

Finite Element Approximation with Hierarchical B-Splines

Christian Apprich¹, Klaus Höllig¹ (✉), Jörg Hörner¹, Andreas Keller²,
and Esfandiar Nava Yazdani¹

¹ IMNG, Fachbereich Mathematik, Universität Stuttgart, Stuttgart, Germany
{`apprich,hoellig,hoerner,navayaz`}@`mathematik.uni-stuttgart.de`
² Allianz AG, Stuttgart, Germany
`andkell@web.de`

Abstract. We review the definition of hierarchical spline spaces and their application to finite element methods. Then we discuss how hierarchical techniques can be implemented using the FEMB program package. Subdivision algorithms play a crucial role and lead to a very simple program structure. A numerical example illustrates the substantial gains in accuracy for the adaptive strategy, in particular for higher degree B-splines.

Keywords: Finite elements · B-splines · Hierarchical bases · Subdivision

1 Introduction

B-splines have become standard tools in approximation, computer aided design, and graphics. A systematic application to finite element analysis is fairly recent. Essentially two different strategies have been proposed: weighted [1] and isogeometric [2] methods. The text books [3, 4] give a comprehensive description of the relevant theory for these novel approaches. Which technique is best suited for a particular problem depends to some extent on the representation and topological form of the simulation domain D . Isogeometric methods use parametrizations of subsets of D over rectangles and cuboids which are often provided by NURBS models for CAD/CAM applications. Weighted methods can handle domains well which have a natural implicit description, e.g., domains constructed from elementary sets with Boolean operations. There are also problems for which a combination of both methods might be appropriate [5].

Using B-splines as finite elements bridges the gap between geometry description and numerical simulation. Compared to conventional finite element methods on unstructured grids, spline-based techniques have several advantages: arbitrary choice of degree and smoothness, exact representation of boundary conditions, and simple data structure (one parameter per grid point). As a consequence, B-splines often yield significantly better results than classical finite element schemes when highly accurate solutions are sought.

Tensor product B-splines span the simplest multivariate spline spaces and can utilize univariate algorithms very efficiently. However, unlike in one variable, knot insertion does not provide truly local flexibility; adding knots has a global effect. The natural remedy is hierarchical refinement. Using nested grids with uniform B-splines of different grid width combines well with adaptive approximation methods.

Given the crucial importance of hierarchical techniques for tensor product spline spaces, it is not surprising that such methods have been subject of intensive research. Building upon the classical articles [6, 7], hierarchical splines have been analyzed by a number of authors; the articles [8–11] serve as a few examples which are most relevant to our application. Moreover, several novel concepts, such as T-splines [12] and splines over box-partitions [13], have been introduced.

In this article we consider hierarchical finite element approximation with B-splines, a topic of key importance if B-splines are to compete successfully with classical mesh-based trial functions. For weighted methods, hierarchical B-spline elements were already described in [3], but first implemented in [14] for a special case. For isogeometric elements, adaptive B-spline approximations were recently studied in [15–17] and in the context of T-splines in [18–22]. Our objective is not to improve upon the by now well established theory. Instead, we describe how hierarchical methods can be easily implemented using the FEMB program package [23]. Clearly, incorporating existing software for uniform grids eliminates a great deal of redundant programming effort. As an illustration of our algorithms, we document the substantial gains in accuracy for a typical model problem with a singular solution.

After briefly reviewing some essential components of the FEMB routines in Sect. 2, we define hierarchical splines in Sect. 3. We follow the description in [24, 25] where a convenient notation has been suggested, covering many of the concepts introduced so far. Section 4 discusses grid transfer operations based on B-spline subdivision. Then we explain in Sect. 5 how to combine these tools with our programs for assembling and solving Ritz-Galerkin systems for uniform grids. Finally, Sect. 6 illustrates the error behavior of hierarchical B-spline elements at a reentrant corner.

2 FEMB Program Package

The FEMB programs implement finite element algorithms with uniform B-splines for two elliptic model problems: the scalar second order equation

$$-\operatorname{div}(p \operatorname{grad} u) + qu = f, \quad (1)$$

where $p > 0$ and $q \geq 0$ are variable coefficients, and the Lamé-Navier system in linear elasticity

$$-\operatorname{div} \sigma(u) = f, \quad (2)$$

where σ is the stress tensor. For each problem the main steps of a finite element simulation are

- determination of integration parameters,
- assembly of the Ritz-Galerkin system,
- conjugate gradient iteration,
- visualization and computation of the residual.

We briefly discuss these steps for a simple special case of problem (1), a Helmholtz equation with homogeneous Neumann boundary conditions:

$$-\Delta u + u = f \text{ in } D, \quad \partial_{\perp} u = 0 \text{ on } \partial D, \quad (3)$$

where $D \subset \mathbb{R}^d$ is a bounded domain in two or three dimensions ($d = 2, 3$), and ∂_{\perp} denotes the normal derivative. This example will be used for illustration purposes throughout the article.

The finite element discretizations of the FEMB programs are based on d -variate uniform tensor product B-splines $b_{k,\xi}$ of a fixed coordinate degree n ; see Fig. 1. The index $k = (k_1, \dots, k_d)$ refers to the lower left grid position, i.e., $b_{k,\xi}$ has the knots $\xi_{\nu,k_{\nu}}, \dots, \xi_{\nu,k_{\nu}+n+1}$ in the ν -th coordinate direction and support

$$(\xi_{1,k_1}, \dots, \xi_{d,k_d}) + [0, n+1]^d h,$$

where $h = \xi_{\nu,\ell+1} - \xi_{\nu,\ell}$ is the grid width.

