11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

ISOGEOMETRIC BOUNDARY ELEMENT METHOD WITH HIERARCHICAL MATRICES

J. Zechner*, B. Marussig*, G. Beer*,†, C. Dünser* and T. P. Fries*

* Institute for Structural Analysis
Graz University of Technology
Lessingstraße 25, 8010 Graz, Austria
e-mail: ifb@tugraz.at, web-page: http://www.ifb.tugraz.at

† School of Engineering
University of Newcastle
Callaghan, NSW 2308, Australia
e-mail: gernot.beer@tugraz.at

Key words: isogeometric analysis, boundary element method, hierarchical matrices, NURBS, BEM, IGA

Abstract. In this work we address the complexity problem of the isogeometric Boundary Element Method by proposing a collocation scheme for practical problems in linear elasticity and the application of hierarchical matrices. For mixed boundary value problems, a block system of matrices – similar to Galerkin formulations – is constructed allowing an effective application of that matrix format. We introduce a strategy for the geometric bisection of surfaces based on NURBS patches. The approximation of system matrices is carried out by means of kernel interpolation. Numerical results are shown that prove the success of the formulation.

1 INTRODUCTION

In the emerging field of isogeometric analysis, Boundary Element Methods (BEM) have gained increasing interest. This is, because for analysis only surface descriptions are required - and Computer Aided Geometric Design (CAD) models are based on such a boundary description. Hence, with this combination the task of domain discretization may be completely avoided. Still, this comes at a prize: the numerical effort of setting up and solving the system of equations is computationally intensive, because the system matrices are fully populated.

Over the last decades much effort has been spent to overcome this barrier. In context of boundary integral techniques, the fast multipole method (FMM) [13], hierarchical matrices ($\mathcal{H}$-matrices) [5], the wavelet method [3] and fast Fourier transformation based

methods [11] reduce the asymptotic numerical complexity significantly, to (almost) linear behavior.

With respect to the analysis with BEM on CAD-surfaces, early reports on the usage of non-uniform rational B-splines (NURBS) have been reported in [12, 17] in the context of electric field equations. In the field of isogeometric analysis, the strategy was applied in [2, 15] to practical problems of elasticity in two dimensions and in [10, 14] to three dimensions. However, there are only few reports [8, 9, 16] of a successful application of fast boundary element techniques in the context of isogeometric analysis.

In this work we present the application of the concept of $\mathcal{H}$-matrices to an isogeometric NURBS-based BEM formulation for problems in elasticity. For the geometric bisection we utilize NURBS-features like knot insertion and the convex hull property. The approximation of far-field matrix blocks is carried out by means of kernel interpolation [7].

2 ISOGEOMETRIC BOUNDARY ELEMENT METHOD

We consider a fixed elastic body subject to external loading. The elastic behavior in terms of displacements u is described by partial differential equation

$$\mathcal{L}u(\boldsymbol{x}) = -\left(\lambda + 2\mu\right) \nabla \cdot \nabla u(\boldsymbol{x}) + \mu \nabla \times \left(\nabla \times u(\boldsymbol{x})\right) = 0 \tag{1}$$

where $\mathcal{L}$ denotes the Lamé-Navier operator. For convenience, the boundary trace

$$\operatorname{Tr} u(\boldsymbol{x}) = \lim_{\mathbf{x} \to \mathbf{y}} u(\boldsymbol{x}) = u(\boldsymbol{y}) \qquad \boldsymbol{x} \in \Omega,\, \boldsymbol{y} \in \Gamma \tag{2}$$

and the conormal derivative

$$\mathcal{T}_y u(\boldsymbol{x}) = \lambda \nabla \cdot u(\boldsymbol{y}) \mathbf{n}(\boldsymbol{y}) + 2\mu \nabla u(\boldsymbol{y}) \cdot \mathbf{n}(\boldsymbol{y}) + \mu \mathbf{n}(\boldsymbol{y}) \times \left(\nabla \times u(\boldsymbol{y})\right) \quad \boldsymbol{x} \in \Omega,\, \boldsymbol{y} \in \Gamma \tag{3}$$

are introduced. The normal $\mathbf{n}$ is defined to always point out of the considered domain. The operator Tr maps displacements $u(\boldsymbol{x})$ to boundary displacements $u(\boldsymbol{y})$. Involving the material law, the conormal derivative $\mathcal{T}_y$ maps $u(\boldsymbol{x})$ to surface traction $t(\boldsymbol{y})$. The boundary can be split into a Neumann and a Dirichlet part such that $\Gamma = \Gamma_N \cup \Gamma_D$ and $\Gamma_N \cap \Gamma_D = 0$. This leads to the following boundary value problem (BVP): Find a displacement field $u(\boldsymbol{x})$ so that

