Archives of Computational Methods in Engineering (2021) 28:473–494
https://doi.org/10.1007/s11831-019-09385-1

ORIGINAL PAPER

# High-Order Shape Functions in the Scaled Boundary Finite Element Method Revisited

Hauke Gravenkamp[1] · Albert A. Saputra[2] · Sascha Duczek[2]

Received: 26 May 2019 / Accepted: 8 December 2019 / Published online: 19 December 2019

**Abstract**
The scaled boundary finite element method (SBFEM) is a semi-analytical approach to solving partial differential equations, in which a finite element approximation is deployed for the domain's boundary, while analytical solutions are sought to describe the behavior in the interior of the domain. Since the inception of SBFEM, a number of different shape functions have been applied to interpolate the solution on the boundary. The overarching goal of this communication is to review the respective advantages and disadvantages of the available interpolants in the context of the SBFEM and develop recommendations regarding their application. In addition, we discuss in detail the discretization employed in the so-called diagonal SBFEM.

**Keywords** Scaled boundary finite element method (SBFEM) · High-order elements · Isogeometric analysis (IGA) · Hierarchical shape functions ($p$-FEM) · Spectral element method (SEM)

## 1 Introduction

The scaled boundary finite element method (SBFEM) is a semi-analytical approach to solving partial differential equations (PDEs) [54]. It is based on transforming the PDE into a set of coupled ordinary differential equations (ODEs) by

1. performing a coordinate transformation $(x, y) \rightarrow (\eta, \xi)$, where $\eta$ denotes a parametrization of the domain's boundary while $\xi$ defines a coordinate pointing from the origin of the coordinate system to the boundary, and
2. applying the method of weighted residuals [57] or the virtual work principle [11] to obtain a weak form of the governing equation with respect to the coordinate $\eta$.

Roughly speaking, the $\eta$-direction is discretized "in the finite element sense" [21, 56, 66], which means that the solution is interpolated using trial functions and a weak form is derived in this coordinate similar to the general concept of finite elements. That being said, we may employ different types of trial or *shape functions*, which basically have to fulfill the same requirement as the ones used in any other variant of the finite element method (FEM). In the early days of the SBFEM, the standard linear and quadratic shape functions as established in the conventional FEM have been used almost exclusively [56, 57, 64, 65]. As alternative classes of interpolants became more popular in finite element approximations (as well as boundary elements, meshless methods, and others), these techniques also found their way into the SBFEM. In particular, Vu & Deeks discussed the application of high-order polynomial shape functions—namely Lagrange polynomials and hierarchical shape functions—in the SBFEM [59]. The use of high-order polynomials is motivated by the attainable higher convergence rates[1] and the consequent reduction of the total number of degrees of freedom (DOF) compared to a low-order interpolation. In this context, it is worthwhile noting that the stiffness and mass matrices computed in the SBFEM for each subdomain are always dense matrices. Thus, the trade-off between a reduction in the number of DOF and the loss of sparsity of the system of equations—which is a crucial consideration in other finite element-based high order methods—is not as significant in the SBFEM.

✉ Hauke Gravenkamp
hauke.gravenkamp@uni-due.de

[1] University of Duisburg-Essen, Universitätsstraße 15, 45141 Essen, Germany

[2] University of New South Wales Sydney, Sydney 2052, Australia

[1] Exponential convergence can be achieved by using a $p$-refinement for sufficiently smooth problems or an $hp$-refinement if singularities are present in the solution.

Since the introduction of high-order polynomial shape functions in the SBFEM, a number of other options have been proposed in separate publications. One of these alternatives is the use of Fourier shape functions, which consist of a series of harmonic functions. Such interpolation is sometimes believed to be advantageous for wave propagation problems since analytical solutions, if available, are typically constructed based on a Fourier series. Within the scope of the SBFEM, the application of Fourier shape functions is discussed in [27]. A rather different class of interpolants is constructed based on the concept of moving least squares. This idea is adopted from meshless Petrov Galerkin [10] and element-free Galerkin [26] methods. Furthermore, B-splines and NURBS (non-uniform rational B-splines) have been used as shape functions in attempts to combine the advantages of the SBFEM and isogeometric analysis (IGA) [20, 41, 42, 47, 61]. In a particular variant, a NURBS-based approximation has been applied in both the radial and circumferential direction in the SBFEM formulation [7, 8, 37]. It may be noted that some of the authors mentioned above choose a terminology different to SBFEM due to them using non-standard elements (e.g., "local Petrov-Galerkin scaled boundary method" [10]) or solving the resulting ODE numerically (e.g., "hybrid collocation-Galerkin method" [37]). However, to keep confusion to a minimum, we will use the term SBFEM throughout this paper, since the different types of interpolation will be discussed in detail.

Another variant of the SBFEM has been published in a series of papers under the name "diagonal SBFEM" [31–36, 46, 67]. This approach uses a set of particular polynomials in conjunction with nodal quadrature with the aim to decouple the set of ODEs, hence describing each node individually. The interaction of the nodes on the boundary is then approximated by a very different solution procedure that sums up the forces acting on the scaling center and distributes them to the nodes. While we are not employing this solution procedure, we shed some light on this approach and discuss in detail the class of polynomial shape functions used therein. We demonstrate that these shape functions are not suitable to be used in a standard FE or SBFE approach. We also provide some insight as to why in certain simple cases, the diagonal SBFEM may potentially still give reasonable results (when employing the special solution procedure) even though the shape functions do not possess the requirements necessary to achieve convergence.

While basic information on all these different interpolations can be found in the literature, direct comparisons in the context of the SBFEM are scarce. For instance, the high-order Fourier shape functions have only been compared to conventional low order finite elements [26], though they can be expected to have more similarity with, e.g., *p*-refined Lagrange elements. Similarly, it is not apparent how the meshless approaches [10, 26] compare to high-order interpolations. High-order NURBS-based interpolants are also usually compared against low-order finite elements [41, 47, 61] or against high-order Lagrange shape functions for very specific cases [20]. In this paper, we clarify the differences between the available interpolants both from a theoretical and practical viewpoint. We also show the results of several numerical experiments and derive some general statements and recommendations that will help the reader to make an informed decision on the implementation of a suitable approach in the context of the SBFEM.

## 2 Summary of the SBFEM

Since the topic of this contribution will be of relevance for readers who are already familiar with the concept and basic formulation of the SBFEM, we shall abstain from presenting a lengthy summary of this method and only provide the key equations that will be required in the scope of this paper. Details can be found, e.g., in Refs. [19, 53, 54, 57]. Also, for brevity, we will restrict the discussion to the two-dimensional case even though most of the statements can be adapted for three-dimensional problems straightforwardly. In a nutshell, the PDE that is to be solved on a domain is transformed into a set of ODEs by employing the concept of semi-discretization: The solution $u(x, y)$—say a displacement or temperature field[2]—is approximated as

$$u(x(\eta, \xi), y(\eta, \xi)) = \mathbf{N}(\eta)\mathbf{u}(\xi) \tag{1}$$

where $\mathbf{N}(\eta)$ denotes a set of shape functions in one spatial coordinate and $\mathbf{u}(\xi)$ are the corresponding "nodal" solutions which are a priori unknown functions of the second spatial coordinate. The chosen coordinate system is one where $\eta$ describes a parametrization of the boundary and $\xi$ is a linear parameter that equals 1 at the boundary and 0 at the origin, which is usually inside the domain. The relationship between the Cartesian coordinates $(x, y)$ and the *scaled boundary* coordinates $(\eta, \xi)$ is typically defined as

$$x = \xi\hat{\mathbf{N}}(\eta)\mathbf{x} \tag{2a}$$

$$y = \xi\hat{\mathbf{N}}(\eta)\mathbf{y} \tag{2b}$$

where $\mathbf{x}, \mathbf{y}$ are vectors of nodal coordinates and $\hat{\mathbf{N}}(\eta)$ are shape functions used to interpolate the boundary. Most implementations of the SBFEM assume $\hat{\mathbf{N}}(\eta) = \mathbf{N}(\eta)$, i.e., isoparametric elements. Applying the method of weighted residuals [57] or the virtual work principle [11, 17] in the $\eta$

[2] We assume here for ease of notation that the solution $u(x, y)$ is a scalar field. The extension to vector fields is straightforward and analogous to other finite element approaches.

direction results in a *semi-analytical* matrix ODE, here given in the frequency domain ($\omega \geq 0$):

$$\mathbf{E}_0\xi^2\mathbf{u}_{,\xi\xi} + (\mathbf{E}_0 - \mathbf{E}_1 + \mathbf{E}_1^{\mathrm{T}})\xi\mathbf{u}_{,\xi} - \mathbf{E}_2\mathbf{u} + \omega^2\xi^2\mathbf{M}_0\mathbf{u} = 0 \tag{3}$$

The coefficient matrices are somewhat similar to stiffness and mass matrices in the FEM but involve only integrations over the boundary. For the common case of linear elasticity, they are given as

$$\mathbf{E}_0 = \int_{-1}^{1} \mathbf{B}_1^{\mathrm{T}}\mathbf{D}\mathbf{B}_1|\mathrm{J}|\mathrm{d}\eta \tag{4a}$$

$$\mathbf{E}_1 = \int_{-1}^{1} \mathbf{B}_2^{\mathrm{T}}\mathbf{D}\mathbf{B}_1|\mathrm{J}|\mathrm{d}\eta \tag{4b}$$

$$\mathbf{E}_2 = \int_{-1}^{1} \mathbf{B}_2^{\mathrm{T}}\mathbf{D}\mathbf{B}_2|\mathrm{J}|\mathrm{d}\eta \tag{4c}$$

$$\mathbf{M}_0 = \rho\int_{-1}^{1} \mathbf{N}^{\mathrm{T}}\mathbf{N}|\mathrm{J}|\,\mathrm{d}\eta \tag{4d}$$

with the mass density $\rho$, elasticity/constitutive matrix $\mathbf{D}$, and

$$\mathbf{B}_1 = \mathbf{b}_1\mathbf{N}, \qquad \mathbf{B}_2 = \mathbf{b}_2\mathbf{N}_{,\eta} \tag{5}$$

$$\begin{aligned} \mathbf{b}_1 &= \frac{1}{|\mathrm{J}|}\begin{bmatrix} y_{,\eta} & 0 \\ 0 & -x_{,\eta} \\ -x_{,\eta} & y_{,\eta} \end{bmatrix}; \\ \mathbf{b}_2 &= \frac{1}{|\mathrm{J}|}\begin{bmatrix} -y & 0 \\ 0 & x \\ x & -y \end{bmatrix} \end{aligned} \tag{6}$$

Rather than solving the differential Eq. (3) directly, it is often transformed into an equation for the (dynamic) stiffness $\mathbf{S}(\omega)$ on the boundary, which reads [53]

$$(\mathbf{S}(\omega) - \mathbf{E}_1)\mathbf{E}_0^{-1}(\mathbf{S}(\omega) - \mathbf{E}_1^{\mathrm{T}}) - \mathbf{E}_2 + \omega\mathbf{S}(\omega)_{,\omega} + \omega^2\mathbf{M}_0 = 0 \tag{7}$$

In the static case, Eq. (7) reduces to a Riccati equation for the static stiffness matrix $\mathbf{K}$, which can be solved using standard techniques:

$$(\mathbf{K} - \mathbf{E}_1)\mathbf{E}_0^{-1}(\mathbf{K} - \mathbf{E}_1^{\mathrm{T}}) - \mathbf{E}_2 = 0 \tag{8}$$

For dynamic cases, it is now quite common to solve Eq. (7) by applying the continued fraction expansion detailed in Refs. [5, 53] of the form

$$\begin{aligned} \mathbf{S}(\omega) = &\mathbf{K} - \omega^2\mathbf{M} - \omega^4\mathbf{X}^{(1)}(\mathbf{S}_0^{(1)} - \omega^2\mathbf{S}_1^{(1)} \\ &- \omega^4\mathbf{X}^{(2)}(\mathbf{S}_0^{(2)} - \omega^2\mathbf{S}_1^{(2)} - \cdots - \omega^4\mathbf{X}^{(M_{cf})}(\mathbf{S}_0^{(M_{cf})} \\ &- \omega^2\mathbf{S}_1^{(M_{cf})})^{-1}[\mathbf{X}^{(M_{cf})}]^{\mathrm{T}})^{-1}[\mathbf{X}^{(2)}]^{\mathrm{T}})^{-1}[\mathbf{X}^{(1)}]^{\mathrm{T}}, \end{aligned} \tag{9}$$

where $M_{cf}$ denotes the order of the expansion. The matrices $\mathbf{S}_0^{(1)} \ldots \mathbf{S}_0^{(M_{cf})}$ and $\mathbf{S}_1^{(1)} \ldots \mathbf{S}_1^{(M_{cf})}$ are computed recursively starting from the static stiffness and mass matrices $\mathbf{K}, \mathbf{M}$ [53]. The matrices $\mathbf{X}^{(1)} \ldots \mathbf{X}^{(M_{cf})}$ have been introduced for preconditioning [5]. For computations in the time domain, high-order stiffness and mass matrices $\mathbf{K}_{\mathrm{h}}, \mathbf{M}_{\mathrm{h}}$ are constructed using the terms of the continued fraction expansion

$$\mathbf{K}_{\mathrm{h}} = \mathrm{diag}(\mathbf{K}, \mathbf{S}_0^{(1)}, \mathbf{S}_0^{(2)}, \ldots, \mathbf{S}_0^{(M_{cf})}) \tag{10a}$$

$$\mathbf{M}_{\mathrm{h}} = \begin{bmatrix} \mathbf{M} & -\mathbf{X}^{(1)} & 0 & \cdots & 0 \\ -[\mathbf{X}^{(1)}]^{\mathrm{T}} & \mathbf{S}_1^{(1)} & -\mathbf{X}^{(2)} & \cdots & 0 \\ 0 & -[\mathbf{X}^{(2)}]^{\mathrm{T}} & \mathbf{S}_1^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{S}_1^{(M_{cf})} \end{bmatrix} \tag{10b}$$

which can be utilized to form the equation of motion.