The domain is described in implicit form,

$$D : w(x) > 0,$$

with a weight function w supplied by the user (cf. [3] and [26] for some principal construction techniques). This description allows to incorporate the geometry information into a B-spline discretization in very simple form, combining the efficiency of regular grids with the flexibility of free-form boundary representations.

Definition 1 (Uniform Splines). *A uniform spline on a domain D is a linear combination*

$$p^{\xi} = \sum_{k \sim \xi} u_k^{\xi} b_{k,\xi},$$

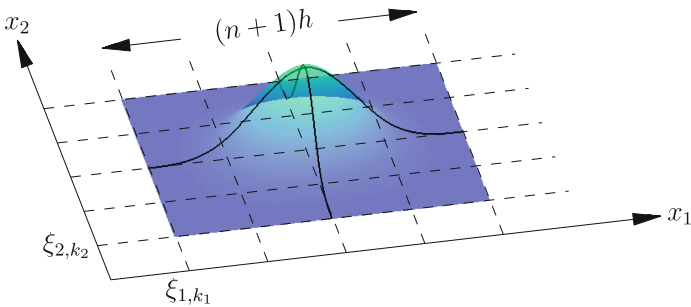


Fig. 1. Uniform tensor product B-spline of coordinate degree $n = 3$ (Color figure online)

where ξ is a d -variate uniform knot sequence,

$$\xi_\nu : \xi_{\nu,0}, \dots, \xi_{\nu,m_\nu+n}, \quad \xi_{\nu,\ell+1} = \xi_{\nu,\ell} + h, \quad 1 \leq \nu \leq d,$$

with parameter hyperrectangle

$$D_\xi^n = [\xi_{1,n}, \xi_{1,m_1}] \times \dots \times [\xi_{d,n}, \xi_{d,m_d}] \supset D.$$

The notation $(k_1, \dots, k_d) \sim \xi$ indicates that the sum is taken over all B-splines corresponding to ξ , i.e., $k_\nu = 0, \dots, m_\nu - 1$, with the convention that the coefficients u_k^ξ of B-splines with no support in D (irrelevant B-splines) are set to zero.

In Fig. 2, the B-splines $b_{k,\xi}$ corresponding to the knot sequence ξ are marked at the center of their support, using dots for relevant and circles for irrelevant B-splines. Keeping irrelevant B-splines, as described in the above definition, avoids index lists which are necessary for approximations on unstructured grids.

While splines are adequate finite elements for natural boundary conditions, essential or mixed boundary conditions must be incorporated into the finite element subspace. In the FEMB package this is done by using weighted B-splines $w b_{k,\xi}$. Since the hierarchical techniques in this more general case (standard splines correspond to the trivial choice $w = 1$) are completely analogous, we have chosen to focus on the model problem (3), which allows us to explain the algorithms in the simplest setting without any modification of the spline space.

The coefficients u^ξ of a finite element approximation p^ξ satisfy the Ritz-Galerkin system

$$\sum_{j \sim \xi} a(b_{k,\xi}, b_{j,\xi}) u_j^\xi = f_k^\xi, \quad k \sim \xi,$$

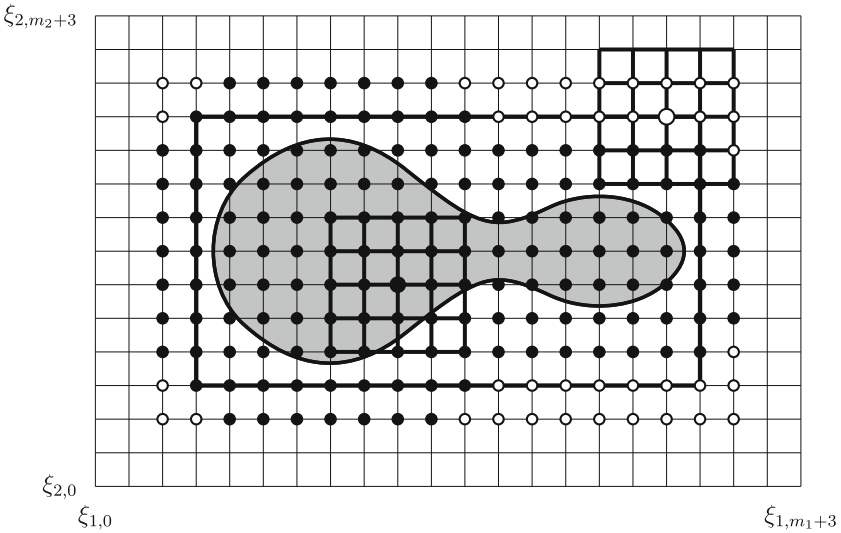


Fig. 2. Relevant and irrelevant cubic B-splines for a bounded domain

where

$$a(\varphi, \psi) = \int_D (\text{grad } \varphi \text{ grad } \psi + \varphi \psi), \quad f_k^\xi = \int_D f b_{k,\xi}$$

for Helmholtz equation (3). We abbreviate this system by

$$G_\xi u^\xi = f^\xi.$$

The matrix G_ξ and the vector f^ξ are assembled by adding the contributions to the integrals from each grid cell using predetermined Gauß parameters (cf. [3, 25] for details). Then the coefficients u^ξ are computed with a preconditioned conjugate gradient iteration. An important feature of the final FEMB evaluation and visualization routine is the return of the residual. Unlike for conventional finite element solvers, the spline approximation is continuously differentiable for degree $n > 1$ and can be substituted into the partial differential equation. This provides a very reliable error measure.