$$\begin{aligned} \mathcal{L}u(\boldsymbol{x}) &= 0 && \forall \boldsymbol{x} \in \Omega \\ \mathcal{T}_y u(\boldsymbol{x}) &= t(\boldsymbol{y}) = g_N(\boldsymbol{y}) && \forall \boldsymbol{y} \in \Gamma_N \\ \operatorname{Tr} u(\boldsymbol{x}) &= u(\boldsymbol{y}) = g_D(\boldsymbol{y}) && \forall \boldsymbol{y} \in \Gamma_D. \end{aligned} \tag{4}$$

Here, g_N is the prescribed Neumann data in terms of surface tractions and g_D represents the prescribed Dirichlet data in terms of displacements.

2.1 Boundary Integral Equation

The BVP (4) can be stated in terms of an boundary integral equation

$$\left(\mathcal{C} + \mathcal{K}\right) u(\boldsymbol{x}) = \mathcal{V} t(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Gamma \tag{5}$$

with the weakly singular single layer operator

$$(\mathcal{V}t)(\boldsymbol{x}) = \int_\Gamma \mathsf{U}(\boldsymbol{x},\boldsymbol{y}) t(\boldsymbol{y}) \,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}} \qquad \forall \boldsymbol{x},\boldsymbol{y} \in \Gamma \tag{6}$$

and the strongly singular double layer operator

$$(\mathcal{K}u)(\boldsymbol{x}) = \int_\Gamma \mathsf{T}(\boldsymbol{x},\boldsymbol{y}) u(\boldsymbol{y}) \,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}} \qquad \forall \boldsymbol{x},\boldsymbol{y} \in \Gamma \setminus B_\varepsilon(\boldsymbol{x}). \tag{7}$$

In case of elasto-static problems $\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$ is *Kelvin's* fundamental solution for displacements and $\mathsf{T}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{T}_y \mathsf{U}(\boldsymbol{x},\boldsymbol{y})$ that for tractions [1]. In (7) the integral only exists as a *Cauchy principal value*, where the radius r_ε of a sphere B_ε around $\boldsymbol{x}$ is treated in a limiting process $r_\varepsilon \to 0$. The remainder of that process is an integral free term which is

$$\mathcal{C}u(\boldsymbol{x}) = cu(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \Gamma \tag{8}$$

with $c = {}^1\!/_2$ on smooth surfaces.

2.2 Discretization with NURBS

In the context of isogeometric Boundary Element analysis, the geometry is discretised by NURBS-patches

$$\Gamma = \Gamma_h = \bigcup_{e=1}^{E} \tau_e \tag{9}$$

which are, in case of three dimensions ($d = 3$), surface patches. Note the equal sign for the geometry description Γ and its discretization Γ_h as a unique feature: the geometry error is zero and thus the subscript is dropped for the remainder of the text. The function

$$\mathcal{X}_\tau(\boldsymbol{r}) : \mathbb{R}^{d-1} \mapsto \mathbb{R}^d \tag{10}$$

is a coordinate transformation mapping local coordinates $\boldsymbol{r} = (r_1, \ldots, r_{d-1})^\intercal$ of the reference NURBS patch to the global coordinates $\boldsymbol{x} = (x_1, \ldots, x_d)^\intercal$ in the Cartesian system.

B-splines form the basis of a mathematical description of the mapping (10) by means of NURBS. Univariate B-splines are described by a knot vector $\Xi = \{r_0, \ldots, r_{i+p+1}\}$, which is a non-decreasing sequence of coordinates in the parametric space, and recursively defined basis functions

$$N_{i,p}(r) = \frac{r - r_i}{r_{i+p} - r_i} N_{i,p-1}(r) + \frac{r_{i+p+1} - r}{r_{i+p+1} - r_{i+1}} N_{i+1,p-1}(r). \tag{11}$$

Here, p denotes the polynomial order of the B-spline and i defines the number of the knot span $[r_i, r_{i+1})$. The initial constant basis functions are

$$N_{i,0}(r) = \begin{cases} 1 & \text{if } r_i \leqslant r < r_{i+1} \\ 0 & \text{else.} \end{cases} \tag{12}$$

NURBS are piece-wise rational functions

$$R_{i,p}(r) = \frac{N_{i,p}(r) w_i}{\sum_{j=0}^{n} N_{j,p}(r) w_j} \tag{13}$$

based on B-splines (11) weighted with w_i. The basis functions $N_{i,p}$ and $R_{i,p}$ have local support and are entirely defined by $p+2$ knots. Multivariate basis functions are simply defined by tensor products of (13). For surfaces they are defined by

$$R_{\mathfrak{i},\mathfrak{J}}(\boldsymbol{r}) = \prod_{n=1}^{d-1} R^n_{\mathfrak{i}_n,\mathfrak{J}_n}(r_n) \tag{14}$$

with multi-indices for the knot span $\mathfrak{i} = \{i_1, \dots, i_{d-1}\}$ and for the order $\mathfrak{J} = \{p_1, \dots, p_{d-1}\}$ in each parametric direction.

