Inexact IETI-DP for Conforming Isogeometric Multi-Patch Discretizations

Rainer Schneckenleitner

Institute of Computational Mathematics, Johannes Kepler University
Altenberger Str. 69, 4040 Linz, Austria

Stefan Takacs

Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences
Altenberger Str. 69, 4040 Linz, Austria

A–4040 LINZ, Altenbergerstraße 69, Austria

Technical Reports before 1998:

1995

95-1	Hedwig Brandstetter	
	Was ist neu in Fortran 90?	March 1995
95-2	G. Haase, B. Heise, M. Kuhn, U. Langer	
	Adaptive Domain Decomposition Methods for Finite and Boundary Element Equations.	August 1995
95-3	Joachim Schöberl	
	An Automatic Mesh Generator Using Geometric Rules for Two and Three Space Dimensions.	August 1995

1996

96-1	Ferdinand Kickinger	
	Automatic Mesh Generation for 3D Objects.	February 1996
96-2	Mario Goppold, Gundolf Haase, Bodo Heise und Michael Kuhn	
	Preprocessing in BE/FE Domain Decomposition Methods.	February 1996
96-3	Bodo Heise	
	A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.	February 1996
96-4	Bodo Heise und Michael Jung	
	Robust Parallel Newton-Multilevel Methods.	February 1996
96-5	Ferdinand Kickinger	
	Algebraic Multigrid for Discrete Elliptic Second Order Problems.	February 1996
96-6	Bodo Heise	
	A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.	May 1996
96-7	Michael Kuhn	
	Benchmarking for Boundary Element Methods.	June 1996

1997

97-1	Bodo Heise, Michael Kuhn and Ulrich Langer	
	A Mixed Variational Formulation for 3D Magnetostatics in the Space $H(rot) \cap H(div)$	February 1997
97-2	Joachim Schöberl	
	Robust Multigrid Preconditioning for Parameter Dependent Problems I: The Stokes-type Case.	June 1997
97-3	Ferdinand Kickinger, Sergei V. Nepomnyaschikh, Ralf Pfau, Joachim Schöberl	
	Numerical Estimates of Inequalities in $H^{\frac{1}{2}}$.	August 1997
97-4	Joachim Schöberl	
	Programmbeschreibung NAOMI 2D und Algebraic Multigrid.	September 1997

Inexact IETI-DP for conforming isogeometric multi-patch discretizations

Rainer Schneckenleitner* and Stefan Takacs†

Abstract

In this paper, we investigate Dual-Primal Isogeometric Tearing and Interconnecting (IETI-DP) methods for conforming Galerkin discretizations on multi-patch computational domains with inexact subdomain solvers. Recently, the authors have proven a condition number estimate for a IETI-DP method using sparse LU factorizations for the subdomain problems that is explicit, among other parameters, in the grid size and the spline degree. In the present paper, we replace the sparse LU factorizations by fast diagonalization based preconditioners to get a faster IETI-DP method while maintaining the same explicit condition number bound.

1 Introduction

The discretization of partial differential equations leads usually to large-scale linear systems. We are interested in a fast solver for linear systems that are obtained from the discretization of boundary value problems using Isogeometric Analysis (IgA; [4]) schemes. We consider computational domains that are composed of multiple non-overlapping patches, for which FETI-DP type algorithms are a canonical choice. Adaptations of FETI-DP, introduced in [1], have already been made to IgA, see, e.g., [6, 2]. This approach is sometimes called Dual-Primal Isogeometric Tearing and Interconnecting (IETI-DP). Recently, a convergence analysis for IETI-DP methods that is

*schneckenleitner@numa.uni-linz.ac.at, Institute of Computational Mathematics, Johannes Kepler University Linz, Austria

†stefan.takacs@ricam.oeaw.ac.at, Johann Radon Institute Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria

explicit in the grid sizes, the patch diameters, the spline degree and other parameters like the smoothness of the splines within the patches or the number of patches was carried out for a conforming Galerkin IgA discretization, see [8]. There, the authors considered a Schur complement IETI-DP method, where the subdomain problems are solved with sparse direct solvers. Large subdomain problems slow down the overall algorithm and require a lot of memory resources. The saddle point formulation of IETI-DP allows the use of inexact local solvers. The successful use of inexact solvers for FETI-DP has already been demonstrated in [3, 5]. In this paper, we use the fast diagonalization (FD) method introduced in [7] as solver for the subdomain-local subproblems. We show that the inexact IETI-DP version obeys the same condition number bound that also holds for the standard IETI-DP method based on exact solvers for the local problems from [8].

