Bembel: The Fast Isogeometric Boundary Element C++ Library for Laplace, Helmholtz, and Electric Wave Equation

J. Dölz, H. Harbrecht, S. Kurz, M. Multerer, S. Schöps, F. Wolf
Bembel: The Fast Isogeometric Boundary Element C++ Library for Laplace, Helmholtz, and Electric Wave Equation

J. Dölz\textsuperscript{a}, H. Harbrecht\textsuperscript{b}, S. Kurz\textsuperscript{c}, M. Multerer\textsuperscript{d}, S. Schöps\textsuperscript{c}, F. Wolf\textsuperscript{c,*}

\textsuperscript{a}TU Darmstadt, Department of Mathematics
\textsuperscript{b}Universität Basel, Department of Mathematics and Computer Science
\textsuperscript{c}TU Darmstadt, Institute TEMF & Centre for Computational Engineering
\textsuperscript{d}Università della Svizzera italiana, Institute of Computational Science

Abstract

In this article, we present Bembel, the C++ library featuring higher order isogeometric Galerkin boundary element methods for Laplace, Helmholtz, and Maxwell problems. Bembel is compatible with geometries from the Octave NURBS package, and provides an interface to the Eigen template library for linear algebra operations. For computational efficiency, it applies an embedded fast multipole method tailored to the isogeometric analysis framework and a parallel matrix assembly based on OpenMP.

Keywords: BEM, IGA, Laplace, Helmholtz, Maxwell, C++, FMM, $H^2$-matrix.

1. Introduction

The boundary element method (BEM) or method of moments (MoM) is a widely used tool for the solution of partial differential equations (PDEs) in engineering applications such as acoustic and electromagnetic scattering problems in homogeneous media. It is accepted in industry and implementations are available as software packages. Prominent examples are BEM++ [1] which is accessible open source and BETL [2] which is freely available for academic use. A commonality of these codes is that they are relying on mesh generators to create triangulation-based surface meshes consisting of flat triangles or lower order parametric elements. Thus, the order of convergence of (higher-order) boundary element methods is limited by the approximation error of the underlying mesh.

With the emergence of isogeometric analysis, see [3], boundary element methods have received increased attention by the community, since computer aided design (CAD) tools directly provide parametric representations of surfaces in terms of non-uniform rational B-splines (NURBS). This has recently led to a flourishing development of software concerning NURBS, see, e.g., [4, 5].

*Corresponding author
Email addresses: doelz@mathematik.tu-darmstadt.de (J. Dölz), helmut.harbrecht@unibas.ch (H. Harbrecht), kurz@gsc.tu-darmstadt.de (S. Kurz), michael.multerer@usi.ch (M. Multerer), schoeps@temf.tu-darmstadt.de (S. Schöps), wolf@gsc.tu-darmstadt.de (F. Wolf)
Since the extraction of volume mappings from surface descriptions is an active research area with unsolved open problems, the use of isogeometric finite element methods is quite challenging in practice. Boundary element methods are the methods of choice in this setting, see also [6, 7, 8, 9, 10, 11, 12]. However, their implementation usually requires a significant amount of expert knowledge, which can lead non-experts to refrain from their usage.

The software library Bembel, **Boundary Element Method Based Engineering Library**, is written in C and C++ [13]. It solves boundary value problems governed by the Laplace, Helmholtz or electric wave equation within the isogeometric framework. The development of the software started in the context of wavelet Galerkin methods on parametric surfaces, see [14], where the integration routines for the Green’s function of the Laplacian have been developed and implemented. It was then extended to hierarchical matrices ($\mathcal{H}$-matrices) in [15] and to $\mathcal{H}^2$-matrices and higher-order B-splines in [16]. With support of B-splines and NURBS for the geometry mappings, the Laplace and Helmholtz code became isogeometric in [7]. Finally, in [17], it has been extended to the electric field integral equation.