Dropping the order-multi-index $\mathfrak{J}$, the geometrical mapping (10) is now expressed by

$$\mathcal{X}_\tau(\boldsymbol{r}) = \boldsymbol{x}(\boldsymbol{r}) = \sum_{\mathfrak{i}} R_{\mathfrak{i}}(\boldsymbol{r}) \boldsymbol{p}_{\mathfrak{i}} \tag{15}$$

in terms of NURBS functions and their corresponding control points $\boldsymbol{p} = (p_1, \dots, p_d)^\intercal$. In addition, Cauchy data is discretised by the same methodology. Different to Lagrange type basis functions, NURBS do not utilize the Kronecker delta property, hence physical values $\mathbf{u} = (u_1, \dots, u_d)^\intercal$ and $\mathbf{t} = (t_1, \dots, t_d)^\intercal$ are mapped to values in $\boldsymbol{p}$, which are $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ marked by a tilde. Hence, the discretization is given by

$$\begin{aligned} u(\boldsymbol{x}(\boldsymbol{r})) \approx \mathbf{u}(\boldsymbol{r}) &= \sum_{\mathfrak{i}} \varphi_{\mathfrak{i}}(\boldsymbol{r}) \tilde{\mathbf{u}}_i \quad \varphi \in S_h \\ t(\boldsymbol{x}(\boldsymbol{r})) \approx \mathbf{t}(\boldsymbol{r}) &= \sum_{\mathfrak{i}} \psi_{\mathfrak{i}}(\boldsymbol{r}) \tilde{\mathbf{t}}_j \quad \psi \in S_h^- \end{aligned} \tag{16}$$

where φ and ψ are basis functions of type (14) and S_h denotes the space of basis functions, which are at least C^0-continuous. With respect to physics, we choose the Ansatz for the tractions to be discontinuous at edges or corners. Hence, S_h^- is the space of discontinuous basis functions which are taken where the surface description (15) exploits C^0-continuity.

2.3 System of Equations

By using collocation, the discretised boundary integral equation (5) is enforced at distinct points. Each of these points are related to a basis function. The location of collocation points is defined by the Greville abscissa [10] except for basis functions with C^{-1} continuity. In that case, the collocation points are slightly indented in order to avoid rank deficient system matrices. By splitting the boundary into a Neumann Γ_N and Dirichlet part Γ_D and by separating known from unknown Cauchy data (16), a block system of equations

$$\begin{matrix} \boldsymbol{x} \in \Gamma_D: \\ \boldsymbol{x} \in \Gamma_N: \end{matrix} \quad \begin{pmatrix} \mathbf{V}_{DD} & -\mathbf{K}_{DN} \\ \mathbf{V}_{ND} & -\mathbf{K}_{NN} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{t}}_D \\ \tilde{\mathbf{u}}_N \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{DD} & -\mathbf{V}_{DN} \\ \mathbf{K}_{ND} & -\mathbf{V}_{NN} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{g}}_D \\ \tilde{\mathbf{g}}_N \end{pmatrix} \tag{17}$$

with the discrete forms of (6) and (7) is created (see [19]). As a consequence of using NURBS, it is possible to approximate known Cauchy data relatively coarsely and differently to the unknown.

The first subscript of the system matrices in (17) denotes the location of collocation point and the second the boundary of the involved NURBS patches. The entries of the system matrices are

$$\mathbf{V}[i,j] = (\mathcal{V}\psi_j)(\boldsymbol{x_i}) \quad \text{and} \quad \mathbf{K}[i,j] = ((\mathcal{C}+\mathcal{K})\varphi_j)\,(\boldsymbol{x_i}) \tag{18}$$

for the i-th collocation point and the j-th basis function. If the value of the basis function is zero at the collocation point, the matrix entries are evaluated by means of standard Gauss quadrature. For singular integrals regularisation schemes for numerical integration are applied [1]. Once the matrix entries are calculated and the known Cauchy values mapped to the control points, the system of equation may be solved by a block LU-factorisation or by means of a direct or iterative Schur-complement solver [18].