## 3 High-Order Interpolation

In this section, we will present the different classes of high-order interpolants that have been applied in the context of the SBFEM and discuss their properties. The interpolation functions (shape functions) are defined in a local coordinate system that represents a parametrization of a part of the boundary. According to standard finite element notation, each section of the boundary is mapped onto a local coordinate on the interval $\eta \in [-1, 1]$. In some cases, we will instead use the parametrization $s \in [0, 1]$ in order to conform with the corresponding literature. To avoid notational overload, we will use the same symbols in the different subsections, e.g., a set of shape functions is denoted as $N_i(\eta)$ or $N_i(s)$ irrespective of the class of interpolation currently addressed.

### 3.1 Lagrange Polynomials

Lagrange interpolation polynomials are probably still the most popular high-order interpolants within the field of Computational Mechanics. In the framework of the SBFEM, these interpolations have been introduced in Ref. [59] and since then have been employed in countless applications (e.g., [44, 55]) due to their comparably simple and stable computation for almost arbitrary order. Rather detailed discussions of Lagrange shape functions of a very high order in the SBFEM can be found in Refs. [17, 20, 22]. Given a set of interpolation points (nodes), denoted by $\eta_i$, Lagrange polynomials of order $p$ are defined as [1]:

$$N_i(\eta) = \prod_{j=1, j\neq i}^{p+1} \frac{\eta - \eta_j}{\eta_i - \eta_j}\,, \qquad i = 1, 2, \ldots, p+1\,. \tag{11}$$

Such shape functions constitute a rather consistent extension of conventional low order (linear and quadratic) finite elements, inasmuch as they exhibit similar properties, in particular:

- *Kronecker-delta-property:*

$$N_i(\eta_j) = \delta_{ij} \tag{12}$$

- *partition of unity:*

$$\sum_i N_i(\eta) = 1 \tag{13}$$

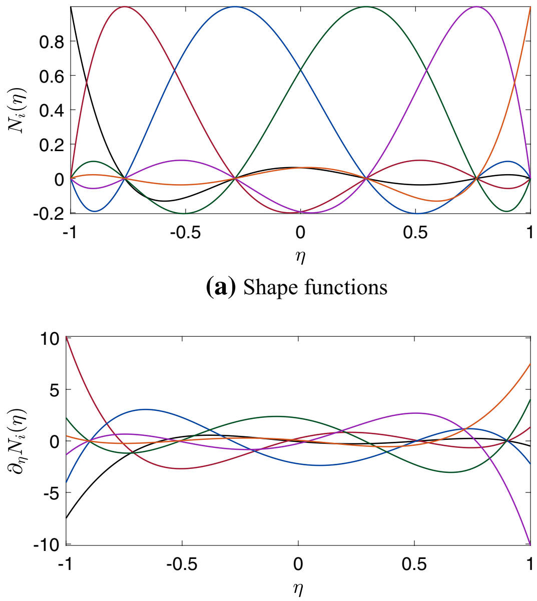
Equation (12) also implies that these shape functions are 'node-based' since $u(\eta_i, \xi) = u_i(\xi)$, i.e., the unknowns retain a physical meaning. Therefore, the post-processing and the application of Dirichlet boundary conditions are straightforward. While Lagrange interpolation polynomials can, in principle, be defined for any set of distinct nodes, their properties depend critically on the nodal positions.

In the wide body of literature, there are several possibilities to distribute these points within the reference interval $[-1, 1]$, see Ref. [51]. In the context of the FEM and SEM, the following distributions are commonly used: equidistant (EQ), Gauss-Lobatto-Chebyshev (GLC), and Gauss-Lobatto-Legendre (GLL) points:

1. Equidistant points [69]:
   - $\eta_i^{\mathrm{EQ}} = -1 + 2\frac{i-1}{p}\,, \quad i = 1, 2, \ldots, p+1\,.$
2. GLC points [48]:
   - $\eta_i^{\mathrm{GLC}} = -\cos\left(\frac{i-1}{p}\,\pi\right), \quad i = 1, 2, \ldots, p+1\,.$
3. GLL points [29]:
   - $\eta_i^{\mathrm{GLL}} = \begin{cases} -1 & \text{for } i = 1 \\ \hat{\eta}_i & \text{for } i = 2, 3, \ldots, p \\ +1 & \text{for } i = p+1 \end{cases}.$

In the above definition, $\hat{\eta}_i$ denotes the roots of the Lobatto polynomial $L_{o_{p-1}}(\eta)$ of order $p-1$, which is defined as the first derivative of the Legendre polynomial $L_p(\eta)$ of order $p$ [51]

$$L_{o_{p-1}}(\eta) = \frac{\mathrm{d}L_p(\eta)}{\mathrm{d}\eta}\,. \tag{14}$$



**(a)** Shape functions

**(b)** Derivatives

**Fig. 1** Lagrange shape functions defined on GLL points, $n = 6$

Note that the equidistant nodal distribution suffers from the so-called Runge effect [51], i.e., the quality of the interpolation deteriorates close to the interval boundaries due to significant oscillations. This problem can be circumvented easily by utilizing non-equidistant distributions such as GLL and GLC points. Here, it has been shown that the results are stable for very high orders ($p \geq 100$) [17, 20]. Figure 1 shows an example of Lagrange interpolation of order 5 ($n = 6$ denotes the number of degrees of freedom) based on GLL points. The most prominent drawback of this set of shape functions is that when changing the order $p$, all shape functions are modified and therefore, the coefficient matrices have to be computed anew. Lagrange interpolation polynomials are commonly used in the spectral element method (SEM), especially for wave propagation phenomena [9, 52]. In this context, the main advantage is the possibility to diagonalize the mass matrix (mass lumping) without loss of accuracy, i.e., retaining optimal convergence rates [13, 14]. Similarly, when using the SBFEM, the coefficient matrices $\mathbf{E}_0$ and $\mathbf{M}_0$ can be lumped without loss of accuracy [18, 23, 53, 59]. Consequently, the inversion of $\mathbf{E}_0$—which is performed for each subdomain separately—becomes trivial (Eq. (7)); however, for most applications, this does not constitute a very significant advantage. The mass matrix $\mathbf{M}$ (Eq. (9)) is a dense matrix independent of the selected type of shape functions.

***Remark 1*** Lumping of the mass matrix (SEM) or the coefficient matrices $\mathbf{E}_0$ and $\mathbf{M}_0$ (SBFEM) is often achieved by employing a so-called nodal quadrature, i.e., by choosing the nodes to coincide with the quadrature points. A popular variant uses the GLL points listed above in conjunction with the corresponding GLL quadrature. In doing so, the matrices $\mathbf{E}_0$ and $\mathbf{M}_0$ are slightly "under-integrated" by at least one order, meaning that (assuming a constant Jacobian) these matrices involve integrals of polynomials of order $2p$, while the corresponding GLL quadrature integrates polynomials of order $2p-1$ exactly. Alternatively, the matrices can be integrated exactly (using Gauss-Legendre quadrature or more integration points) and then lumped by row-summing. In most cases, both approaches lead to identical or very similar results; the reason for this behaviour is explained in detail in Ref. [14]. In the current paper, we will not distinguish between these cases and always use full integration when dealing with Lagrange shape functions.

## 3.2 Hierarchical Shape Functions

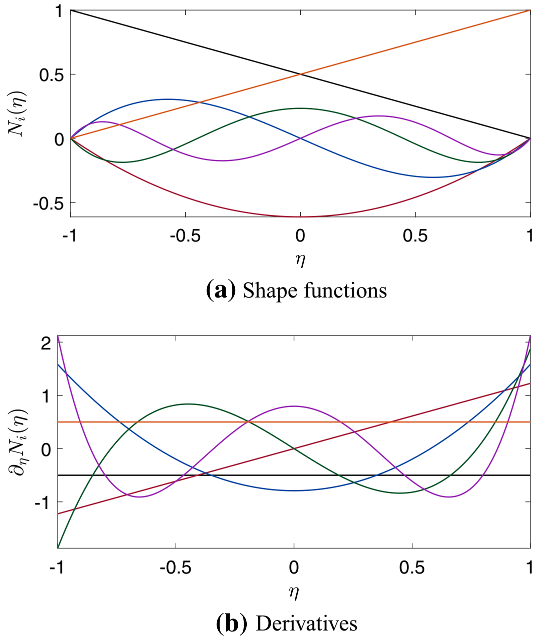
The concept of hierarchical shape functions is linked to the $p$-version of the FEM [16, 58]. In this context, hierarchical means that all shape functions of order $p$ are contained in the set of shape functions of order $p+1$. Though the term *hierarchical shape functions* usually refers to the specific type of interpolation discussed here, the property of being hierarchical is, strictly speaking, common between various kinds of interpolations. To avoid confusion (and create a concise abbreviation), we will often refer to this class of interpolants as *p-FEM*. The construction of these shape functions is based on the normalized integrals of the Legendre polynomials:

$$N_i(\eta) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{\eta} L_{i-1}(x)\mathrm{d}x\,, \quad i = 2, 3, \ldots, p\,, \tag{15}$$

with $L_{i-1}(x)$ denoting the Legendre polynomial of order $i-1$. The functions defined by Eq. (15) contain polynomials of order 2 up to $p$ and vanish at the interval limits. To create a set of complete polynomials, these interpolants are augmented by the standard linear finite element shape functions:

$$N_1(\eta) = \frac{1}{2}(1-\eta)\,, \tag{16a}$$

$$N_{p+1}(\eta) = \frac{1}{2}(1+\eta)\,. \tag{16b}$$



**(a)** Shape functions

**(b)** Derivatives

**Fig. 2** $p$-FEM shape functions, $n = 6$

Figure 2 shows the hierarchical shape functions up to an order of $p = 5$. The Legendre polynomials can be computed using Bonnet's recursion expression [15] or Rodriguez' formula

$$L_i(x) = \frac{1}{2^i i!}\frac{\mathrm{d}^i}{\mathrm{d}x^i}\left[(x^2-1)^i\right], \quad i \in \mathbb{N}_0 \tag{17}$$

Note that only the points at the interval limits possess the Kronecker delta property and therefore, only the DOFs associated with these modes retain physical meaning. Consequently, these shape functions are often referred to as modal functions [51]. This results in an additional step in the post-processing and the application of Dirichlet boundary conditions (BCs). The linear shape functions alone possess both the Kronecker delta and the partition of unity (PUM) properties while the higher order modes ensure the completeness of the ansatz for arbitrary polynomial orders. The hierarchical property implies that the order of interpolation can be increased by including additional shape functions without modifying the existing ones. In the SBFEM, the coefficient matrices inherit this hierarchy, similar to the stiffness and mass matrices in the FEM [15]. Details on adaptive refinement procedures in the context of the SBFEM are published in Ref. [60].

## 3.3 Fourier Shape Functions

Another approach to constructing shape functions of arbitrary order consists in using a series of harmonic functions. Their application in the SBFEM has been proposed by He et al. [27] and was inspired by the works of Guan et al. [24] and Khaji et al. [25, 30] who employed similar concepts in the framework of the FEM and BEM, respectively. The resulting interpolants are referred to as *Fourier shape* *functions* since they resemble an approximation of the resulting displacement field by a Fourier series. To ensure that a subset of the basis functions forms a partition of unity and is capable of representing constant strains, the constant term of the standard Fourier series is replaced by two linear functions:

$$\begin{aligned} u_h(s) = {} & a_0 \frac{L-s}{L} + b_0 \frac{s}{L} \\ & + \sum_{j=1}^{m} \left( a_j \cos\left(\frac{j\pi}{L}s\right) + b_j \sin\left(\frac{j\pi}{L}s\right)\right) \end{aligned} \tag{18}$$

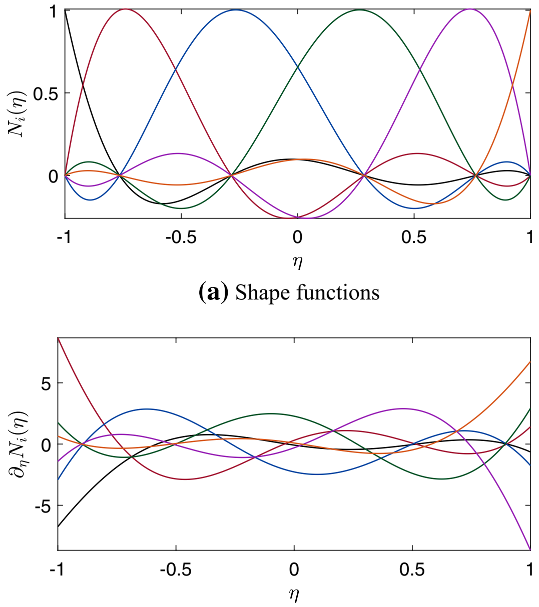
Here $a_j, b_j$ are the unknown coefficients of the basis functions. The reason why this approach may seem advantageous is mainly due to the simplicity of computing the required basis of sine and cosine functions. Furthermore, it is sometimes assumed that these shape functions may perform particularly well in dynamic (vibration, wave propagation) problems, where analytical solutions typically involve a superposition of harmonic functions which are solutions to the linear wave equation. Regarding the latter assumption, Fourier shape functions in our experience do not live up to the expectations. Except for those cases where the solution can be represented exactly by a truncated Fourier series, an approximation by harmonic functions does not seem to be superior to complete polynomials. Also, the ease of computing the harmonic basis functions comes at a cost: Using a standard Fourier series would lead to several shape functions being nonzero at the element limits (the cosine terms in our notation) which would make assembling elements and applying Dirichlet boundary conditions overly cumbersome. To circumvent this problem, two variants of Fourier shape functions can be found in the literature and are summarized as follows:

