\ell(D)$ to $H^\ell(\mathbb{R}^m)$, as described by Stein ([Ste73]). The operator $\mathcal{Q}$ is a tensor product quasi-interpolant.

Let K' be the subset of the set of relevant indices K such that $D_{k,h} \subseteq \operatorname{supp} b_{k',h}$ for $k' \in K'$. Let $p_k \in \mathbb{P}_1(\mathcal{W}_k)$, where $\mathcal{W}_k = \bigcup_{k'\in K'} \operatorname{supp} b_{k',h}$, be the approximation to $\widetilde{u}|_{\mathcal{W}_k}$ over $\mathcal{W}_k$. Then

$$\begin{aligned} \|u - \mathcal{Q}u\|_{H^\ell(D_{k,h}\cap D)} &\le \|u - p_k\|_{H^\ell(D_{k,h}\cap D)} + \|\mathcal{Q}u - p_k\|_{H^\ell(D_{k,h}\cap D)} \\ &= \|u - p_k\|_{H^\ell(D_{k,h}\cap D)} + \|\mathcal{Q}(u - p_k)\|_{H^\ell(D_{k,h}\cap D)} \end{aligned} \tag{3.3}$$

for $\ell \in \{0, 1\}$, where we used that $\mathcal{Q}p_k = p_k$ by a standard property of quasi-interpolants. By using Bramble-Hilbert's theorem, we obtain for the first term of the right hand side of (3.3)

$$\|u - p_k\|_{H^\ell(D_{k,h}\cap D)} \le \|\widetilde{u} - p_k\|_{H^\ell(\mathcal{W}_k)} \preccurlyeq h^{2-\ell}\|\widetilde{u}\|_{H^2(\mathcal{W}_k)}. \tag{3.4}$$

By the boundedness of the quasi-interpolant, the second term gives

$$\begin{aligned} \|\mathcal{Q}(u - p_k)\|_{H^\ell(D_{k,h}\cap D)} &= \|\sum_{j\in K'} \mathcal{Q}_j(\widetilde{u} - p_k) b_j\|_{H^\ell(D_{k,h}\cap D)} \\ &\le \sum_{j\in K'} |\mathcal{Q}_j(\widetilde{u} - p_k)|\ \|b_j\|_{H^\ell(D_{k,h}\cap D)} \\ &\le \sum_{j\in K'} \|\mathcal{Q}_j\| \|\widetilde{u} - p_k\|_{L^2(\mathcal{W}_k)} \|b_j\|_{H^\ell(D_{k,h}\cap D)}. \end{aligned} \tag{3.5}$$

We estimate the three factors in turn. By standard properties of quasi-interpolants,

$$\|\mathcal{Q}_j\| \preccurlyeq h^{-1}.$$

Moreover, we can deduce from (3.4) that

$$\|\widetilde{u} - p_k\|_{L^2(\mathcal{W}_k)} \preccurlyeq h^2 \|\widetilde{u}\|_{H^2(\mathcal{W}_k)}.$$

Finally,

$$\|b_j\|^2_{H^\ell(D_{k,h}\cap D)} \preccurlyeq h^{1-2\ell}.$$

Substituting all the values in (3.5),

$$\|\mathcal{Q}(u - p_k)\|_{H^\ell(D_{k,h}\cap D)} \preceq h^{2-\ell}\|\widetilde{u}\|_{H^2(\mathcal{W}_k)}.$$

Now squaring both sides of the inequality and summing this inequality over all grid cells $D_{k,h}, k \in K$, completes the proof for $\ell \in \{0, 1\}$. □

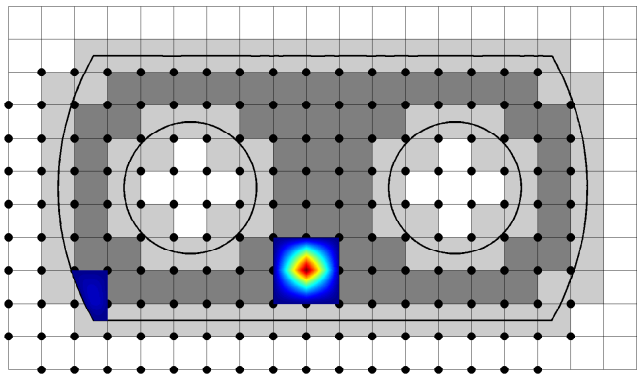

Figure 3.6: Weighted linear B-spline basis for a two-dimensional domain.

3.4 Weighted linear B-splines

We now adapt the linear splines to curved boundaries. To this end, we represent the simulation domain in implicit form:

$$D: \ w > 0.$$

Suitable weight functions can be defined by any of the methods described in section 3.1.

Definition 3.4.1 (Weighted linear splines)**.** The weighted linear splines $w\mathbb{B}_h(D)$ are defined as the space spanned by weighted linear B-splines i.e.,

$$w\mathbb{B}_h(D) = \text{span}_{k\in K}\, wb_{k,h}$$

where K is the set of relevant indices.

The positivity of the weight function is essential. If $\omega(x) = 0$ for some $x \in D$, this will affect the approximation order of weighted spline spaces. Similarly, it is also essential that $\omega = 0$ with minimum order on ∂D.

Figure 3.6 illustrates the relevant indices of linear B-splines over a bounded domain. The splines are marked by dots at the lower left corner of their support.

Results for linear splines $\mathbb{B}_h(D)$ easily extend to the weighted space $w\mathbb{B}_h(D)$. For example, if u is a smooth function which vanishes on the smooth boundary of a domain D, then, for a

smooth weight function of order 1,

$$\inf_{wp_h \in w\mathbb{B}_h(D)} \|u - wp_h\|_\ell = O(h^{2-\ell}), \quad \ell = 0, 1.$$

This estimate is an immediate consequence of theorem 3.3.2, since $v = u/w$ is smooth on $\overline{D}$. Hence,

$$\|u - wp_h\|_\ell = \|wv - wp_h\|_\ell \leq \text{const}(w)\|v - p_h\|_\ell.$$

3.5 Hierarchical refinement

When the domain is discretized into a finite element mesh, it is necessary to create a new finer mesh with an improved resolution. An alternative to remeshing is to adjust the density of the mesh by performing local refinement of the existing mesh. This means that in some regions finite elements are split to decrease their size, in other regions they are merged to reduce the resolution.

When using adaptive remeshing, we should keep in mind that it should be efficient and it should not become a bottleneck of the adaptive computations. B-splines are ideally suited for such grid refinement procedures. As is apparent from the following definition, adjusting the B-splines to a finer grid is very simple.

Lemma 3.5.1 (Subdivision)**.** *The linear B-spline with grid width h can be expressed as a linear combination of linear B-splines with grid width $h/2$:*

$$b_{k,h} = \frac{1}{2} \sum_{l=0}^{2} \begin{pmatrix} 2 \\ l \end{pmatrix} b_{2k+l,h/2}.$$

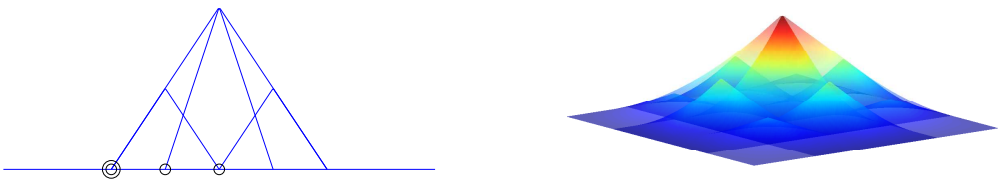

Figure 3.7: Subdivision of a univariate linear B-spline and a bivariate linear B-spline.

The univariate subdivision formula can easily be extended to the tensor product of B-splines. The definition (3.2.2) implies

Lemma 3.5.2 (Bivariate Subdivision)**.** *The relevant B-splines $b_{k,h}$ with grid width h can be expressed as linear combination of B-splines with grid width $h/2$,*

$$\begin{aligned} b_{k,h}(x) &= b_{k_1,h}(x_1) b_{k_2,h}(x_2) \\ &= \left(\tfrac{1}{2} \sum_{l_1=0}^{2} \begin{pmatrix} 2 \\ l_1 \end{pmatrix} b_{2k_1+l_1,h/2}(x_1) \right) \left(\tfrac{1}{2} \sum_{l_2=0}^{2} \begin{pmatrix} 2 \\ l_2 \end{pmatrix} b_{2k_2+l_2,h/2}(x_2) \right) \\ &= \sum_{l=0}^{2} s_l b_{2k+l,h/2}(x), \end{aligned}$$

where

$$s_l = \prod_{\nu=1}^{2} \frac{1}{4} \begin{pmatrix} 2 \\ l_\nu \end{pmatrix}.$$

Let U_h denote the coefficients of a spline function $u_h = \sum_k u_{k,h} b_{l,h}$ on the coarse grid as defined in (3.2.3). The spline u_h on the coarser grid is represented on the fine grid as

$$u_h = \sum_{l \in K_h} u_{l,h} b_{l,h}(x) = \sum_{l \in K_h} \left(\sum_{k \in K_{h/2}} s_{k-2l} u_{l,h} \right) b_{k,h/2}(x) = \sum_{k \in K_{h/2}} u_{k,h/2} b_{k,h/2}, \tag{5.6}$$

where the coefficients are related by

$$U_{h/2} = P U_h, \quad p_{k,l} = s_{k-2l}. \tag{5.7}$$

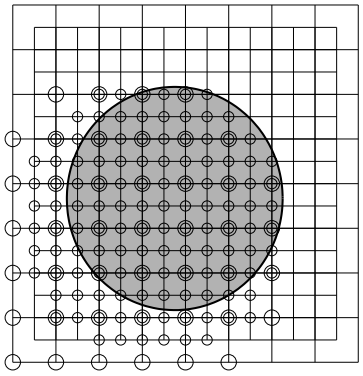

Figure 3.8: Relevant bilinear B-splines $b_{k,h}$ on the coarse grid and $b_{k,h/2}$ on the fine grid.