In the FEMB package it is assumed that $D \subseteq (0, 1)^d$ and, accordingly, $\xi_{\nu,n} = 0$, $\xi_{\nu,m_\nu} = 1$, $h = 1/(m_\nu - n)$. Changing the relevant program codes slightly, we can relax this requirement. We can provide a MATLAB¹ function which assembles G_ξ and f^ξ for arbitrary uniform knot sequences ξ . In particular, the parameter hyperrectangle D_ξ^n need not contain the simulation region D . This subroutine will be the essential tool for implementing a hierarchical finite element solver.

3 Hierarchical Splines

Roughly speaking, a hierarchical spline is a sum of uniform splines p^ξ with different knot sequences ξ . Typically, such approximations are constructed with an adaptive process. In regions with large error, B-splines $b_{k,\xi}$ are subdivided by halving the grid width and replaced by the resulting B-splines on the finer grid. These refinement steps are repeated until a prescribed tolerance for the error is met. Figure 3 shows an example for bilinear B-splines ($n = 1$).

A knot sequence ξ with grid width $h = 1/4$ and parameter rectangle $D_\xi^1 = [0, 1]^2$ (thick grid lines), covering a heart-shaped domain, is refined near the two corners of the domain boundaries. The knot sequence η near the reentrant corner is further refined leading to a knot sequence ζ with grid width $h = 1/16$. The bilinear B-splines belonging to the hierarchical basis are marked with circles at the center of their support. As mentioned before, B-splines which can be represented on finer grids do not belong to the hierarchical basis as well as B-splines with no support in the domain D . Nevertheless, such B-splines are included with zero coefficients in linear combinations to facilitate programming.

The adaptive construction, leading to a hierarchical approximation, corresponds to a tree structure of knot sequences. To make this more precise, some notation is helpful. We define the extent of a d -variate knot sequence as the smallest rectangle containing all break points:

$$[\xi] = [\xi_{1,0}, \xi_{1,m_1+n}] \times \cdots \times [\xi_{d,0}, \xi_{d,m_d+n}].$$

¹ MATLAB[®] is a registered trademark of The MathWorks, Inc., Natick, MA, U.S.A.

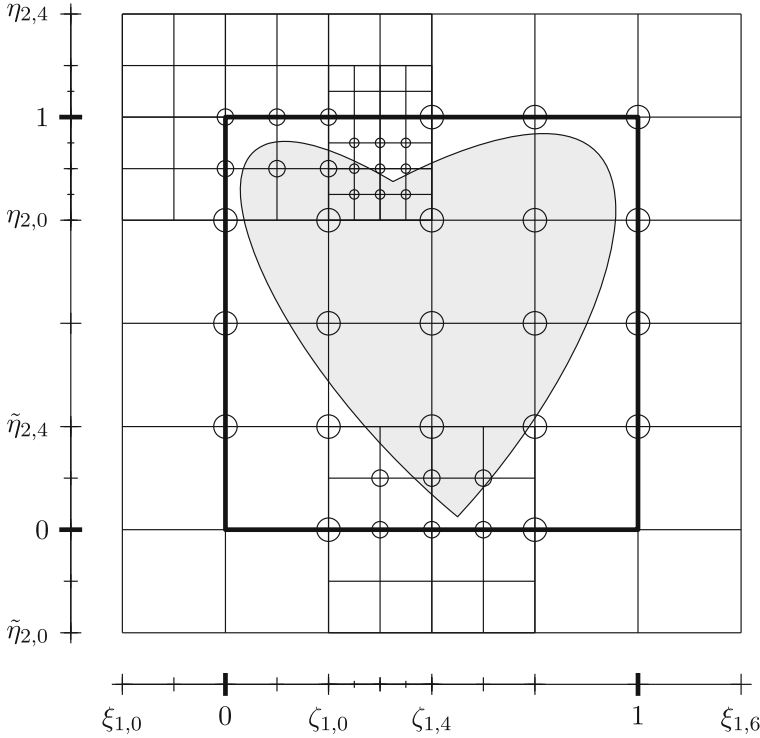


Fig. 3. Grid refinement for bilinear hierarchical B-splines

Moreover, we say that η is the refinement of ξ if η_ν , $\nu = 1, \dots, d$, is obtained from ξ_ν by adding midpoints in all knot intervals. Finally, we speak of a local refinement, if the knot sequences η_ν are refinements of subsequences of consecutive knots of ξ_ν .

Definition 2 (Hierarchical Spline). A hierarchical spline p^Ξ is a sum of uniform splines p^ξ corresponding to knot sequences ξ which are nodes of a tree Ξ :

$$p^\Xi = \sum_{\xi \sim \Xi} p^\xi, \quad p^\xi = \sum_{k \sim \xi} u_k^\xi b_{k,\xi}.$$

It is required that the children of each node ξ are local refinements of ξ with disjoint extents. Moreover, coefficients u_k^ξ of irrelevant B-splines $b_{k,\xi}$ or B-splines which can be represented on refined grids are set to zero.