The structure of the paper is as follows. Section 2 is devoted to the introduction of the model problem and the IETI-DP solver. Numerical results are presented in Section 3.

2 Model problem and its solution

Let $\Omega \subset \mathbb{R}^2$ be an open bounded Lipschitz domain with boundary $\partial\Omega$ and $f \in L_2(\Omega)$ be a given source function. We consider the following model problem: Find $u \in H_0^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x = \int_\Omega f v \, \mathrm{d}x \qquad \text{for all } v \in H_0^1(\Omega). \tag{1}$$

We assume that Ω is composed of K non-overlapping patches $\Omega^{(k)}$ that are parameterized with geometry mappings

$$G_k : \widehat{\Omega} := (0,1)^d \to \Omega^{(k)} := G_k(\widehat{\Omega}),$$

where the interface between any two patches is either a common vertex or a common edge, cf. [8, Ass. 2]. Additionally, we assume that the number of patches sharing a vertex is uniformly bounded, cf. [8, Ass. 3]. Moreover, we assume that there is a constant C_G such that $\|\nabla G_k\|_{L_\infty(\widehat{\Omega})} \le C_G \operatorname{diam}(\Omega^{(k)})$ and $\|\nabla G_k^{-1}\|_{L_\infty(\widehat{\Omega})} \le C_G \operatorname{diam}(\Omega^{(k)})^{-1}$, see [8, Ass. 1], and that we have quasi-uniform grids [8, Ass. 4]. The local discretization spaces on the parameter domain $\widehat{\Omega}$ are tensor-product B-splines spaces, obtained using the Cox-de Boor formula. The local discretization spaces on the physical patches $\Omega^{(k)}$ are obtained by the pull-back principle. We assume that the geometry mappings

as well as the discretizations agree on all interfaces between the patches, cf. [8, Ass. 5]. So, we are able to set up a fully matching discretization with the function space

$$V = \{v \in H_0^1(\Omega) : v \circ G_k \text{ is a B-spline function}\}.$$

The corresponding discrete problem is obtained by restricting (1) to this space.

In the following, we introduce the IETI-DP solver. The patches from the definition of the computational domain provide a canonical choice of substructures which we use to set up the solver. By assembling the variational problem (1) on the individual patches separately, we obtain independent linear systems

$$A^{(k)}\underline{u}^{(k)} = \underline{f}^{(k)} \quad \text{for} \quad k = 1, \dots, K,$$

where $A^{(k)}$ is the local stiffness matrix and $\underline{f}^{(k)}$ is the local source vector. We use the vertex values as primal degrees of freedom, see [8, Alg. A]. By eliminating the primal degrees of freedom, we obtain the matrices $A_{\Delta\Delta}^{(k)}$ and the vectors $\underline{f}_{\Delta}^{(k)}$ and $\underline{u}_{\Delta}^{(k)}$. Moreover, we introduce jump matrices $B^{(k)}$, where $\sum_{k=1}^{K} B^{(k)}\underline{u}^{(k)} = 0$ ensures the continuity between the patches (except the continuity at the corners), in the usual way, see [8, Section 3]. The matrices $B_{\Delta}^{(k)}$ are obtained from $B^{(k)}$ again by eliminating the primal degrees of freedom.

The $A^{(k)}$-orthogonal primal basis representation $\Psi^{(k)}$ is characterized by the linear system

$$\begin{pmatrix} A^{(k)} & (C^{(k)})^\top \\ C^{(k)} & \end{pmatrix} \begin{pmatrix} \Psi^{(k)} \\ \Delta^{(k)} \end{pmatrix} = \begin{pmatrix} 0 \\ R_c^{(k)} \end{pmatrix},$$

where $C^{(k)}$ is a matrix that selects the coefficients corresponding to the primal degrees of freedom and $R_c^{(k)}$ is a binary local-to-global mapping of the primal constraints. We compute $\Psi^{(k)}$ using a MINRES solver, preconditioned with