Figure 3.8 illustrates the grid refinement. We showed two consecutive grids with grid width h and $h/2$. The relevant B-splines are marked with large and small circles, respectively.

Adaptive finite element computations rely on adjustment of the spatial resolution of the domain discretization to deliver high accuracy where it is needed. It requires the insertion of new basis functions with smaller grid width than the coarse basis functions have. Some of the basis functions on the coarse grid must be removed to keep the finite elements linearly independent. Now we explain the process of the construction of the adaptive space for two levels of nested domains.

Let $D : D_h \supseteq D_{h/2}$ be two nested domains with the following properties:

$$\begin{aligned} D_{h/2} &= \overline{D} \cap \cup_{k \in K'} \operatorname{supp} b_{k,h} \\ \partial D_h \cap \partial D_{h/2} &= \emptyset \end{aligned}$$

From the B-splines $b_{k,h}$, we select those B-splines with support not contained in $D_{h/2}$ as basis functions. In other words, we set

$$K' \subset K_h := \{k \in \mathbb{Z}^2 : \overline{D} \cap \operatorname{supp} b_{k,h} \subseteq D_h \text{ and } \overline{D} \cap \operatorname{supp} b_{k,h} \nsubseteq D_{h/2}\}.$$

We replace the relevant B-splines $b_{k,h}$ with $k \notin K_h$ by finer B-splines via subdivision. Then the refined space is spanned by

$$b_{k,h}, \quad k \in K_h, \qquad b_{l,h/2}, \quad l \in K_{h/2}$$

where $K_{h/2}$ contains the indices k' with $\overline{D} \cap \operatorname{supp} b_{k,h/2} \subseteq D_{h/2}$. By successive repetition of this procedure, we can define the hierarchical space $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$ as follows (cf. [HÖ3]):

Definition 3.5.1 (Hierarchical spline space)**.** A basis for a hierarchical spline space $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$ consists of the bivariate linear B-splines

$$b_{k,h}, \ k \in K_h, \ h \in \{h^+, h^+/2, \cdots, h^-\}. \tag{5.8}$$

The refinement is determined by a nested sequence of domains

$$\mathbb{D} := D_{h^+} \supset D_{h^+/2} \supset \ldots \supset D_{h^-}. \tag{5.9}$$

The set K_h contains the indices (k_1, k_2) for which $\overline{D} \bigcap \operatorname{supp}\ b_{k,h}$ is a subset of $\overline{D_h}$ (with non zero measure), but, not contained in $\overline{D_{h/2}}$ for $h > h^-$.

As is illustrated in figure 3.9, each set $\overline{D_{h/2}}$ is the intersection of $\overline{D}$ with the union of supports of B-splines with grid width h. These B-splines are not part of the hierarchical basis, but are replaced, via subdivision, by B-splines $b_{k,h/2}$ on the finer grid.

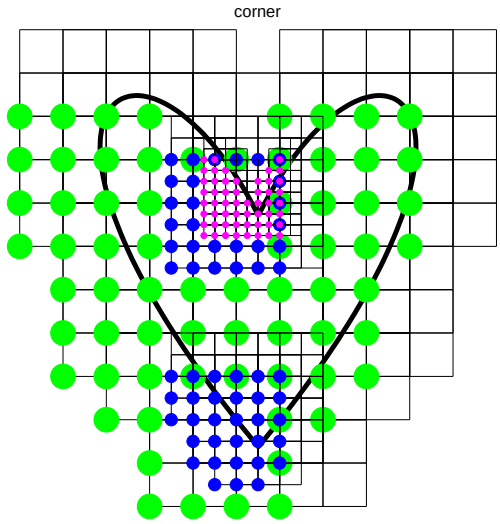

Figure 3.9: Bilinear hierarchical B-spline basis.

Example 3.5.1. A typical example of a hierarchical spline space results from refinement at the corner of a domain as occurs, e.g., in finite element approximation of singularities. As is shown in figure 3.10, for $D = (0,1)^2$, we choose $h^+ = 1/4$, $h^- = h^+/8$ and $D_h = (0, 4h_1) \times (0, 4h_2)$. The B-splines which belong to the corresponding hierarchical basis are marked at the lower left corners of their supports with the size of the circles indicating the grid level. On the coarsest grid, the B-splines

$$b_{k,1/4}, \quad k \in K_{1/4} : \quad -1 \leq k_\mu < 4, \ k_1 > 0 \vee k_2 > 0. \tag{5.10}$$

have support in $D_{1/4}$ and do not vanish on $D_{1/4} \backslash D_{1/8}$. The B-splines

$$b_{k,1/4}, \quad -1 \leq k_\mu < 1, \tag{5.11}$$

are replaced by B-splines $b_{k,1/8}$ on the finer grid. They become part of the hierarchical basis if their support intersected with $\overline{D}$ is not contained in $\overline{D}_{1/16}$, i.e.

$$K_{1/8} : \quad -1 \leq k_\mu < 3, \ k_1 > 0 \vee k_2 > 0. \tag{5.12}$$

Continuing in this fashion, we see that for this example the subsequent index set $K_{1/16}$ has the same form. For the finest grid level, $K_{1/32} = \{-1, ..., 2\}^2$.

Theorem 3.5.3. *The B-splines spanning $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$ are linearly independent.*

Proof. If

$$\sum_h \sum_{k \in K_h} c_{k,h} b_{k,h}(x) = 0 \quad \forall x \in D, \tag{5.13}$$

we can show inductively for $h = h^+, h^+/2, ..., h^-$, that the coefficients $c_{k,h}$ are zero. First, we restrict x to the open set $U = D_{h^+} \backslash \overline{D_{h^+/2}}$. By definition of the hierarchical spline space, all

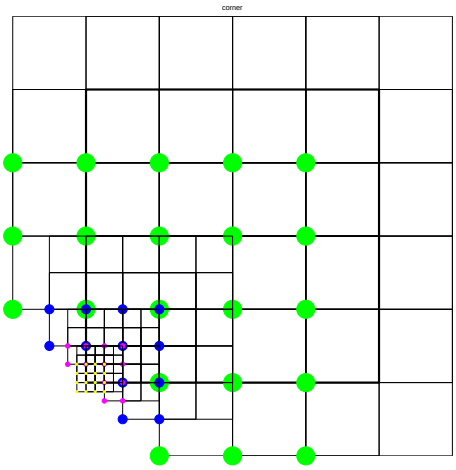

Figure 3.10: Basis for hierarchical splines on a rectangular domain.

B-splines with grid width $h < h^+$ vanish on this set. On the other hand, for each B-spline $b_{k,h^+},\ k \in K_{h^+}$, there exists a point x, and hence also a neighborhood in U, where this B-spline is nonzero. Hence, by the local linear independence of the B-splines with grid width h^+, all coefficients c_{k,h^+} must be zero.

Now we repeat the argument, restricting x successively to the set $D_h \backslash \overline{D_{h/2}},\ h = h^+/2, h^+/4, ..., h^-$, and setting $D_{h^-/2} = \emptyset$. Arguing as above, we conclude that the coefficients $c_{k,h}$ for all levels are zero. □

3.6 Error estimate

In this section we derive a local error estimate. This is facilitated if the number of B-splines which are non-zero at any given point is uniformly bounded with respect to the number of grid levels. To this end, we need some restriction on the choice of the domains.

Definition 3.6.1 (Strongly nested domains)**.** The domains D_h which characterize the refinement of a hierarchical spline space $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$ are strongly nested if the support of any B-spline $b_{k,2h}$ in the basis does not intersect $\overline{D_{h/2}}$.

Strongly nested domains have the following properties:

- B-splines in the basis with non-empty intersection of their supports must belong to two consecutive hierarchical levels.
- The distance between the boundary ∂D_h and $\partial D_{h/2}$ is greater than $2h$.

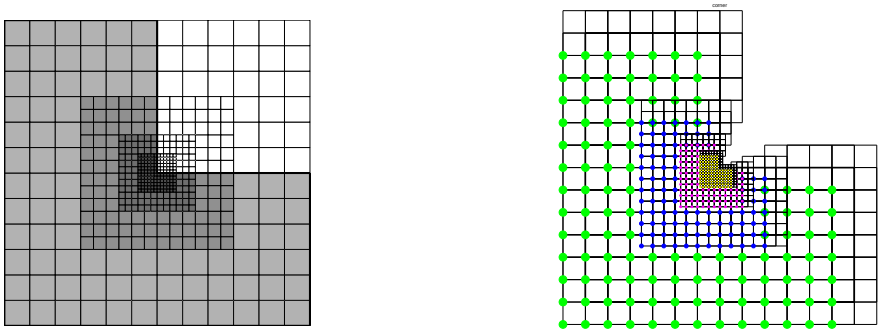

Figure 3.11: Strongly nested domains with four levels of refinement.

Figure 3.11 shows a sequence of strongly nested domains over an L-shaped domain up to four levels.

Theorem 3.6.1. *Assume that the domains D_h, which characterize the refinement of a hierarchical spline space $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$, are strongly nested. Then, for any smooth bivariate function f, there exists a hierarchical spline p with*

$$|f(x) - p(x)| \leq \text{const } \; h^2 \sum_{\nu=1}^{2} \|\partial_\nu^2 \widetilde{f}\|_{\infty, D_{x,h}}, \quad x \in D_h \backslash D_{h/2}, \tag{6.14}$$

where we set $D_{h^-/2} = \emptyset$, $\widetilde{f}$ is a smooth extension of f to $\mathbb{R}^2$, and $D_{x,h}$ denotes a disc with radius $\leq \text{const}\, h$.

Proof. The proof is divided into three steps: some preliminary considerations, a construction of a projector onto $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$, and a local Taylor approximation.