In MATLAB, the relevant data for a hierarchical spline p^Ξ can be stored in a cell vector `HS` of structures, corresponding to the nodes ξ of the tree Ξ , with fields

```
.parent, .children, .h, .knots, .u.
```

The first two fields refer to the cell vector indices of the parent and children nodes ξ , `.h` is the grid width, the field `.knots` specifies the knot sequence using the lower left and upper right coordinates of $[\xi]$ divided by the grid width,

$$[\xi_{1,0} \xi_{1,m_1+n}; \dots; \xi_{d,0} \xi_{d,m_d+n}]/h,$$

and `.u` is the d -dimensional array of coefficients with size

$$m_1 \times \dots \times m_d.$$

This natural storage model is illustrated for the example in Fig. 3 above:

```
HS{1}.parent = []; HS{1}.children = [2 3]; HS{1}.knots = [-1 5;-1 5];
HS{1}.h=1/4; HS{1}.u = [0 *** 0;*** 0;0 ****;*****;0 ****];
HS{2}.parent = 1; HS{2}.children = [4]; HS{2}.knots = [-2 4;6 10];
HS{2}.h = 1/8; HS{2}.u = [0 0 0;*** 0;*** 0;0 0 0];
HS{3}.parent = 1; HS{3}.children = []; HS{3}.knots = [2 6;-2 2];
HS{3}.h = 1/8; HS{3}.u = [0 **;0 **;0 **];
HS{4}.parent = 2; HS{4}.children = []; HS{4}.knots = [4 8;12 18];
HS{4}.h = 1/16; HS{4}.u = [*** 0 0;*** 0 0;*** 0 0];
```

We note the forced zeros in the coefficient arrays due to irrelevant or refined B-splines. For example, the B-spline $b_{(4,0),\xi}$ with center at $x = (1, 0)$ has support outside of D and hence $\text{HS}\{1\}.u(5, 1) = 0$. The B-spline $b_{(4,0),\eta}$ with center at $x = (3/8, 7/8)$ is a linear combination of the B-splines $b_{k,\zeta}$, $0 \leq k_1, k_2 \leq 2$. Consequently, $\text{HS}\{2\}.u(5, 1) = 0$. Note that MATLAB indexing starts at 1 while 0 is more convenient for the theory. Moreover, columns in the matrices $\text{HS}\{i\}.u$ correspond to horizontal directions in the figure (analogously as for the commands `ndgrid` and `meshgrid` in MATLAB).

In addition to the assumptions on the tree Ξ made in Definition 2, the following property, already proposed in [14], simplifies programming considerably and also has theoretical relevance for stability and the construction of quasi-interpolants.

Definition 3 (Nested Tree). *A tree Ξ is nested if the grid widths of any two B-splines in the hierarchical basis, which are both nonzero at some common point x , are equal or differ by a factor two.*

Requiring a tree to be nested is not a severe restriction. The knot sequences of a tree generated via an adaptive procedure can always be refined to meet the additional requirement. We illustrate this for the example in Fig. 3. Clearly, the tree formed by the knot sequences ξ , η , $\tilde{\eta}$, ζ is not nested since the B-splines $b_{(1,3),\xi}$ (grid width $1/4$, centered at $(1/4, 3/4)$) and $b_{(0,0),\zeta}$ (grid width $1/16$, centered at $(5/16, 13/16)$) are both nonzero for all x in $(2/8, 3/8) \times (6/8, 7/8)$. To conform to the requirement of Definition 3, we add a layer of B-splines with grid width $1/8$ to separate the coarse- and fine-grid B-splines. We replace the knot sequence η by η^* with knots

$$\eta_{1,0}^* = -2/8, \dots, 6/8 = \eta_{1,8}^*, \quad \eta_{2,0}^* = 4/8, \dots, 10/8 = \eta_{2,6}^*.$$

All B-splines $b_{k,\xi}$ which overlap B-splines $b_{k,\zeta}$ from the finest grid are now replaced by linear combinations of B-splines from η^* with grid width $1/8$. This leads to the following changes in the MATLAB structure listed above.

```
HS{1}.u = [0 ** 0 0;*** 0 0;0 ** 0 0;*****;0 *****];
HS{2}.knots = [-2 6;4 10];
HS{2}.u = [0 0 0 0 0;*** * 0;*** * 0;*** * 0;*** 0 0 0;*** * 0;*** * 0];
```

The other structure variables remain unchanged.

4 Grid Transfer Operations

For approximation with hierarchical splines grid transfer operations are essential. Subdivision strategies, systematically introduced by Boehm [27] and Cohen, Lyche, and Riesenfeld [28], provide the canonical mechanism.

Theorem 1 (Subdivision of B-Splines). *A d -variate B-spline $b_{k,\xi}$ with grid width h can be represented as linear combination of B-splines $b_{k,\eta}$ with grid width $h/2$ ($\xi_k = \eta_{2k}$):*

$$b_{k,\xi} = \sum_{\nu_1=0}^{n+1} \cdots \sum_{\nu_d=0}^{n+1} s_\nu b_{2k+\nu,\eta}, \quad s_\nu = 2^{-nd} \binom{n+1}{\nu_1} \cdots \binom{n+1}{\nu_d}; \quad (4)$$

see Fig. 4. When manipulating sums it is often convenient to extend the range of summation to all integer vectors $\nu = (\nu_1, \dots, \nu_d)$. This is possible in view of the convention that the binomial coefficient $\binom{n+1}{\alpha}$ vanishes if $\alpha \notin \{0, \dots, n+1\}$.

By linearity, the subdivision rule extends to splines as well. It can be used for grid transfer in both directions. We consider each case in turn.