The publication of this software package aims at making isogeometric boundary element methods available for a broader audience. Therefore, we aim at an easy to use C++ API, which streamlines the access to the underlying routines, with compatibility to the Eigen template library for linear algebra, see [18]. This is achieved, while still providing black-box $\mathcal{H}^2$-compression of the boundary element system matrices and OpenMP parallelized matrix assembly. The $\mathcal{H}^2$-compression yields an almost linear complexities in the number of unknowns for assembly, storage requirements and matrix-vector multiplication for the system matrix. For the representation of geometries, Bembel features arbitrary parametric mappings, most prominently given as NURBS-mappings and can directly be imported from files generated by the Octave NURBS package [4].

The structure of this document is as follows. Section 2 shortly recalls the main concepts of isogeometric boundary element methods. Section 3 is concerned with design considerations for the library, whereas Section 4 deals with the implementation. Afterwards, in Section 5, we discuss an example program, whereas Section 6 discusses the significance of the code. Finally, in Section 7, we will conclude our review.

## 2. Isogeometric Boundary Element Methods

Bembel is able to treat potential, acoustic scattering, and electromagnetic scattering problems governed by the well-known Laplace, Helmholtz, and electric wave equations as stated in Appendix A. For the Helmholtz and the electric wave equation, Bembel assumes non-resonant wave-numbers.

The boundary value problems can be solved by single layer potential ansatzes, see Appendix B, which require the solution of boundary integral equations by means of a numerical discretization. This and similar approaches are commonly known as boundary element methods. Bembel implements these through the use of a conforming Galerkin scheme.

One of the inherent advantages of boundary element methods over classical finite element techniques in the volume is that they can act directly on surface descriptions by NURBS from CAD programs, see Appendix C. This, in connection with corresponding spline spaces, leads to so-called isogeometric boundary
element methods. Bembel implements this and assumes that these surface
descriptions fulfill the requirements stated in Appendix D. The spline spaces for the
Galerkin method are constructed as isogeometric multi-patch B-spline spaces,
see Appendix E.

It is well known that in all cases solvability and uniqueness of the solution,
both of the continuous problem as well as of its discrete counterpart, can be
 guaranteed by imposing reasonable assumptions on the boundary values, essen-
tially guaranteeing “physicality” of the input data, see e.g. [19, 20].

The computation of the system matrix requires the evaluation of singular
integrals, see [20]. For the numerical quadrature of these integrals, we employ
regularization techniques as described in [21]. The compression of the resulting
densely populated system matrices is based on the embedded fast multipole
method (FMM), which is tailored to the framework of isogeometric analysis, see
[16, 17], and fits into the framework of $\mathcal{H}^2$-matrices. Its particular advantage
is that the matrix compression is directly applied on the reference geometry,
that is, the unit square. Hence, the employed compression scheme profits from
the inherently two-dimensional structure of the problem. The complexity for
assembly, storage requirement and matrix-vector multiplication for the system
matrix are almost linear in the number of unknowns, see [16, 17]. Moreover,
this compression technique provably maintains the convergence behaviour for
increasingly finer discretizations, cf. [16]. An in-depth mathematical analysis of
the implemented approach together with numerical studies based on previous
versions of Bembel is available in [7, 16, 17].

3. Design Considerations

Most modern three-dimensional boundary element codes with built-in ma-
trix compression are written in C or C++, which are proven to lead to efficient
implementations. One of the central aims of Bembel is to provide a computa-
tionally efficient isogeometric boundary element code with a plain and simple
user interface. Therefore, the API of Bembel is designed in modern C++ and
provides an interface for the Eigen template library for numerical linear algebra.
This allows the user for a programming experience similar to Matlab and Octave
and for the use of all matrix-free algorithms from the Eigen library. Particularly,
all iterative solvers provided by Eigen can be employed without further modifi-
cations. In contrast, the low-level routines of Bembel are written in optimized
C code which is tuned towards efficiency. Shared memory parallelism by the
use of OpenMP is provided and hence aims at multicore machines.