(i) To avoid a special treatment of the domain boundary, we work with a smooth extension $\widetilde{f}$ to $\mathbb{R}^2$ and, accordingly, we replace D by $\mathbb{R}^2$ as well. Clearly, restricted to the original domain, the B-spline basis for the hierarchical spline space remains unchanged.

(ii) An approximation of the asserted optimal order is obtained as a projection

$$f \mapsto p = Qf = \Sigma_{h^- \leq h \leq h^+} \Sigma_{k \in K_h} (\mathcal{Q}_{k,h} f)\; b_{k,h}. \tag{6.15}$$

For the construction of an appropriate projector Q we use the functionals $Q_{k,h}$ of standard projectors onto the spline space $\mathbb{B}_h(\mathbb{R}^2)$. As is possible, we choose, for $k \in K_h$, the support of these functionals in grid cells $D_{k,h}$ which are outside but closest to $D_{h/2}$. Denoting by $K_{k,2h}$ the indices k' in K_{2h} for which the support of $b_{k',2h}$ intersects $D_{k,h}$, we now define

$$\mathcal{Q}_{k,h} f = Q_{k,h} \left(f - \sum_{k' \in K_{k,2h}} (Q_{k',2h} f)\; b_{k',2h} \right). \tag{6.16}$$

In order to check that these functionals define indeed a projector we have to show that

$$c = \mathcal{Q}_{k,h} b_{k'',h''} = \delta_{k,k''}\delta_{h,h''} \tag{6.17}$$

for all relevant indices and grid widths. We consider four cases:

(a) $h'' < h$: Since the support of all functionals appearing in the definition of $\mathcal{Q}_{k,h}$ lies outside of the set $D_{h/2}$, which contains the support of $b_{k'',h''}$, $c = 0$.

(b) $h'' = h$: Since the support of the functionals $Q_{k',2h}$ lies outside of D_h,

$$c = Q_{k,h}(b_{k,h''} - 0) = \delta_{k,k''} \tag{6.18}$$

by the orthogonality on a fixed grid level.

(c) $h'' = 2h$: If $\operatorname{supp} b_{k'',2h}$ does not intersect $D_{k,h}$, $Q_{k,h} b_{k'',2h} = 0$ and $k'' \notin K_{k,2h}$. Hence, $c = 0$. If on the other hand the intersection is not empty, $\sum_{k' \in K_{k,2h}} \ldots = b_{k'',2h}$ by the orthogonality of the functionals $Q_{k',2h}$. Hence, $c = 0$ also in this case.

(d) $h'' > 2h$: For $k' \in K_{k,2h}$ the support of the B-spline $b_{k',2h}$ intersects D_h. Hence, $D_{k',2h}$ touches the boundary of D_h. This implies $c = 0$, since for strongly nested domains $\operatorname{supp} b_{k'',h''}$ does not touch D_h and, as a consequence,

$$Q_{k',2h} b_{k'',h''} = 0 \quad \forall k' \in K_{k,2h}.$$

Moreover, since $\operatorname{supp} b_{k'',h''} \cap D_{k,h} = \emptyset$, $Q_{k,h} b_{k'',h''} = 0$, too.

(iii) With q, the Taylor polynomial of coordinate degree 1 at x,

$$\begin{aligned} |f(x) - p(x)| &\leq \underbrace{|f(x) - q(x)|}_{0} + |\mathcal{Q}q - \mathcal{Q}f(x)| \\ &\leq \sum_{h'} \sum_{K_{h'}} |\mathcal{Q}_{k,h'}(f - q)| b_{k,h'}(x). \end{aligned}$$

Because of the small support of the B-spline, only few terms contribute to the sum. For $x \in D_h \backslash D_{h/2}$, $h' = h$ or $h' = 2h$, the number of relevant B-splines is $\leq \text{const}_1$. Similarly, the number of functionals $Q_{k,h''}$ in the definition of $\mathcal{Q}_{k,h'}$ is $\leq \text{const}_2$ and $h'' \in \{h, 2h, 4h\}$. Moreover,

$$\text{supp } b_{k,h''} \bigcap \text{supp } b_{k,h'} \neq \emptyset \wedge x \in \text{supp } b_{k,h'} \tag{6.19}$$

which implies that the support of any of the functionals $Q_{k,h''}$ is contained in an open disc $D_{x,h}$ with radius $\leq \text{const}_3\, h$ and center x. Hence, it follows that

$$|f(x) - p(x)| \leq \text{const}_1 \text{const}_2 \sup_{k,h''} \|Q_{k,h''}\| \; \|f - p\|_{\infty, D_{x,h}}. \tag{6.20}$$

By the uniform boundedness of the functionals of standard projectors and the error estimate for the Taylor polynomial of coordinate degree 1, we obtain the asserted error bound. □

The theorem easily extends to weighted hierarchical approximations wp_h, $p_h \in \mathbb{B}_{(h^+,h^-)}(D)$. Clearly, if $u = wv$ with smooth w and v, then the error $wp_h - u$ can be estimated in terms of bounds for $p_h - v$.

Chapter 4

Implementation

In the previous chapter we have defined the hierarchical B-spline basis and described the construction of the FE space. In this chapter we shall discuss the data structure for the implementation of FE schemes.

Matlab provides a convenient way to construct the data structure to implement the FE discretizations. First, we shall discuss how to store the information about the hierarchical basis in Matlab. Then, we describe the assembly of the Ritz-Galerkin matrix. Numerical integration plays a crucial role. Appropriate procedures are explained in section 3. Sections 4 and 5 are devoted to matrix assembly and adaptive refinement. Finally, numerical examples are given.

4.1 Grid data structure

Mesh based computations require a data structure to store the geometric information. The information should be generated at the preprocessing stage, so that the processing module knows in particular

- the vertices belonging to each element
- the spatial coordinates of each vertex
- the elements or vertices whose supports are intersected by ∂D.

Since the hierarchical B-spline space involves B-splines of different support sizes at different hierarchical levels, the data structure requires more information. When an adaptive refinement is performed, the data structure becomes more complex.

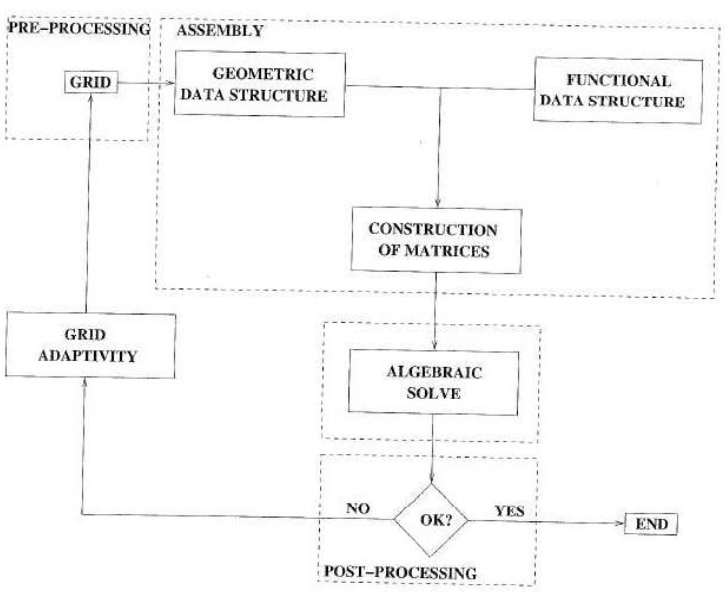

Figure 4.1: Phases of an adaptive FE code.

The parameters for a hierarchical spline space are stored in an array of structures $HB(\ell),\ \ell = 1, 2, \cdots, L$:

- $HB(\ell).h$: grid width
- $HB(\ell).R$: rectangular region $[r_1 h, r_2 h] \times [r_3 h, r_4 h]$ comprising the support of the B-splines
- $HB(\ell).C$: coefficients $[c_{1,1}, \cdots, c_{m,n}]$ of the B-splines corresponding to R ($m = r_2 - r_1 - 1,\ n = r_4 - r_3 - 1$)
- $HB(\ell).L$: indices $[\ell_1, \cdots]$ of the substructures generated via refinement.

Initially, the matrix of coefficients C is filled with zeros. If some portion of region R at level ℓ is further subdivided, then by the construction of hierarchical B-spline space, this subregion of R must be the union of supports of B-splines at level ℓ. As these B-splines are removed from the spline space, the coefficients of these B-splines are fixed as 'NaN'.

The data structure for figure 4.2 can be described as follows. Initially, the first level of discretization is stored in a simple manner under the fields $HB(1).h = 1$, $HB(1).R = [0, 6, 0, 6]$, $HB(1).C = O_{5\times 5}$, and $HB(1).L = [\ ]$. To proceed to the next level, suppose we are interested in two sets of rectangular regions to be subdivided which comprise $\operatorname{supp} b_{(1,1),h}$ and

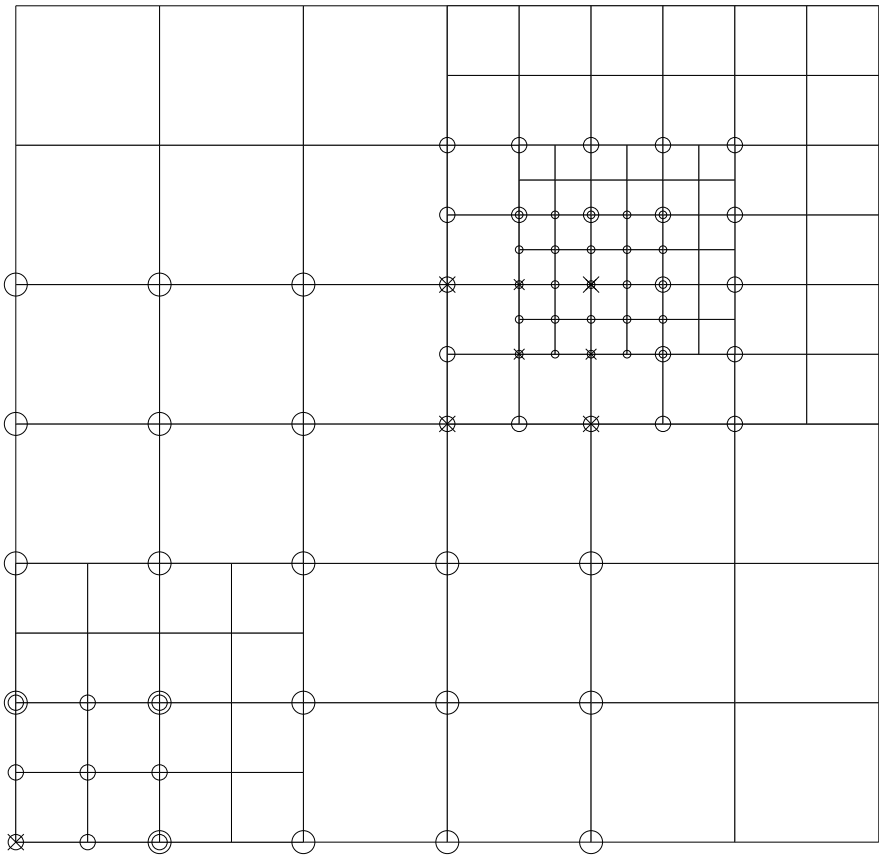

Figure 4.2: Hierarchical grid with three refinements.