### 3.3.1 Complete (Node-Based) Fourier Shape Functions

He et al. [27] suggested using the harmonic basis given in Eq. (18) to create node-based shape functions by enforcing the Kronecker-delta-property at uniformly distributed points throughout the element. First, the basis of interpolants corresponding to Eq. (18) is written as

$$\psi_i(s) = \begin{cases} \frac{L-s}{L} & i = 1 \\ \cos\left(\frac{(i-1)\pi}{L}s\right) & 2 \le i \le m+1 \\ \sin\left(\frac{(i-1-m)\pi}{L}s\right) & m+2 \le i \le 2m+1 \\ \frac{s}{L} & i = 2m+2 \end{cases} \tag{19}$$

It should be noted that since the total number of basis functions is $n = 2m + 2$, we will only deal with even numbers of shape functions for this type of interpolation.[3] The basis provided in Eq. (19) is then used to construct a set of shape functions that fulfill the Kronecker-delta-property at uniformly distributed points, denoted as $s_i$. For this purpose, a transfer matrix $\mathbf{T}$ is assembled as



**(a)** Shape functions

**(b)** Derivatives

**Fig. 3** Complete Fourier shape functions, $n = 6$

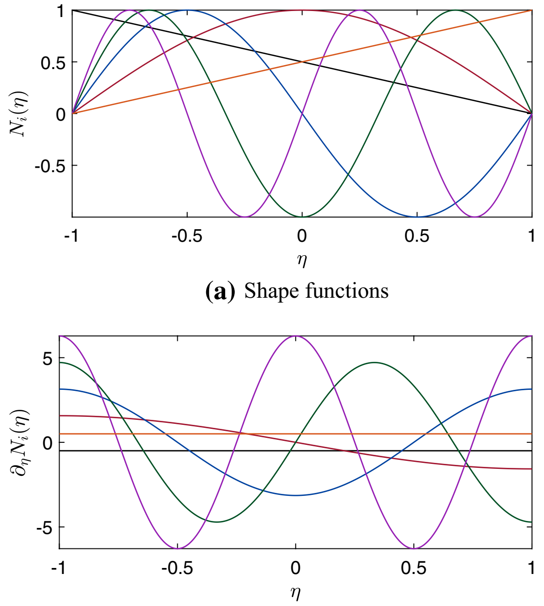
$$T_{ij} = \psi_j(s_i) \quad i,j = 1, 2, \dots, 2m+2. \tag{20}$$

The node-based shape functions follow as

$$\mathbf{N}(s) = \boldsymbol{\psi}(s)\mathbf{T}^{-1}\,. \tag{21}$$

Figure 3 shows the resulting shape functions for a 6-noded element. While this approach does work, it requires the solution of a system of equations and breaks the hierarchial property of the basis. Furthermore, as the shape functions are not polynomial, we may require more integration points to obtain accurate results. Hence, it is not obvious whether such a basis should be preferred over a polynomial one. While in Ref. [27], a relatively fast convergence is demonstrated when increasing the number of terms in the Fourier series (similar to $p$-refinement), the results are only compared against $h$-refinement using conventional low-order (linear and quadratic) Lagrange shape functions. For a fair comparison, we shall contrast the results we obtained by this approach with other interpolations using $p$-refinement.

[3] While it would generally be possible to omit one of the sine or cosine terms to create an odd-numbered basis of shape functions, this would lead to ill-conditioning in the steps that follow.



**(a)** Shape functions

**(b)** Derivatives

**Fig. 4** Incomplete Fourier shape functions, $n = 6$

### 3.3.2 Incomplete (Hierarchic) Fourier Shape Functions

Another approach to avoiding the problem of creating connectivity of multiple shape functions at the element limits simply ignores the cosine terms of the Fourier series altogether. The remaining linear and sine functions compose a set of hierarchical shape functions which are straightforward to compute. This approach is only rarely used, and the main area of application is vibration analysis [38, 39, 45]. To our knowledge, these shape functions have not been applied in the context of the SBFEM; however, they are included in this discussion since they constitute a simple modification of the aforementioned harmonic interpolation. We will refer to these interpolants as *incomplete Fourier shape functions* throughout this paper. Omitting the cosine terms in Eq. (19), the remaining functions constitute a hierarchical set of shape functions, i.e.,

$$N_i(s) = \begin{cases} \frac{L-s}{L} & i = 1 \\ \sin\left(\frac{(i-1)\pi}{L} s\right) & 2 \le i \le m+1 \\ \frac{s}{L} & i = m+2 \end{cases} \tag{22}$$

Figure 4 shows the first six shape functions of this class. In principle, the same disadvantages as mentioned in Sect. 3.2 apply. The most appealing advantage of using this kind of function is its simplicity. It can be implemented without any difficulties in existing codes. However, due to the incomplete ansatz, the attainable convergence rate is lower compared to that of high-order polynomials [12].

## 3.4 B-splines/NURBS

Splines provide another class of interpolants that can be used to construct trial and test spaces in the framework of the FEM. Most commonly, so-called basis splines or *B-splines* are employed which are piece-wise polynomial functions of one parameter, say $\zeta$. In our application, $\zeta$ will be used as a parametrization of the boundary, but it is not necessarily identical to the standard local coordinates $\eta$ or $s$. A set of spline basis functions is defined by their polynomial order $p$ and a set of $\zeta$-values, so-called *knots*:

$$\boldsymbol{\zeta} = \left\{\zeta_1, \zeta_2, \ldots, \zeta_{n+p+1}\right\} \tag{23}$$

Constructing a spline of order $p$ requires at least $p+2$ knots—not all of which need to be distinct. Repeating a knot $k$ times results in the spline basis functions to be of class $C^{p-k}$, i.e., they are $p-k$ times continuously differentiable at this particular knot. As an example, to construct a set of 6 basis functions of order 5 on the interval $[-1, 1]$, we can use the knot vector

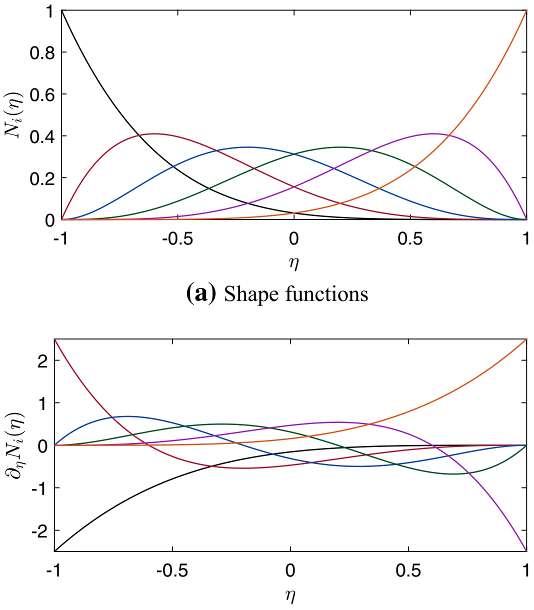
$$\boldsymbol{\zeta} = \{-1, -1, -1, -1, -1, -1, +1, +1, +1, +1, +1, +1\}\,. \tag{24}$$

For a given knot vector, the corresponding set of basis functions $B_i^p(\zeta)$ can be computed recursively using the well-known Cox-de Boor formula:

$$\begin{aligned} p = 0\ : \qquad & B_i^0(\zeta) = \begin{cases} 1 & \text{if } \zeta_i \le \zeta < \zeta_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ p > 0\ : \qquad & B_i^p(\zeta) = \frac{\zeta - \zeta_i}{\zeta_{i+p} - \zeta_i} B_i^{p-1}(\zeta) \\ & \qquad + \frac{\zeta_{i+p+1} - \zeta}{\zeta_{i+p+1} - \zeta_{i+1}} B_{i+1}^{p-1}(\zeta) \end{aligned} \tag{25}$$

Details can be found in the literature on computer-aided design (CAD) or isogeometric analysis (IGA), e.g., in Refs. [28, 50].

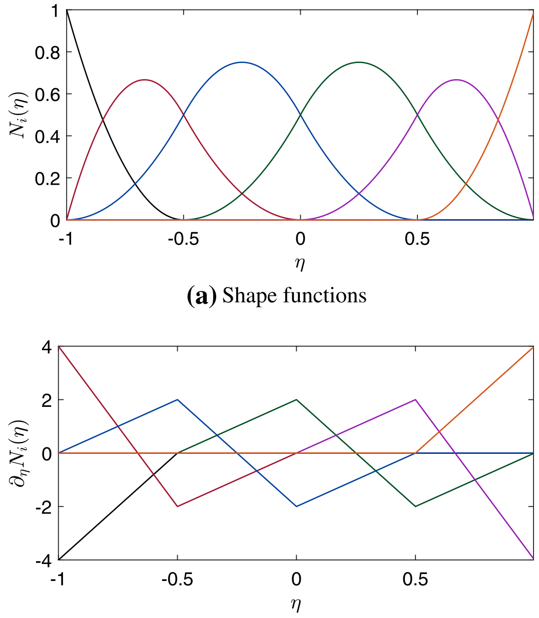
***Remark 2*** Since the literature usually presents the Cox-de Boor equation in this compact form, it may be worthwhile noting that two important pieces of information are commonly omitted: Firstly, the number of basis functions reduces by one in each step of the iteration. Hence, if $m$ is the number of knots, the index variable $i$ is given as $i = 1\ \ldots\ m - (p+1)$. In our example, Eq. (24), the 12 knots will thus yield 6 splines of order 5 as we would expect for a complete set of shape functions. Secondly, if knots are repeated, some of the terms in Eq. (25) may be undefined, as they contain a term of the form $1/(\zeta_{i+p} - \zeta_i)$ with $\zeta_{i+p} = \zeta_i$. These terms are set to zero. In other words, Eq. (25) generally implies that "all undefined terms are ignored" to avoid distinguishing different cases.



**(a)** Shape functions

**(b)** Derivatives

**Fig. 5** B-splines, $n = 6$



**(a)** Shape functions

**(b)** Derivatives

**Fig. 6** B-splines, $p = 2, n = 6$

Such a complete set of basis functions can be employed as shape functions in any FEM or SBFEM framework. Figure 5 presents those functions for the knot vector given by Eq. (24). Due to the repeated knots at the interval limits, the interpolated solution will be $C^0$-continuous and interpolatory at these points. Hence, such finite elements based on spline shape functions can be coupled directly in order to perform $h$-refinement. For a given element, $p$-refinement can be performed while increasing the multiplicity of the interval's endpoints in the knot vector. On the other hand, splines offer another type of refinement that has no direct analog in other finite element approaches: By inserting (unrepeated) knots within the interval for a higher order spline, we increase the number of shape functions of the same order without reducing the continuity at these additional points. For instance, the knot vector

$$\zeta = \left\{-1, -1, -1, -\tfrac{1}{2}, 0, +\tfrac{1}{2}, +1, +1, +1\right\} \tag{26}$$

leads to 6 basis functions of order 2, see Fig. 6. Inasmuch as these shape functions are continuously differentiable within the element, this so-called $k$-refinement is not directly comparable to neither $h$- nor $p$-refinement in the conventional finite element context.

Notwithstanding the fact that B-spline basis functions can directly be used as shape functions in the FEM or SBFEM, there are strong arguments for working instead with a generalization called *non-uniform rational B-splines (NURBS)*. NURBS are rational functions and are derived from a basis of B-splines as

$$N_i(\zeta) = \frac{B_i^p(\zeta) w_i}{\sum\limits_{j=1}^{n} B_j^p(\zeta) w_j} \tag{27}$$

with weight factors $w_i > 0$. The motivation for using NURBS instead of simple B-splines mainly stems from their flexibility in representing the geometry (rather than their capabilities in interpolating the displacement field). The advantages of NURBS are:

1. By choosing adequate weight factors $w_i$, certain geometries like circular arcs and ellipses can be represented exactly. This is demonstrated for the case of a circular arc in Fig. 7 where a NURBS representation is compared to an interpolation using quadratic Lagrange polynomials. While NURBS are capable of representing the boundary exactly, quadratic polynomials introduce small errors in the geometry approximation that may affect the convergence.
2. NURBS are commonly used in computer-aided design. Hence, if the geometry to be modeled has been created in a CAD software, the geometry definition can, in principle, be used directly to define the trial and test functions of the weak form. This concept is the core idea of isogeometric analysis (IGA).

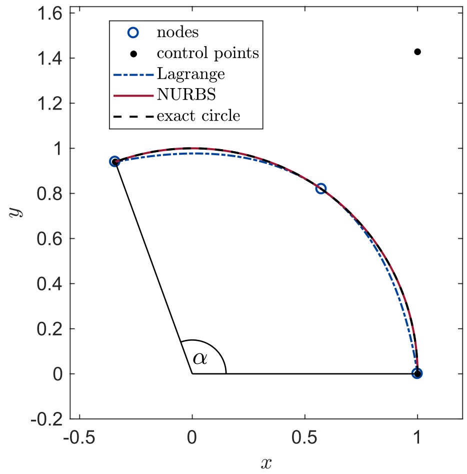


**Fig. 7** Representation of a circular arc ($\alpha = 110°$) using NURBS and Lagrange interpolation of order 2

On the downside, NURBS, in general, require a larger number of integration points compared to polynomial shape functions—except for the case where all $w_j$ equal 1, i.e., NURBS are identical to B-splines. Like in any other finite element approximation, the geometry is interpolated based on a set of given Cartesian coordinates, referred to as control points:

$$\mathbf{x}(\zeta) = \sum_{i=1}^{n} N_i(\zeta)\mathbf{x}_i \,. \tag{28}$$

However, since the weights determine the geometry representation, they are typically stored with the coordinates:

$$\mathbf{P}_i = \left[x_i, y_i, z_i, w_i\right]^{\mathrm{T}} = \begin{bmatrix}\mathbf{x}_i \\ w_i\end{bmatrix} \qquad i = 1, \ldots, n \tag{29}$$

Whether the IGA paradigm can be exploited effectively depends on the problem statement, in particular, how the geometry is defined. We shall not veer into a discussion of the pros and cons of this approach and refer instead to the abundance of literature on this matter. Within the scope of the SBFEM, the use of NURBS shape functions has been reported for classical elastostatic problems [41] and linear fracture mechanics [47]. It was demonstrated that $k$-refined NURBS can lead to a faster convergence than $h$-refined Lagrange elements. This can, of course, be ascribed to the exact geometry representation and higher continuity when using NURBS. In Ref. [20], NURBS shape functions have been employed for the simulation of guided elastic waves. The results therein indicate that while it may not be advisory to employ B-splines of a very high order, $k$-refinement in conjunction with B-splines of a moderately large order (say 6) can at least in some cases outperform $p$-refined Lagrange elements. Furthermore, it has been found that NURBS tend to be somewhat more robust at high frequencies, for the special case studied there. Variants of the SBFEM with NURBS shape functions are also used in Refs. [42, 43, 61].[4] In another modification of the SBFEM, the semi-analytical paradigm has been replaced with a fully numerical model by employing collocation or a Galerkin-based weak form in the radial direction [6–8, 37].

