A Block ILUT Smoother for Multipatch Geometries in Isogeometric Analysis

Roel Tielen, Matthias Möller, and Kees Vuik

Abstract Since its introduction in [20], Isogeometric Analysis (IgA) has established itself as a viable alternative to the Finite Element Method (FEM). Solving the resulting linear systems of equations efficiently remains, however, challenging when high-order B-spline basis functions of order $p > 1$ are adopted for approximation. The use of Incomplete LU (ILU) type factorizations, like ILU(k) or ILUT, as a preconditioner within a Krylov method or as a smoother within a multigrid method is very effective, but costly [37]. In this paper, we investigate the use of a block ILUT smoother within a p-multigrid method, where the coarse grid correction is obtained at $p = 1$, and compare it to a global ILUT smoother in case of multipatch geometries. A spectral analysis indicates that the use of the block ILUT smoother improves the overall convergence rate of the resulting p-multigrid method. Numerical results, obtained for a variety of two dimensional benchmark problems, illustrate the potential of this block ILUT smoother for multipatch geometries.

1 Introduction

Isogeometric Analysis [20] has established itself as a viable alternative to the Finite Element Method (FEM). The use of high-order B-spline basis functions allows for an accurate description of curved geometries, thereby bridging the gap between Computer Aided Design (CAD) and Finite Element Analysis (FEA). Furthermore, a higher accuracy per degree of freedom can be obtained with high-order B-spline basis functions compared to standard FEM [21].

Solving the resulting linear systems of equations efficiently remains, however, a challenging task. The condition number of the resulting system matrices grows

R. Tielen (✉) · M. Möller · K. Vuik
Delft University of Technology, Delft, The Netherlands
e-mail: r.p.w.m.tielen@tudelft.nl

M. Möller
e-mail: m.moller@tudelft.nl

K. Vuik
e-mail: c.vuik@tudelft.nl

C. Manni and H. Speleers (eds.), *Geometric Challenges in Isogeometric Analysis*,
Springer INdAM Series 49, https://doi.org/10.1007/978-3-030-92313-6_11

exponentially with the approximation order p [12], making the use of (standard) iterative solvers not straightforward. Over the years, special preconditioners have been developed [3, 19, 34] to obtain convergence rates independent of the approximation order p. A viable alternative are multigrid methods [5, 15], which can be designed to be of optimal complexity for elliptic problems. Multigrid methods combine the use of a smoother (e.g. Gauss-Seidel) and a correction obtained on coarser levels. Typically, this correction is obtained based on a coarser discretization, leading to so-called h-multigrid methods. The use of h-multigrid methods in IgA has been investigated in detail, see for example [10, 12, 17, 18, 27, 32].

Alternatively, for high-order methods, the correction can be determined at a low-order level as well, resulting in p-multigrid methods [11, 16, 22, 23, 31]. Recently, the use of p-multigrid methods has been investigated in the context of Isogeometric Analysis [35–38]. Throughout this paper, p-multigrid methods will be adopted as a stand-alone solver or as a preconditioner within a Krylov solver. More precisely, a two-level method is considered, where the second level (i.e. the "coarse" level) corresponds to a discretization of order $p = 1$. Furthermore, k-refinement is applied throughout this paper, implying that the considered spaces are non-nested.

In recent years, Incomplete LU (ILU) factorizations have gained increasing popularity within Isogeometric Analysis. As a preconditioner, ILU(k) has shown to be very efficient [9]. When applied as a smoother in multigrid methods [37], convergence rates independent of p have been observed. A drawback of ILU factorizations are, however, the high setup costs, making their use as a smoother or preconditioner within a solver costly [37].

In this paper, we focus on the use of ILUT factorizations as a smoother within multigrid methods. In particular, we propose the use of a block ILUT smoother in case of multipatch geometries. The smoother is based on a splitting of the domain in multiple patches and has been applied successfully in the context of domain decomposition methods for reservoir simulation problems [26].