4. Implementation

4.1. Geometry

With Bembel, geometry representations can directly be imported from files
generated by the Octave NURBS package [4]. However, we should mention that
the implementation of the Geometry class is rather flexible such that the NURBS
mappings may be replaced by arbitrary smooth parametric mappings defined
by the user.
4.2. PDE

The PDE to solve and the potential ansatz to use are defined by a derivation of the base class \texttt{PDEproblem}. The currently available derivations are \texttt{LaplaceSingle}, \texttt{HelmholtzSingle}, and \texttt{MaxwellSingle}, which implement the boundary integral equations (BIE) for Laplace, Helmholtz and Maxwell problems listed in Appendix B. The derived classes contain all necessary information for the PDE and the boundary integral equation under consideration, see Appendix A and Appendix B. Note that, for \texttt{HelmholtzSingle} and \texttt{MaxwellSingle}, the construction of a class instance requires a wave number described as \texttt{std::complex<double>}.

4.3. Discretization

Given a \texttt{Geometry} object and a class inheriting from \texttt{PDEproblem}, e.g. \texttt{LaplaceSingle}, an object of the template class \texttt{Discretization<LaplaceSingle>} describes a conforming boundary element space on refinement level \(l\) and polynomial degree \(p\). The type of the B-spline boundary element space is automatically defined through the class inheriting from \texttt{PDEproblem}, e.g. \texttt{LaplaceSingle}.

4.4. Boundary Values

The load vectors of the Galerkin systems can be obtained via quadrature from a \texttt{std::function} object describing the boundary values. For the Laplace case, \texttt{computeRhs} takes a \texttt{Discretization} and a \texttt{std::function<double(Eigen::Vector3d)>>} object and returns the load vector as an \texttt{Eigen::VectorXd}. For the Helmholtz and Maxwell case, the return value is an \texttt{Eigen::VectorXcd}. Note that in this case the \texttt{std::function} object has to be modified accordingly, i.e. it must accept arguments of type \texttt{Eigen::Vector3cd}. Bembel provides exemplary functions. They can be wrapped into this framework, e.g., via C++ lambdas.

4.5. System Matrix

Given a specialized \texttt{Discretization}, e.g. \texttt{Discretization<LaplaceSingle>}, and a polynomial degree for the \(H^2\)-compression, a corresponding specialization of the template class \texttt{HierarchicalMatrix}, e.g. \texttt{HierarchicalMatrix<LaplaceSingle>}, may be used to assemble the compressed system matrix of the Galerkin system. Note that the type of the PDE and the geometry are directly encoded into the discretization, such that no further information is required here.

The template class \texttt{HierarchicalMatrix<T>} inherits from the corresponding specialization of the \texttt{EigenBase} class, i.e. \texttt{EigenBase<HierarchicalMatrix<T>>}. Therefore, a \texttt{HierarchicalMatrix} object may use the Eigen interface and can be passed to all functions which accept \texttt{EigenBase<Derived>} objects as parameters.

The Eigen methods \texttt{rows()} and \texttt{cols()} are specialized in order to return the dimensions. Moreover, to facilitate the multiplication of \texttt{Eigen::VectorX} objects, the specialization \texttt{Product<HierarchicalMatrix, Rhs, AliasFreeProduct>} of the Eigen \texttt{Product} class is provided. In particular, the operator \(\ast\) is overloaded to support expressions like \(H \ast x\), where \(H\) is a \texttt{HierarchicalMatrix<PDEproblem>} object, and \(x\) an \texttt{Eigen::VectorX} object, with a fitting \texttt{Scalar} type.
4.6. Solution of the System of Linear Equations

The solution of linear systems within Bembel can be performed by the matrix-free solvers from Eigen. To that end, the template class HierarchicalMatrix inherits the Eigen::internal::traits of an Eigen::SparseMatrix with the same Scalar type as in the corresponding example from the Eigen documentation on matrix-free solvers. We opted for the Eigen::SparseMatrix traits, since they seem to impose the least requirements. Hence, in view of the implemented general matrix-vector (GEMV) product $H \cdot x$, all Eigen iterative solvers which solely rely on this operation may be used.