### 3.5 Element-Free Galerkin Scaled Boundary Method

A rather different type of interpolation is utilized in the *element-free Galerkin scaled boundary method*. This variant was introduced in an attempt to incorporate features of meshless methods in the SBFEM. He et al. [26] suggested the element-free Galerkin approach as an improvement to the previously used *meshless Petrov-Galerkin* technique [10], which according to the authors is not always robust and results in a non-symmetric stiffness matrix. The element-free Galerkin approach has been implemented along the lines of the work presented by Belytschko [4].

Before summarizing the formulation of this type of interpolation, we shall discuss the motivation for this work. The core idea of meshless methods is to avoid the difficult task of decomposing the computational domain into non-overlapping elements and determining their connectivity. Instead, the domain is described by a number of points (referred to as nodes like in finite elements) which are associated with an interpolant with local support. The interpolating functions are somewhat similar to shape functions in conventional finite element approximations; however, they can overlap in virtually arbitrary ways, thus removing the necessity to specify a particular connectivity for these points. The pros and cons of this general approach in various applications are discussed in great detail in the literature. On the other hand, the reader may wonder how this concept applies in the context of the SBFEM. At least in two dimensions, we only require interpolation along the boundaries of subdomains. These boundaries need to be parametrized in order to formulate the scaled boundary coordinate transformation and perform the numerical integration. Hence, employing a “meshless” approach would imply only that we may distribute nodes arbitrarily along the boundary. However, once the parametrization of each subdomain’s boundary is established, the placement of nodes is a rather straightforward task. In Ref. [26], the nodes are distributed uniformly

[4] While other authors refer to the combination of IGA and SBFEM as *SBIGA* [41], *SIGA* [42, 43], or *IGA-SBFEM* [61] we will stick to the term *SBFEM* since in the context of this discussion the approaches differ only in the choice of shape functions.

along each boundary section. In other words, the "meshless property" of this approach is not exploited, and thus this type of interpolation merely constitutes a different type of shape function that can be compared directly against the other methods summarized before.

Roughly speaking, the interpolation in this approach is obtained by (low order) polynomials that are modified by a weight function. The weight function determines the size of the local support for the nodes as well as the smoothness of the interpolation. The shape functions are computed by a moving least squares (MLS) formulation. Let $\mathbf{p}(s)$ denote a complete monomial basis in the local coordinate $s$—for instance $\mathbf{p}(s) = [1, s]^{\mathrm{T}}$ or $\mathbf{p}(s) = [1, s, s^2]^{\mathrm{T}}$ for a linear or quadratic basis, respectively. Furthermore, $w_i(s)$ is a weight function associated with node $i$ (different types of weight functions will be presented below). The set of interpolation functions is computed as

$$\boldsymbol{\psi}(s) = \mathbf{p}(s)^{\mathrm{T}}\mathbf{A}(s)^{-1}\mathbf{B}(s) \tag{30}$$

with

$$\mathbf{A}(s) = \sum_{i=1}^{n} w_i(s)\mathbf{p}(s_i)\mathbf{p}^{\mathrm{T}}(s_i) \tag{31}$$

and

$$\mathbf{B}(s) = \left[w_1(s)\mathbf{p}(s_1),\ w_2(s)\mathbf{p}(s_2),\ \ldots,\ w_N(s)\mathbf{p}(s_n)\right] \tag{32}$$

where $s_i$ denote the nodal coordinates. Since in meshless methods, the interpolation functions may overlap more or less arbitrarily, they will not form a partition-of-unity nor fulfill the Kronecker-delta property. In Ref. [26], the authors construct a set of nodal shape functions using the same approach as for the case of Fourier shape functions (cf. Sect. 3.3.1), thus we obtain[5]

$$\mathbf{N}(s) = \boldsymbol{\psi}(s)\mathbf{T}^{-1} = \mathbf{p}(s)^{\mathrm{T}}\mathbf{A}(s)^{-1}\mathbf{B}(s)\mathbf{T}^{-1} \tag{33}$$

with

$$T_{ij} = \psi_j\left(s_i\right) \quad i,j = 1, 2, \ldots, n\,. \tag{34}$$

The spatial derivatives of the shape functions are computed as

$$\mathbf{N}(s)_{,s} = \boldsymbol{\psi}(s)_{,s}\mathbf{T}^{-1} \tag{35}$$

with

$$\boldsymbol{\psi}_{,s} = \mathbf{p}_{,s}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^{\mathrm{T}}(\mathbf{A}_{,s}^{-1}\mathbf{B} + \mathbf{A}^{-1}\mathbf{B}_{,s}) \tag{36}$$

and

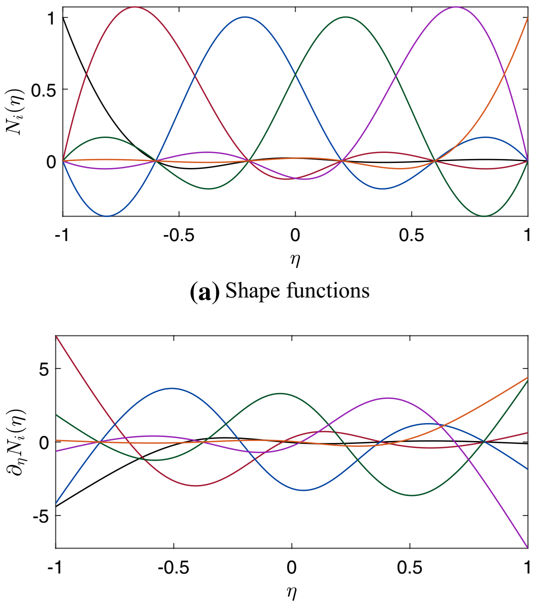
$$\mathbf{A}_{,s}^{-1} = -\mathbf{A}^{-1}\mathbf{A}_{,s}\mathbf{A}^{-1}\,. \tag{37}$$

The remaining question is how to choose the weight functions $w_i(s)$, which determine not only the domain of influence (the size of the local support) but also the smoothness of the shape functions. If we were to use piece-wise constant weight functions on each edge, we would recover the standard finite element shape functions. Commonly used weight functions include

1. Conical weight function:
$$w_i(s) = \begin{cases} 1 - (d_i/r_i)^2, & d_i \le r_i \\ 0, & d_i > r_i \end{cases}$$
2. Exponential weight function:
$$w_i(s) = \begin{cases} \frac{\exp[-(d_i/c_i)^2] - \exp[-(r_i/c_i)^2]}{1 - \exp[-(r_i/c_i)^2]}, & d_i \le r_i \\ 0, & d_i > r_i \end{cases}$$
3. Spline weight function:
$$w_i(s) = \begin{cases} 1 - 6\left(\frac{d_i}{r_i}\right)^2 + 8\left(\frac{d_i}{r_i}\right)^3 - 3\left(\frac{d_i}{r_i}\right)^4, & d_i \le r_i \\ 0, & d_i > r_i \end{cases}$$

where $d_i(s) = |s - s_i|$ denotes the distance from the $i$th node; $r_i$ is the size of the local support, and $c_i$ is a constant that determines the decay of the exponential weight function. Generally, it can be a difficult task to choose optimal values for the parameters $r_i, c_i$. On the other hand, since we are not exploiting the meshless features in the SBFEM we shall for now choose $r_i = 1$, thus selecting the whole edge as the domain of influence. This will result in all nodes along each edge to be coupled, similar to what happens when performing $p$-refinement with other types of shape functions. Furthermore, we will choose $c_i = r_i/4$ as recommended in Ref. [68]. In the context of the SBFEM, only the spline weight function has been tested by He et al. [26]. Belytschko et al. compared exponential and conical weight functions and concluded that the exponential function "performs far better than a conical weight function" [4]. Zhu et al. used both Gaussian and spline weight functions within a boundary element framework and found the Gaussian weight functions to be superior [68]. Other authors used slightly different versions of the aforementioned weight functions, see e.g. [40, 49]. We implemented and tested all three of the weight functions presented above. However, we also found the exponential weight functions to lead to the smallest errors in our examples and hence, we will only include those results in the comparison. The shape functions for order 6 using the exponential weight functions are exemplarily depicted in Fig. 8.

[5] It can be noted that—compared to Fourier shape functions—the basis functions $\boldsymbol{\psi}(s)$ are relatively expensive to compute, which may be a drawback when constructing a nodal basis of high order.



**(a)** Shape functions

**(b)** Derivatives

**Fig. 8** EFG shape functions (moving least squares), $n = 6$

### 3.6 Diagonal SBFEM

A very peculiar type of shape functions has been proposed by Khodakarami et al. and is applied in a series of papers on the so-called 'diagonal SBFEM' [31–36, 46]. To be clear: We do *not* recommend these shape functions to be employed in a general SBFEM framework (or any other finite element based approach). They are included in the context of this discussion merely for completeness as well as to shed some light on an approach that has puzzled many researchers who have worked on the SBFEM. We will refer to these shape functions—for want of a better name—as 'diagonal shape functions,' as they have been constructed with the purpose of creating diagonal coefficient matrices in the SBFEM.

The underlying idea is the following: Remember that when employing spectral elements, the concept of nodal quadrature requires evaluating the shape functions exclusively at each node, where only one of the shape functions has a nonzero value (*Kronecker-delta-property*) [51]. This leads to the mass matrix—or in the case of SBFEM the matrices $\mathbf{M}_0$ and $\mathbf{E}_0$—being diagonal. The matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, on the other hand, are not diagonal since they involve the derivatives of shape functions. It may now seem like an obvious idea to come up with shape functions whose derivatives either also fulfill the Kronecker-delta-property or (as Khodakarami et al. opted for in the diagonal SBFEM) vanish at *all* nodes. This would cause all coefficient matrices to be diagonal if not zero. Do such shape functions exist? Yes and no. It is, of course, possible to construct polynomials that exhibit the mentioned properties, and this is precisely what has been done in the papers on the diagonal SBFEM. On the other hand, such polynomials will not be promising candidates to be used as general shape functions.

Shape functions of this type can be constructed for a given configuration of nodes by choosing a sufficiently large basis of polynomials and enforcing the aforementioned conditions at the nodes. If the element is defined by $n$ nodes, there are in total $2n$ conditions for each shape function and its first derivative, which requires a polynomial basis of order $2n - 1$. Hence, assuming polynomials of the form

$$N_i(\eta) = \sum_{k=0}^{2n-1} a_k^{(i)} \eta^k \tag{38}$$

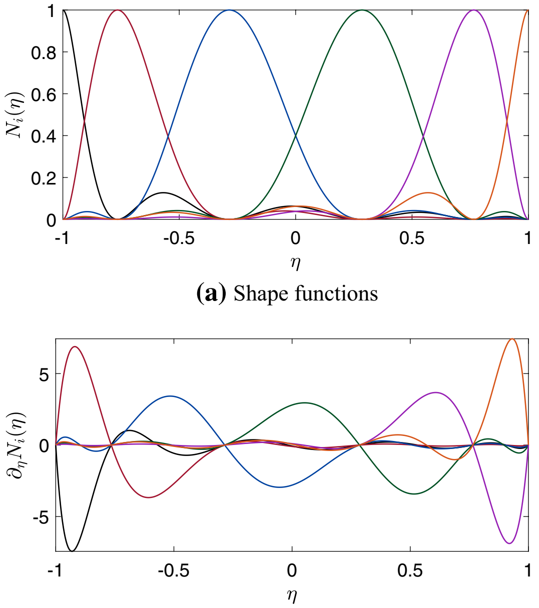
the coefficients $a_k^{(i)}$ of the $i$th shape function are obtained by solving the linear system of equations

$$\left[\begin{array}{cccccc} 1 & \eta_1 & \eta_1^2 & \dots & & \eta_1^{2n-1} \\ 1 & \eta_2 & \eta_2^2 & \dots & & \eta_2^{2n-1} \\ \vdots & & & \vdots & & \\ 1 & \eta_n & \eta_n^2 & \dots & & \eta_n^{2n-1} \\ \hdashline 0 & 1 & 2\eta_1 & \dots & & (2n-1)\eta_1^{2n-2} \\ 0 & 1 & 2\eta_2 & \dots & & (2n-1)\eta_2^{2n-2} \\ \vdots & & & \vdots & & \\ 0 & 1 & 2\eta_n & \dots & & (2n-1)\eta_n^{2n-2} \end{array}\right] \left[\begin{array}{c} a_0^{(i)} \\ a_1^{(i)} \\ \vdots \\ a_{n-1}^{(i)} \\ \hdashline a_n^{(i)} \\ a_{n+1}^{(i)} \\ \vdots \\ a_{2n-1}^{(i)} \end{array}\right] = \mathbf{e}_i \tag{39}$$

where $\eta_i$ denote the nodal positions and $\mathbf{e}_i$ is a unit vector with the $i$th component equal to 1. The first $n$ equations enforce the Kronecker-delta-property of the shape functions while the remaining equations ensure that the derivatives vanish at all nodes. Once the coefficients are computed for each polynomial, these shape functions can, in principle, be employed in the conventional SBFEM formulation. If, in addition, a nodal quadrature rule is applied as suggested in Refs. [31–36, 46], the SBFEM coefficient matrices will indeed be diagonal. Furthermore, the matrix $\mathbf{E}_2$ vanishes (as it only involves the derivatives of the shape functions at the nodes), leading to the modified SBFEM equation

$$\xi \tilde{\mathbf{E}}_\mathbf{0} \mathbf{u}_{\mathbf{n},\xi\xi} + \tilde{\mathbf{E}}_\mathbf{1} \mathbf{u}_{\mathbf{n},\xi} + \xi \mathbf{F}_\mathbf{b} = 0 \tag{40}$$

where $\mathbf{F}_\mathbf{b}$ denotes a vector of body loads and $\tilde{\mathbf{E}}_\mathbf{0}$, $\tilde{\mathbf{E}}_\mathbf{1}$ are the diagonal coefficient matrices that we distinguish from the general (fully integrated or weakly under-integrated) ones in the previous sections. The diagonal shape functions derived for 6 nodes are exemplarily depicted in Fig. 9.