$\cup_{i,j=4}^{5} \operatorname{supp} b_{(i,j),h}$. The data for these two regions are stored in $HB(2)$ and $HB(3)$. The subdivision process affects the initial setting of $HB(1)$. The entries $c_{1,1}, c_{4,4}, c_{4,5}, c_{5,4}, c_{5,5}$ in the coefficient matrix $HB(1).C$ are now replaced by 'NaN', and the field $HB(1).L$ becomes $[2, 3]$ depicting two sub data fields $HB(2)$ and $HB(3)$. After the subdivision process, the data for the first level become:

$HB(1).h = 1$

$HB(1).R = [0, 6, 0, 6]$

$HB(1).C = O_{5\times 5}, c_{1,1} = c_{4,4} = c_{4,5} = c_{5,4} = c_{5,5} = NaN$

$HB(1).L = [2, 3]$

Since $HB(2)$ has no further fields of interest, the data are

$HB(2).h = 1/2$

$HB(2).R = [0, 4, 0, 4]$

$HB(2).C = O_{3\times 3}$

$HB(2).L = [\,]$

The field $HB(3)$ has a further region to be subdivided and stored in the subdata field $HB(4)$. The rectangular region in $HB(4)$ is $\cup_{i,j=2}^{3} \operatorname{supp} b_{(i,j),h}$, where $b_{(i,j),h}$ are B-splines of the second level. The entries of the coefficient matrix in $HB(3).C$ which are associated with $b_{(i,j),h}, i, j = 2, 3$ are replaced by 'NaN'. The final data fields are

$HB(3).h = 1/2$

$HB(3).R = [6, 12, 6, 12]$

$HB(3).C = O_{5\times 5}, c_{2,2} = c_{2.3} = c_{3.2} = c_{3,3} = NaN$

$HB(3).L = [4]$

$HB(4).h = 1/4$

$HB(4).R = [14, 20, 14, 20]$

$HB(4).C = O_{5\times 5},$

$HB(4).L = [\,]$

In figure 4.2, the active B-splines are marked with circles of different sizes at the lower left corners of their support which correspond to different hierarchical levels , while the inactive or subdivided B-splines are marked with an '×' at the lower left corner of their support.

4.2 Assembly of the Ritz-Galerkin system

Having discretized the domain, the next step is to generate and assemble the stiffness matrix and load vector by using the hierarchical linear B-spline basis. In this phase, we construct the functional data structure starting from the geometric data structure obtained by the mesh.

Each basis function can be identified by the indices (ν, μ) of its grid position within a rectangle $HB(k).R$ of the structure array. We recall that we have set the coefficients of irrelevant B-splines equal to 'NaN'. We eliminate all irrelevant B-splines from the list of indices. Only the relevant B-splines take part in the further calculations.

We represent the relevant B-splines by a list of triplets (k, ν, μ). From this list, the exact position of each B-spline is easily determined. The position of the B-spline is identified by the

lower left corner of its support. For instance, let b_ℓ be determined by the triplet (k, ν, μ), then the lower left corner of the support of the B-spline b_ℓ is $(r_1 + \nu - 1, r_3 + \mu - 1)h$, where h and r_1, r_3 can be obtained from the data structure $HB(k)$.

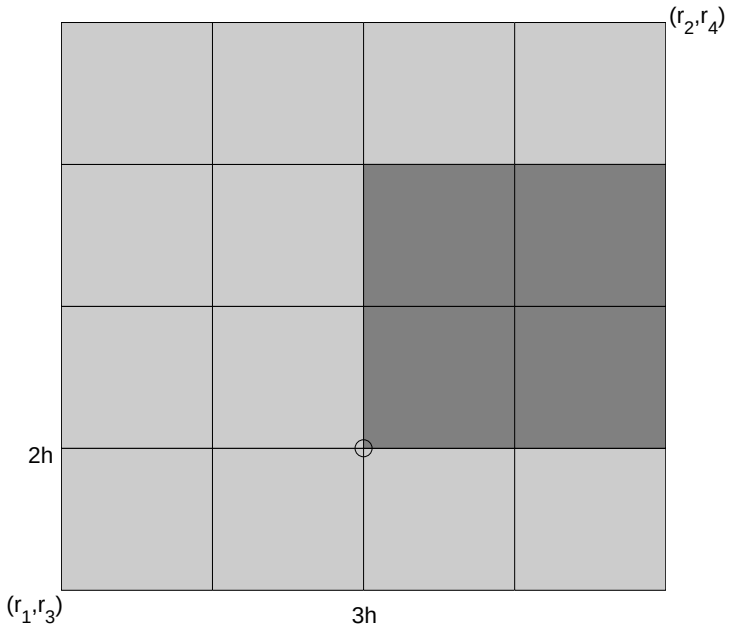

Figure 4.3: Support of the B-spline $b_{(3,2),h}$ of the k-th rectangular region.

As an example, the lower left corner of the support of the B-spline with index $(3, 2)$ is $(r_1 + 2, r_3 + 1)h$. The support of the B-spline is shown in figure 4.3 as the darker region. The lower left corner of the support is marked by a circle.

The hierarchical spline space in figure 4.2 with four rectangular domains has 20+9+21+25 = 75 relevant B-splines. Scanning the grids in the natural order, the list has the following form:

$$HBLIST = \begin{bmatrix} 1,2,1 \\ \cdots \\ 1,3,5 \\ 2,1,1 \\ \cdots \\ 2,3,3 \\ 3,1,1 \\ \cdots \\ 3,5,5 \\ 4,1,1 \\ \cdots \\ 4,5,5 \end{bmatrix}$$

4.3 Numerical integration

To assemble the Ritz-Galerkin system, we need to compute integrals over subsets of the domain D. This is done by summing the contributions from each grid cell $D_{\ell,h}$; i.e., the integrals have the form

$$\int_{D_{\ell,h} \cap D} \phi,$$

where ϕ depends on the basis functions, coefficients of the differential equation and other parameters.

There are different methods for numerical integration. However, for our application, Gauss quadrature yields the most efficient approximation.

For a univariate function ϕ, the integral is the weighted sum of the values of ϕ on M points, i.e.,

$$\int_a^b \varphi(x)dx \approx \sum_{l=1}^{M} w_l \varphi(x_l) \tag{3.1}$$

The points can be constructed with the help of orthogonal polynomials. The resulting Gauss Legendre formula is exact for polynomials of degree $\leq 2M-1$, i.e.,

$$\int_a^b p(x)dx = \sum_{l=1}^{M} w_l p(x_l) \quad \forall p \in \mathbb{P}_{2M-1}([a,b]).$$

If $[a,b]$ is not the standard interval $[-1,1]$, then we can transform the Gauss points and the weights via:

$$\begin{array}{rcl} \widetilde{x}_l & := & \frac{1}{2}(a+b+(b-a)x_l) \quad \text{and} \\ \widetilde{w}_l & := & \frac{1}{2}(b-a)w_l \end{array}$$

for all $l \in \{1, \cdots, M\}$.

Let $W = [a, b] \times [c, d]$ be a rectangle on $\mathbb{R}^2$. Then the integral over W can be approximated by the product rule:

$$\int_W \varphi(x)dx = \int_a^b \int_c^d \varphi(x, y)dxdy \approx \sum_{k=1}^{M} \sum_{l=1}^{M} v_k w_l \varphi(x_k, y_l). \tag{3.2}$$

The bivariate Gauss Legendre formula is exact for all bivariate polynomials of degree $\leq 2M - 1$.

We now turn to the discussion of the domain integral $\int_{D_{\ell,h} \cap D} \phi$. The grid cells $D_{\ell,h}$ can be classified into inner, boundary and outer cells. For inner grid cells, the Gauss product formula can be used directly.

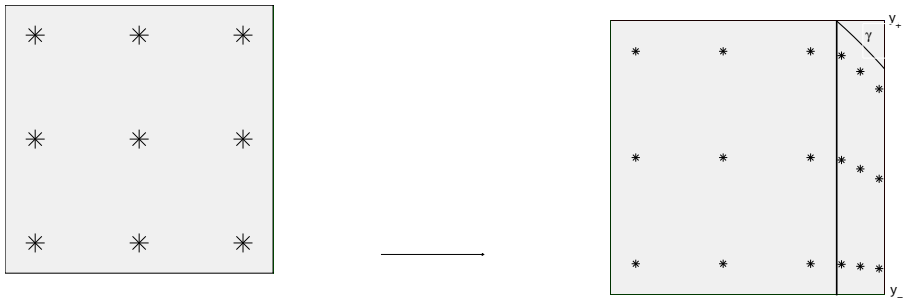

Figure 4.4: Transformation of Gauss points.