4.7. Evaluation of the Solution

Having the density vector $\rho$ from the iterative solver, the solution to the PDE in either the exterior or the interior domain can be evaluated by the function Sol::evalSolution(gridpoints, $\rho$, myDisc). In here, gridpoints should be an Eigen::Matrix<double, Eigen::Dynamic, 3> matrix of a three-dimensional points in the domain of interest. Depending on the PDE, $\text{sol}$ is either an Eigen::VectorXd, Eigen::VectorXcd or Eigen::Matrix<std::complex<double>, Eigen::Dynamic, 3> object, where each row contains the solution at the corresponding gridpoint.

4.8. Visualization

The Bembel::Vis::plotDiscretizationToVTK functions can be used to visualize $\rho$ and the geometry in vtk-format. Their functionality is explained extensively in the src/examples/example_Visualization.cpp file.

5. Example Program

We discuss a full example for the solution of an electromagnetic scattering problem, which is located in src/examples/example_MaxwellSingle.cpp.

```cpp
int main() {
  using namespace Bembel;
  using namespace Eigen;

  const int multipoleDegree = 16;
  const int knotRepetition = 1;
  const std::complex<double> wavenumber(3, 0);

  Geometry geom("geometry.dat");
  MaxwellSingle pde(wavenumber);
  MatrixXd gridpoints = Util::makeSphereGrid(6, 10);

  const std::function<Vector3cd(Vector3d, std::complex<double>)> fun = ...
    ; // Define incident wave

  for (int P = 1; P < 10; ++P)) {
    for (int L = 0; L < 6; ++L) {
      Discretization<MaxwellSingle> disc(geom, pde, P, knotRepetition, L);
      VectorXcd rhs = Rhs::computeRhs(disc, fun);
      HierarchicalMatrix<MaxwellSingle> hmat(disc, multipoleDegree);
      GMRES<HierarchicalMatrix<MaxwellSingle>,
        IdentityPreconditioner> gmres;
      gmres.compute(hmat);
      VectorXcd rho = gmres.solve(rhs);
    }
  }
}
```
MatrixXcd pot = Sol::evalSolution(gridpoints, rho, disc);
} }
return (0);
}

At the beginning of the program, the polynomial degree for the $H^2$-compression, the knot repetition of the B-spline spaces, and the wave number are defined. The Geometry object is loaded from the file "geometry.dat", and the PDE is defined for a given wave number. Util::makeSphereGrid(6,10) generates ten evaluation points on ten lines of latitude for the scattered field on a sphere around the origin with radius 6. Afterwards, an incident wave for the scattering problem is defined in fun.

The rest of the example program loops over different polynomial degrees $P$ and refinement levels $L$. In the loop, the discrete space is defined as a Discretization object, the load vector and the system matrix are assembled and solved with the GMRES solver from the Eigen (unsupported) package. Finally, the scattered field is evaluated at the previously defined points.

Note that the program is straightforwardly adapted to the Laplace and Helmholtz cases as indicated by the example files in the src/examples folder. These examples also provide error measurements of the solution to the underlying PDE and output of these errors into log-files. These were not included to the above example for a better understanding of the main functionality of the code. \LaTeX-files creating error plots such as illustrated in Figure 1 can be found in the LaTeX folder. The routines can visualize the intermediate quantity rho as illustrated in Figure 1.

6. Impact

The implementation of boundary element methods in three spatial dimensions is a non-trivial task due to the necessary numerical evaluation of singular integrals and the required matrix compression to achieve computational
efficiency. While this is already true for lowest-order implementations for the Laplace equation on boundary triangulations with flat elements, difficulties increase on isogeometric (or parametric) surfaces, isogeometric B-spline (or higher-order) boundary element spaces, and more involved electromagnetic problems. These implementation-related issues lead many people to refrain from the use of boundary element methods, even when a specific engineering problem is known to be solved best therewith. Our software package aims to provide a state-of-the-art boundary element toolbox for engineers who would like to apply competitive isogeometric boundary element methods in a black-box fashion. This allows users to freely employ boundary element methods as a tool in involved engineering applications. To the best of our knowledge, the functionality of higher-order B-spline boundary element spaces is the first open source implementation available for three dimensions.