This paper is organized as follows. In Sect. 2, we present a model problem that will be considered throughout this paper. Section 3 presents the adopted p-multigrid method. The global and block ILUT smoother considered in this paper are presented in Sect. 4. The performance of both ILUT smoothers is compared for several two dimensional benchmark problems in Sect. 5. Furthermore, both smoothers are applied to a challenging benchmark, a Yeti footprint consisting of 21 patches, in Sect. 6. Finally, conclusions are drawn in Sect. 7.

2 Model Problem

As a model problem, we consider the convection-diffusion-reaction (CDR) equation on a connected, Lipschitz domain $\Omega \subset \mathbb{R}^d$ (where d is the spatial dimension):

$$-\nabla \cdot (\mathbf{D}\nabla u) + \mathbf{v} \cdot \nabla u + Ru = f, \quad \text{on } \Omega. \tag{1}$$

Here, $\mathbf{D}$ denotes the diffusion tensor, $\mathbf{v}$ a divergence-free velocity field and R a reaction term. Furthermore, we have $f \in L^2(\Omega)$ and $u = 0$ on the boundary $\partial\Omega$. Let $\mathscr{V} = H_0^1(\Omega)$ be defined as the Sobolev space $H^1(\Omega)$ with functions that vanish on $\partial\Omega$. The variational form of the CDR-equation is obtained by multiplying Eq. (1) with a test function $v \in \mathscr{V}$, integrating over the domain Ω and applying integration by parts to the second-order term:

$$\int_\Omega (\mathbf{D}\nabla u) \cdot \nabla v + (\mathbf{v} \cdot \nabla u)v + Ruv \, \mathrm{d}\Omega = \int_\Omega f v \, \mathrm{d}\Omega \qquad \forall v \in \mathscr{V}. \tag{2}$$

The physical domain Ω is then parameterized by a geometry function that describes an invertible mapping connecting the parameter domain $\Omega_0 = (0, 1)^d$ with the physical domain Ω:

$$\mathbf{F} : \Omega_0 \to \Omega, \qquad \mathbf{F}(\boldsymbol{\xi}) = \mathbf{x} \in \Omega \quad \forall \boldsymbol{\xi} \in \Omega_0. \tag{3}$$

Many physical domains of practical relevance cannot be modelled by a single geometry function, e.g. a channel with one or more obstacles. In that case, Ω is divided into a collection of non-overlapping subdomains $\Omega^{(k)}$ such that

$$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}^{(k)} \tag{4}$$

and we refer to Ω as a multipatch geometry consisting of K patches. For each $\Omega^{(k)}$, a geometry function $\mathbf{F}^{(k)}$ is then defined to parameterize each subdomain individually.

$$\mathbf{F}^{(k)} : \Omega_0 \to \Omega^{(k)}, \qquad \mathbf{F}^{(k)}(\boldsymbol{\xi}) = \mathbf{x} \in \Omega^{(k)} \quad \forall \boldsymbol{\xi} \in \Omega_0. \tag{5}$$

Figure 1 shows a multipatch geometry, consisting of 4 patches. For the spatial discretization, multivariate B-spline basis functions are adopted which are constructed as tensor product of univariate B-spline basis functions. The latter ones are defined on the parameter domain (0, 1) and are uniquely determined by their underlying knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{N+p}, \xi_{N+p+1}\}$, consisting of a sequence of non-decreasing knots $\xi_i \in (0, 1)$. Here, N denotes the number of basis functions and p is their polynomial order. B-spline basis functions are defined recursively by the Cox-de Boor formula [4].

The resulting B-spline basis functions $\phi_{i,p}$, $(i = 1, \ldots, N)$, are non-zero on the interval $[\xi_i, \xi_{i+p+1})$, implying a compact support that increases with p. As a consequence, system matrices resulting from the substitution of B-spline basis functions into Galerkin formulations of differential equations are sparse, although the number of non-zero entries increases with p. Furthermore, at every knot ξ_i the basis functions are C^{p-m_i}-continuous, where m_i denotes the multiplicity of knot ξ_i. Finally, the basis functions possess the partition of unity property and are non-negative:

$$\sum_{i=1}^{N} \phi_{i,p}(\xi) = 1 \quad \forall \xi \in [\xi_1, \xi_{n+p+1}], \tag{6}$$

$$\phi_{i,p}(\xi) \geq 0 \quad \forall \xi \in [\xi_1, \xi_{n+p+1}]. \tag{7}$$

As a consequence, mass matrices can be lumped within the variational formulation.

