Inexact Dual-Primal Isogeometric Tearing and Interconnecting Methods

Christoph Hofer, Ulrich Langer, and Stefan Takacs

1 Introduction

Isogeometric Analysis (IgA), cf. Hughes et al. [2005], Beirão da Veiga et al. [2014], is a variant of the Galerkin method where both the geometry of the computational domain and the solution of the partial differential equation (PDE) are represented by B-splines or Non Uniform Rational B-splines (NURBS). One of the strengths of IgA consists in its capability of creating high-order smooth function spaces, while keeping the number of degrees of freedom relatively small. Originally, IgA was formulated by means of one *global geometry mapping*, which restricts the method to simple domains being topologically equivalent to the unit square or the unit cube. More complicated domains are represented as a non-overlapping composition of such simple domains, called *patches*. In such a *multi-patch* setting, each of the patches has its own geometry mapping, and all of the patches are discretized separately.

We are interested in fast solvers for linear systems arising from the discretization of elliptic PDEs in such a multi-patch setting. The local discretization on each patch has typically tensor-product structure.

We use a non-overlapping domain decomposition (DD) method to couple the problem across the patches, namely the dual-primal IsogEometric Tearing and Interconnecting (IETI-DP) method, a variant of the FETI-DP method, see Kleiss et al. [2012]. In general, the geometry mapping does not exhibit more than C^0-continuity across the interfaces. Thus, we only aim to guarantee C^0-continuity of the solution across the interfaces. Moreover, also for

Christoph Hofer, Ulrich Langer
Johannes Kepler University (JKU), Altenbergerstr. 69, A-4040 Linz, Austria, e-mail: {christoph.hofer,ulrich.langer}@jku.at

Ulrich Langer, Stefan Takacs
Austrian Academy of Sciences, RICAM, Altenbergerstr. 69, A-4040 Linz, Austria, e-mail: {ulrich.langer,stefan.takacs}@ricam.oeaw.ac.at

a decomposition of the patches into smaller subpatches, e.g., for parallelization, the choice of C^0 continuity is reasonable if the number of inner dofs stays large enough, cf. Hofer [2017]. The IETI method is closely related to the BDDC method, see Toselli and Widlund [2005], Beirão da Veiga et al. [2013, 2017] and references therein.

So far, the local problems have been solved using direct solvers. Since we want to choose the given patches also as subdomains of the DD-method, the local problems become large if the discretization is refined. In this case, inexact solvers for the local subproblems, as introduced in Klawonn and Rheinbach [2007], could be superior to direct solvers. The aim of this work is to investigate such approaches in combination with the p-robust multigrid solvers, which were proposed by Hofreither and Takacs [2017], as inexact solvers.

In the present paper, we consider the Poisson problem on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$, with $d \in \{2, 3\}$, as model problem: For a given, sufficiently smooth f, find $u \in V_0 := H^1_0(\Omega)$ such that

$$a(u, v) := (\nabla u, \nabla v)_{L^2(\Omega)} = (f, v)_{L^2(\Omega)} =: \langle F, v \rangle \quad \forall v \in V_0. \tag{1}$$

2 Isogeometric Analysis and IETI-DP

On the unit interval, for any spline degree p and number of basis functions M, we define the basis $(\widehat{N}_{i,p})_{i=1}^M$ of univariate B-splines of maximum smoothness C^{p-1} via Cox-de Boor's algorithm. A basis for the *parameter domain* $\widehat{\Omega} := (0, 1)^d$, is realized by the tensor product of such basis functions, again denoted by $\widehat{N}_{i,p}$, where $i = (i_1, \dots, i_d) \in \mathcal{I} := \{1, \dots, M_1\} \times \dots \times \{1, \dots, M_d\}$ and $p = (p_1, \dots, p_d)$ are multi-indices.

In standard (single-patch) IgA, the *physical domain* Ω is given as the image of the parameter domain under the *geometry mapping* $G : \widehat{\Omega} \to \mathbb{R}^d$, defined by $G(\xi) := \sum_{i \in \mathcal{I}} P_i \widehat{N}_{i,p}(\xi)$, with the control points $P_i \in \mathbb{R}^d$, $i \in \mathcal{I}$.