$$\widetilde{P}^{(k)} = \begin{pmatrix} (\widehat{A}_M^{(k)})^{-1} & \\ & (C^{(k)}(\widehat{A}_M^{(k)})^{-1}C^{(k)})^{-1} \end{pmatrix},$$

where $\widehat{A}_M^{(k)} := \widehat{A}^{(k)} + \gamma_k \widehat{M}^{(k)}\underline{e}_h^{(k)}(\underline{e}_h^{(k)})^\top \widehat{M}^{(k)}$ is the stiffness matrix on the parameter domain $\widehat{A}^{(k)}$, corrected by a term involving the mass matrix $\widehat{M}^{(k)}$ on the parameter domain and the vector $\underline{e}_h^{(k)}$, which represents the constant function with value 1. We choose $\gamma_k = 1$ if $\Omega^{(k)}$ does not contribute to the

3

Dirichlet boundary $\partial\Omega$ and $\gamma_k = 0$ otherwise. The application of $(\widehat{A}_M^{(k)})^{-1}$ is realized with the fast diagonalization (FD) method, see [7]. After the computation of all $\Psi^{(k)}$, we obtain the global primal basis representation matrix Ψ by canonical mappings. Next, we set up the primal stiffness matrix, the primal source vector and the corresponding jump matrix given by

$$A_\Pi := \sum_{k=1}^{K} (\Psi^{(k)})^\top A^{(k)} \Psi^{(k)}, \quad \underline{f}_\Pi := \Psi^\top \underline{f} \quad \text{and} \quad B_\Pi := \sum_{k=1}^{K} B^{(k)} \Psi^{(k)}.$$

The overall IETI-DP saddle point system reads as follows:

$$\begin{pmatrix} A_{\Delta\Delta}^{(1)} & & & & (B_\Delta^{(1)})^\top \\ & \ddots & & & \vdots \\ & & A_{\Delta\Delta}^{(K)} & & (B_\Delta^{(K)})^\top \\ & & & A_\Pi & B_\Pi^\top \\ B_\Delta^{(1)} & \cdots & B_\Delta^{(K)} & B_\Pi & 0 \end{pmatrix} \begin{pmatrix} \underline{u}_\Delta^{(1)} \\ \vdots \\ \underline{u}_\Delta^{(K)} \\ \underline{u}_\Pi \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \underline{f}_\Delta^{(1)} \\ \vdots \\ \underline{f}_\Delta^{(K)} \\ \underline{f}_\Pi \\ 0 \end{pmatrix}. \tag{2}$$

We solve (2) with a preconditioned MINRES method using the preconditioner

$$\mathcal{P} := \text{diag}\left(Q^{(1)}(\widehat{A}_M^{(1)})^{-1}(Q^{(1)})^\top, \cdots, Q^{(K)}(\widehat{A}_M^{(K)})^{-1}(Q^{(K)})^\top, A_\Pi^{-1}, \widehat{M}_{\text{sD}}\right), \tag{3}$$

where $Q^{(k)}$ is the $A^{(k)}$-orthogonal projection from the local function space into the space $V_\Delta^{(k)} := \{\underline{u}^{(k)} : C^{(k)}\underline{u}^{(k)} = 0\}$ of functions with vanishing primal degrees of freedom and $\widehat{A}_M^{(k)}$ is defined as above and realized using the FD method. A_Π^{-1} is realized by a direct solver. $\widehat{M}_{\text{sD}}$ is an inexact scaled Dirichlet preconditioner, defined by

$$\widehat{M}_{\text{sD}} := B_\Gamma D^{-1} \widehat{S} D^{-1} B_\Gamma^\top, \tag{4}$$

where $\widehat{S} := \text{diag}(\widehat{S}^{(1)}, \ldots, \widehat{S}^{(K)})$ with $\widehat{S}^{(k)} := \widehat{A}_{\Gamma\Gamma}^{(k)} - \widehat{A}_{\Gamma I}^{(k)}(\widehat{A}_{II}^{(k)})^{-1}\widehat{A}_{I\Gamma}^{(k)}$. The diagonal matrix D is based on the principle of multiplicity scaling, cf. [8]. The index Γ refers to the rows/columns of $\widehat{A}^{(k)}$ and the columns of $B^{(k)}$ that refer to basis functions with non-vanishing trace, the index I refers to the remaining rows/columns.