Throughout this paper, B-spline basis functions are considered based on an open uniform knot vector with knot span size h, that is, the first and last knots are repeated $p+1$ times. Therefore, the basis functions considered are C^{p-1} continuous and interpolatory only at the two end points.

Multivariate B-spline basis functions $\Phi_{i,p}$ $(i = 1, \ldots, N_{\text{dof}})$ are then defined as the tensor product of univariate basis functions of order p. Let

$$\mathscr{V}_{h,p} := \text{span}\left\{\Phi_{i,p} \circ \mathbf{F}^{-1}\right\}_{i=1,\ldots,N_{\text{dof}}}. \tag{8}$$

Then, the Galerkin formulation of (2) becomes: Find $u_{h,p} \in \mathscr{V}_{h,p}$ such that

$$a(u_{h,p}, v_{h,p}) = (f, v_{h,p}) \qquad \forall v_{h,p} \in \mathscr{V}_{h,p}. \tag{9}$$

where

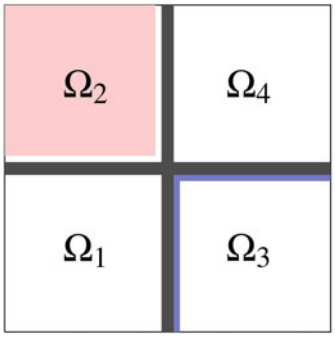
$$a(u, v) = \int_{\Omega} (\mathbf{D}\nabla u) \cdot \nabla v + (\mathbf{v} \cdot \nabla u)v + Ruv \, \mathrm{d}\Omega, \qquad (f, v) = \int_{\Omega} fv \, \mathrm{d}\Omega. \tag{10}$$

The discretized problem can be written as a linear system

$$\mathbf{A}_{h,p}\mathbf{u}_{h,p} = \mathbf{f}_{h,p}, \tag{11}$$

where $\mathbf{A}_{h,p}$ denotes the system matrix. For a more detailed description of the spatial discretization in Isogeometric Analysis, we refer to [20].

Figure 1 illustrates a multipatch geometry consisting of 4 patches $\Omega^{(k)}$, $k = 1, \ldots, 4$, and the resulting block structure of the system matrix $\mathbf{A}_{h,p}$. The first 4 diagonal blocks are associated with the interior degrees of freedom on Ω_i, $i = 1, \ldots, 4$, while the right-bottom block, shown in grey, denotes the degrees of freedom at the interface Γ. Finally, the off-diagonal blocks denote the coupling between degrees of freedom at the interior and the interface, resulting in an arrowhead matrix. It should be noted that this structure is related to a particular numbering of the degrees of freedom. Note that this block structure is common within domain decomposition (DD) methods [33], which decompose the global domain Ω into subproblems and solve small boundary value problems per subdomain individually using data at the interface as boundary conditions. A coarse problem is then typically solved to exchange information about the solution between the subdomains. When non-overlapping subdomains are considered, the resulting block structure is similar to the structure shown in Fig. 1. The definition of the block ILUT smoother considered in this paper is based on this structure and will be discussed in detail in Section 4.



$$\begin{bmatrix} \mathbf{A_{11}} & 0 & 0 & 0 & \mathbf{A_{\Gamma 1}} \\ 0 & \mathbf{A_{22}} & 0 & 0 & \mathbf{A_{\Gamma 2}} \\ 0 & 0 & \mathbf{A_{33}} & 0 & \mathbf{A_{\Gamma 3}} \\ 0 & 0 & 0 & \mathbf{A_{44}} & \mathbf{A_{\Gamma 4}} \\ \mathbf{A_{1\Gamma}} & \mathbf{A_{2\Gamma}} & \mathbf{A_{3\Gamma}} & \mathbf{A_{4\Gamma}} & \mathbf{A_{\Gamma\Gamma}} \end{bmatrix}$$