In a multi-patch setting, the domain Ω (*multipatch domain*) is composed of non-overlapping patches $\Omega^{(k)}$, $k = 1, \dots, N$, such that $\overline{\Omega} := \bigcup_{k=1}^N \overline{\Omega}^{(k)}$. Each patch $\Omega^{(k)} := G^{(k)}(\widehat{\Omega})$ is represented by its own geometry mapping. We call $\Gamma := \bigcup_{k>l} \partial\Omega^{(k)} \cap \partial\Omega^{(l)}$ the *interface*, and denote its restriction to one of the patches $\Omega^{(k)}$ by $\Gamma^{(k)} := \Gamma \cap \partial\Omega^{(k)}$. Throughout the paper, the superscript (k) denotes the restriction of the underlying symbol to $\Omega^{(k)}$.

We use B-splines not only for defining the geometry, but also for representing the approximate solution of (1). Once the basis functions are defined on the parameter domain $\widehat{\Omega}$, we define the bases on the patches $\Omega^{(k)}$ via the pull-back principle, and obtain the basis functions $N_{i,p} := \widehat{N}_{i,p} \circ G^{-1}$.

The main idea of IETI-DP is to decouple the patches by tearing the interface unknowns which introduces additional degrees of freedom (dofs). We denote the resulting space by V_h. Then, continuity is again enforced using

Lagrange multipliers λ. Hence, the local subproblems on each patch are essentially pure Neumann problems (at least for interior patches). Due to the presence of a kernel, a straight-forward Schur complement formulation is not possible. In order to overcome this problem, certain continuity conditions are enforced by incorporating them into the space V_h, (*strongly enforced continuity conditions*) which yields the smaller space $\widetilde{V}_h$. There, we formulate the following problem. Find $(u, \lambda) \in \widetilde{V}_h \times \Lambda$ such that

$$\begin{bmatrix} \widetilde{K} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{f} \\ 0 \end{bmatrix}, \tag{2}$$

where $\widetilde{K}$ is the stiffness matrix, $\widetilde{B}$ the jump operator, and $\widetilde{f}$ the right hand side. Here and in what follows, we do not distinguish between the IgA functions and their vector representation with respect to the chosen basis.

Now, we split V_h into interior dofs and interface dofs, which yields an interface space W. By splitting $\widetilde{V}_h$ analogously, we obtain the space $\widetilde{W}$. Based on this splitting, we formulate the problem using the Schur complement of the stiffness matrix K in V_h with respect to the interface dofs: $S := K_{BB} - K_{BI}K_{II}^{-1}K_{IB}$, where the subindices B and I denote the boundary and interior dofs, respectively. The restriction of S to $\widetilde{W}$ is denoted by $\widetilde{S}$, which yields the following saddle-point formulation: Find $(w, \lambda) \in \widetilde{W} \times \Lambda$ such that

$$\begin{bmatrix} \widetilde{S} & \widetilde{B}^T \\ \widetilde{B} & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} \widetilde{g} \\ 0 \end{bmatrix}, \tag{3}$$

where $\widetilde{g} := \widetilde{I}^T(f_B - K_{BI}K_{II}^{-1}f_I)$ and $\widetilde{I} : \widetilde{W} \to W$ is the canonical embedding. We denote the subspace of $\widetilde{W}$ satisfying the strongly enforced continuity conditions homogeneously by W_Δ and the S-orthogonal complement by W_Π. In the literature, our choice of W_Π is often called *energy minimizing primal subspace*. Finally, we can define the Schur complement F of the saddle-point problem (3), and obtain the problem: Find $\lambda \in \Lambda$ such that

$$F\lambda := (\widetilde{B}\widetilde{S}^{-1}\widetilde{B}^T)\lambda = \widetilde{B}\widetilde{S}^{-1}\widetilde{g} := d. \tag{4}$$

Equation (4) is solved by means of the conjugate gradient (CG) method using the scaled Dirichlet preconditioner $M_{sD}^{-1} := B_D S B_D^T$, where B_D is a scaled version of the jump operator B on V_h. Note that we can approximate $\widetilde{S}^{-1}$ because $\widetilde{S}$ can be represented (by reordering of the dofs) as a block diagonal matrix of matrices $S_{\Delta\Delta}^{(k)}$ for each patch and the matrix $S_{\Pi\Pi}$. For a summary of the algorithm and a more detailed explanation, we refer, e.g., to Toselli and Widlund [2005], Hofer and Langer [2017] and references therein.