Boundary cells have to be partitioned into deformed rectangles. Very few cuts are required for the partition. Figure 4.4 shows an example.

The standard Gauss formula is easily mapped onto the subdomain. For example, if $t \longmapsto \gamma(t), 0 \leq t \leq 1$ parametrizes the curved boundary segment of the left domain D in the figure 4.4, then

$$[0, 1]^2 \ni \begin{pmatrix} s \\ t \end{pmatrix} \mapsto \alpha(s, t) = \begin{pmatrix} \gamma_1(s) \\ y_- + (\gamma_2(s) - y_-)t \end{pmatrix}$$

is a parameterization of ∂D. We can apply the tensor product formula in a straightforward way and obtain

$$\int_{D_{\ell,h} \cap D} \varphi(x)dx = \int_0^1 \int_0^1 \varphi(\alpha(s, t)) \left| \gamma_1'(s) (\gamma_2(s) - y_-) \right| dsdt,$$

and we can use the standard Gauss formula for $[0, 1]^2$ to approximate the integral over D.

4.4 Matrix assembly

First, we consider the assembly of Ritz-Galerkin matrix. Since the bilinear form $a(\cdot, \cdot)$ is defined in terms of integrals, it can be computed by adding the contributions from each grid cell. In

our data structure, we represented each B-spline as a triple (k, ν, μ), where k is the index of the structure array and (ν, μ) is the index of the B-spline corresponding to the rectangular region stored as $HB(k).R$. To assemble the Galerkin matrix G we loop over the list of B-splines, i.e.,

$$G = [g_{\ell,\ell'}], \quad g_{\ell,\ell'} = \int_D \nabla(wb_\ell)\nabla(wb_{\ell'}).$$

Let b_ℓ and $b_{\ell'}$ be determined by the triplets (k, ν, μ) and (k', ν', μ'), respectively. Their support intersects only when the difference between their lower left corners is $< h_k + h_{k'}$, where h_k and $h_{k'}$ are the grid width of b_ℓ and $b_{\ell'}$. We add the contributions from the grid cells in the intersection.

In the list of B-splines there are also some irrelevant B-splines whose support is outside the domain D. The entries in the Galerkin matrix corresponding to such B-splines produce a zero row and a zero column. We add 1 on the diagonal to make the system non-singular and 0 on the corresponding entry of the right hand side.

The right hand side of the Galerkin system is

$$F = [f_{\ell'}], \quad f_{\ell'} = \int_D f w b_{\ell'}.$$

To compute this integral, we loop over the list of B-splines and add the contributions of each grid cell.

4.5 Adaptive refinement

Adaptive refinement is a very crucial step in finite element approximation. An appropriate strategy yields a reduction in error. Here we first describe the refinement procedure and then it is explained in a one dimensional example.

Consider an approximation in a weighted hierarchical spline space

$$w\mathbb{B}_{(h^+,h^-)} \ni p = w \sum_{h=h^+}^{h^-} \sum_{k \in K_h} c_{k,h} b_{k,h}$$

over a nested sequence of domains

$$\mathbb{D} : D_{h^+} \supset D_{h^+/2} \supset \cdots \supset D_{h^-}.$$

The key step of an adaptive method is to decide in which region further refinement is needed. To achieve this goal, we refine the existing grid and compute an approximation on the refined grid. Let the new approximation be

$$w\mathbb{B}_{(h^+/2,h^-/2)} \ni \widetilde{p} = w \sum_{h=h^+/2}^{h^-/2} \sum_{k \in K_h} \widetilde{c}_{k,h} b_{k,h}.$$

Here it is important to note that the domains $D_{h^+}, D_{h^+/2}, \cdots, D_{h^-}$ have not been changed in size but in grid width. A comparison of $\widetilde{p}$ and p is facilitated by converting to the same representation via subdivision. We express p in terms of the basis for $w\mathbb{B}_{(h^+/2,h^-/s)}(\mathbb{D})$ i.e.,

$$p = p' = w \sum_{h=h^+/2}^{h^-/2} \sum_{k \in K_h} c'_{k,h} b_{k,h}.$$

Now, both p and $\widetilde{p}$ are represented on the same weighted hierarchical spline space and can be compared more easily. We compare them with respect to the coefficients of the basis functions. We refine the domain where $|c'_{k,h} - \widetilde{c}_{k,h}|$ is greater than the given tolerance. This means that we enlarge the domain, $D_h \rightarrow D'_h$, so that it covers the support of all B-splines for which $|c'_{k,h} - \widetilde{c}_{k,h}|$ exceeds the tolerance tol. For $h = h^-/2$, a new domain $D_{h^-/2}$ can be defined in this fashion. As a result we obtain a new sequence of nested domains

$$\mathbb{D} : D_{h^+} \supset D'_{h^+/2} \supset \cdots \supset D'_{h^-} \supset D_{h^-/2},$$

describing the refinement.

We illustrate this strategy for a simple univariate example, for simplicity without weight function.

Let a function f be defined on a domain $D = (0, 6)$. The domain D is discretized into a nested sequence of domains up to two levels with grid width $h^+ = 1$ and $h^- = 1/2$.

Let $D_1 = (-1, 7), D_{1/2} = (-1/2, 3)$. The corresponding sets of indices are $K_1 = \{2, 3, 4, 5\}, K_2 = \{-1, 0, 1, 2, 3, 4\}$. There are ten relevant B-splines that form hierarchical B-spline space over D. Let $p \in \mathbb{B}_{1,1/2}(\mathbb{D})$ be an approximation to f.

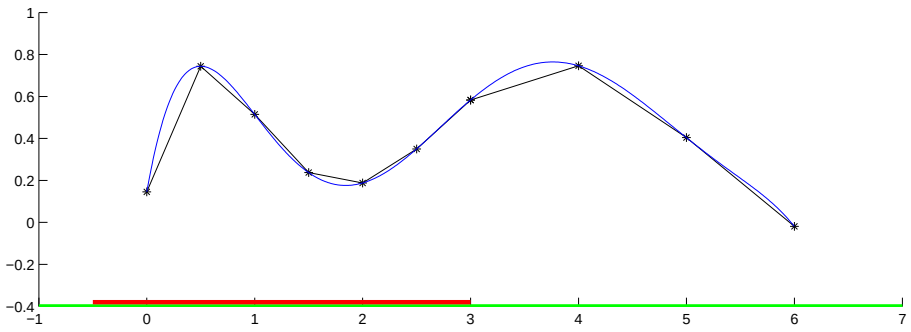

Figure 4.5: The function f (blue), its approximation p (black *) on D_1 (green) and $D_{1/2}$ (red).

The figure 4.5 shows the hierarchical discretization D_1 (green) and $D_{1/2}$ (red). The function f (blue) and an approximation p (black *) are also shown in the figure. We consider the three steps of refinement procedure in turn. First comes the refinement of the hierarchical grid and the subdivision of the existing approximation $p \in \mathbb{B}_{1,1/2}(\mathbb{D})$ by the subdivision algorithm. The refinement $p' = p$ is now expressed as an element of the refined hierarchical space $\mathbb{B}_{1/2,1/4}(\mathbb{D})$. The second step is to find an approximation $\widetilde{p}$ from the refined space $\mathbb{B}_{1/2,1/4}(\mathbb{D})$ and compare its coefficient with the approximation p'. The figure 4.6 shows the refinement p' in black while the approximation $\widetilde{p}$ from the refined space is shown in red.

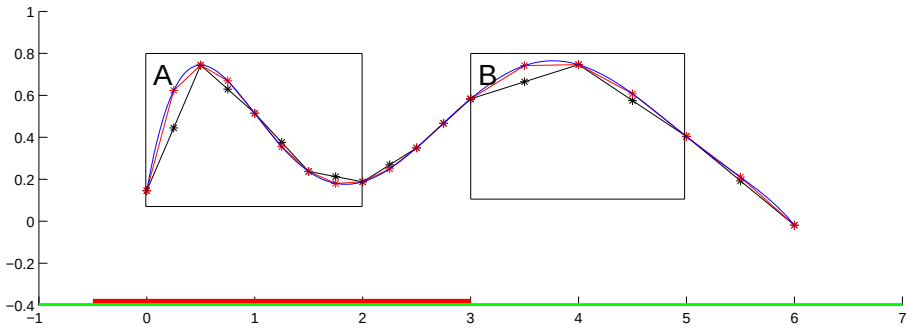

Figure 4.6: The function f (blue), subdivision of the approximation $p = p'$ (black *) and approximation $\widetilde{p}$ (red *) on refined grid D_1 (green) and $D_{1/2}$ (red).

The refinement p' and the approximation $\widetilde{p}$ are now defined on the same refined space, therefore they can be compared in terms of their coefficients. The discrepancy of coefficients $> tol$ is indicated by the rectangular regions A and B.

The third step is to refine the grid according to the rectangular regions A and B. First we look at the rectangular region B. Errors $|c'_{k,h} - \widetilde{c}_{k,h}| > tol$ occur for $k = 6$ and $k = 8$. The corresponding B-splines $b_{6,1/2}$, $b_{8,1/2}$ have supports $[3, 4]$ and $[4, 5]$. The domain $D_{1/2}$ has to be enlarged so that its closure covers these intervals, i.e. $D_{1/2} \to (-1/2, 5)$. Similarly for the region A, the discrepancy occurs for the B-splines $b_{k,1/4},\ k = 0, 1, \cdots, 6$, with $\cup_{k=0}^{6} \operatorname{supp} b_{k,1/4} = [0, 2]$. Accordingly, a new domain $D'_{1/4} = (0, 2)$ is defined. This leads to the new hierarchical spline space $\mathbb{B}_{(1,1/4)}(\mathbb{D}')$ defined on the nested domains

$$\mathbb{D}' : D_1 \supset D'_{1/2} \supset D'_{1/4}.$$

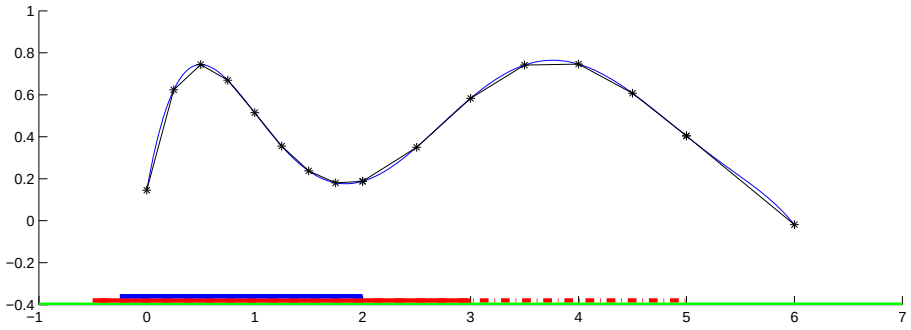

Figure 4.7: The function f (blue), its approximation p (black *) on D_1 (green), $D'_{1/2}$ (red), and $D'_{1/4}$ (blue).