**(a)** Shape functions

**(b)** Derivatives

**Fig. 9** Shape functions used in the 'diagonal SBFEM', $n = 6$

***Remark 3*** When employing nodal quadrature, the matrix $\tilde{\mathbf{E}}_\mathbf{1}$ vanishes unless an element on the boundary is curved.[6] Thus, for the case of straight elements and in the absence of body loads, Eq. (40) reduces to

$$\xi \tilde{\mathbf{E}}_\mathbf{0} \mathbf{u}_{\mathbf{n},\xi\xi} = 0 \tag{41}$$

which is solved by linear functions in $\xi$. Keeping in mind that $\tilde{\mathbf{E}}_\mathbf{0}$ is diagonal, Eq. (41) constitutes a set of equations of the form $E_i u_{i,\xi\xi} = 0$. Hence, roughly speaking, the diagonal formulation can be interpreted as replacing the continuum by a set of springs connecting the nodes with the scaling center.

We would like to stress again that in the current contribution we are solely interested in the shape functions and their applicability in the general SBFEM without discussing in too much detail other particularities of the diagonal SBFEM. In the system of ODEs (41), the DOFs are decoupled; hence, they move independently. This behavior obviously does not represent the physical system to be modeled; thus, the straightforward solution of this system of equations does not yield very meaningful results. To overcome this problem, additional steps are introduced in the solution procedure of the diagonal SBFEM, which (in our understanding) involve the computation of stresses at the scaling center due to external loads, interpolating these stresses over the domain, reinterpreting those interpolated stresses as body loads, and finally solving Eq. (40) *again* while considering these body loads in the vector $\mathbf{F}_\mathbf{b}$. These additional steps are omitted in most publications dealing with the diagonal SBFEM and are only briefly mentioned in Ref. [33]. The details, however, remain obscure to us and we were, unfortunately, unable to reproduce any of the results presented in the previous publications [31–36, 46].

Nevertheless, we can—and we should—have a closer look at the proposed shape functions. We shall keep in mind that we always have the option to fully integrate the coefficient matrices by using a sufficient number of quadrature points. This will lead to fully populated (i.e., coupled) systems such that the same solution procedures can be applied that are well-known to work in the SBFEM. Regarding the 'diagonal' shape functions, there exist, in our view, the following main issues:

1. The polynomial basis obtained from solving Eq. (39) is incomplete. If the element consists of $n$ nodes, there will be only $n$ shape functions of order $2n - 1$. Consequently, not every polynomial of order $2n - 1$ can be written as a superposition of such shape functions.
2. To obtain the desirable feature of diagonal coefficient matrices, nodal quadrature has to be employed. In doing so, the shape functions are very severely under-integrated. For instance, the matrix $\mathbf{E}_0$ contains the products of shape functions which are polynomials of order $4n - 2$, while the $n$ point GLL quadrature (as employed in [33]) is only exact for polynomials up to an order of $2n + 1$. Instead of GLL integration, a Clenshaw-Curtis quadrature rule is employed in several other publications. In this case, the issue of under-integration is even more severe, since Clenshaw-Curtis quadrature only integrates polynomials up to a degree of $n - 1$ exactly. In addition, the Jacobian $|J(\eta)|$ may be non-constant, which deteriorates the accuracy of the quadrature even further.

To clarify, the former of these issues does in itself not necessarily guarantee that these shape functions are bound to fail since incomplete polynomials could in general still lead to converging interpolations with the convergence rate being determined by the highest *complete* polynomial. However, in the case of the diagonal shape functions, the severity of this issue can be demonstrated by a simple example: Consider a 2-noded element, in which case the shape functions are given as two polynomials of order 3 (see also Table 1 in Ref. [34]):

$$N_1 = 0.5 - 0.75\eta + 0.25\eta^3 \tag{42}$$

[6] This is because $\tilde{\mathbf{E}}_\mathbf{1}$ involves second derivatives of the Cartesian coordinates w.r.t. $\eta$ (see Eqs. (36), (20), (11) in [34]), and the Coordinates are interpolated using standard finite element shape functions. All other terms in $\tilde{\mathbf{E}}_\mathbf{1}$ involve derivatives of the diagonal shape functions and thus vanish when nodal quadrature is applied.

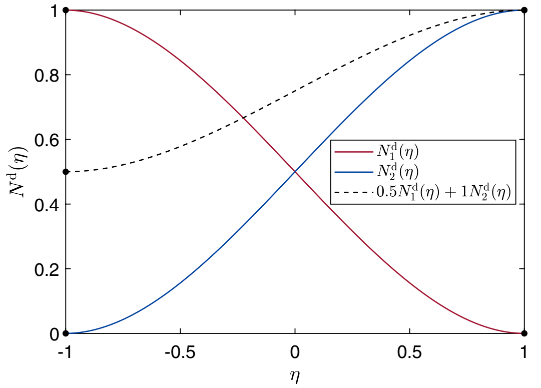


**Fig. 10** Shape functions of a 2-noded element as used in the 'diagonal SBFEM'. The solution between the nodes is interpolated by a cubic function (dashed line)

$$N_2 = 0.5 + 0.75\eta - 0.25\eta^3 \tag{43}$$

The shape functions are plotted in Fig. 10 together with an example of an interpolation between two arbitrary nodal values. Obviously, a superposition of the form $u = u_1N_1 + u_2N_2$ cannot represent an arbitrary complete cubic or quadratic function, as the shape functions do not involve any quadratic terms. But more importantly, the shape functions also cannot represent linear functions because of the additional cubic terms. The only case where the cubic terms cancel out is obtained for $u_1 = u_2$, in which case the resulting function is constant over the element. For this reason, this type of shape functions is not even capable of representing a constant strain field (linear displacement), which would be a requirement to pass the simplest patch test.

***Remark 4*** Since these sets of polynomials lack the most fundamental properties that would justify their use as valid shape functions, the reader may wonder whether the diagonal ODE (41) is at all meaningful or does at least represent a comprehensible approximation. To answer this, let us look at this approach from a different angle and focus on the way these shape functions are integrated to obtain the coefficient matrices. Similar to the concept of spectral elements detailed in Sect. 3.1, the nodal quadrature technique is applied, which means that the shape functions and their derivatives are only evaluated *at the nodes*. Think about what this means: Each shape function is constructed such that it equals one at the corresponding node, where simultaneously all other shape functions vanish (*Kronecker-delta-property*). The standard spectral elements—and in general all node-based elements for that matter—share the same property. Consequently, integrating the shape functions (not their derivatives!) by nodal quadrature gives *exactly the same results*, no matter if we assume conventional Lagrange interpolation polynomials or the shape functions proposed for the diagonal SBFEM.[7] Hence, if we use the same nodal positions and the same quadrature scheme as for conventional spectral elements, the diagonal SBFEM will compute the identical $\mathbf{E_0}$ and $\mathbf{M_0}$ matrices as they only involve integrations of products of shape functions.[8] The derivatives $\frac{\mathrm{d}}{\mathrm{d}\eta}N(\eta)$ of the shape functions, on the other hand, vanish at all nodes by construction, which is in contrast to all valid shape functions that are known to us. Hence, employing nodal quadrature leads to all terms in $\mathbf{E}_1$ and $\mathbf{E}_2$ that contain these derivatives to vanish. In other words, instead of constructing the 'diagonal' shape functions with all their confusing properties, we can reinterpret this approach as employing conventional nodal shape functions and assuming their derivatives to vanish. Consequently, Eq. (40) provides an approximation in which the degrees of freedom are decoupled simply by neglecting the derivative of the solution in the direction along the boundary.

## 4 Numerical examples

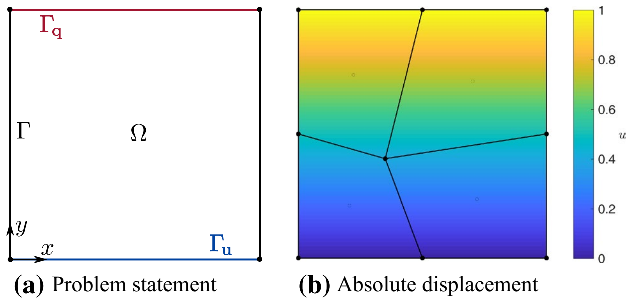
In this section, selected examples are presented to assess and compare the performances of the different shape functions in the framework of the SBFEM. We shall restrict ourselves to rather basic benchmark examples that are simple enough for the results to be reproduced while still highlighting the different features of the various interpolants that we compare. The benchmark examples are also chosen such that the results provide an unbiased comparison without involuntarily favoring one or the other approach which can easily happen when arbitrarily choosing a more complex geometry and problem statement.

### 4.1 Square Geometry

We begin by analyzing a simple geometry, namely a square of unit size, and perform several quite standard numerical experiments to evaluate the properties of the different high order interpolations. The general problem statement is shown in Fig. 11a. The computational domain is

[7] This argument, of course, assumes that both approaches use the same nodal positions and quadrature scheme. In Ref. [33], GLL-points are used to construct the shape functions just like in SEM. In most other publications on the diagonal SBFEM, GLC points are employed which would be a possible yet highly uncommon choice in other approaches.

[8] The interested reader is referred to Ref. [14], in particular Sect. 4.1, which contains a similar argument to demonstrate the equivalence of different mass lumping techniques.



**(a)** Problem statement **(b)** Absolute displacement

**Fig. 11** Square geometry: **a** problem statement, **b** mesh used for the patch test and displacement solution

$$\Omega = \left\{ (x, y) \in \mathbb{R}^2 \;\middle|\; 0 \le x \le 1,\ 0 \le y \le 1 \right\} \tag{44}$$

The boundary of the domain is denoted by $\Gamma$, while $\Gamma_u$ and $\Gamma_q$ refer to those parts of the boundary where Dirichlet and nonzero von-Neumann boundary conditions are applied.

### 4.1.1 Patch Test

For a linear patch test, we choose the mesh depicted in Fig. 11b and solve the boundary value problem[9]

$$\Delta u = 0 \qquad \text{in } \Omega \tag{45}$$

$$u = 0 \qquad \text{on } \Gamma_u \tag{46}$$

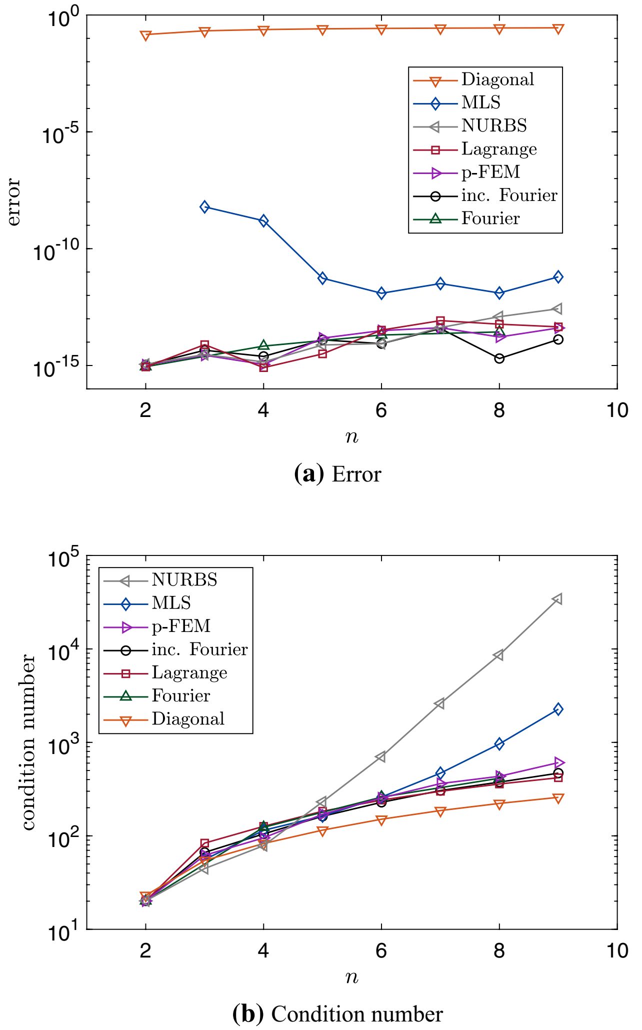
$$\frac{\partial u}{\partial n} = 1 \qquad \text{on } \Gamma_q \tag{47}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on } \Gamma \setminus (\Gamma_u \cup \Gamma_q) \tag{48}$$

with the exact solution

$$u(x, y) = y \tag{49}$$

(also shown in Fig. 11b). We compare the $L_2$ norm of the relative error in the nodal solutions when using different interpolations. Figure 12a shows the resulting errors when increasing the number of degrees of freedom $n$ on each edge. For most interpolants, this corresponds to $p$-refinement, i.e., an increase in the order of the shape functions. However, in the case of moving least squares (as used in the element-free Galerkin approach), the number of overlapping functions is increased while their order remains constant. On the other hand, since the resulting degrees of freedom are fully coupled on each edge, this type of refinement is effectively more similar to $p$- than to $h$-refinement in terms of implementation, sparsity of the coefficient matrices, and continuity of the trial space. In the case of Fourier shape functions, this type of refinement corresponds to increasing the number of harmonic terms (which, strictly speaking, is not $p$-refinement as $p$ denotes the order of a polynomial). Lastly, it should be noted that in the case of the 'diagonal' shape functions, the highest order of the (incomplete) polynomials is increased by two in each refinement step. Notwithstanding these differences between the interpolants, we shall refer to the type of refinement as described above generally as $p$-refinement for easier reference.