3 Incorporating Multigrid in IETI-DP

We investigate different possibilities to incorporate a multigrid solver into the IETI-DP algorithm. The application of the IETI-DP algorithm requires the solution of local Neumann and Dirichlet problems.

3.1 Local Dirichlet problems

We have to solve linear systems with the system matrix $K_{II}^{(k)}$ in the application of S in the preconditioner and when calculating the right hand side $\widetilde{g}$. These linear systems are Dirichlet problems (up to boundary conditions). The right hand side $\widetilde{g}$ has to be computed very accurately, i.e., at least up to discretization error. However, for the preconditioner, a few MG V-cycles are usually enough, since we only have to ensure the spectral equivalence of the inexact scaled Dirichlet preconditioner to the exact one, cf. Klawonn et al. [2016] and references therein.

3.2 Local Neumann problems

Local Neumann problems appear in the construction of the S-orthogonal basis for W_Π and in the application of $S_{\Delta\Delta}$. In order to construct the nodal and S-orthogonal basis $\{\phi_j^{(k)}\}_j$ of $W_\Pi^{(k)}$, we have to solve

$$\begin{bmatrix} S^{(k)} & C^{(k)^T} \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} \phi_j^{(k)} \\ \mu_j^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{e}_j^{(k)} \end{bmatrix}, \quad \forall j \in \{1, \dots, n_\Pi^{(k)}\}, \tag{5}$$

where $\boldsymbol{e}_j^{(k)} \in \mathbb{R}^{n_\Pi^{(k)}}$ is the j-th unit vector, and the matrix $C^{(k)}$ realizes the $n_\Pi^{(k)}$ strongly enforced continuity conditions contributing to the patch $\Omega^{(k)}$. Instead of solving (5) directly, we solve

$$\begin{bmatrix} K^{(k)} & C^{(k)^T} \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} \overline{\phi}_j^{(k)} \\ \mu_j^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{e}_j^{(k)} \end{bmatrix}, \quad \forall j \in \{1, \dots, n_\Pi^{(k)}\}, \tag{6}$$

and obtain the desired basis functions by $\phi_j = \overline{\phi}_j|_{\Gamma^{(k)}}$. Note that $\{\overline{\phi}_j^{(k)}\}_j$ is a K-orthogonal basis. The system is solved with the *Schöberl-Zulehner* (SZ) preconditioner, see Schöberl and Zulehner [2007].

The SZ preconditioner for (6) requires preconditioners $\hat{K}^{(k)}$ and $\hat{H}^{(k)}$ for the upper left block $K^{(k)}$ and its inexact Schur complement $H^{(k)} := C^{(k)}(\hat{K}^{(k)})^{-1}C^{(k)^T}$, respectively. The preconditioner $K^{(k)}$ is realized by a few

MG V-cycles. It is required that $\hat{K}^{(k)} > K^{(k)}$, which implies that $\hat{K}^{(k)}$ has to be positive definite. In order to handle also the case where $K^{(k)}$ is singular, we need to set up MG based on a regularized matrix $K_M^{(k)} := K^{(k)} + \alpha \widehat{M}^{(k)}$, where α is chosen to be 10^{-2} and $\widehat{M}^{(k)}$ is the mass matrix on the parameter domain. Note that we can exploit the tensor product structure to efficiently assemble the mass matrix $\widehat{M}^{(k)}$. Secondly, the SZ preconditioner requires that $\hat{H}^{(k)} < H^{(k)}$. Since in our case the number of rows of $C^{(k)}$ is given by $n_\Pi^{(k)}$, a small number that does not change during refinement, we calculate the inexact Schur complement exactly. This can be performed by applying $\left(\hat{K}^{(k)}\right)^{-1}$ to $n_\Pi^{(k)}$ vectors. Finally, by a suitable scaling, e.g., $\hat{H}^{(k)} := 0.99\, H^{(k)}$, we obtain the desired matrix inequality.