In figure 4.7 we show D_1 (green), $D'_{1/2}$ (red solid and doted) and $D'_{1/4}$ (blue) along with the functions f (blue) and an approximation p (black *) to f on D_1, $D'_{1/2}$ and $D'_{1/4}$.

The application of the refinement strategy to the solution of the Poisson problem

$$\begin{array}{ccc} -\Delta u = f & \text{in} & D \\ u = 0 & \text{on} & \partial D \end{array}$$

is straightforward. For the refinement step we compare a refined Galerkin solution with the Galerkin solution from the refined space. Some examples will be given in the next section.

4.6 Numerical examples

In this section, we discuss some numerical examples. We compare the solutions obtained from the WEB and WB methods with the solutions and error estimates obtained from our adaptive method.

Example 4.6.1. Our first example deals with the solution of the Poisson problem

$$\begin{array}{ccc} -\Delta u = 1 & \text{in} & D \\ u = 0 & \text{on} & \partial D \end{array}$$

where D is an L-shaped domain. The domain D has a reentrant corner at $(1, 1)$.

The solution shows a singularity at the reentrant corner. We approximated the solution by the WEB method and the adaptive method. First, we choose a sequence of strongly nested domains (figure 4.8, left) with an appropriate refinement near the reentrant corner. Second, we illustrate one step of automatic refinement as described in section 4.5.

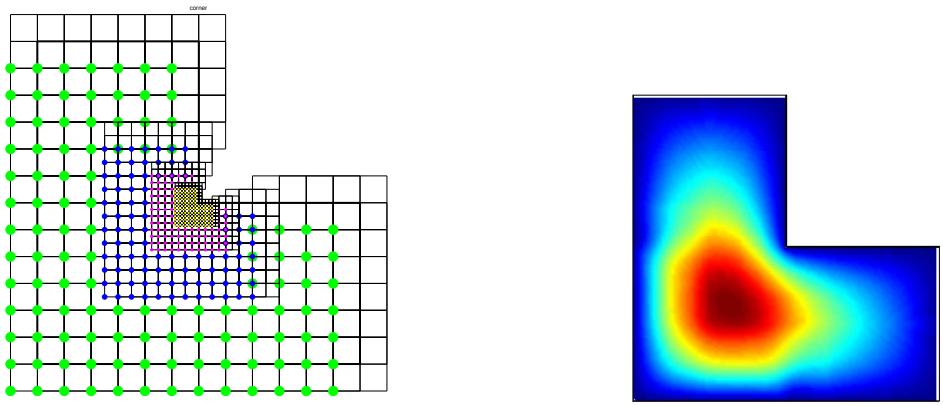

Figure 4.8: L-shaped domain and solution of Poisson's equation.

The figure 4.8 shows the relevant splines over the strongly nested domains near the reentrant corner $(1, 1)$ with grid widths $h^+ = 1/6$ and $h^- = 1/48$. The right figure shows the solution of the Poisson problem.

h	DOF(WEB)	H_0 error	CondG	DOF(WB)	CondG
1/2	12	-	7.6360e+00	32	1.1477e+04
1/4	48	0.0745	4.5163e+01	84	7.1285e+04
1/8	192	0.0270	1.4410e+02	260	3.1590e+05
1/16	768	0.0095	5.2558e+02	900	1.2810e+06
1/32	3072	0.0031	2.0291e+03	3332	5.1446e+06
1/64	12288	0.0010	7.9956e+03	12804	2.0604e+07

Table 4.1: Grid width h, Degree of freedom (DOF), H_0-error and condition number of Galerkin matrix for WEB and WB methods.

The first four columns of table 4.1 show the grid width h, degrees of freedom (DOF), H_0-error and the condition number of the Galerkin matrix obtained from the WEB method. The last two columns of the table show the DOF and the condition number of the Galerkin matrix obtained from the WB method. The rate of convergence, estimated from $\ell d(0.0745/0.0270)$, $\ell d(0.0270/0.0095)$, $\cdots$ is less than the optimal order 2 as is expexted in view of the singularity.

The table 4.2 shows the DOF and H_0-error of the Galerkin solution obtained from hierarchical approximation. Since, for strongly nested domains, the distance between the boundary ∂D_h and $\partial D_{h/2}$ is greater than $2h$, we fix this distance as $3h$. We approximate the solution up to six

Hierarchical levels	DOF(WHB)	H_0-error
1	133	-
2	231	0.028107
3	329	0.010267
4	427	0.003495
5	525	0.001161
6	623	0.000394

Table 4.2: H_0-error for adaptive method over strongly nested domain with $h^+ = 1/6$.

hierarchical levels. The resulting sequence of domain is

$$\mathbb{D} = D_{h^+} \supset D_{h^+/2} \supset \cdots \supset D_{h^-}.$$

where $h^+ = 1/6$ and $h^- = 1/192$. As is expected, the results are slightly better. For example, with 623 DOF an error of 0.000394 is achieved which is less than 0.0095 as obtained with 768 WEB basis functions.

Hierarchical levels	DOF(Adaptive Method)	H_0-error
1	133	-
2	187	0.013549
3	339	0.006276

Table 4.3: H_0-error for adaptive method with $h^+ = 1/6$ and $tol. = 10^{-2}$.

The last table 4.3 shows the DOF and the H_0-error obtained from the adaptive scheme discussed in section 4.5. Here we show only two refinement steps which indicate a slight improvement over a user-defined domain selection.

Example 4.6.2. Our second example is the domain consisting of a 3/4-th portion of the unit circle with center at $(0, 0)$. The exact solution is

$$u(x, y) = (1 - x^2 - y^2)(x + y - \sqrt{x^2 + y^2})$$

with $f = 8x + 8y - 9\sqrt{x^2 + y^2} + \frac{1}{\sqrt{x^2+y^2}}$.

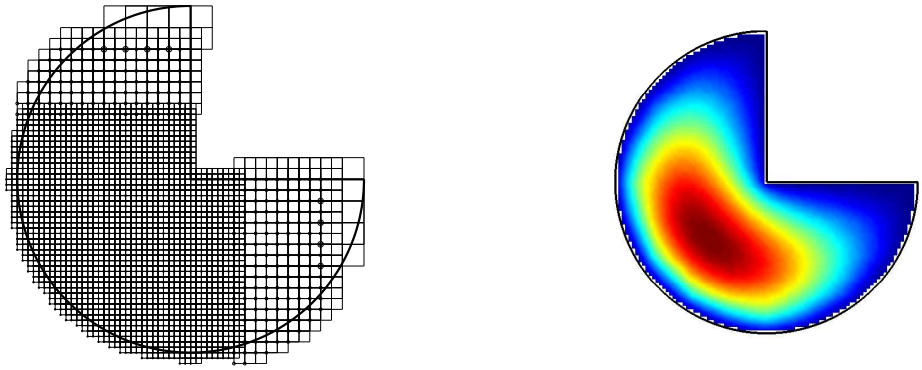

Figure 4.9: Domain with reentrant corner and solution of Poisson's equation.

Figure 4.9 shows the B-splines over domains of different hierarchical levels and the solution of the Poisson problem.

The first four columns of table 4.4 show the grid width h, degree of freedom (DOF), H_0-error and the condition number of the Galerkin matrix obtained from the WEB method. The last two columns of the table show the DOF and condition number of the Galerkin matrix obtained from the WB method.

h	DOF(WEB)	H_0 error	CondG	DOF(WB)	CondG
1/2	7	0.013086148	8.3984e+00	29	5.7269e+03
1/4	37	0.003289313	3.1660e+01	73	2.7415e+08
1/8	156	0.000616409	1.2855e+02	224	6.3688e+08
1/16	609	0.000145764	6.4816e+02	741	1.5162e+11
1/32	2421	0.000035975	3.3423e+03	2681	2.8203e+13
1/64	9638	0.000008925	1.1870e+04	10154	2.0407e+19

Table 4.4: Grid width h, Degree of freedom (DOF), H_0-error and condition number of the Galerkin matrix for WEB and WB methods.

The table 4.5 shows the data for solving the Poisson problem via the adaptive scheme for strongly nested domains with up to four hierarchical levels with refinement near $(0, 0)$. The error is calculated with respect to the exact solution of the Poisson problem.

Hierarchical levels	DOF(WHB)	H_0-error
1	136	0.0011424
2	264	0.0004612
3	403	0.0000822
4	555	0.0000248

Table 4.5: H_0-error for adaptive method over strongly nested domains with $h_0 = 1/6$.

Hierarchical levels	DOF(Adaptive Method)	H_0-error
1	136	0.0011424
2	427	0.0007591
3	1563	8.2702 e -6

Table 4.6: H_0-error for adaptive method with $h^+ = 1/6$ and $tol. = 10^-2$.

Table 4.6 shows the DOF and the H_0-error for the solution of the Poisson problem solved over the nested domain obtained by comparing the approximation u_h with the exact solution u. The nested domains have the grid widths $h^+ = 1/6$ and $h^- = 1/24$.