A major strongpoint of Bembel is the direct integration into the Eigen Linear Algebra Library. This allows the user for a straightforward pre or post processing of data with a clean user interface. The matrix-free algorithms of Eigen provide different kinds of iterative solvers for the Galerkin systems or eigenvalue problems.

7. Conclusion

Bembel is an open-source library enabling users to apply isogometric boundary element methods for potential, acoustic, electromagnetic, and many other problems in a black-box fashion. This is achieved through an easy-to-use API and compatibility with the Eigen template library for linear algebra. Moreover, it incorporates state-of-the-art compression techniques for large problems, as well as OpenMP parallelization.

Acknowledgement

This work is supported by DFG Grants SCH01562/3-1 and KU1553/4-1 within the project Simulation of superconducting cavities with isogeometric boundary elements (IGA-BEM). The work of Jürgen Dölz was partially supported by SNSF Grants 156101 and 174987, as well as the Graduate School of Computational Engineering at TU Darmstadt and the Excellence Initiative of the German Federal and State Governments and the Graduate School of Computational Engineering at TU Darmstadt. Michael Multerer was partially supported by SNSF Grant 137669. The work of Felix Wolf is supported by the Excellence Initiative of the German Federal and State Governments and the Graduate School of Computational Engineering at TU Darmstadt.

References


### Metadata

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Code metadata description</th>
<th>Please fill in this column</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Current code version</td>
<td>v0.9</td>
</tr>
<tr>
<td>C2</td>
<td>Permanent link to code/repository used for this code version</td>
<td><a href="http://www.bembel.eu">http://www.bembel.eu</a>, cf. [13]</td>
</tr>
<tr>
<td>C3</td>
<td>Legal Code License</td>
<td>GPL3</td>
</tr>
<tr>
<td>C4</td>
<td>Code versioning system used</td>
<td>Git</td>
</tr>
<tr>
<td>C5</td>
<td>Software code languages, tools, and services used</td>
<td>C, C++, OpenMP</td>
</tr>
<tr>
<td>C6</td>
<td>Compilation requirements, operating environments &amp; dependencies</td>
<td>CMake, C++11, Eigen Linear Algebra Library</td>
</tr>
<tr>
<td>C7</td>
<td>If available Link to developer documentation/manual</td>
<td><a href="http://www.bembel.eu/Documentation">http://www.bembel.eu/Documentation</a></td>
</tr>
<tr>
<td>C8</td>
<td>Support email for questions</td>
<td><a href="mailto:info@bembel.eu">info@bembel.eu</a></td>
</tr>
</tbody>
</table>

Table 1: Code metadata


Appendix A. Partial Differential Equations

In the following, let $\Omega \subset \mathbb{R}^3$ be a bounded domain with Lipschitz boundary $\Gamma := \partial \Omega$. Moreover, we define the exterior domain $\Omega^c := \mathbb{R}^3 \setminus \overline{\Omega}$. Bembel can treat the Laplace equation

$$-\Delta u_L = 0 \quad \text{in } \Omega,$$
$$u_L = g_L \quad \text{on } \Gamma,$$  \hfill (LP)
and the Helmholtz equation
\[-\Delta u_H - \kappa_H^2 u_H = 0 \quad \text{in } \Omega^c,\]
\[u_H = g_H \quad \text{on } \Gamma,\]  
\[(\text{HP})\]
with Sommerfeld radiation conditions towards infinity. In both cases, the boundary values have to be understood in the usual sense of traces, see, e.g., [20]. In addition, Bembel can treat the electric wave equation
\[\text{curl} \, \text{curl} \, E_M - \kappa_M^2 E_M = 0 \quad \text{in } \Omega^c,\]
\[n \times E_M = g_M \quad \text{on } \Gamma,\]  
\[(\text{MP})\]
with Silver-Müller radiation conditions towards infinity, which can be derived from Maxwell’s equations. Again, the boundary values have to be understood in the sense of traces, see [19].