The second type of Neumann problem appears in the application of F. We look for a solution of the system $S_{\Delta\Delta}^{(k)} w_\Delta^{(k)} = f_\Delta^{(k)}$, which can be written as

$$\begin{bmatrix} S^{(k)} & {C^{(k)}}^T \\ C^{(k)} & 0 \end{bmatrix} \begin{bmatrix} w_\Delta^{(k)} \\ \mu^{(k)} \end{bmatrix} = \begin{bmatrix} f^{(k)} \\ 0 \end{bmatrix}. \tag{7}$$

Certainly, one can use the same method as above. However, we can utilize the fact that we search for a minimizer of $\frac{1}{2}(S^{(k)} w^{(k)}, w^{(k)}) - (w^{(k)}, f^{(k)})$ in the subspace given by $C^{(k)} w^{(k)} = 0$. This solution can be computed by first solving the unconstrained problem, and then projecting the minimizer into the subspace using a energy-minimizing projection. The projection is trivial because the decomposition of $\widetilde{W}$ into W_Π and W_Δ is S-orthogonal.

Note that the CG algorithm, when applied to a positive semidefinite matrix, stays in the factor space with respect to the kernel and computes one of the minimizers. The solution of the constrained minimization problem is, as outlined above, obtained by applying the projection. As long as the number of CG iterations is not too large, numerical instabilities are not observed when applying CG to a positive semidefinite problem.

The S-orthogonal basis has to be computed very accurately in order to maintain the orthogonality. Since the equation $S_{\Delta\Delta}^{(k)} w_\Delta^{(k)} = f_\Delta^{(k)}$ appears in the system matrix F, its solution also requires an accuracy of at least the discretization error.

3.3 Variants of inexact formulations

From the discussion above, we deduce four (reasonable) versions:

(D-D) The classical IETI-DP method, using direct solvers everywhere.

(D-MG) We use MG in the scaled preconditioner for the solution of the local Dirichlet problems and the transformation of the right hand side, see Section 3.1. As already mentioned, the required accuracy for computing $\widetilde{g}$ has

to be of the order of discretization error, whereas a few V-cycles are enough for the preconditioner.

(MG-MG) We use MG for all patch-local problems, i.e., the local Dirichlet and Neumann problems. This implies that also the calculation of the basis for W_Δ is performed by means of MG, which turns out to be very costly. Moreover, for each application of F, we have to solve a local Neumann problem in W_Δ with the accuracy in the order of the discretization error.

(MG-MG-S) To overcome the efficiency problem of the requirement of solving a linear system with MG very accurately, we use the saddle point formulation instead of F. On the one hand, at each iteration step, we only have to apply a given matrix instead of solving a linear system. On the other hand, we now have to deal with a saddle point problem. Moreover, the iteration is not only applied to the interface dofs, but also to the dofs in the whole domain.

We will always assume that the considered multipatch domain has only a moderate number of patches, such that the coarse problem can still be handled by a direct solver. For extensions to inexact version for the coarse problem, we refer to Klawonn and Rheinbach [2007].

For the first three methods, we use the CG method to solve $F\lambda = d$ as outer iteration. For (MG-MG-S), we have to deal with the saddle point problem (2), which we solve using the Bramble-Pasciak CG (BPCG) method, cf. Bramble and Pasciak [1988]. The building blocks for this method are a preconditioner $\widehat{\widetilde{K}}$ for $\widetilde{K}$ and $\hat{F}$ for the Schur complement F. The construction of $\widehat{\widetilde{K}}$ follows the same steps as in the previous section, but we only apply a few MG V-cycles. Concerning $\hat{F}$, a good choice is the scaled Dirichlet preconditioner M_{sD}^{-1}, cf. Klawonn and Rheinbach [2007].