**(a)** Error

**(b)** Condition number

**Fig. 12** Square geometry, patch test: **a** $L_2$ norm of the relative displacement error for varying element order; **b** condition number of the stiffness matrix

The results demonstrate that Lagrange interpolation, $p$-FEM, Fourier shape functions, and NURBS in conjunction with the SBFEM pass the patch test. This is of course not surprising as all these types of shape functions include the linear terms and also the analytical solution in the radial

[9] Physically, this case corresponds to a steady-state heat conduction problem with unit thermal diffusivity in the absence of body loads.

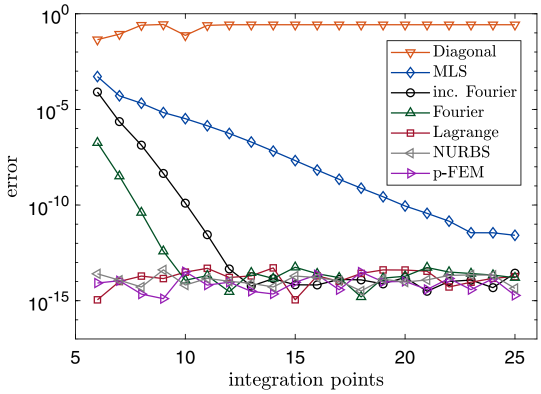


**Fig. 13** Square geometry, patch test: convergence of the relative error when increasing the number of integration points

coordinate $\xi$ is capable of representing a linear solution exactly. When employing moving least squares (MLS), the patch test is not passed exactly since the final shape functions are not polynomials; nevertheless, accurate results (errors between $10^{-7}$ and $10^{-12}$) are obtained. As anticipated, based on the discussion in Sect. 3.6, the shape functions used in the diagonal SBFEM do not pass the patch test and lead to false results. Since in all examples we tested, the errors resulting from using diagonal shape functions are of the order of 100% we will omit these results in the following sections and conclude immediately that these shape functions do not constitute valid interpolants.

Figure 12b presents the condition number of the stiffness matrix. With regard to this aspect, most of the tested interpolations show quite similar behavior. Only when using MLS or NURBS does the condition number increase much faster with the number of DOFs. In the case of NURBS, this problem can be avoided by applying $k$-refinement with splines of a moderately high order as will be demonstrated in the following sections.

Another essential difference between the classes of shape functions lies in the number of quadrature points required to integrate the SBFEM coefficient matrices. As an example, we plot the error obtained in the patch test against the number of integration points (standard Gauss quadrature is employed) for $n = 6$, see Fig. 13. When using Lagrange interpolation, $p$-FEM, and B-splines, the shape functions consist of polynomials of order $p = 5$. The coefficient matrices $\mathbf{E}_0, \mathbf{E}_1, \mathbf{E}_2$ require the integration of polynomials of order $2p$, $2p - 1$, $2p - 2$, respectively and are thus integrated exactly when using at least $p + 1$ integration points.[10] Fourier shape functions and MLS are not integrated exactly by a Gauss quadrature. Still, in the case of Fourier shape functions, the error converges to almost machine precision when using ten integration points. MLS shape functions require significantly more integration points to obtain fully converged solutions.

### 4.1.2 Modal Analysis

For our next test, we perform a modal analysis on the same square geometry as before. We consider the scalar wave equation with vanishing natural boundary conditions, i.e.,

$$\Delta u = \partial_{tt} u \qquad \text{in } \Omega \tag{50}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on } \Gamma\,. \tag{51}$$

This problem statement describes a rectangular closed acoustic cavity. The exact solution of the eigenfrequencies are given as

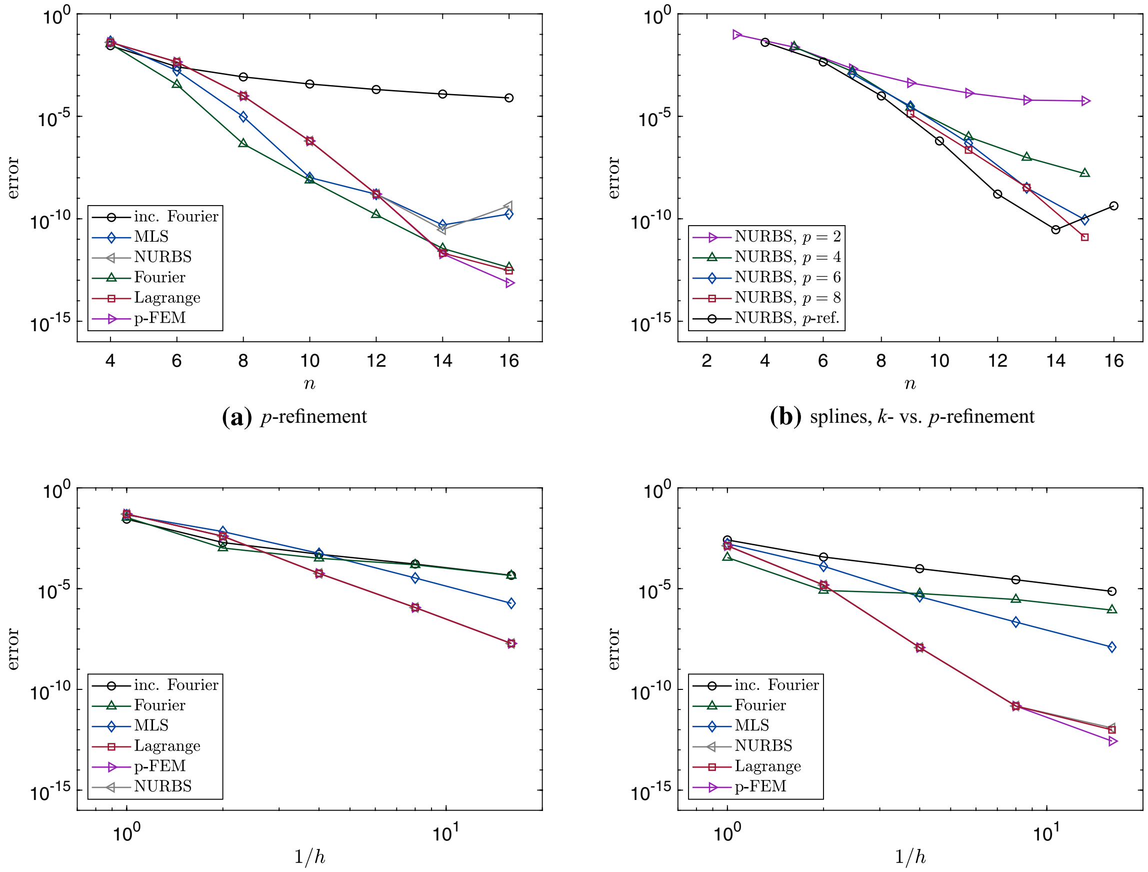
$$\omega_{mn} = 2\pi\left[\left(\frac{m}{2}\right)^2 + \left(\frac{n}{2}\right)^2\right]^{\frac{1}{2}}, \qquad m, n \in \mathbb{N}_0\,. \tag{52}$$

We apply the solution procedure based on continued fractions with preconditioning (Eq. (10), [5, 53]). To study the convergence behavior, we compute the first ten nonzero eigenfrequencies and analyze the $L_2$ norm of the relative error w.r.t. the exact solution. Figure 14a shows the resulting errors when performing $p$-refinement[11] on each of the domain's four edges. For this smooth problem, exponential convergence is achieved when employing a set of complete polynomials (such as Lagrange polynomials or $p$-FEM). The polynomial shape functions show optimal convergence to an error level of about $10^{-12}$ at which point small numerical errors accumulated in the continued fraction expansion begin to dominate the deviation from the reference solution. B-splines generally show a very similar trend compared with polynomial shape functions while the solution begins to deviate when increasing the order above 11. This finding is consistent with previous studies showing that it is generally not recommended to increase the order of B-splines arbitrarily but rather to employ $k$-refinement with a moderately large order, say 6 or 8 [20].

It is interesting to note that for this particular problem complete Fourier shape functions indeed result in slightly smaller errors compared to their polynomial counterparts—at least for moderately large element orders. This can be understood since, in the current test case, the exact solution

[10] Again, it should be noted that in this example, where the geometry consists of straight edges only, the NURBS shape functions reduce to simple B-splines.

[11] Again assuming the somewhat loose definition of $p$-refinement as discussed in Sect. 4.1.1.



**(a)** *p*-refinement

**(b)** splines, *k*- vs. *p*-refinement

**(c)** *h*-refinement, $p = 3$

**(d)** *h*-refinement, $p = 5$

**Fig. 14** Square geometry, modal analysis: relative error of the first 10 nonzero eigenfrequencies of the acoustic cavity

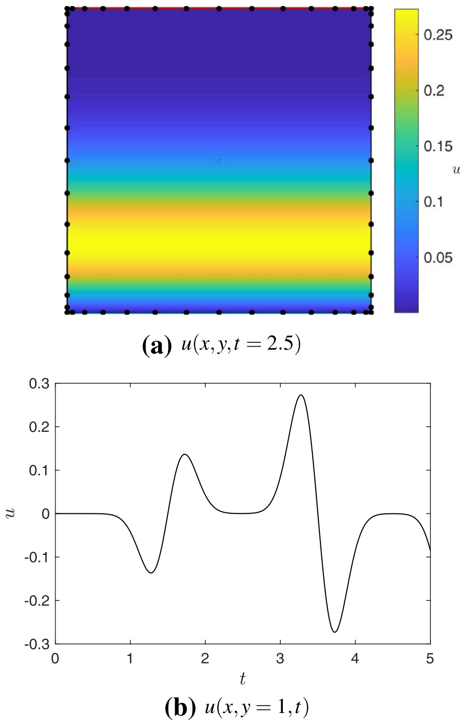
for each mode is given as a product of harmonic functions in the two spatial coordinates

$$u_{mn}(x, y) = A_{mn} \cos(\pi m x) \cos(\pi n y) \tag{53}$$

where $A_{mn}$ are arbitrary amplitudes. Thus, the solution along the $\eta$ direction can, in principle, be represented exactly by the Fourier shape functions. In fact, when employing Fourier shape functions in a conventional high order finite element method, this problem is solved to machine precision, provided that a sufficient number of terms are included to represent the highest mode of interest. On the other hand, since this special example constitutes the best case scenario for applying Fourier shape functions in the SBFEM, the improvement compared to polynomial shape functions is rather small and comes at the cost of requiring a larger number of integration points. The further examples will demonstrate that in more general cases, usually no gain in accuracy can be expected when using Fourier shape functions. In this example also the MLS shape functions show slightly better performance compared to polynomial shape functions up to 10 DOFs per edge. The incomplete Fourier shape functions lead to significantly slower convergence and hence cannot be recommended despite their convenient implementation. Again, using 'diagonal' shape functions did not yield correct results.

In Fig. 14b we present a comparison between *p*- and *k*-refinement when using NURBS (or B-splines in this case). Along each edge, *k*-refinement leads to a set of continuously differentiable shape functions of the given order *p*. Using *k*-refinement and an order of 8 leads to a convergence rate similar to *p*-refinement. The very similar behavior of $p = 6$ and $p = 8$ indicate that it is not advantageous to increase the order further.

Figure 14c, d illustrates the results of *h*-refinement. Here, the error is plotted against the inverse of the element length on the boundary *h*. Using Lagrange polynomials, *p*-FEM, or splines in the SBFEM, we obtain very similar convergence rates compared to a high order finite element formulation (not shown in the figure). Fourier shape functions and MLS lead to sub-optimal convergence rates.



**(a)** $u(x, y, t = 2.5)$

**(b)** $u(x, y = 1, t)$

**Fig. 15** Square geometry, transient analysis: solution and exemplary mesh

### 4.1.3 Transient Analysis

As a transient problem, we consider once more the scalar wave equation of the square domain where we choose a time-dependent load described by a Ricker pulse. The problem statement reads

$$\Delta u = \partial_{tt} u \qquad \text{in } \Omega \quad \text{for } 0 \leq t \leq t_{\max} \tag{54}$$

$$u = 0 \qquad \text{on } \Gamma_u \tag{55}$$

$$\frac{\partial u}{\partial n} = P(t) \qquad \text{on } \Gamma_q \tag{56}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on } \Gamma \setminus (\Gamma_u \cup \Gamma_q) \tag{57}$$

where we define

$$P(t) = (1 - 2\pi^2 f_0^2 (t - t_0)^2) \exp(-\pi^2 f_0^2 (t - t_0)^2) \tag{58}$$