Corollary 1 (Coarse-to-Fine Extension). *If the knot sequence η is the refinement of ξ and $\sum_{k \sim \xi} u_k^\xi b_{k,\xi} = \sum_{j \sim \eta} v_j^\eta b_{j,\eta}$, then the transformation of*

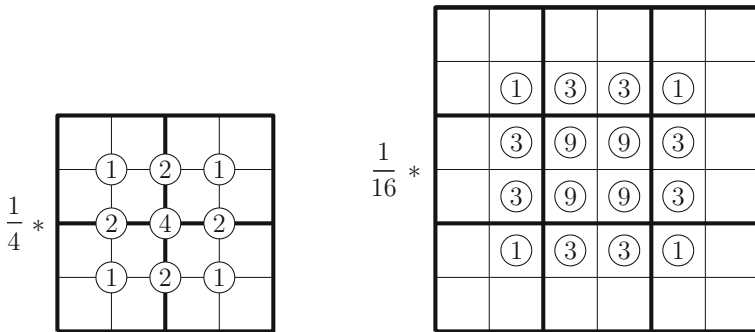


Fig. 4. Subdivision coefficients of a bilinear and biquadratic B-spline

the coefficient vectors has the form

$$v^\eta = S_\xi^\eta u^\xi : \quad v_j^\eta = \sum_k s_{j-2k} u_k^\xi.$$

In pseudo MATLAB code, this extension operation can be implemented as

initialize V with zeros

for $\nu_1, \dots, \nu_d = 0 : n + 1$

$V(2K + \nu) \leftarrow V(2K + \nu) + s_\nu U$

end

where K denotes the relevant range of indices for the B-splines corresponding to the knot sequence ξ .

The assertions follow from the definitions. Substituting the subdivision formula (4) for B-splines and changing the summation index yields

$$\sum_k u_k^\xi b_{k,\xi} = \sum_\nu \sum_k s_\nu u_k^\xi b_{2k+\nu,\eta} = \sum_j \sum_k s_{j-2k} u_k^\xi b_{j,\eta}.$$

The middle expression corresponds to the MATLAB pseudo code and the right expression to the extension operator S_ξ^η .

We now consider the reverse operation.

Corollary 2 (Fine-to-Coarse Restriction). *Let the knot sequence η be the refinement of ξ . If, for a bilinear form $a(\cdot, \cdot)$ and a function φ , $v_j^\eta = a(b_{j,\eta}, \varphi)$, $j \sim \eta$, then $u_k^\xi = a(b_{k,\xi}, \varphi)$, $k \sim \xi$, can be computed with the transformation*

$$u^\xi = S_\eta^\xi v^\eta : \quad u_k^\xi = \sum_j s_{j-2k} v_j^\eta.$$

In pseudo MATLAB code, this restriction operation can be implemented as

initialize U with zeros

for $\nu_1, \dots, \nu_d = 0 : n + 1$

$U \leftarrow U + s_\nu V(2K + \nu)$

end

where K denotes the relevant range of indices for the B-splines corresponding to the knot sequence ξ .

Again, the assertions follow by substituting the subdivision formula (4) for B-splines and changing the summation index:

$$u_k^\xi = a(b_{k,\xi}, \varphi) = \sum_\nu s_\nu a(b_{2k+\nu,\eta}, \varphi) = \sum_\nu s_\nu v_{2k+\nu}^\eta = \sum_j s_{j-2k} v_j^\eta,$$

with the second to last expression corresponding to the MATLAB code.

The grid transfer operations simplify the assembly and solution of the Ritz-Galerkin system considerably. As will be described in the next section, it suffices to compute integrals involving only B-splines on the same grid level.

5 Solving Hierarchical Systems

We recall from Sect. 3 that a hierarchical spline has the form

$$u^\Xi = \sum_{\xi \sim \Xi} \sum_{k \sim \xi} u_k^\xi b_{k,\xi}.$$

Accordingly, the Ritz-Galerkin matrix G has block structure with entries

$$G_\eta^\xi : \quad a(b_{k,\xi}, b_{j,\eta}), \quad k \sim \xi, \quad j \sim \eta.$$

We could proceed in a standard fashion, i.e., assemble G using an appropriate storage scheme, and solve the Ritz-Galerkin system

$$GU = F \quad \Leftrightarrow \quad \sum_{\eta \sim \Xi} G_\eta^\xi u^\eta = f^\xi, \quad \xi \sim \Xi,$$

by some iterative method. However, since the standard solvers just require the implementation of multiplication by the system matrix, we can avoid the explicit assembly of G and use the routines of the FEMB package for the uniform case in an elegant fashion. To this end we examine the matrix/vector multiplication

$$U \mapsto F = GU, \quad f^\xi = \sum_{\eta \sim \Xi} G_\eta^\xi u^\eta,$$

in more detail.

The equation, relating the blocks of the hierarchical vectors U and F , simplifies if we require that Ξ is nested, i.e., if for B-splines in the hierarchical basis only supports of adjacent levels overlap. Since we are also assuming that the extents of knot sequences with the same grid width are disjoint, the matrices G_η^ξ are zero unless $\xi = \eta$, or η is the parent or a child of ξ . Hence, for implementing the multiplication by G_η^ξ , i.e., for computing

$$v^\xi = G_\eta^\xi u^\eta \quad \Leftrightarrow \quad v_k^\xi = \sum_{j \sim \eta} a(b_{k,\xi}, b_{j,\eta}) u_j^\eta, \quad (5)$$

there are three cases to consider.

(i) $\eta = \xi$: Since $G_\xi = G_\xi^\xi$ is a Ritz-Galerkin matrix of B-splines of the same grid level, the FEMB routines are applicable with minor modifications (ξ corresponds in general to a grid covering only a subset of the simulation domain D).