4 Numerical Experiments

We solve the model problem (1) on a two and a three dimensional computational domain. In the two dimensional case, we use the quarter annulus divided into $32 = 8 \times 4$ patches, as illustrated in Fig. 1(left). The three dimensional domain is the twisted quarter annulus, decomposed into $128 = 4 \times 4 \times 8$ patches as presented in Fig. 1(right). We use B-splines of maximal smoothness inside a patch and C^0-coupling across the patch interfaces.

We have chosen the continuity of the vertex values and the edge averages for the two dimensional example, and the continuity of the edge averages for the three dimensional example as strongly enforced continuity conditions.

For the examples with polynomial degree $p = 2$, we use a standard MG method based on a hierarchy of nested grids keeping p fixed and use a standard Gauss Seidel (GS) smoother. For the examples with higher polynomial degree ($p = 4$ or 7), we have used $p = 1$ on all grid levels but the finest grid.

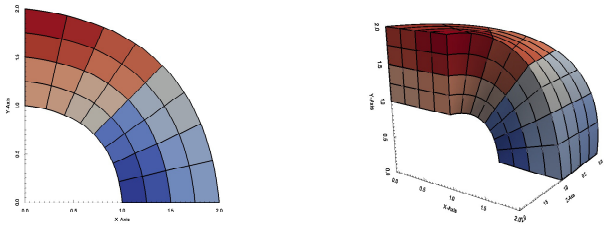

Fig. 1 Quarter annulus in 2d (left), twisted quarter annulus in 3d (right).

This does not yield nested spaces. Thus, we cannot use the canonical embedding and restriction. Instead, we use L^2-projections to realize them. On the finest grid, we use a MG smoother suitable for high-order IgA, namely a variant of the subspace-corrected mass smoother proposed and analyzed in Hofreither and Takacs [2017]. For this smoother, it was shown that a resulting MG method is robust with respect to both the grid size and the polynomial degree. However, for $p = 1$ or 2, standard approaches are more efficient. Thus, we again use this smoother only for the finest level, while for all other grid levels we use standard GS smoothers. To archive better results, we have modified the subspace-corrected mass smoother by incorporating a rank-one approximation of the geometry transformation.

For the outer CG or BPCG iteration, we use a zero initial guess, and the reduction of the initial residual by the factor 10^{-6} as stopping criterion. The local problems related to the calculation of the S-orthogonal basis are solved up to a tolerance of 10^{-12}. In case of the (MG-MG) version, the local Neumann problems (7) in W_Δ are solved up to a relative error of 10^{-10}. The number of MG cycles in the preconditioner is fixed. For the local Dirichlet problems in the scaled Dirichlet preconditioner, we use 2 V-cycles. The local Neumann problems, which appear in the preconditioner of the (MG-MG-S) version, are approximately solved by 3 V-cycles. In the following, we report on the number of CG iterations to solve (4) and BPCG iterations for (2) and the total time in seconds, which includes the assembling, the IETI-DP setup and solving phase. For the weak scalability tests in Table 1 and Table 2, we observe in all cases a polylogarithmic growth of the outer iterations and a quasi-optimal behavior of the computation time.

The algorithm is realized with the open source C++ library G+Smo[1] We utilize the PARDISO 5.0.0 Solver, cf. Kuzmin et al. [2013], for performing the LU factorizations. To allow a better comparison of the different variants, we only perform serial computations.[2]

[1] G+Smo (Geometry plus Simulation modules) v0.8.1, `http://gs.jku.at/gismo`.

[2] Our code is compiled with the `gcc 4.8.3` compiler with optimization flag `-O3`. The results are obtain on the RADON1 cluster at Linz. We use a single core of a node, equipped with 2x Xeon E5-2630v3 "Haswell" CPU (8 Cores, 2.4Ghz, 20MB Cache) and 128 GB RAM.