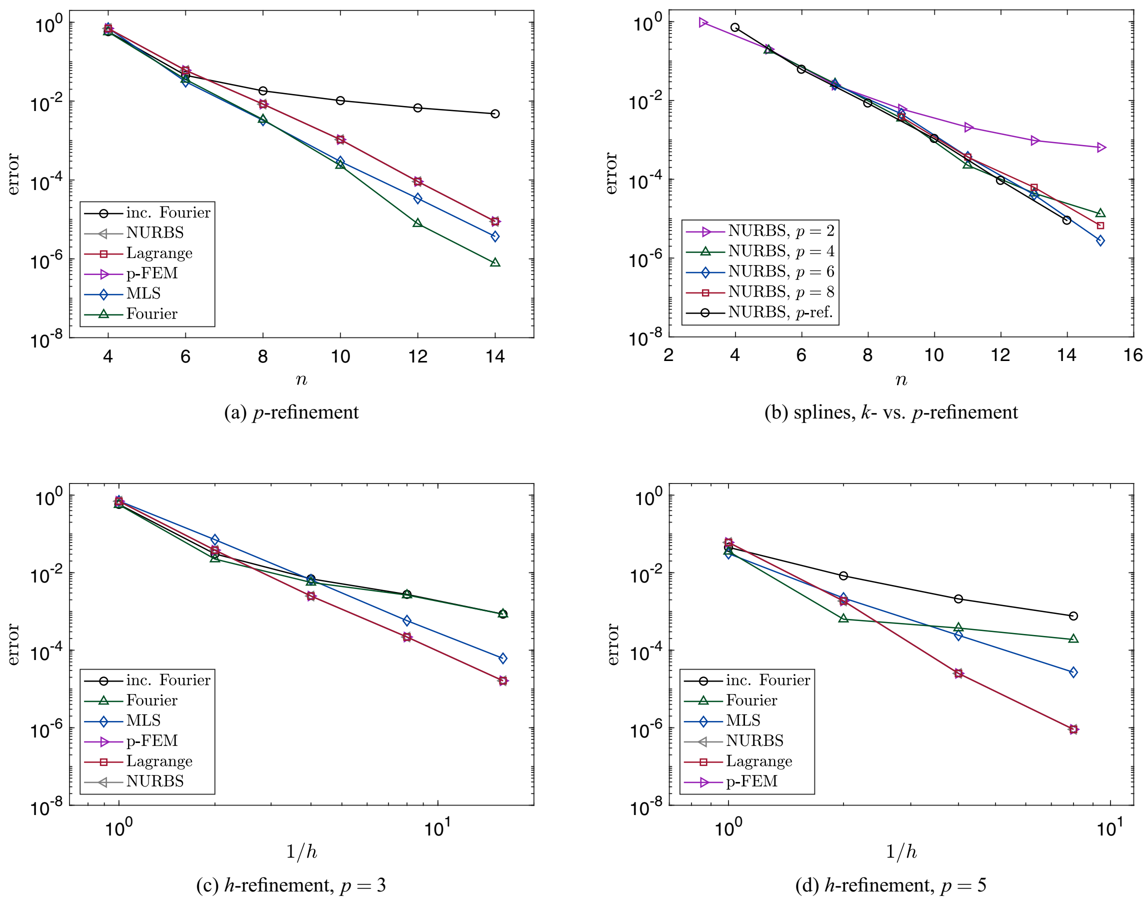
with $t_0 = 1.5$, $f_0 = 1$, and we choose $t_{\max} = 3$. Since this is essentially a one-dimensional problem, an accurate reference solution has been computed using a one-dimensional discretization consisting of 32 conventional spectral elements of order 8. The problem is solved in the time domain by employing an explicit second-order central difference scheme. To ensure that the error in the time integration is negligible, we use a very small time step $\Delta t = 5 \cdot 10^{-6}$ to compute the reference solution.

For the comparison of different interpolations, the boundary of the square geometry has again been discretized using line elements of uniform size. Figure 15a shows an example of a mesh consisting of four elements of order 14 (in this example, the nodes are positioned at the GLL points). The figure also shows a snapshot of the solution at the time $t = 2.5$. The continued fraction expansion is used to obtain the equation of motion. A continued fraction order of 8 has been chosen, which has been tested to be sufficient such that the approximation in the scaling direction $\xi$ does not affect the results. The equation of motion has been integrated with the same central difference scheme as the reference solution but with a larger time step of $\Delta t = 10^{-4}$. This was done to verify once more that the time integration is converged and does not introduce relevant errors.

Figure 15b shows the computed solution on the top of the geometry (extracted at the position (1, 1)). The accuracy is evaluated by computing the $L_2$ norm of the relative deviation from the reference solution at the vertices and all time steps. Results are presented in Fig. 16. Again, Lagrange polynomials, $p$-FEM, and NURBS lead to practically identical results for reasonable element order. When performing $p$-refinement, complete Fourier shape functions and moving least squares result in slightly smaller error levels for this wave propagation problem, while incomplete Fourier shape functions lead to very poor convergence. When performing $h$-refinement, however, complete Fourier shape functions show a similarly slow convergence compared to the incomplete ones. Considering NURBS, rapid convergence can be achieved by employing $k$-refinement instead of $p$-refinement, which is consistent with the findings in Ref. [63] in the context of high order FEMs. In this example, a polynomial degree of 4 suffices to obtain very accurate results (Fig. 16b).

## 4.2 Plate with Hole

As a further benchmark example that involves a curved boundary, we investigate an infinite elastic plate with a circular hole of radius $a$ as depicted in Fig. 17a. A uniform tension $T$ is applied in the $x$-direction and a plane stress state is assumed. We discretize a finite section of a quarter of the plate and apply symmetric boundary conditions on the boundary sections $\Gamma_u$. The analytical solution for the displacements $u_i$ and corresponding stresses $\sigma_{ij}$ in a polar coordinate system $(r, \theta)$ is presented in numerous textbooks and papers, e.g., Refs. [2, 3]. A particularly compact form can be found online [62]:



(a) $p$-refinement

(b) splines, $k$- vs. $p$-refinement

(c) $h$-refinement, $p = 3$

(d) $h$-refinement, $p = 5$

**Fig. 16** Square geometry, transient analysis: relative error of the solution

$$u_x^a(r,\theta) = \frac{Ta}{8\mu}\left[\frac{r}{a}(\kappa+1)\cos\theta + \frac{2a}{r}((1+\kappa)\cos\theta + \cos 3\theta) - \frac{2a^3}{r^3}\cos 3\theta\right] \tag{59}$$

$$u_y^a(r,\theta) = \frac{Ta}{8\mu}\left[\frac{r}{a}(\kappa-3)\sin\theta + \frac{2a}{r}((1-\kappa)\sin\theta + \sin 3\theta) - \frac{2a^3}{r^3}\sin 3\theta\right] \tag{60}$$

$$\begin{aligned}\sigma_{xx}^a(r,\theta) &= -T\frac{a^2}{r^2}\left(\frac{3}{2}\cos 2\theta + \cos 4\theta\right) + T\frac{3a^4}{2r^4}\cos 4\theta + T\\ \sigma_{yy}^a(r,\theta) &= -T\frac{a^2}{r^2}\left(\frac{1}{2}\cos 2\theta - \cos 4\theta\right) - T\frac{3a^4}{2r^4}\cos 4\theta\\ \tau_{xy}^a(r,\theta) &= -T\frac{a^2}{r^2}\left(\frac{1}{2}\sin 2\theta + \sin 4\theta\right) + T\frac{3a^4}{2r^4}\sin 4\theta\end{aligned} \tag{61}$$

In the above equations, $\kappa = (3-\nu)/(1+\nu)$ and $\mu = E/2(1+\nu)$ denote the Lamé parameters, with Young's modulus $E$ and Poisson's ratio $\nu$. For our example, we choose $E = 1$, $\nu = 0.3$. The computed absolute displacements are plotted in Fig. 17b which also shows an exemplary mesh, consisting of 14 four-noded elements. We apply a traction boundary condition based on the analytical solution at the boundary sections $\Gamma_\mathrm{q}$ and evaluate the numerical error with respect to the analytical solution of the displacement at the vertices. Hence, the problem statement reads

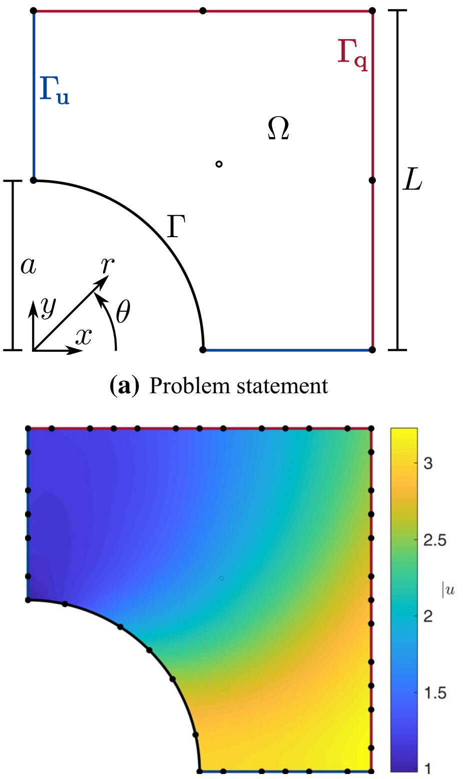
$$\mathscr{L}^\mathrm{T}\mathbf{D}\mathscr{L}\boldsymbol{u} = \mathbf{0} \qquad \text{in } \Omega \tag{62}$$

$$\boldsymbol{n}^\mathrm{T}\boldsymbol{u} = 0 \qquad \text{on } \Gamma_\mathrm{u} \tag{63}$$

$$\boldsymbol{n}^\mathrm{T}\boldsymbol{\sigma} = \boldsymbol{n}^\mathrm{T}\boldsymbol{\sigma}^a \qquad \text{on } \Gamma_\mathrm{q} \tag{64}$$

$$\boldsymbol{n}^\mathrm{T}\boldsymbol{\sigma} = \mathbf{0} \qquad \text{on } \Gamma \setminus (\Gamma_\mathrm{u} \cup \Gamma_\mathrm{q}) \tag{65}$$

with



**(a)** Problem statement

**(b)** Absolute displacement

**Fig. 17** Plate with hole, static analysis: **a** problem statement, **b** exemplary mesh and displacement solution

$$\mathscr{L} = \begin{pmatrix} \partial_x & 0 \\ 0 & \partial_y \\ \partial_y & \partial_x \end{pmatrix}. \tag{66}$$

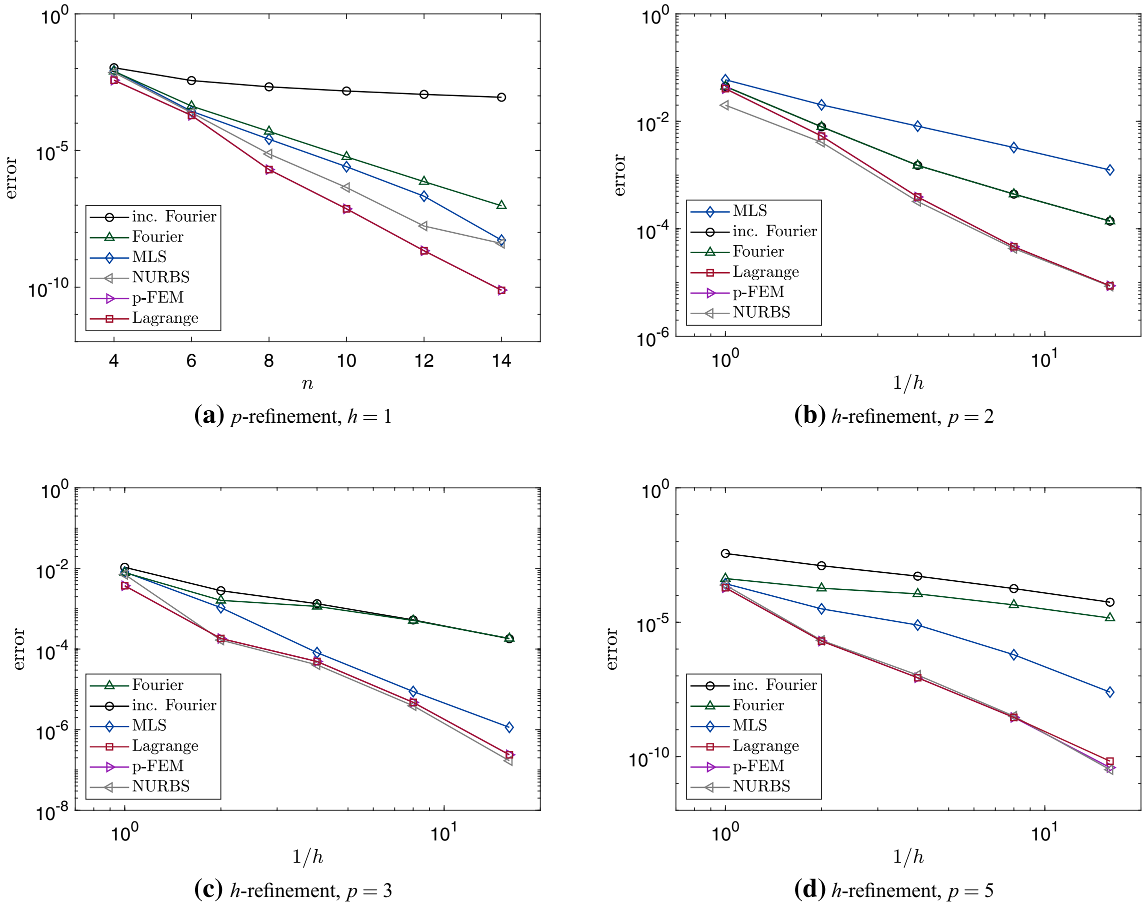
Figure 18 shows the computed error in displacements when performing $h$- and $p$-refinement. The results are consistent with the previous test cases. However, it should be noted that when employing NURBS, the geometry is described exactly, whereas in all other cases, Lagrange interpolation has been applied to approximate the geometry, leading to an additional geometry error. On the other hand, this effect is rather small, in fact, it is only notable when comparing the NURBS-based interpolation to a low-order Lagrange interpolation. For this reason, we included the element order $p = 2$ in this comparison. Only then can we find a slightly smaller error when using NURBS compared to polynomial shape functions for coarse discretizations (Fig. 18b). When applying high-order interpolation, the error due to the geometry approximation is not significant in this example.