**Appendix B. Boundary Integral Equations**

The three boundary value problems (LP), (HP), and (MP) can each be solved by means of a single layer potential ansatz, i.e., setting
\[u_L(x) = (\tilde{S}_L \rho_L)(x) := \int_\Gamma \frac{1}{4\pi ||x - y||^2} \rho_L(y) \, d\sigma_y, \quad x \in \Omega,\]  
\[(\text{LS})\]
\[u_H(x) = (\tilde{S}_H \rho_H)(x) := \int_\Gamma \frac{e^{-i\kappa_H ||x - y||}}{4\pi ||x - y||^2} \rho_H(y) \, d\sigma_y, \quad x \in \Omega^c,\]  
\[(\text{HS})\]
\[E_M(x) = (\tilde{S}_M \tilde{j}_M)(x) := \int_\Gamma \frac{e^{-i\kappa_M ||x - y||}}{4\pi ||x - y||^2} \tilde{j}_M(y) \, d\sigma_y + \frac{1}{\kappa_M^2} \nabla x \int_\Gamma \frac{e^{-i\kappa_M ||x - y||}}{4\pi ||x - y||^2} \text{div}_\Gamma \tilde{j}_M(y) \, d\sigma_y, \quad x \in \Omega^c.\]  
\[(\text{MS})\]
It can be shown that (LS), (HS), and (MS) solve the boundary value problems (LP), (HP), and (MP) for appropriate density functions \(\rho_L, \rho_H, \) and \(\tilde{j}_M,\) see [19, 20]. Taking the proper trace operators one arrives at the three boundary integral equations
\[S_L \rho_L = g_L, \quad S_H \rho_H = g_H, \quad S_M \tilde{j}_M = g_M \quad \text{(BIE)}\]
on \(\Gamma\) to determine the unknown density functions in (LS), (HS), and (MS). We note that \(S_L\) and \(S_H\) are isomorphisms from \(H^{-1/2}(\Gamma)\) to \(H^{1/2}(\Gamma)\), whereas \(S_M\) is an isomorphism on \(H^{-1/2} (\text{div}_\Gamma, \Gamma)\), see [19, 20] for details. Thus, in view of a conforming Galerkin method, piecewise polynomial boundary element spaces are sufficient for the boundary element based solution of (LP) and (HP), whereas (MP) requires divergence conforming boundary element spaces. Unique solvability of the corresponding linear systems can be proven, see [19, 20].

**Remark Appendix B.1.** In fact, the described single layer potential approaches solve the interior and the exterior problems of (LP), (HP), and (MP) simultaneously. Thus, \(S_H \rho_H\) and \(S_M \tilde{j}_M\) satisfy (HP) and (MP) also in \(\Omega\), whereas \(S_L \rho_L\) satisfies (LP) also in \(\Omega^c\), with additional radiation conditions \(|u_L(x)| = \mathcal{O}(||x||^{-1})\) and \(||\nabla u_L(x)|| = \mathcal{O}(||x||^{-2})\) towards infinity, see [19, 20].
Appendix C. B-Splines and NURBS

Let $p$ and $k$ be two fixed integers such that $0 \leq p < k$ and let $\Xi$ be a locally quasi uniform knot vector with knots in $[0, 1]$, see [22]. The B-spline basis $\{b^p_j\}_{0 \leq j < k}$ is then defined by recursion as

$$b^p_j(x) = \begin{cases} x - \xi_j - p & \text{if } p = 0, \\ \frac{x - \xi_j}{\xi_{j+p} - \xi_j} b^{p-1}_j(x) + \frac{\xi_{j+p+1} - x}{\xi_{j+p+1} - \xi_{j+1}} b^{p-1}_{j+1}(x), & \text{else}, \end{cases}$$

where $\chi_M$ denotes the indicator function for the set $M$. Having the B-spline basis at our disposal, we define the spline space $S^p(\Xi) := \text{span}(\{b^p_j\}_{0 \leq j < k})$.