(ii) η is the parent of ξ : In Eq. (5) we refine the coarse grid B-splines $b_{j,\eta}$ and obtain by Corollary 1

$$\sum_{j \sim \eta} u_j^\eta b_{j,\eta} = \sum_{\ell \sim \zeta} w_\ell^\zeta b_{\ell,\zeta}, \quad w^\zeta = S_\eta^\zeta u^\eta,$$

with the knot sequence ζ being the refinement of η . This yields the desired conversion to multiplication with a Ritz-Galerkin matrix for B-splines on the same grid level:

$$v_k^\xi = \sum_{\ell \sim \zeta} a(b_{k,\xi}, b_{\ell,\zeta}) w_\ell^\zeta \quad \Leftrightarrow \quad v_k^\xi = (G_\zeta w^\zeta)_{\bar{k}},$$

where \tilde{k} is the index of the B-spline $b_{k,\xi}$ with respect to ζ ($b_{k,\xi} = b_{\tilde{k},\zeta}$). While ξ_ν are subsequences of ζ_ν , the indices do, in general, not match since the labeling starts from 0 for all knot sequences (cf. Fig. 3).

A final adjustment is necessary. Multiplying w^ζ by G_ζ results in a larger vector than needed. Since ξ is a local refinement of η , $[\xi] \subseteq [\eta] = [\zeta]$. As a consequence, some B-splines of the knot sequence ζ are, in general, irrelevant for ξ . Hence, we have to extract the entries relevant for ξ :

$$v_k^\zeta, k \sim \zeta \quad \mapsto \quad v_k^\xi, k \sim \xi.$$

We denote this truncation operation, which also incorporates the index adjustment, by $v^\xi = I_\zeta^\xi v^\zeta$. Summarizing, we obtain the following procedure.

Theorem 2 (Coarse-to-Fine Multiplication). *If the knot sequence η is the parent of ξ , then*

$$v^\xi = G_\eta^\xi u^\eta \quad \Leftrightarrow \quad v^\xi = I_\zeta^\xi G_\zeta S_\eta^\zeta u^\eta$$

with ζ the refinement of η .

(iii) η is a child of ξ : As in the previous case, we rewrite Eq. (5) in terms of the Ritz-Galerkin matrix G_ζ , where ζ is the refinement of the coarse knot sequence ξ . To this end let $u^\zeta = I_\eta^\zeta u^\eta$ denote the extension of the coefficient vector u^η with coefficients of the B-splines $b_{j,\zeta}$, $j \sim \zeta$, which do not correspond to the smaller knot sequence η ($[\eta] \subseteq [\xi] = [\zeta]$) set to zero, and define

$$\varphi = \sum_{j \sim \eta} u_j^\eta b_{j,\eta} = \sum_{j \sim \zeta} u_j^\zeta b_{j,\zeta}.$$

Then, according to Corollary 2, $v_k^\xi = a(b_{k,\xi}, \varphi)$ can be computed from $w_\ell^\zeta = a(b_{\ell,\zeta}, \varphi)$ with the restriction operator:

$$v^\xi = S_\zeta^\xi w^\zeta, \quad w^\zeta = G_\zeta u^\zeta.$$

We summarize this procedure as follows.

Theorem 3 (Fine-to-Coarse Multiplication). *If the knot sequence η is a child of ξ , then*

$$v^\xi = G_\eta^\xi u^\eta \quad \Leftrightarrow \quad v^\xi = S_\zeta^\xi G_\zeta I_\eta^\zeta u^\eta$$

with ζ the refinement of ξ .

Summarizing, we have reduced the multiplication with the hierarchical matrix G to multiplications with Ritz-Galerkin matrices G_ξ , involving only B-splines with the same grid width. These matrices can be precomputed using the FEMB package before an iterative solver is started. If we employ, e.g., the conjugate gradient method, we have to provide in addition routines for adding hierarchical vectors, multiplication with scalars, and forming of scalar products. This is straightforward.

As a final comment on the programming details, we note that the hierarchical matrix G contains zero rows and columns since we keep irrelevant B-splines for ease of programming. This does not cause any problem though, since the corresponding sections in the coefficient vectors are set to zero. In effect, the iteration “sees” only the invertible part of G .

6 Accuracy of Adaptive Approximations

As a sample problem for testing the proposed method, we chose Helmholtz equation (3) with right-hand side $f(x) = \sin(\pi x_1 x_2)$ on the B-shaped domain shown in Fig. 5. D is the intersection of the square $(0, 1)^2$ with the union of the two ellipses

$$E_1 : x_1^2 + \frac{(x_2 - a)^2}{a^2} < 1 \quad \text{and} \quad E_2 : \frac{9x_1^2}{4} + \frac{(x_2 - b)^2}{(1 - b)^2} < 1,$$

where $a = \frac{11}{10}(2 - \sqrt{3})$ and $b = \frac{1}{5}(1 + \sqrt{7})$. The boundary of D has a reentrant corner at $x^* = (0.5, 0.55)$ with an interior angle of width $\alpha \approx 1.8089\pi$, causing the solution u of (3) to form a singularity at this point. More precisely, u is of asymptotic order $O(r^{\pi/\alpha}) \approx O(r^{0.5528})$ in the region near the corner, where r denotes the distance to x^* , cf. [29]. No other singularities occur at the corners $(0, 0)$ and $(0, 1)$ since, due to symmetry, u can be regarded as restriction to D of the solution on the entire domain $E_1 \cup E_2$, which has a smooth boundary near these two points.

The shading in Fig. 5 illustrates the growth of $|\text{grad } u|$ in the vicinity of the reentrant corner, indicating that the gradient of u — yet still being square integrable on D — becomes infinite when approaching x^* .