As for the previous example, the tables confirm a slightly better performance of our hierarchical approximations compared to the WEB method.

Chapter 5

Summary and Discussion

This chapter outlines the contents of the thesis which discussed the finite element approximation of an elliptic boundary value problem by using hierarchical weighted B-splines. It was often asked about the adaptivity of WEB-splines for boundary value problems. We developed a scheme of adaptivity for weighted linear B-splines.

In the second chapter, we gave a short introduction to the theory of the finite element method. As a model problem, we considered Poisson's equation with Dirichlet boundary conditions

$$\begin{aligned} -\Delta u &= f \quad \text{in} \quad D \\ u &= 0 \quad \text{on} \quad \partial D \end{aligned}$$

on a bounded domain $D \subset \mathbb{R}^2$.

Since, in general, there does not exist a classical solution of this boundary value problem, it is solved in a generalized function space, the Sobolev space, which is discussed in section 2.2. Its definition uses a weak derivative, a generalization of the classical derivative via Lebesgue integration by parts.

The variational formulation of our model problem is

$$\int_D \nabla u . \nabla v \; dD = \int_D f v \; dD \quad \forall v \in H_0^1(D).$$

The Lax-Milgram theorem guarantees the existence of a unique solution u of the variational problem

$$\begin{aligned} J(u) &= min_{v \in V} J(v), \\ J(v) &= \frac{1}{2} a(v,v) - F(v) \end{aligned}$$

and hence of our model problem.

For the computation of an approximate solution u_h, the Ritz-Galerkin method employs a finite subspace V_h of $H_0^1(D)$ using basis functions b_k with the local support. Then u_h has a unique representation of the form $u_h = \sum_{k=1}^{N_h} c_k b_k$, where $N_h = dim(V_h)$. Substituting u_h in variational formulation and restricting v to $\{b_1, \cdots, b_{N_h}\}$, one obtains the Galerkin system

$$G\mathbf{u} = \mathbf{f}$$

where the entries of G are given by (4.16). Lemma 2.4.1 shows the positive definiteness of the stiffness matrix G. According to Cea's lemma 2.4.2, the error of the Ritz-Galerkin solution u_h corresponds to the error of the best approximation of u in V_h.

B-splines play an important role. We use B-splines as finite element basis functions. Two major problems arise when using B-splines as basis functions in finite element approximations on arbitrary domains. One is the fulfillment of homogeneous Dirichlet boundary conditions and the other is the stability problem.

We can remove these difficulties by simple modifications. The homogeneous Dirichlet boundary condition causes, in general, all the coefficients near the boundary to be zero. As a result, the approximation power is lost. This deficit can easily be removed by multiplying the B-splines with a sufficiently smooth weight function which vanishes on the boundary and is positive inside D. For domains with smooth boundary ∂D, the distance function can be used as a weight function. An alternative approach for generating weight functions is the R-function method. This technique is useful for the domains designed by CSG techniques. In this case, the R-function method supplies set operations to construct a global weight function from the boundary segments.

Weighting the B-splines leads us to the weighted B-spline basis (WB-basis), which incorporates essential boundary conditions to the finite element basis. The Bramble-Hilbert's lemma helps us to find the error bound for $u \in H^n(D)$ using the orthogonal projection $L^n u$ onto polynomial $\mathbb{P}_{n-1}(D)$ of total degree $< n$ on D. The error bound for the Sobolev norm on $H^\ell(D)$ for $0 \leq \ell < k \leq n$ is

$$\|u - L^n u\|_{H^\ell(D)} \leq const(D, n)\varrho^{k-\ell}\|u\|_{H^k(D)} \quad \text{for } 0 \leq \ell < k \leq n,$$

where ϱ is the diameter of the domain D. For the spline function $u_h = \sum_{k\in K} c_k b_{k,h}$, the error bound for $u \in H^2(D)$ is

$$\|u - \sum_{k\in K} c_k b_{k,h}\|_{H^\ell(D)} \preceq h^{2-\ell}\|u\|_{H^2(D)} \quad for \;\; 0 \le \ell < 2.$$

In section 3.5, we explain the adaptive refinement with hierarchical bases. For the refinement step some of the B-splines on the coarser grid are subdivided. The subdivision is accomplished by the formula

$$b_{k,h}(x) = \sum_{l=0}^{2} s_l b_{2k+l,h/2}(x)$$

where

$$s_l = \prod_{\nu=1}^{2} \frac{1}{4} \begin{pmatrix} 2 \\ l_\nu \end{pmatrix}.$$

If by U_h we denote the coefficients of a spline function $u_h = \sum_k u_{k,h} b_{l,h}$ on the coarser grid, then u_h is represented on the fine grid by the representation (5.6).

The coefficients on the fine grid and on the coarse grid are related by

$$U_{h/2} = PU_h, \quad p_{k,l} = s_{k-2l}.$$

In adaptive finite element computations, we adjust the spatial resolution of the domain discretization to obtain high accuracy on some subregions of D. To fulfill this requirement we insert a new set of basis functions having a smaller grid width than the coarser basis functions.

We define hierarchical splines in definition 3.5.1 corresponding to a nested sequence of domains

$$\mathbb{D} : D_{h^+} \supset D_{h^+/2} \supset ... \supset D_{h^-}.$$

We replace the relevant B-splines $b_{k,h}$ with $k \in K_h$ by the finer B-splines via subdivision. The spanning set is then

$$b_{k,h}, \quad k \in K_h, \quad h \in \{h^+h^+/2, \cdots, h^-\}.$$

The hierarchical B-splines which span $\mathbb{B}_{(h^+,h^-)}(\mathbb{D})$ are linearly independent. We prove the linear independence in theorem 3.5.3.

To keep the basis uniformly conditioned, we impose some restrictions on the nested sequence of domains. We introduced strongly nested domains in definition 3.6.1.

Strongly nested domains obey the following properties:

- B-splines in the basis with non-empty intersection of their supports must belong to two consecutive hierarchical levels.
- The distance between the boundary ∂D_h and $\partial D_{h/2}$ is greater than $2h$.

The error of a hierarchical approximation p to a smooth bivariate function f is bounded by

$$|f(x) - p(x)| \leq \text{const } h^2 \sum_{\nu=1}^{2} \|\partial_\nu^2 \widetilde{f}\|_{\infty, D_{x,h}}, \quad x \in D_h \backslash D_{h/2}.$$

A proof of this statement is given in theorem 3.6.1.

Chapter 4 is devoted to the numerical implementation of an adaptive scheme. We used MATLAB as a software package. In section 4.1 we explain the data structure to store the information obtained from the adaptive grid discretization and the resulting spline space. The grid data consists of an array of structures $HB(\ell)$, where ℓ denotes the discretization level. To assemble the Ritz-Galerkin system we make a list of all relevant B-splines in the form of triplets (k, ν, μ) containing the level k and the index (ν, μ) of the B-spline. The entries of the Galerkin system

$$g_{k,l} = \int_D \operatorname{grad} b_k \operatorname{grad} b_l$$

are computed with the help of numerical integration. The techniques for numerical integration over grid cells is explained in section 4.3. The Galerkin matrix is obtained by adding the contributions from each grid cell. We conclude by describing our refinement strategy and some numerical examples.

5.1 Possible generalizations

An adaptive refinement can be generalized in different ways. The possible generalization concerns the selection of the finite element basis function and the refinement strategy.

As far as the finite element basis is concerned, splines of arbitrary degree can be used as finite element basis functions. The basis functions may be defined on three or higher dimensional domains. The discretization and the adaptivity for such domains are difficult to program.

One can classify the inner and outer B-splines for different hierarchical levels and construct the WEB splines as finite element basis functions. WEB splines also permit an adaptive refinement strategy. Error estimates for our refinement strategy are also an open problem.

As far as the refinement strategy is concerned, the assumption of a "strongly nested domain" can be removed and one can form a strategy that deals with the general sequence of domains. The stability of the hierarchical basis is also included in our future work. Moreover, the Galerkin system constructed over a nested sequence of domains can be solved with multigrid solvers.

Appendix A

Appendix A

Definition A.0.1. (Quasi-Interpolant).

A quasi-interpolant $\mathcal{Q} : L^2(\mathbb{R}^2) \to \mathbb{B}_h(\mathbb{R}^2)$ with uniform B-splines $b_{k,h}$ of coordinate degree n on the grid $h\mathbb{Z}^2$ has the form

$$\mathcal{Q}f = \sum_{k\in\mathbb{Z}^2} (\mathcal{Q}_k f) b_{k,h}$$

where $\mathcal{Q}_k : L^2(\operatorname{supp} b_{k,h}) \to \mathbb{R}$ are continuous linear functionals. Moreover,

- $|\mathcal{Q}_k f| \leq \|\mathcal{Q}\| \ \|f\|_{L^2(\operatorname{supp} b_{k,h})} \quad \forall\ k \in \mathbb{Z}^2$
- $\mathcal{Q}f = f \quad \forall f \in \mathbb{P}_n(\mathbb{R}^2)$.

The uniform boundedness and the reproduction of polynomials in particular imply that $\mathcal{Q}$ has the optimal approximation order.

Definition A.0.2. (Extension Operator). Let D be a Lipschitz domain in $\mathbb{R}^m$. For $\ell \in \mathbb{N}$, a linear operator $\mathcal{E}$ mapping $H^\ell(D)$ into $H^\ell(\mathbb{R}^m)$ is called an extension operator for D if there exists a constant K such that for every $u \in H^\ell(D)$ the following holds:

- $\mathcal{E}u(x) = u(x)$ for $x \in D$
- $\|\mathcal{E}u\|_{H^\ell(\mathbb{R}^m)} \leq K\|u\|_{H^\ell(D)}$.

An extension operator $\mathcal{E}$, valid for any ℓ, was constructed by Stein ([Ste73]), improving the earlier result by Calderón ([Cal61]) where the extensions were not independent of the order of the differentiability.