Under the presented assumptions, the condition number of the system (2) preconditioned with (3) is bounded by

$$C\, p \left(1 + \log p + \max_{k=1,\ldots,K} \log \frac{H_k}{h_k}\right)^2,$$

where p is the spline degree, H_k is the diameter of $\Omega^{(k)}$, and h_k the grid size on $\Omega^{(k)}$, and the constant C only depends on the constant C_G, the quasi-uniformity constant and the maximum number of patches sharing a vertex, see [8].

3 Numerical results

In this section, we show numerical results of the proposed inexact IETI-DP method on the computational domains as given in Fig. 1. The first domain is a quarter annulus consisting of 32 patches and the second one is the Yeti-footprint decomposed into 84 patches.
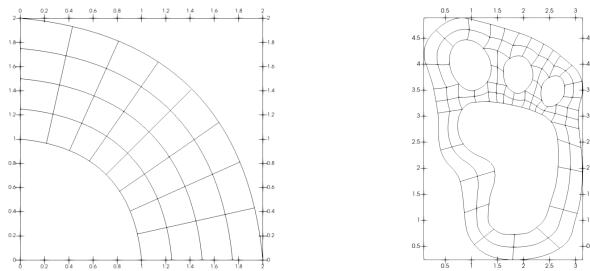

Figure 1: Computational domains; Quarter annulus (left); Yeti-footprint (right)

We consider the model problem

$$-\Delta u(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y) \quad \text{for} \quad (x, y) \in \Omega$$

with homogeneous Dirichlet boundary conditions. Within each patch, we use B-splines of degree p and maximum smoothness C^{p-1}. The coarsest discretization space ($r = 0$) is the space of patchwise global polynomials, only the more rectangular patches of the Yeti-footprint have one inner knot on each of the longer sides of the patches. The subsequent refinements $r = 1, 2, \ldots$ are obtained via uniform refinement steps.

All numerical experiments have been carried out using the C++ library G+Smo[1], the CPU times have been recorded on the Radon1[2] Cluster in Linz.

We compare three different IETI-DP solvers: the proposed solver as introduced in Section 2 (=MFD), a IETI-DP solver for the saddle point system (2) without the primal degrees of freedom eliminated with direct solvers for the local subproblems (=MLU), and the Schur complement based approach as introduced in [8] (=CGLU). We use MINRES as outer solver in the cases MFD and MLU and conjugate gradient as outer solver for the case CGLU. For MLU and CGLU, we use sparse direct LU solvers from the Pardiso project[3] for the local subproblems and for computing the bases $\Psi^{(k)}$. For

---

[1] https://github.com/gismo/gismo
[2] https://www.ricam.oeaw.ac.at/hpc/
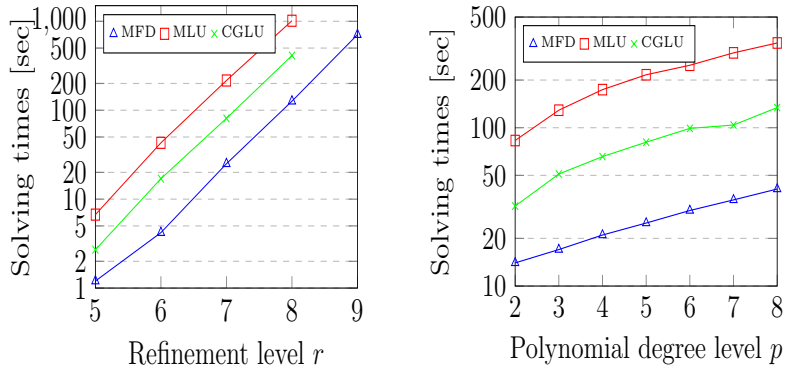[3] https://www.pardiso-project.org/