Corner singularities strongly affect the performance of both conventional finite element schemes and methods based on uniform tensor product B-splines,

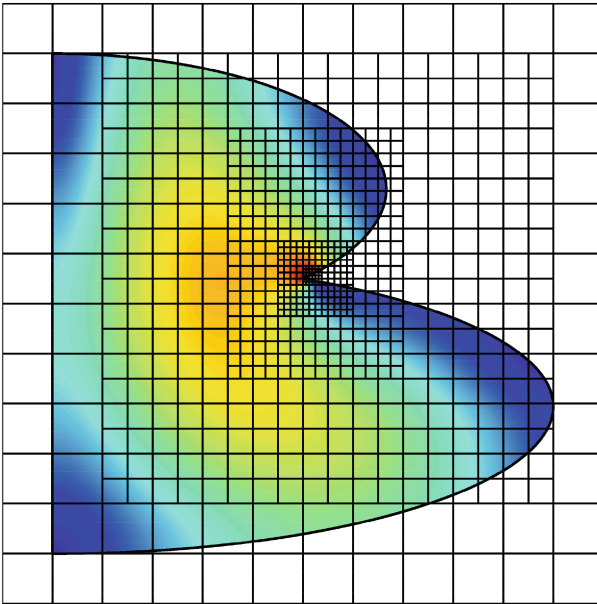


Fig. 5. Singular solution with hierarchical grid (Color figure online)

as they require the solution to belong to some higher order Sobolev space in order to yield optimal rates of convergence. If, as in our example, u has no square integrable second derivatives, the uniform bases cannot release their full approximation power, which results in a slow convergence of numerical solutions. Not even the use of higher spline degrees pays off, as can be seen from Fig. 6 where, regardless of the degree, all uniform spline approximations exhibit about the same poor convergence rate of $O(h^{3/2})$, or $O(N^{-3/4})$ if expressed in terms of the dimension N of the respective spline spaces.

For comparison, we computed approximate solutions to (3) in sequences of hierarchical spline spaces, defined by nested trees with different numbers of levels. Local grid refinement was employed (manually) in regions where the pointwise error of a corresponding uniform approximation was large with respect to the average error. A typical choice of the grid structure for a quadratic spline space is depicted in Fig. 5.

The significant improvement in efficiency achieved through hierarchical refinement near the critical corner is apparent from Fig. 6. The number of parameters necessary to attain similar accuracy is, on an average, smaller by a factor

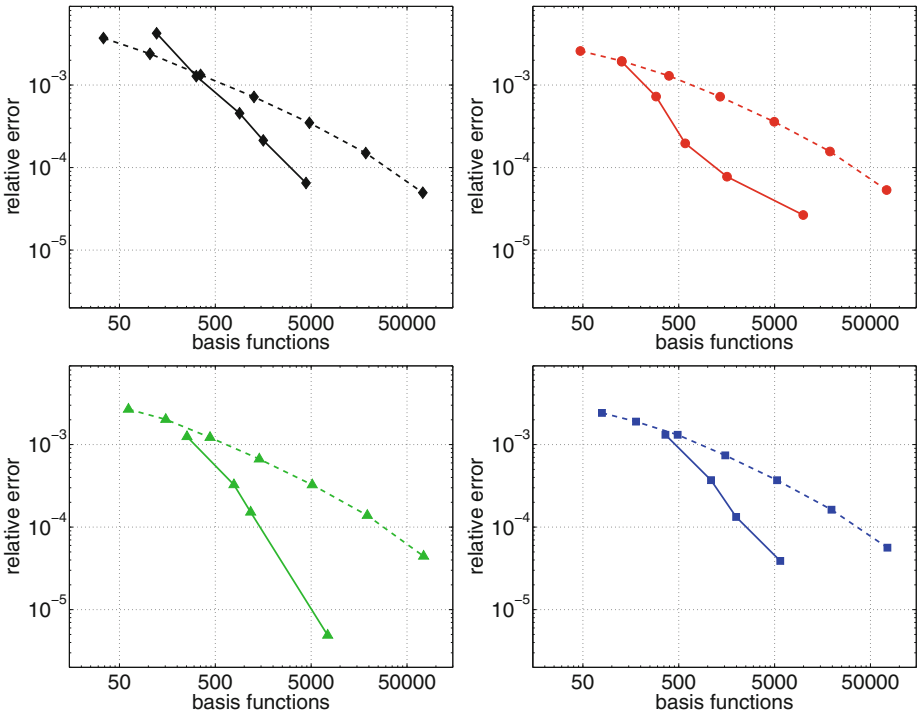


Fig. 6. Relative L^2 error for uniform (dashed) and hierarchical approximations (solid) as a function of the dimension for degrees $n = 1$ (\blacklozenge), 2 (\bullet), 3 (\blacktriangle) and 4 (\blacksquare) (Color figure online)

of more than 20, compared with approximations on uniform grids. For example, quadratic elements meet a tolerance of $7.8 \cdot 10^{-5}$ with 1593 B-splines on five hierarchical levels, whereas the error for a finite element space spanned by 18900 quadratic B-splines on a uniform grid of 160×160 cells is still greater than $1.5 \cdot 10^{-4}$.

7 Conclusion

We have described the implementation of hierarchical finite element methods with B-splines using the FEMB package. B-spline subdivision plays a crucial role and leads to a very simple program structure. The numerical results for Helmholtz's equation as a model problem document the substantial gains in accuracy due to adaptive refinement.