	D-D		**MG-D**		**MG-MG**		**MG-MG-S**	
$p = 2$\Dofs	It.	Time	It.	Time	It.	Time	It.	Time
134421	9	10	9	8	9	13	14	14
530965	10	45	10	37	10	54	15	90
2110485	11	224	11	172	11	272	16	568
8415253	11	1005	11	762	11	1181	15	3394
33607701	OoM		OoM		13	5070	OoM	
$p = 7$\Dofs	It.	Time	It.	Time	It.	Time	It.	Time
45753	10	26	10	27	10	57	14	54
155961	11	108	11	110	11	225	15	211
572985	12	498	12	495	12	1048	17	1013
2193465	13	2384	13	2265	14	4427	18	4344
8580153	OoM		OoM		15	18484	20	19958

Table 1 Numerical results for the quarter annulus in 2d.

	D-D		**MG-D**		**MG-MG**		**MG-MG-S**	
$p = 2$\Dofs	It.	Time	It.	Time	It.	Time	It.	Time
14079	11	3	11	3	11	8	25	7
86975	12	19	12	19	12	59	26	59
606015	14	213	14	197	14	484	30	616
4513343	OoM		16	2764	16	5244	35	11657
$p = 4$\Dofs	It.	Time	It.	Time	It.	Time	It.	Time
40095	13	30	13	33	13	112	23	104
160863	15	234	15	254	15	659	28	633
849375	16	2237	17	2356	17	5403	32	5298
5390559	OoM		OoM		19	45243	37	52831

Table 2 Numerical results for the twisted quarter annulus in 3d.

In Table 1, we summarize the results for the two dimensional domain for $p = 2$ and 7. The size of the coarse space W_{Π} is 73. We observe that replacing the direct solver in the preconditioner with two MG V-cycles does not change the number of outer iterations. Moreover, going from the Schur complement to the saddle point formulation and using BPCG there, leads only to a minor increase in the number of outer iterations. In all cases, the logarithmic dependence of the condition number on h is preserved. The advantage of the formulation using only MG, especially (MG-MG), is its smaller memory footprint, therefore, the possibility of solving larger systems. However, the setting with the best performance is (MG-D). Concluding, for small polynomial degrees and using the GS smoother, (MG-MG) gives reasonable trade off between performance and memory usage and for larger polynomial degrees, this setting can be still recommended if memory consumption is an issue.

In the case $p = 2$, for the inner iterations, we have observed that the CG needed on average 8 iterations to compute $\widetilde{g}$, the calculation of the S-orthogonal basis needed on average 14 iterations, and the solution of (7)

required on average 10 iterations. For the second case, $p = 7$, we needed 9 iterations to compute $\widetilde{g}$, 13 iterations for the calculation of the S-orthogonal basis and 10 iterations for the solutions of (7). Here and in what follows, we have taken the average over the patches, the individual levels and the individual steps of the outer iteration. We mention that the number of inner iterations was only varying slightly.

In Table 2, we summarize the results for the three dimensional domain and for $p = 2$ and 4. The size of the coarse space W_Π is 240. We observe that replacing the direct solver in the preconditioner with two MG V-cycles does not change the number of outer iterations. We further observe that the results are similar to the one of the two dimensional case. However, the number of iterations almost doubled when using BPCG for (MG-MG-S). In all cases, the logarithmic dependence of the condition number on h is preserved. The advantage of the formulation using only MG, especially (MG-MG), is its smaller memory footprint, therefore the possibility of solving larger systems. The best performance is obtained sometimes by (D-D) and sometimes by (MG-D), where both approaches are comparable in all cases.

Concerning the inner iterations, for $p = 2$, we need on average 15 CG iterations to compute $\widetilde{g}$, 22 CG iterations to build up each S-orthogonal basis function, and 18 CG iterations to solve (7). In the case of $p = 4$, we needed on average only 10 iterations to compute $\widetilde{g}$, 14 iterations for the construction of the S-orthogonal basis functions, and 11 iterations for solving (7).