## 5 Conclusion

In this contribution, we have presented an overview of the different higher-order interpolants that are being used within the framework of the scaled boundary finite element method. Due to the large number of different techniques, it may not be possible to cover each and every detail within one paper. Nevertheless, we hope that this article did provide valuable insights into the theory and application of the numerous approaches and may serve as a useful reference to guide the reader to a suitable choice of interpolation and facilitate the implementation of the SBFEM. Based on our theoretical considerations and numerical examples, three different classes of shape functions can generally be recommended and should be chosen based on the application:

1. **Lagrange interpolation polynomials** are the most commonly employed shape functions in the SBFEM. They are node-based (which facilitates post-processing and the application of Dirichlet boundary conditions), relatively easy to compute and stable for large element orders. Since Lagrange shape functions are complete polynomials, they can lead to optimal convergence rates for smooth problems.
2. **Hierarchical shape functions** ("$p$-FEM") yield the same convergence rates as Lagrange shape functions and require similar effort in the implementation. The main distinction lies in their hierarchical nature, which has both benefits and drawbacks. The application of nonzero Dirichlet boundary conditions may be considered somewhat inconvenient, and in many cases, post-processing will be slightly more involved. On the other hand, hierarchical shape functions allow adjusting the order of interpolation without computing the coefficient matrices from scratch. However, while this is an elegant feature, applications, where the (repeated) computation of the coefficient matrices constitutes a significant bottleneck, may be scarce.
3. **NUBRS** are the interpolation of choice when employing an isogeometric concept. If the geometry is already defined by such splines, they can be used as both trial and test functions in the SBFEM. The attainable convergence rates are typically similar to the ones obtained by Lagrange polynomials or $p$-FEM. However, if high-order interpolation is desired, it is typically more robust to use $k$-refinement based on splines of moderate order.

Some of the shape functions we tested—even though they are functional and yield converging results—do not seem to offer any particular benefits that would justify their implementation within the SBFEM. The idea of Fourier



**(a)** $p$-refinement, $h = 1$ **(b)** $h$-refinement, $p = 2$

**(c)** $h$-refinement, $p = 3$ **(d)** $h$-refinement, $p = 5$

**Fig. 18** Plate with hole, static analysis: relative error in displacements

shape functions may appear attractive due to the straightforward computation of the underlying harmonic basis functions. However, when employing a complete basis, we need to enforce the Kronecker-delta property at the element endpoints, which is not only computationally more expensive and unstable for large element orders but also destroys the hierarchical property of the basis functions. Furthermore, only even numbers of shape functions should be used for stability. On the other hand, we can use an incomplete basis of harmonic functions, which is hierarchical and does not require an additional effort but leads to very slow convergence. MLS interpolants are adopted from the concept of meshless methods where they have been proven to allow the discretization of surfaces and volumes without requiring an explicit connectivity of subdomains. Within the (two-dimensional) SBFEM, however, we require only parametrization of boundaries. Using a meshless interpolation along the parametrized boundaries does not provide any additional advantages. Since these interpolations pose further challenges regarding the choice of weight functions and the integration and do not generally lead to higher accuracy compared to polynomial shape functions, we currently do not see any motivation for their application.

We also discussed in detail the shape functions that are employed in the so-called diagonal SBFEM. We explained why these shape functions are incomplete polynomials, do not pass the patch test, and—when applied in the standard SBFEM formulation—will not lead to correct results. Furthermore, we provided an alternative interpretation of the diagonal SBFEM, which avoids formulating these shape functions in the first place.

## Compliance with Ethical Standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

1. Abramowitz M, Stegun IA (1972) Handbook of mathematical functions, 10th edn. No. 55 in Applied Mathematics Series. National Bureau of Standards

2. Apostolatos A, Schmidt R, Wüchner R, Bletzinger KU (2014) A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. Int J Numer Methods Eng 97:473–504
3. Barber JR (2004) Elasticity, 2nd edn. Kluwer, Dordrecht
4. Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. Int J Numer Methods Eng 37:229–256
5. Birk C, Prempramote S, Song C (2012) An improved continued-fraction-based high-order transmitting boundary for time-domain analyses in unbounded domains. Int J Numer Methods Eng 89:269–298
6. Chasapi M, Klinkel S (2018) A scaled boundary isogeometric formulation for the elasto-plastic analysis of solids in boundary representation. Comput Methods Appl Mech Eng 333:475–496
7. Chen L, Dornisch W, Klinkel S (2015) Hybrid collocation-Galerkin approach for the analysis of surface represented 3D-solids employing SB-FEM. Comput Methods Appl Mech Eng 295:268–289
8. Chen L, Simeon B, Klinkel S (2016) A nurbs based Galerkin approach for the analysis of solids in boundary representation. Comput Methods Appl Mech Eng 305:777–805
9. Dauksher W, Emery AF (2000) The solution of elastostatic and elastodynamic problems with Chebyshev spectral finite elements. Comput Methods Appl Mech Eng 188:217–233
10. Deeks AJ, Augarde CE (2005) A meshless local Petrov–Galerkin scaled boundary method. Comput Mech 36:159–170
11. Deeks AJ, Wolf JP (2002) A virtual work derivation of the scaled boundary finite-element method for elastostatics. Comput Mech 28:489–504
12. Duczek S (2014) Higher order finite elements and the fictitious domain concept for wave propagation analysis. VDI Fortschritt-Berichte Reihe 20 Nr. 458. https://opendata.uni-halle.de//handle/1981185920/11873
13. Duczek S, Gravenkamp H (2019) Critical assessment of different mass lumping schemes for higher order serendipity finite elements. Comput Methods Appl Mech Eng 350:836–897
14. Duczek S, Gravenkamp H (2019) Mass lumping techniques in the spectral element method: on the equivalence of the row-sum, nodal quadrature, and diagonal scaling methods. Comput Methods Appl Mech Eng 353:516–569
15. Düster A (2002) High order finite elements for three-dimensional, thin-walled nonlinear continua. Berichte aus dem Bauwesen, Shaker
16. Düster A, Rank E, Szabó B (2018) The p-version of the finite element and finite cell methods, chap. 4. Wiley, New York, pp 1–55
17. Gravenkamp H (2014) Numerical methods for the simulation of ultrasonic guided waves. Ph.D. thesis, TU Braunschweig
18. Gravenkamp H, Birk C, Song C (2014) The computation of dispersion relations for axisymmetric waveguides using the Scaled Boundary Finite Element Method. Ultrasonics 54:1373–1385
19. Gravenkamp H, Natarajan S (2018) Scaled boundary polygons for linear elastodynamics. Comput Methods Appl Mech Eng 333:238–256
20. Gravenkamp H, Natarajan S, Dornisch W (2017) On the use of nurbs-based discretizations in the scaled boundary finite element method for wave propagation problems. Comput Methods Appl Mech Eng 315:867–880
21. Gravenkamp H, Prager J, Saputra AA, Song C (2012) The simulation of Lamb waves in a cracked plate using the scaled boundary finite element method. J Acoust Soc Am 132(3):1358–1367
22. Gravenkamp H, Saputra AA, Song C, Birk C (2017) Efficient wave propagation simulation on quadtree meshes using SBFEM with reduced modal basis. Int J Numer Methods Eng 110:1119–1141
23. Gravenkamp H, Song C, Prager J (2012) A numerical approach for the computation of dispersion relations for plate structures using the scaled boundary finite element method. J Sound Vib 331:2543–2557
24. Guan Y, Pourboghrat F, Yu WR (2006) Fourier series based finite element analysis of tube hydroforming. Eng Comput Int J Comput Aided Eng Softw 23(7):697–728
25. Hamzeh Javaran S, Khaji N, Moharrami H (2011) A dual reciprocity BEM approach using new Fourier radial basis functions applied to 2D elastodynamic transient analysis. Eng Anal Bound Elem 35:85–95
26. He Y, Yang H, Deeks AJ (2012) An element-free Galerkin (EFG) scaled boundary method. Finite Elem Anal Des 62:28–36
27. He Y, Yang H, Deeks AJ (2014) Use of Fourier shape functions in the scaled boundary method. Eng Anal Bound Elem 41:152–159
28. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195
29. Karniadakis GE, Sherwin SJ (2005) Spectral/hp element methods for computational fluid dynamics. Oxford Science Publications, Oxford
30. Khaji N, Hamzehei Javaran S (2013) New complex Fourier shape functions for the analysis of two-dimensional potential problems using boundary element method. Eng Anal Bound Elem 37(2):260–272
31. Khaji N, Khodakarami MI (2011) A new semi-analytical method with diagonal coefficient matrices for potential problems. Eng Anal Bound Elem 35(6):845–854
32. Khaji N, Khodakarami MI (2012) A semi-analytical method with a system of decoupled ordinary differential equations for three-dimensional elastostatic problems. Int J Solids Struct 49(18):2528–2546
33. Khodakarami MI, Fakharian M (2015) A new modification in decoupled scaled boundary method with diagonal coefficient matrices for analysis of 2d elastostatic and transient elastodynamic problems. Asian J Civ Eng 16(5):709–732
34. Khodakarami MI, Khaji N (2011) Analysis of elastostatic problems using a semi-analytical method with diagonal coefficient matrices. Eng Anal Bound Elem 35:1288–1296
35. Khodakarami MI, Khaji N (2014) Wave propagation in semi-infinite media with topographical irregularities using Decoupled Equations Method. Soil Dyn Earthq Eng 65:102–112
36. Khodakarami MI, Khaji N, Ahmadi M (2012) Modeling transient elastodynamic problems using a novel semi-analytical method yielding decoupled partial differential equations. Comput Methods Appl Mech Eng 213–216:183–195
37. Klinkel S, Chen L, Dornisch W (2015) A NURBS based hybrid collocation-Galerkin method for the analysis of boundary represented solids. Comput Methods Appl Mech Eng 284:689–711
38. Leung AYT, Chan JKW (1997) Fourier p-element for the analysis of beams and plates. J Sound Vib 212:179–185
39. Leung AYT, Zhu B (2003) Hexahedral Fourier p-elements for vibration of prismatic solids. Int J Struct Stab Dyn 4:125–138
40. Liew KM, Cheng Y, Kitipornchai S (2006) Boundary element-free method (BEFM) and its application to two-dimensional elasticity problems. Int J Numer Methods Eng 65(8):1310–1332
41. Lin G, Zhang Y, Hu ZQ, Zhong H (2014) Scaled boundary isogeometric analysis for 2D elastostatics. Sci China Phys Mech Astron 57:286–300
42. Liu Y, Han Q, Liang Y, Xu G (2018) Numerical investigation of dispersive behaviors for helical thread waveguides using the semi-analytical isogeometric analysis method. Ultrasonics 83:126–136
43. Liu Y, Lin S, Li Y, Li C, Liang Y (2019) Numerical investigation of Rayleigh waves in layered composite piezoelectric structures using the SIGA-PML approach. Compos Part B Eng 158:230–238

44. Man H, Song C, Xiang T, Gao W, Tin-Loi F (2013) High-order plate bending analysis based on the scaled boundary finite element method. Int J Numer Methods Eng 95:331–360
45. Milsted MG, Hutchinson JR (1973) Use of trigonometric terms in the finite element method with application to vibrating membranes. J Sound Vib 32:327–346
46. Mirzajani M, Khaji N, Khodakarami MI (2016) A new global nonreflecting boundary condition with diagonal coefficient matrices for analysis of unbounded media. Appl Math Model 40:2845–2874
47. Natarajan S, Wang J, Song C, Birk C (2015) Isogeometric analysis enhanced by the scaled boundary finite element method. Comput Methods Appl Mech Eng 283:733–762
48. Patera AT (1984) A spectral element method for fluid dynamics: laminar flow in a channel expansion. J Comput Phys 54:468–488
49. Peng M, Cheng Y (2009) A boundary element-free method (BEFM) for two-dimensional potential problems. Eng Anal Bound Elem 33:77–82
50. Piegl L, Tiller W (1997) The NURBS book, 2nd edn. Springer, Berlin
51. Pozrikidis C (2014) Introduction to finite and spectral element methods using MATLAB, 2nd edn. Chapman and Hall/CRC, London
52. Seriani G, Priolo E (1994) Spectral element method for acoustic wave simulation in heterogeneous media. Finite Elem Anal Des 16:337–348
53. Song C (2009) The scaled boundary finite element method in structural dynamics. Int J Numer Methods Eng 77:1139–1171
54. Song C (2018) The scaled boundary finite element method: introduction to theory and implementation. Wiley, New York
55. Song C, Tin-Loi F, Gao W (2010) A definition and evaluation procedure of generalized stress intensity factors at cracks and multi-material wedges. Eng Fract Mech 77:2316–2336
56. Song C, Wolf JP (1996) Consistent infinitesimal finite-element cell method: three-dimensional vector wave equation. Int J Numer Methods Eng 39:2189–2208
57. Song C, Wolf JP (1997) The scaled boundary finite-element method—alias consistent infinitesimal finite-element cell method—for elastodynamics. Comput Methods Appl Mech Eng 147:329–355
58. Szabó B, Babuška I (1991) Finite element analysis. Wiley, New York
59. Vu TH, Deeks AJ (2006) Use of higher-order shape functions in the scaled boundary finite element method. Int J Numer Methods Eng 65:1714–1733
60. Vu TH, Deeks AJ (2008) A p-hierarchical adaptive procedure for the scaled boundary finite element method. Int J Numer Methods Eng 73:47–70
61. Wang W, Peng Y, Wei Z, Guo Z, Jiang Y (2019) High performance analysis of liquid sloshing in horizontal circular tanks with internal body by using IGA-SBFEM. Eng Anal Bound Elem 101:1–16
62. Wikiversity.org: plate with hole in tension. https://en.wikiversity.org/wiki/Introduction_to_Elasticity
63. Willberg C, Duczek S, Vivar Perez J, Schmicker D, Gabbert U (2012) Comparison of different higher order finite element schemes for the simulation of Lamb waves. Comput Methods Appl Mech Eng 241–244:246–261
64. Wolf JP, Song C (1996) Static stiffness of unbounded soil by finite-element method. J Geotech Eng 122:267–273
65. Wolf JP, Song C (1998) Unit impulse response of unbounded medium by scaled boundary finite-element method. Comput Methods Appl Mech Eng 159:355–367
66. Yang ZJ, Zhang ZH, Liu GH, Ooi ET (2011) An h-hierarchical adaptive scaled boundary finite element method for elastodynamics. Comput Struct 89:1417–1429
67. Yazdani M, Khaji N, Khodakarami MI (2016) Development of a new semi-analytical method in fracture mechanics problems based on the energy release rate. Acta Mech 227:3529–3547
68. Zhu T, Zhang JD, Atluri SN (1998) A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach. Comput Mech 21:223–235
69. Zienkiewicz OC, Taylor RL (2000) The finite element method, vol 1. The basis. Butterworth Heinemann, Oxford