For classical mesh-based finite element methods, adaptive techniques have been studied intensively. Hence, for the relatively novel application to B-splines, a number of topics remain to be investigated. An interesting question is, whether the possibility of computing pointwise residuals of the partial differential equations (B-splines are sufficiently smooth) can be exploited. Moreover, deriving optimal error estimates for typical singularities is a major project. These are just two examples of a longer list of open problems to which we hope to contribute in the future.

References

1. Höllig, K., Reif, U., Wipper, J.: Weighted extended B-spline approximation of Dirichlet problems. *SIAM J. Numer. Anal.* **39**, 442–462 (2001)
2. Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Meth. Appl. Mech. Eng.* **194**, 4135–4195 (2005)
3. Höllig, K.: *Finite Element Methods with B-Splines*. SIAM, Philadelphia (2003)
4. Cottrell, J.A., Hughes, T.J.R., Bazilevs, Y.: *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester (2009)
5. Höllig, K., Hörner, J., Hoffacker, A.: Finite element analysis with B-splines: weighted and isogeometric methods. In: Boissonnat, J.-D., Chenin, P., Cohen, A., Gout, C., Lyche, T., Mazure, M.-L., Schumaker, L. (eds.) *Curves and Surfaces 2011*. LNCS, vol. 6920, pp. 330–350. Springer, Heidelberg (2012)
6. Forsey, D.R., Bartels, R.H.: Hierarchical B-spline refinement. *Comput. Graph.* **22**, 205–212 (1988)
7. Forsey, D.R., Bartels, R.H.: Surface fitting with hierarchical splines. *ACM Trans. Graph.* **14**, 134–161 (1995)
8. Kraft, R.: Adaptive and linearly independent multilevel B-splines. In: Le Méhauté, A., Rabut, C., Schumaker, L. (eds.) *Surface Fitting and Multiresolution Methods*, pp. 209–218. Vanderbilt University Press, Nashville (1997)
9. Greiner, G., Hormann, K.: Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In: Le Méhauté, A., Rabut, C., Schumaker, L. (eds.) *Surface Fitting and Multiresolution Methods*, pp. 163–172. Vanderbilt University Press, Nashville (1997)

10. Rabut, C.: Locally tensor product functions. *Numer. Algorithm.* **39**, 325–348 (2005)
11. Giannelli, C., Jüttler, B., Speleers, H.: Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.* **40**, 459–490 (2014)
12. Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A.: T-splines and T-NURCCS. *ACM Trans. Graph.* **22**, 477–484 (2003)
13. Dokken, T., Lyche, T., Pettersen, K.F.: Polynomial splines over locally refined box-partitions. *Comput. Aided Geom. Des.* **30**, 331–356 (2013)
14. Mustahsan, M.: Finite element methods with hierarchical WEB-splines. Dissertation, Universität Stuttgart (2011)
15. Vuong, A.V., Giannelli, C., Jüttler, B., Simeon, B.: A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Meth. Appl. Mech. Eng.* **200**, 3554–3567 (2011)
16. Schillinger, D., Rank, E.: An unfitted *hp*-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry. *Comput. Meth. Appl. Mech. Eng.* **200**, 3358–3380 (2011)
17. Bornemann, P.B., Cirak, F.: A subdivision-based implementation of the hierarchical b-spline finite element method. *Comput. Meth. Appl. Mech. Eng.* **253**, 584–598 (2013)
18. Bazilevs, Y., Calo, V.M., Cottrell, J.A., Evans, J.A., Hughes, T.J.R., Lipton, S., Scott, M.A., Sederberg, T.W.: Isogeometric analysis using T-splines. *Comput. Meth. Appl. Mech. Eng.* **199**, 229–263 (2010)
19. Dörfel, M.R., Jüttler, B., Simeon, B.: Adaptive isogeometric analysis by local *h*-refinement with T-splines. *Comput. Meth. Appl. Mech. Eng.* **199**, 264–275 (2010)
20. Scott, M.A., Borden, M.J., Verhoosel, C.V., Sederberg, T.W., Hughes, T.J.R.: Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int. J. Numer. Meth. Eng.* **88**, 126–156 (2011)
21. Scott, M.A., Li, X., Sederberg, T.W., Hughes, T.J.R.: Local refinement of analysis-suitable T-splines. *Comput. Meth. Appl. Mech. Eng.* **213–216**, 206–222 (2012)
22. Schillinger, D., Dedè, L., Scott, M.A., Evans, J.A., Borden, M.J., Rank, E., Hughes, T.J.R.: An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Meth. Appl. Mech. Eng.* **249–252**, 116–150 (2012)
23. Höllig, K., Hörner, J.: Finite element methods with B-splines: supplementary material (2012). <http://www.siam.org/books/fr26/>
24. Höllig, K., Hörner, J.: Approximation and Modeling with B-Splines. SIAM, Philadelphia (2013)
25. Höllig, K., Hörner, J.: Programming finite element methods with B-splines. To appear in: *Comput. Math. Appl., Special Issue on High-Order Finite Element and Isogeometric Methods (HOFEIM 2014)* (2015)
26. Rvachev, V.L., Sheiko, T.I.: R-functions in boundary value problems in mechanics. *Appl. Mech. Rev.* **48**, 151–188 (1995)
27. Boehm, W.: Inserting new knots into B-spline curves. *Comput. Aided Des.* **12**, 199–201 (1980)
28. Cohen, E., Lyche, T., Riesenfeld, R.F.: Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Comput. Graph. Image Proc.* **14**, 87–111 (1980)
29. Nazarov, S.A., Plamenevsky, B.A.: Elliptic Problems in Domains with Piecewise Smooth Boundaries. de Gruyter, Berlin (1994)