Due to the non-local fundamental solution $\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$ the system matrices are fully populated so that the numerical effort for storage and the matrix-vector-product is $\mathcal{O}(n^2)$. To overcome this non-optimal complexity we apply the concept of $\mathcal{H}$-matrices to (17). In the context of NURBS functions, this is explained in the following section.

3 HIERARCHICAL MATRICES

In terms of the described isogeometric BEM formulation, different approximation errors have been introduced. Firstly, by the approximations introduced by discretization of (5), where the residual is minimized in a finite number of collocation points, and by the errors introduced evaluating integrals (18) numerically. Secondly, the approximation of the Cauchy data (16). Finally, the residual of iterative solver is allowed to have a certain tolerance. Consequently, it is reasonable to approximate the system of equations (17) itself with a similar magnitude of error. This motivated the development of the $\mathcal{H}$-matrix technique by Hackbusch [5]. This matrix format provides linear complexity

up to a logarithmic factor $\mathcal{O}(n \log^{\alpha} n)$ in terms of storage and matrix operations. For isogeometric problems of reasonable sizes the logarithmic term is acceptable.

The matrix approximation is based on the fact, that for asymptotically smooth integral kernels matrix blocks of well separated variables $\boldsymbol{x}$ and $\boldsymbol{y}$ have low rank. Therefor, a partition of the system matrices with respect to the geometry is needed. That is, indices of matrix rows $i \in I$ and columns $j \in J$ are resorted such that their offset corresponds somehow to their geometric distance. Naturally, the splitting is done block-wise and categorised into *near field* and *far field*. For the latter type the variables are far away from each other and hence, the matrix block is a candidate for approximation.

3.1 Geometric Bisection

Almost every fast summation method deploys a tree to represent the partition of matrices with general structure. The cluster tree in context of $\mathcal{H}$-matrices is a binary tree and created by splitting the geometry recursively. As shown in Figure 1(a), the indices

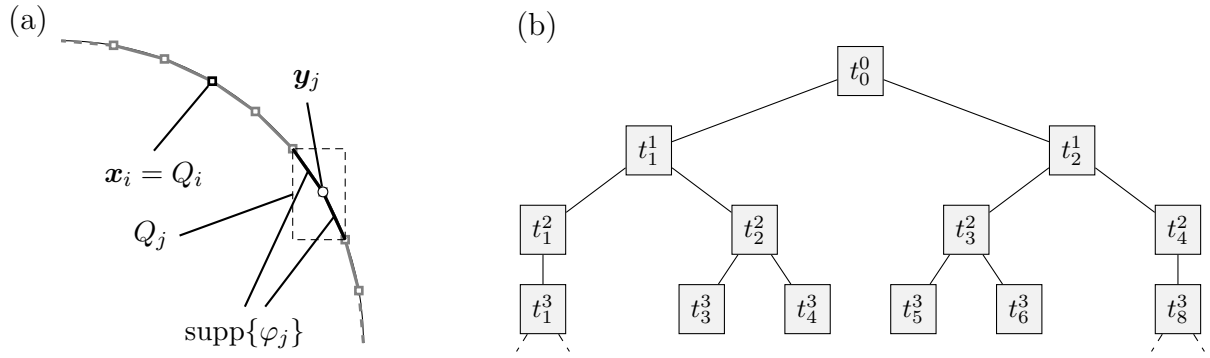

Figure 1: (a) Characteristic points with local bounding boxes for collcation points Q_i and the support of a linear NURBS function Q_j and (b) general binary-tree structure of a cluster tree T

i and j are assigned to characteristic points $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ with local axis parallel bounding boxes Q_i and Q_j. Row indices of the system matrices in (17) correspond to collocation points. Therefore Q_i reduces to the characteristic point. In case of column indices, Q_j defines a bounding box around the support of the NURBS basis function. All indices are collected to the index sets I and J. A cluster is the union of one or more indices of a set including additional information stored in a label. For each set, a labeled binary cluster tree T is constructed. The nodes of the tree are clusters where t_0^0 denotes the root cluster and is labeled by all indices i.e. I, their associated positions $\boldsymbol{x}_i$ and their bounding boxes Q_i. Furthermore, a cluster bounding box B_t^ℓ is created out of all Q_i which is then geometrically split once: t_0^0 gets exactly two children - the clusters t_1^1 and t_2^1. The superscript denotes the level ℓ in T. The splitting is continued recursively until a stopping criterion

$$\text{size}(t) = \#t \leq n_{min} \tag{19}$$

is fulfilled which is characterized by the minimum leaf size n_{min} denoting the minimal amount of indices in a cluster. In Figure 1(b) the general structure of a binary tree with clusters is shown. If a cluster does not have any child, it is called a leaf. In that example this is the case i.e. for t_3^3. The same procedure is applied to column indices j resulting to clusters s and a cluster tree T_J.