Figure 2: Solving times for $p = 5$ (left) and $r = 7$ (right); MFD (blue lines); MLU (red lines); CGLU (green lines); Annulus
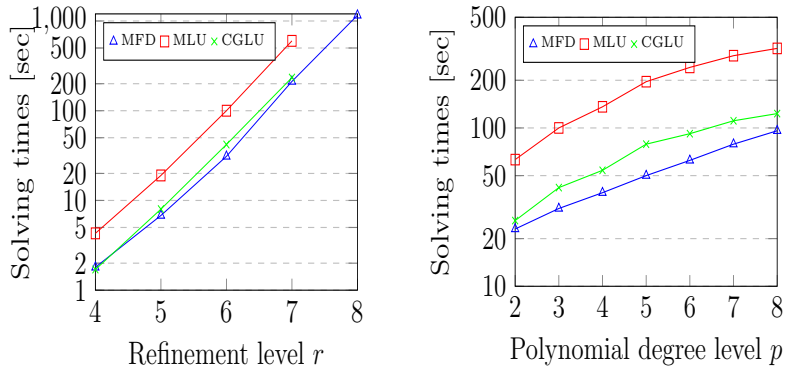

Figure 3: Solving times for $p = 3$ (left) and $r = 6$ (right); MFD (blue lines); MLU (red lines); CGLU (green lines); Yeti-footprint

MFD, the primal bases $\Psi^{(k)}$ are solved with MINRES up to an accuracy of 10^{-8}. We present the times required for computing a primal basis Ψ which we indicate in the tables with the same letter, the accumulated setup and application times of the different local preconditioners for all patches K, abbreviated by Θ_S and Θ_A, respectively as well as the solving times and the number of iterations (it.) required by the main Krylov space solver. We start all numerical experiments with zero initial guess and stop the iterations if the ℓ_2-norm of the residual vector is reduced by a factor of 10^{-6} compared to the ℓ_2-norm of the right-hand side vector.

The Tables 1 and 2 present the timings of the algorithms on the quarter annulus domain. We observe that the solving and total times required by

	r	Ψ	Θ_S	Θ_A	solving	total	it.
MFD	6	1.2	0.1	0.4	4.2	5.5	71
MLU		0.8	7.6	21.2	43.0	51.4	37
CGLU		0.8	7.6	9.5	17.0	15.4	15
MFD	7	8.0	0.3	4.9	25.0	33.3	80
MLU		4.0	42.3	106.4	216.0	262.3	39
CGLU		4.0	42.3	45.4	81.0	127.3	15
MFD	8	35.5	2.0	26.9	126.3	163.8	88
MLU		18.1	243.6	503.8	1015.0	1276.7	41
CGLU		18.0	242.1	225.3	412.0	672.1	17

Table 1: Alg. A; $p = 5$; timings in sec.; Annulus

	r	Ψ	Θ_S	Θ_A	solving	total	it.
MFD	6	2.5	0.2	0.5	8.2	10.9	76
MLU		1.2	17.9	33.8	72.0	91.1	39
CGLU		1.2	17.8	14.4	26.0	45.0	15
MFD	7	15.1	0.4	5.4	41.0	56.5	84
MLU		5.8	105.0	163.3	343.0	453.8	41
CGLU		5.8	105.0	73.8	134.0	244.8	17
MFD	8	60.1	2.3	29.9	200.0	262.4	94
MLU				OoM			
CGLU				OoM			

Table 2: Alg. A; $p = 8$; timings in sec.; Annulus

MFD are three to five times smaller compared to the other methods MLU and CGLU. MFD is much faster than the other methods despite the fact that the required number of iterations are up to approximately six times larger. One disadvantage of MFD is the computation of the primal basis Ψ. The tables show a larger computation time when using MFD. This is a weakness of the classical preconditioned MINRES method when applied to problems with multiple right-hand sides. In general, we have to solve systems with four right-hand sides to compute $\Psi^{(k)}$. Moreover, we see that the setup and application of the FD based preconditioner is much faster compared to the factorizations of

$$\begin{pmatrix} A^{(k)} & (C^{(k)})^\top \\ C^{(k)} & \end{pmatrix}$$

and their application. Table 2 shows another advantage of the MFD method. Since its memory footprint is smaller, it also provides a solution vector to the considered linear system for the refinement level $r = 8$. The plots in Fig. 2 vi-

	r	Ψ	Θ_S	Θ_A	solving	total	it.
MFD	5	0.7	0.1	0.8	6.8	7.6	213
MLU		0.3	2.3	9.4	19.0	21.6	45
CGLU		0.3	2.3	4.5	8.0	10.6	20
MFD	6	3.4	0.2	5.0	31.0	34.6	242
MLU		1.2	10.5	49.6	100.0	111.7	51
CGLU		1.2	10.4	23.1	42.0	53.6	22
MFD	7	21.8	1.2	52.7	212.0	235.0	274
MLU		6.8	50.3	295.2	600.5	657.6	55
CGLU		6.8	50.1	126.1	234.0	290.9	22