References

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. The design and analysis of computer algorithms. *Addison Wesley, Reading, Mass*, 1974.

[ANW67] J. H. Ahlberg, E. N. Nielson, and J. L. Walsh. The theory of splines and their applications. *Academic Press, New York*, 1967.

[App01] C. Apprich. Der Penalty Ansatz bei Spline Galerkin Approximation des Poisson-Problems, Diplomarbeit. *Mathematisches Institut A, Universität Stuttgart*, 2001.

[Arg60] H. J. Argyris. Energy theorems and structure analysis. *Butterworth, London*, 1960.

[Arg64] H. J. Argyris. Recent Advances in Matrix Methods of Structural Analysis. *Pergamon Press, Elmsford, NY.*, 1964.

[Bab73a] I. M. Babuska. The Finite Element Method with Lagrangian Multipliers. *Numerische Mathematik*, 20:179–192, 1973.

[Bab73b] I. M. Babuska. The Finite Element Method with Penalty. *Mathematics of Computation*, 27(122):221–228, 1973.

[Ban87] R. E. Bank. The Hierarchical Basis Multigrid Method. *Konrad-Zuse-Zentrum für Informationstechnik*, Apr. 1987.

[Ber13] S. Bernstein. Definition du theoreme de Weierstrass, fondee sur le calcul des probabilites. *Comm. Soc. Math. Kharkaw*, 13:1–2, 1912-1913.

[BG04] H. J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica, Cambridge University Press*, 2004.

[BH70] J. H. Bramble and S. R. Hilbert. Estimation of Linear Functional on Sobolev Spaces with Application to Fourier Transforms and Spline Interpolation. *SIAM J. Numer. Anal.*, 7(1):112–124, 1970.

[BR78] I. Babuska and W. C. Rheinboldt. Error Estimates for Adaptive Finite Element Computations. *SIAM J. Numer. Anal.*, 15(4), Aug. 1978.

[Bun92] H. J. Bungartz. Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. *Ph.D. thesis, Institut für Mathematik, Technische Universität München*, 1992.

[Cal61] A. P. Calderon. Lebesgue spaces of differentiable functions and distributions. *Proc. Symp. in Pure Math.*, 4:33–49, 1961.

[CE07] R. W. Clough and L. W. Edward. Early finite element research at Berkeley, Retrieved on 2007-10-25. 2007.

[CK01] Z. Cai and S. Kim. A Finite Element Method using Singular Functions for the Poisson Equation: Corner Singularities. *SIAM J. Numer. Anal.*, 39(1):286–299, 2001.

[Cou43] R. L. Courant. Variational methods for the solution of problems of equilibrium and variation. *Bulletin of the Amer. Math. Soc.*, 49:1–23, 1943.

[dB72] C. de Boor. On calculating with B-Splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[dB78] C. de Boor. A Practical Guide to Splines. *Springer-Verlag*, 1978.

[DW00] W. Dörfler and O. Wilderotter. An Adaptive Finite Element Method for Linear Elliptic Equations with Variable Coefficients. *ZAMM. Z. Angew. Math. Mech.*, 80(7), 2000.

[EJ91] K. Erikssoin and C. Johnson. Adaptive Finite Element Methods for Parabolic problems I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1), Feb. 1991.

[FB88] D. R. Forsey and R. H. Bartels. Hierarchical B-spline Refinement. *Computer Graphics*, 22(4):205–212, August 1988.

[FB95] D. R. Forsey and R. H. Bartels. Surface fitting with hierarchical splines. *ACM Trans. on Graphics*, 14:134–161, 1995.

[FW98] D. R. Forsey and D. Wong. Multi-resolution surface reconstruction for hierarchical B-splines. *Graphic interface*, 1998.

[GM03] M. Griebel and A. S. Marc. Mesh free Methods for Partial Differential Equations. *No. 26 in lecture notes in Computational Science and Engineering, Springer*, 2003.

[Gri98] M. Griebel. *Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences*. Springer-Verlag, 1998.

[HÖ3] K. Höllig. Finite Element Methods with B-splines. *SIAM, Philadelphia*, 2003.

[Her01] E. Herbert. Geometry and Topology for Mesh Generation. *No. 7 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press*, 2001.

[HHP] K. Höllig, J. Hörner, and M. Pfeil. Parallel Finite Element Methods with Weighted Linear B-Splines. *Universität Stuttgart*.

[Hre41] A. Hrennikoff. Solution of Problems of Elasticity by the Frame-Work Method. *ASME J. Appl. Mech.*, 8:A619–A715, 1941.

[HRW] K. Höllig, U. Reif, and J. Wipper. Weighted extended b-spline approximation of Dirichlet problems. *SIAM J. Numer. Anal.*, 39(2):442–462.

[HRW01] K. Höllig, U. Reif, and J. Wipper. Error estimates for the web-method. *Vanderbilt University Press, Nashville, TN*, pages 195–209, 2001.

[Kip98] A. M. Kipp. Spline-Galerkin-Approximation elliptischer Randwertprobleme. *Dissertation, Universität Stuttgart*, 1998.

[KJ96] K. G. Kajal and L. M. John. A Brief History of the Beginning of the Finite Element Method. *International Journal for Numerical Methods in Engineering*, 39:3761–3774, 1996.

[KK56] L. W. Kantorowitsch and W. I. Krylow. Näherungsmethoden der Höheren Analysis. *VEB Deutscher Verlag der Wissenschaften, Berlin*, 1956.

[KKD06] V. V. K. S. Kumar, B. V. R. Kumar, and P. C. Das. Weighted Extended B-spline methods for the approximation of the stationary Stokes problem. *J. Comp. and Appl. Math.*, 186:335–348, 2006.

[Kra] R. Kraft. Hierarchical B-Splines. *Mathematisches Institut A, Universität Stuttgart*.

[Kra98] R. Kraft. Adaptive und Linear Unabhängige Multilevel B-splines und ihre Anwendungen. *Dissertation, Universität Stuttgart*, 1998.

[Kry02] P. Krysl. Natural Hierarchical Refinement for Finite Element Methods. *International Journal for Numerical Method*, 2002.

[Lam93] J. Laminie. Implementation of finite element non-linear Galerkin methods using hierarchical basis. *Comp. Mech.*, 11, 1993.

[Mal99] S. G. Mallat. A wavelet tour of signal processing. *Academic Press*, 1999.

[Rei95] U. Reif. A unified approach to subdivision algorithms near extraordinary vertices. *CAGD*, 12:153–174, 1995.

[Rei96] U. Reif. On constructing topologically unrestricted B-splines. *ZAMM 76*, (Suppl. 1):73–74, 1996.

[Rei97a] U. Reif. A refinable space of smooth spline surfaces of arbitrary topological genus. *Journal of Approximation Theory*, 90(2):174–199, 1997.

[Rei97b] U. Reif. Orthogonality of cardinal B-Splines in weighted Sobolev spaces. *SIAM J. Math. Anal.*, 28(5):1258–1263, 1997.

[Rei97c] U. Reif. Uniform B-spline approximation in Sobolev spaces. *Numerical Algorithms*, 15:1–14, 1997.

[Rei98] U. Reif. *Orthogonality Relations for Cardinal B-Splines over Bounded Intervals*. Springer-Verlag, 1998.

[Ron00] H. Ronald. Numerische Lösung des Poisson-Problems im 2-dimensionalen Raum mit Lagrange-Multiplikatoren, Diplomarbeit. *Mathematisches Institut A, Universität Stuttgart*, 2000.

[RS95] V. L. Rvachev and T. I. Sheiko. R-functions in boundary value problems in mechanics. *Appl. Mech. Rev.*, 48:151–188, 1995.

[RSST00] V. L. Rvachev, T. I. Sheiko, V. Shapiro, and I. Tsukanov. On completeness of RFM solution structures. *Comp. Mech.*, 25:305–316, 2000.

[Sch46] I. J. Schoenberg. Contribution to the Problem of Approximation of Equidistant Data by Analytic Functions. *Quarterly of Applied Mathematics*, 4:45–99 , 112–141, 1946.

[Sob38] S. L. Sobolev. On a theorem of functional analysis. *Transl. Amer. Math. Soc.*, 34(2):471–497, 1938.

[Ste73] E. M. Stein. Singular Integrals and Differentiability Properties of Functions. *Princeton University Press, Princeton, NJ*, 1973.

[Str73] G. Strang. An Analysis of the Finite Element Method. *Prentice-Hall*, 1973.

[TCMT56] M. J. Turner, R. W. Clough, H. C. Martin, and L. C. Topp. Stiffness and deflection analysis of complex structures. *Aeronaut. Sci.*, 23:805–823, 854, 1956.

[VSVT00] L. R. Vladimir, T. I. Sheiko, S. Vadim, and I. G. Tsukanov. On completeness of RFM solution structures. *Comp. Mech.*, 25:305–316, 2000.

[Wel95] F. Weller. B-spline surfaces with knot segments. *Interner Bereicht Nr. 249/94. Fachbereich Informatik, Universität Kaiserslautern*, 1995.

[Wer80] C. R. Werner. On a Data Structure for Adaptive Finite Element Mesh Refinement. *ACM Transactions on Mathematical software*, 6(2), June 1980.

[Wip05] J. Wipper. Finite-Element-Approximation mit WEB-Splines. *Dissertation, Universität Stuttgart*, 2005.

[YH00] A. Yvart and S. Hahmann. Hierarchical Triangular Splines . *ACM Transactions on Graphics*, V(N), 2000.

[Yse85] H. Yserentant. Hierarchical Basis for Finite Element Spaces in the Discretization of Non-symmetric Elliptic Boundary Value Problems. *Springer-Verlag*, 1985.

[Zen90] C. Zenger. Sparse Grids. *Technical Report SFB-Bericht Nr. 342/18/90 A, Technische Universität München*, 1990.