For different clustering strategies we refer to the textbook of Hackbusch [6]. In the context of this work, it is suitable to use geometrically balanced clustering. Contrary to clustering techniques in FMM, the overall bounding box of a cluster is shrunk to the minimum possible size with respect to the geometry Q_i and Q_j of the cluster-indices. However, to perform the clustering for column indices J a bounding box Q_j for each support of the NURBS functions needs to be constructed. This is done by means of Bézier extraction and the convex hull property.

In our approach we generate an accumulated knot vector $\Xi_H = \Xi_u \cup \Xi_t$ which is determined by the individual approximation of the fields u and t. For a cubic curve, the

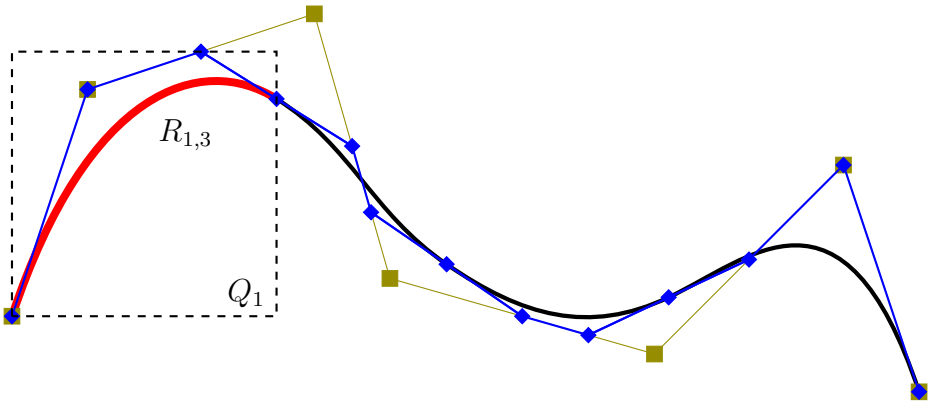

Figure 2: Bézier extraction (blue) of a cubic NURBS curve described by the accumulated knot vector Ξ_H. The dashed box Q_1 denotes the bounding box of the support for the first cubic NURBS-function $R_{1,3}$ (red)

following process is depicted in Figure 2 exemplary. A Bézier extraction is performed by means of knot insertions in Ξ_H until C^0-continuity is reached. The resulting control points (blue) represent a convex hull of the NURBS curve. Hence, for each basis function φ or ψ a bounding box Q of their support is generated easily by taking these control points. For instance, the dashed box in Figure 2 depicts the Q_1 for the first basis function of the description of t or u.

The structure of a $\mathcal{H}$-matrix is then defined by the block cluster $T_{I\times J}$ and its nodes $b = t \times s$. These nodes are constructed for each t and s in the same level where an admissibility condition

$$\min(diam(B_t), diam(B_s)) \leq \eta dist(B_t, B_s) \tag{20}$$

is determined and stored. If (20) is fulfilled, the corresponding matrix block $\mathbf{M}_b$ is related to the far field and therefor, a candidate for approximation. The block cluster tree is now a quad tree and the basis for the partitioned $\mathcal{H}$-matrix. An example for the level-wise

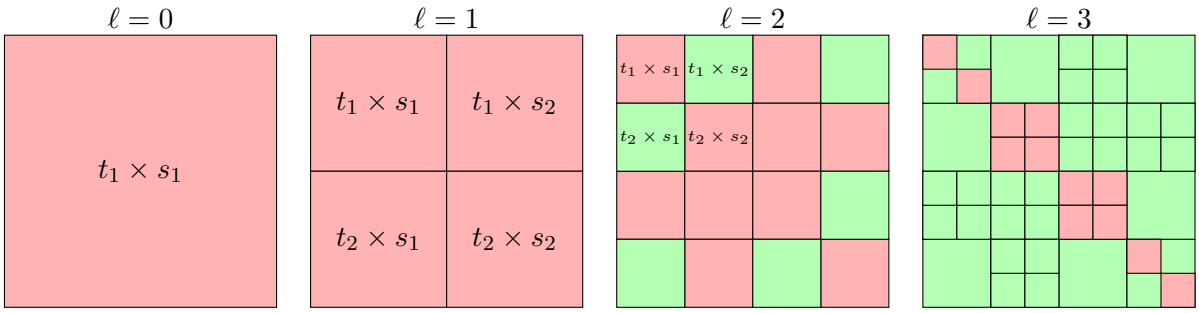

Figure 3: Matrix partition into blocks defined by the block cluster tree $T_{I\times J}$ in up to level $\ell = 3$

definition of the matrix structure is depicted in Figure 3. Here, green matrix blocks denote the far field. For red matrix blocks the level in $T_{I\times J}$ is increased as long as the leaf level in t or s is reached. Finally, the remaining red blocks not fulfilling (20) define the far field. Near field matrix blocks are evaluated with standard BEM techniques whereas far field matrix blocks are subject to approximation. One possibility for that is explained in the upcoming section.