The last test deals with the weak scalability of the method, where we only investigate the two dimensional setting for $p = 7$. We fix the ratio H/h and increase the number of patches. We expect constant number of iterations and a linear increase of the computation time. In Table 3, beside the Dofs, we report the size of the coarse space n_Π and the number of patches N. For each method, we provide the number of iterations and the computation time in seconds. We observe that the number of iterations and computation time behave as expected.

$p = 7$			**D-D**		**MG-D**		**MG-MG**		**MG-MG-S**	
n_Π	N	Dofs	It.	Time	It.	Time	It.	Time	It.	Time
73	32	45753	10	27	10	27	10	62	20	60
337	128	183921	11	111	11	108	11	268	15	234
1441	512	737505	11	446	11	438	11	1111	13	943
5953	2048	2953665	10	1777	10	1729	10	4468	12	3821
24193	8192	11821953	OoM		OoM		10	19691	11	15392

Table 3 Weak scalability of the methods with respect to the number of patches.

Acknowledgements This work was supported by the Austrian Science Fund (FWF) under the grant W1214, project DK4. This support is gratefully acknowledged.

References

L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. BDDC preconditioners for isogeometric analysis. *M³AS*, 23(6):1099–1142, 2013.

L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numer.*, 23:157–287, 2014.

L. Beirão da Veiga, L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. Adaptive selection of primal constraints for isogeometric BDDC deluxe preconditioners. *SISC*, 39(1):A281–A302, 2017.

J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Math. Comput.*, 50(181):1–17, 1988.

C. Hofer. Parallelization of continuous and discontinuous Galerkin dual–primal isogeometric tearing and interconnecting methods. *CAMWA*, 74(7):1607 – 1625, 2017.

C. Hofer and U. Langer. Dual-primal isogeometric tearing and interconnecting solvers for multipatch dG-IgA equations. *CMAME*, 316:2 – 21, 2017.

C. Hofreither and S. Takacs. Robust multigrid for isogeometric analysis based on stable splittings of spline spaces. *SINUM*, 55(4):2004–2024, 2017.

T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *CMAME*, 194:4135–4195, 2005.

A. Klawonn and O. Rheinbach. Inexact FETI-DP methods. *Int. J. Numer. Methods Eng.*, 69(2):284–307, 2007.

A. Klawonn, M. Lanser, and O. Rheinbach. *A Highly Scalable Implementation of Inexact Nonlinear FETI-DP Without Sparse Direct Solvers*, pages 255–264. Springer International Publishing, Cham, 2016.

S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI–isogeometric tearing and interconnecting. *CMAME*, 247:201–215, 2012.

A. Kuzmin, M. Luisier, and O. Schenk. Fast methods for computing selected elements of the Greens function in massively parallel nanoelectronic device simulations. In F. Wolf, B. Mohr, and D. Mey, editors, *Euro-Par 2013*, volume 8097 of *LNCS*, pages 533–544. Springer Berlin Heidelberg, 2013.

J. Schöberl and W. Zulehner. Symmetric Indefinite Preconditioners for Saddle Point Problems with Applications to PDE-Constrained Optimization Problems. *SIMAX*, 29(3):752–773, 2007.

A. Toselli and O. B. Widlund. *Domain decomposition methods – algorithms and theory.* Berlin: Springer, 2005.

Abstract In Isogeometric Analysis (IgA), non-trivial computational domains are often composed of volumetric patches where each of them is discretized by means of tensor-product B-splines or NURBS. In such a setting, the dual-primal IsogEometric Tearing and Interconnecting (IETI-DP) method, that is nothing but the generalization of the FETI-DP method to IgA, has proven to be a very efficient solver for huge systems of IgA equations. Using IETI-DP, basically any patch-local solver can be extended to the global problem. So far, only direct solvers have been considered as patch-local solvers. In the present paper, we compare them with the option of using robust multigrid as patch-local solver. This is of special interest for large-scale patch-local systems or / and for large spline degrees, because the convergence of standard smoothers deteriorates with large spline degrees and the robust multigrid smoother chosen is only available on tensor-product discretizations.