A NURBS mapping $\gamma_i : \square \to \mathbb{R}^3$ on the unit square $\square = [0, 1]^2$ is given by

$$\gamma_j(x, y) := \sum_{0 \leq j_1 < k_1, 0 \leq j_2 < k_2} \sum_{i_1=0}^{j_1-1} \sum_{i_2=0}^{j_2-1} \sum_{j_1,j_2} c_{j_1,j_2} b^{p_1}_{j_1}(x) b^{p_2}_{j_2}(y) w_{j_1,j_2},$$

and described by its control points $c_{j_1,j_2} \in \mathbb{R}^3$ and weights $w_{j_1,j_2} > 0$. For further concepts and algorithmic realization of the NURBS, we refer to [22].

Appendix D. Boundary Representation

Bembel assumes that the boundary representations are the union of several patches $\Gamma_i$, i.e.,

$$\gamma_i : \square \to \Gamma_i \quad \text{with} \quad \Gamma_i = \gamma_i(\square) \quad \text{for } i = 1, 2, \ldots, M,$$

where $\gamma_i$ is given by a NURBS mapping with outward pointing normal. The boundary itself is then the collection of all patches

$$\Gamma = \bigcup_{i=1}^M \Gamma_i,$$

where the intersection $\Gamma_i \cap \Gamma_{i'}$ consists at most of a common vertex or a common edge for $i \neq i'$.

In order to ensure conforming meshes, Bembel also imposes the following matching condition on the parametrizations: For each $x = \gamma_i(s)$ on a common edge of $\Gamma_i$ and $\Gamma_{i'}$, there has to exist a bijective and affine mapping $\Xi : \square \to \square$ such that there holds $\gamma_{i'}(s) = (\gamma_{i'} \circ \Xi)(s)$. This means that the parameterizations $\gamma_i$ and $\gamma_{i'}$ coincide on the common edge except for orientation.

Appendix E. B-Spline Boundary Element Spaces

The boundary element spaces for the Galerkin method are constructed as isogeometric multi-patch B-spline spaces, see [23]. Bembel uses equidistant knot vectors $\Xi_1 = \Xi_2$ with $2^L$ elements and the same polynomial degrees in each direction, i.e., $p_1 = p_2$, on the unit square. The corresponding spline spaces are thus given by

$$S^0_{p, \Xi}(\square) = S_{p_1, \Xi_1([0, 1])} \otimes S_{p_2, \Xi_2([0, 1])},$$

$$S^1_{p, \Xi}(\square) = S_{p_1, \Xi_1([0, 1])} \otimes S_{p_2-1, \Xi_2([0, 1])} \times S_{p_1-1, \Xi_1([0, 1])} \otimes S_{p_2, \Xi_2([0, 1])},$$

$$S^2_{p, \Xi}(\square) = S_{p_1-1, \Xi_1([0, 1])} \otimes S_{p_2-1, \Xi_2([0, 1])},$$

where $\otimes$ denotes the outer product.
for \( \mathbf{p} = (p_1, p_2) \) and \( \Xi = (\Xi_1, \Xi_2) \). Herein, \( \Xi'_i \), for \( i = 1, 2 \), denotes the truncated knot vector, i.e., the knot vector without its first and last element.

Due to the representation of the computational geometry by patches, the spaces constructed on the unit square can easily be lifted to the patches on the boundary by the \textit{Piola transform}. Enforcing continuity conditions across patch boundaries yields conforming discretizations