3.2 Matrix Approximation

Since the fundamental solution $\mathsf{U}(\boldsymbol{x}, \boldsymbol{y})$ is asymptotically smooth, it is possible to separate the variables $\boldsymbol{x}$ and $\boldsymbol{y}$ to approximate the integrals (6) and (7). Usually, such approximations stem from Taylor or multipole expansion as well as spherical harmonics. To avoid higher order derivatives of the kernel function, we use the concept of kernel interpolation introduced to $\mathcal{H}$-matrices by Hackbusch and Börm [7].

The fundamental solution is now interpolated by means of Lagrange polynomials

$$\mathsf{U}(\boldsymbol{x}, \boldsymbol{y}) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} L_\nu(\boldsymbol{x}) \mathsf{U}(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) L_\mu(\boldsymbol{y}) \tag{21}$$

with k support points defined on each of the d-dimensional bounding boxes B_t for $\boldsymbol{x}$ and B_s for $\boldsymbol{y}$. The interpolation functions L_ν and L_μ are represented by the tensor product of the Lagrange polynomials in one dimension. To get the best approximation quality for the integral kernel, roots of Chebyshev polynomials of the first kind are chosen for the support points. The interpolated kernel is then taken for the representation of single layer operator $\mathcal{V}$ leading to

$$\mathcal{V}t(\boldsymbol{x}) \approx \sum_{\nu=1}^{k} \sum_{\mu=1}^{k} L_\nu(\boldsymbol{x}) \mathsf{U}(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \int_\Gamma L_\mu(\boldsymbol{y}) t(\boldsymbol{y}) \, \mathrm{d}\, \mathrm{s}_{\boldsymbol{y}}. \tag{22}$$

As a consequence, the boundary integral in (22) depends only on $\boldsymbol{y}$ and is determined by Lagrange polynomials and the traction representation. After discretization, the resulting low rank approximation of an admissible matrix block $\mathbf{M}_b$ is given by its outer product form

$$\mathbf{M}_b \approx \mathbf{R}_k = \mathbf{A} \cdot \mathbf{S} \cdot \mathbf{B}^T \qquad \mathbf{A} \in \mathbb{R}^{r\times k},\, \mathbf{S} \in \mathbb{R}^{k\times k},\, \mathbf{B} \in \mathbb{R}^{c\times k}. \tag{23}$$

The number of support points k denote the rank of the matrices of which the entries are given by

$$\mathbf{A}[i,\nu] = L_\nu(\boldsymbol{x_i}),\;\; \mathbf{S}[\nu,\mu] = \mathsf{U}_\ell(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \text{ and } \mathbf{B}[j,\mu] = \int_{\Gamma_e} L_\mu(\boldsymbol{y})\varphi_j(\boldsymbol{y})\,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}}. \tag{24}$$

Contrary to the quadratic storage requirement rc of $\mathbf{M}_b$, the requirements for $\mathbf{R}_k$ are only $k(r+c+k)$ which is much smaller if $k \ll \min(r,c)$. Similar holds for the numerical effort of a matrix-vector product. This property is the key point for the overall reduced complexity of $\mathcal{H}$-matrices.

Special care has to be taken if the integral kernel depends on normal derivatives like the fundamental solution $\mathsf{T}(\boldsymbol{x},\boldsymbol{y}) = \mathcal{T}_y\mathsf{U}(\boldsymbol{x},\boldsymbol{y})$ for the double layer operator (7). In that case, the conormal derivative (3) is shifted to the Lagrange polynomial. The interpolated double layer potential becomes

$$\mathcal{K}u(\boldsymbol{x}) \approx \sum_{\nu=1}^{k}\sum_{\mu=1}^{k} L_\nu(\boldsymbol{x})\mathsf{U}(\bar{\boldsymbol{x}}_{\boldsymbol{\nu}}, \bar{\boldsymbol{y}}_{\boldsymbol{\mu}}) \int_{\Gamma} \mathcal{T}_y L_\mu(\boldsymbol{y})u(\boldsymbol{y})\,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}}.$$