Table 3: Alg. A; $p = 3$; timings in sec.; Yeti-footprint

	r	Ψ	Θ_S	Θ_A	solving	total	it.
MFD	5	2.2	0.2	1.2	20.0	22.4	249
MLU		0.6	10.5	23.5	48.0	59.1	51
CGLU		0.6	10.4	10.9	19.0	30.0	22
MFD	6	9.5	0.4	5.9	79.0	88.9	282
MLU		3.0	44.8	140.5	286.0	333.8	57
CGLU		3.0	44.8	60.6	111.0	158.8	23
MFD	7	54.2	1.5	63.2	414.0	469.7	309
MLU		14.1	265.3	739.1	1521.0	1800.4	61
CGLU		13.8	261.7	304.4	570.0	845.5	25

Table 4: Alg. A; $p = 7$; timings in sec.; Yeti-footprint

sualize solving times of the IETI-DP solvers on the quarter annulus domain. We mark the performance of MFD with blue lines and triangles, MLU with red lines and squares and the performance of CGLU is indicated with green lines and crosses. In both graphs, we observe that MFD is the fastest algorithm. In the left plot, we see that for spline degree $p = 5$, the solving times increase rather linearly with respect to the number of unknowns. Moreover, the left graph shows that MFD computes the solution for the linear system even for refinement level $r = 9$ ($\approx 8.5\text{M dofs}$). In the right graph, we present the solving times with respect to the spline degree for refinement level $r = 7$. The solving times for the three IETI-DP solvers increase about linearly with the spline degree.

The plots in Fig. 3 show solving times of the IETI-DP solvers on the Yeti-footprint. We marked the performance of the different IETI-DP solvers as above in the experiments on the quarter annulus. The plot on the left

shows the increase of the solving time with respect to the refinement level with polynomial degree $p = 3$ and the plot on the right shows the increase of the solving time with respect to the polynomial degree, where we have fixed the refinement level to $r = 6$. As on the quarter annulus, we see that MFD is superior compared to MLU and CGLU also on the Yeti-footprint with respect to the solving times and the smaller memory footprint of MFD allows us to consider larger problems. In both plots, we observe similar growth rates of the solving time for all three solvers as in Fig. 2. In the Tables 3 and 4, we present and compare the required timings for the polynomial degrees $p = 3$ and $p = 7$ for different refinement levels on the Yeti-footprint.

To conclude, we presented a fast IETI-DP method which allows the incorporation of inexact solvers for the local subproblems while maintaining the condition number bound as established in [8]. It is beneficial both because of its smaller memory footprint and its faster convergence for the model problems.

References

[1] C. Farhat, M. Lesoinne, P. L. Tallec, K. Pierson, and D. Rixen. FETI-DP: A dual-primal unified FETI method I: A faster alternative to the two-level FETI method. *Int. J. Numer. Methods Eng.*, 50:1523 – 1544, 2001.

[2] C. Hofer and U. Langer. Dual-primal isogeometric tearing and interconnecting solvers for multipatch continuous and discontinuous Galerkin IgA equations. *PAMM*, 16(1):747–748, 2016.

[3] C. Hofer, U. Langer, and S. Takacs. Inexact Dual-Primal Isogeometric Tearing and Interconnecting Methods. In P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, and O. B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXIV*, pages 393–403. Springer International Publishing, 2018.

[4] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.*, 194(39-41):4135 – 4195, 2005.

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

[6] S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI-Isogeometric Tearing and Interconnecting. *Comput. Methods Appl. Mech. Eng.*, 247-248:201 – 215, 2012.

[7] G. Sangalli and M. Tani. Isogeometric preconditioners based on fast solvers for the Sylvester equation. *SIAM J. Sci. Comput.*, 38(6):A3644 – A3671, 2016.

[8] R. Schneckenleitner and S. Takacs. Condition number bounds for IETI-DP methods that are explicit in h and p. *Math. Models Methods Appl. Sci.*, 30(11):2067 – 2103, 2020.

Latest Reports in this series

2009 - 2019

[..]

2020

2021