\[
S^0_{\mathbf{p}, \Xi}(\Gamma) \subset H^{1/2}(\Gamma), \quad S^1_{\mathbf{p}, \Xi}(\Gamma) \subset H^{-1/2}_x(\text{div}\Gamma, \Gamma), \quad S^2_{\mathbf{p}, \Xi}(\Gamma) \subset H^{-1/2}(\Gamma).
\]
<table>
<thead>
<tr>
<th>No.</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017-17</td>
<td>P. Corvaja, D. W. Masser and U. Zannier</td>
<td>Torsion hypersurfaces on abelian schemes and Betti coordinates</td>
</tr>
<tr>
<td>2017-18</td>
<td>F. Caubet, M. Dambrine and H. Harbrecht</td>
<td>A Newton method for the data completion problem and application to obstacle detection in Electrical Impedance Tomography</td>
</tr>
<tr>
<td>2018-01</td>
<td>H. Harbrecht and P. Zaspel</td>
<td>On the algebraic construction of sparse multilevel approximations of elliptic tensor product problems</td>
</tr>
<tr>
<td>2018-02</td>
<td>F. Ghiraldin and X. Lamy</td>
<td>Optimal Besov differentiability for entropy solutions of the eikonal equation</td>
</tr>
<tr>
<td>2018-03</td>
<td>H. Harbrecht and M. Schmidlin</td>
<td>Multilevel quadrature for elliptic problems on random domains by the coupling of FEM and BEM</td>
</tr>
<tr>
<td>2018-04</td>
<td>M. Bugeanu and H. Harbrecht</td>
<td>Parametric representation of molecular surfaces</td>
</tr>
<tr>
<td>2018-05</td>
<td>A. Abdulle, M. J. Grote and O. Jecker</td>
<td>Finite element heterogeneous multiscale method for Elastic Waves in Heterogeneous Media</td>
</tr>
<tr>
<td>2018-06</td>
<td>M. J. Grote and J. H. Tang</td>
<td>On controllability methods for the Helmholtz equation</td>
</tr>
<tr>
<td>2018-07</td>
<td>H. Harbrecht and M. Moor</td>
<td>Wavelet boundary element methods — Adaption and goal-oriented error estimation</td>
</tr>
<tr>
<td>2018-08</td>
<td>P. Balazs and H. Harbrecht</td>
<td>Frames for the solution of operator equations in Hilbert spaces with fixed dual pairing</td>
</tr>
<tr>
<td>2018-09</td>
<td>R. Brügger, R. Croce and H. Harbrecht</td>
<td>Solving a Bernoulli type free boundary problem with random diffusion</td>
</tr>
<tr>
<td>2018-11</td>
<td>H. Harbrecht and P. Zaspel</td>
<td>A scalable $H$-matrix approach for the solution of boundary integral equations on multi-GPU clusters</td>
</tr>
<tr>
<td>No.</td>
<td>Author:</td>
<td>Title</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>2018-12</td>
<td>H. Harbrecht, N. Ilić and M. D. Multerer</td>
<td>Acoustic scattering in case of random obstacles</td>
</tr>
<tr>
<td>2018-13</td>
<td>D. H. Baffet, M. J. Grote, S. Imperiale and M. Kachanovska</td>
<td>Energy decay and stability of a perfectly matched layer for the wave equation</td>
</tr>
<tr>
<td>2018-14</td>
<td>D. Baffet and M. J. Grote</td>
<td>On wave splitting, source separation and echo removal with absorbing boundary conditions</td>
</tr>
<tr>
<td>2019-01</td>
<td>M. Graff, M. J. Grote, F. Nataf and F. Assous</td>
<td>How to solve inverse scattering problems without knowing the source term: a three-step strategy</td>
</tr>
<tr>
<td>2019-02</td>
<td>Z. Gao and P. Habegger</td>
<td>Heights in families of abelian varieties and the geometric Bogomolov conjecture</td>
</tr>
<tr>
<td>2019-04</td>
<td>G. Ciampa, G. Crippa and S. Spirito</td>
<td>On smooth approximations of rough vector fields and the selection of flows</td>
</tr>
<tr>
<td>2019-05</td>
<td>M. Colombo, G. Crippa, M. Graff and L. V. Spinolo</td>
<td>Recent results on the singular local limit for nonlocal conservation laws</td>
</tr>
<tr>
<td>2019-06</td>
<td>M. Colombo, G. Crippa, M. Graff and L. V. Spinolo</td>
<td>On the role of numerical viscosity in the study of the local limit of nonlocal conservation laws</td>
</tr>
<tr>
<td>2019-07</td>
<td>G. Ciampa, G. Crippa and S. Spirito</td>
<td>Smooth approximation is not a selection principle for the transport equation with rough vector field</td>
</tr>
<tr>
<td>2019-08</td>
<td>G. Ciampa, G. Crippa and S. Spirito</td>
<td>Weak solutions obtained by the vortex method for the 2D Euler equations are Lagrangian and conserve the energy</td>
</tr>
</tbody>
</table>