It is remarkable that for both, the discrete single layer and double layer potentials, the kernel evaluations and evaluation of L_ν at the collocation points $\boldsymbol{x_i}$ stay the same. So do the matrices $\mathbf{A}$ and $\mathbf{S}$. The matrix $\mathbf{B}$ is now defined by

$$\mathbf{B}[j,\mu] = \int_{\Gamma_e} \mathcal{T}_y L_\mu(\boldsymbol{y})\varphi_j(\boldsymbol{y})\,\mathrm{d}\,\mathrm{s}_{\boldsymbol{y}}. \tag{25}$$

For the Laplace problem $\mathcal{T}_y L_\mu(\boldsymbol{y}) = \nabla L_\mu(\boldsymbol{y}) \cdot \mathbf{n}$ holds but it can be envisaged that the implementation of the conormal derivative for elastostatic problems (3) is not a straightforward task. Details on the traction operator applied to the Lagrange polynomials are given in the appendix of [18].

Since k is typically chosen by the user in order to fulfil the approximation quality, the rank of $\mathbf{R}_k$ might not be optimal. In order to further reduce the storage requirement, the matrix block is compressed by means of QR decomposition. The procedure is described in [4].

4 NUMERICAL RESULTS

To show the practicability of the described isogeometric fast boundary element method, a numerical example in two dimensions is presented. The approximation quality of the discretised single (6) and double layer operator (7) is tested on a tunnel geometry such as used in [2]. As test setting, we chose several source points outside the domain and apply Kelvin's fundamental solution from that points to the surface as boundary condition. The approximation quality is measured at multiple points inside the domain by means of the maximum norm $\| \bullet \|_\infty$. Figure 4 shows the optimal convergence of the described BEM formulation. As depicted in Figure 5, matrix compression $c_{\mathcal{H}} = \frac{\text{storage}(\mathbf{M})}{\text{storage}(\mathbf{M}_{\mathcal{H}})}$ with almost linear rate is observed while accuracy is still maintained according to the chosen interpolation quality.

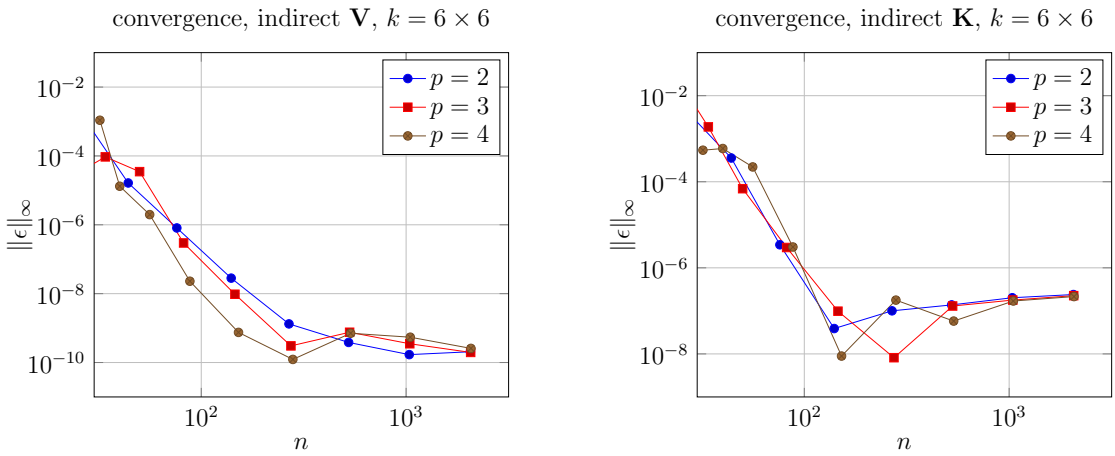

Figure 4: Convergence of $\mathbf{V}$ and $\mathbf{K}$ for NURBS basis functions of order p and 6-th-order kernel interpolation with Lagrange polynomials

5 CONCLUSION

In this work we have shown the application of the concept of $\mathcal{H}$-matrices to a NURBS based, isogeometric collocation BEM. The matrix approximation stems from the interpolation of fundamental solutions over bounding boxes of admissible pairs of indices. For the interpolation of the double layer operator in elasticity, the conormal derivative to the surface is used. For the spatial bisection, bounding boxes enclosing the support of NURBS functions are required. We have shown an evaluation scheme based on knot insertion and Bézier extraction

The authors gratefully acknowledge the financial support of the Austrian Science Fund (FWF), grant no. P24974.

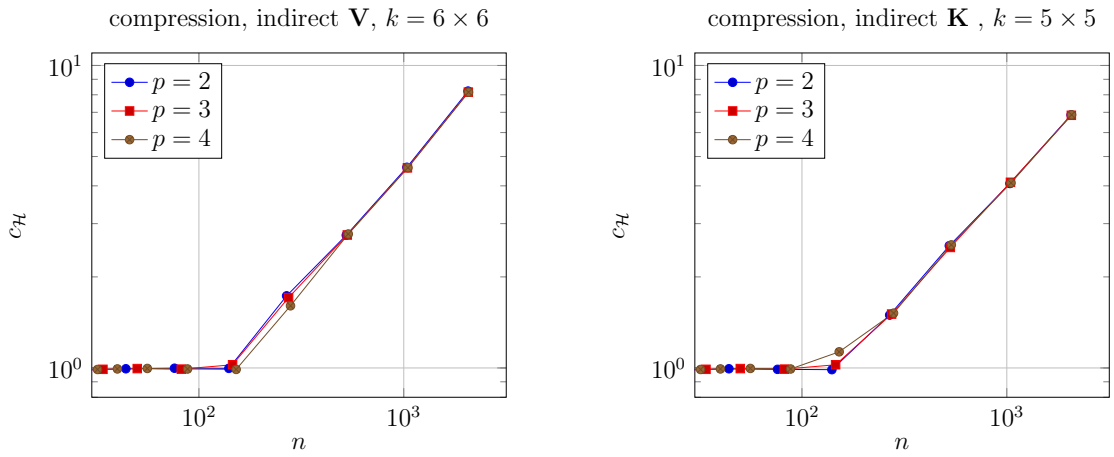

Figure 5: Compression rate of $\mathbf{V}$ and $\mathbf{K}$ for NURBS basis functions of order p and 6-th-order kernel interpolation with Lagrange polynomials

REFERENCES

[1] G. Beer, I. M. Smith, and C. Dünser. *The Boundary Element Method with Programming*. Springer Wien - New York, 2008.

[2] G. Beer, B. Marussig, and C. Duenser. Isogeometric boundary element method for the simulation of underground excavations. *Géotechnique Letters*, 3:108–111, 2013.

[3] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

[4] L. Grasedyck. Adaptive recompression of $\mathcal{H}$-matrices for bem. *Computing*, 74:205–223, 2005.

[5] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. *Computing*, 62: 89–108, 1999.

[6] W. Hackbusch. *Hierarchische Matrizen*. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00221-2. doi: 10.1007/978-3-642-00222-9.

[7] W. Hackbusch and S. Börm. $\mathcal{H}^2$-matrix approximation of integral operators by interpolation. *Applied Numerical Mathematics*, 43(1-2):129 – 143, 2002.

[8] H. Harbrecht and M. Peters. Comparison of fast boundary element methods on parametric surfaces. *Computer Methods in Applied Mechanics and Engineering*, 261–262(0):39 – 55, 2013.

[9] H. Harbrecht and M. Randrianarivony. From computer aided design to wavelet bem. *Computing and Visualization in Science*, 13(2):69–82, 2010.

[10] K. Li and X. Qian. Isogeometric analysis and shape optimization via boundary integral. *Computer-Aided Design*, 43(11):1427–1437, Nov. 2011.

[11] J. R. Phillips and J. White. A precorrected-fft method for capacitance extraction of complicated 3-d structures. In *Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design*, pages 268–271, 1994.

[12] F. Rivas, L. Valle, and M. Cátedra. A moment method formulation for the analysis of wire antennas attached to arbitrary conducting bodies defined by parametric surfaces. *Applied Computational Electromagnetics Society Journal*, 11(2):32–39, 1996.

[13] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, 60(2):187 – 207, 1985.

[14] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, and T. Sederberg. Isogeometric boundary element analysis using unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197 – 221, 2013.

[15] R. Simpson, S. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209–212(0):87–100, feb 2012.

[16] T. Takahashi and T. Matsumoto. An application of fast multipole method to isogeometric boundary element method for laplace equation in two dimensions. *Engineering Analysis with Boundary Elements*, 36(12):1766 – 1775, 2012.

[17] L. Valle, F. Rivas, and M. Catedra. Combining the moment method with geometrical modeling by nurbs surfaces and bezier patches. *IEEE Transactions on Antennas and Propagation*, 42(3):373–381, MAR 1994.

[18] J. Zechner. *A Fast Boundary Element Method with Hierarchical Matrices for Elastostatics and Plasticity*. PhD thesis, Graz University of Technology, Institute for Structural Analysis, 2012.

[19] J. Zechner and G. Beer. A fast elasto-plastic formulation with hierarchical matrices and the boundary element method. *Computational Mechanics*, 51(4):443–453, April 2013.