Fig. 1 A multipatch geometry, consisting of 4 patches and the resulting block structure of the system matrix. The block associated to the interior degrees of freedom of Ω_2 is highlighted in red, while the coupling between the interface Γ and Ω_3 is highlighted in blue. Furthermore, the degrees of freedom at Γ result in the gray block

3 *p*-Multigrid Method

In order to solve the linear system from Eq. (11), we consider the p-multigrid method presented in [37]. With a p-multigrid method, a correction is obtained at level $p = 1$ by a direct projection from the high-order level. Starting from the high-order problem, the following steps are performed:

1. Apply a fixed number ν_1 of presmoothing steps to the initial guess $\mathbf{u}_{h,p}^{(0,0)}$:

$$\mathbf{u}_{h,p}^{(0,m)} = \mathbf{u}_{h,p}^{(0,m-1)} + \mathscr{S}_{h,p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(0,m-1)}\right), \quad m = 1, \ldots, \nu_1, \tag{12}$$

   where $\mathscr{S}_{h,p}$ is a smoothing operator applied to the high-order problem.
2. Determine the residual at level p and project it onto the space $\mathscr{V}_{h,1}$ using the restriction operator $\mathscr{I}_p^1$:

$$\mathbf{r}_{h,1} = \mathscr{I}_p^1\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(0,\nu_1)}\right). \tag{13}$$

3. Solve the residual equation to determine the coarse grid error:

$$\mathbf{A}_{h,1}\mathbf{e}_{h,1} = \mathbf{r}_{h,1}. \tag{14}$$

4. Project the error $\mathbf{e}_{h,1}$ onto the space $\mathscr{V}_{h,p}$ using the prolongation operator $\mathscr{I}_1^p$ and update $\mathbf{u}_{h,p}^{(0,\nu_1)}$:

$$\mathbf{u}_{h,p}^{(0,\nu_1)} := \mathbf{u}_{h,p}^{(0,\nu_1)} + \mathscr{I}_1^p\left(\mathbf{e}_{h,1}\right). \tag{15}$$

5. Apply ν_2 postsmoothing steps of the form (12) to obtain $\mathbf{u}_{h,p}^{(0,\nu_1+\nu_2)} =: \mathbf{u}_{h,p}^{(1,0)}$.

Note that, in contrast to h-multigrid methods, the number of degrees of freedom for the residual equation from Eq. (14) can still be relatively large. Therefore, we apply a single W-cycle of a standard h-multigrid method [15], using canonical prolongation and weighted restriction, to solve Eq. (14) approximately. This corresponds to a low-order Lagrange discretization, and hence, h-multigrid methods (using Gauss-Seidel as a smoother) are known to be both efficient and cheap [15, 39]. The resulting p-multigrid is shown in Fig. 2.

Instead of applying a single W-cycle, the h-multigrid method could be applied until the residual is reduced by a fixed factor as well. By doing so, the h-multigrid method basically becomes a solver, leaving the high-order errors to the (more expensive) ILUT smoother. Numerical results show, however, that the total number of p-multigrid iterations is not affected by this for the benchmarks considered in this paper. Therefore, only a single W-cycle is applied at level $p = 1$ throughout the remainder of this paper.

To project the error to $\mathcal{V}_{h,p}$ and the residual to $\mathcal{V}_{h,1}$, prolongation and restriction operators based on an L_2 projection are adopted which have been used extensively in the literature [6, 7, 30]. The prolongation operator $\mathcal{I}_1^p : \mathcal{V}_{h,1} \to \mathcal{V}_{h,p}$ is given by

$$\mathcal{I}_1^p(\mathbf{v}_1) = (\mathbf{M}_p)^{-1}\mathbf{P}_1^p\,\mathbf{v}_1. \tag{16}$$

Here, the mass matrix $\mathbf{M}_p$ and transfer matrix $\mathbf{P}_1^p$ are defined as follows:

$$(\mathbf{M}_p)_{(i,j)} := \int_\Omega \Phi_{i,p}\Phi_{j,p}\,\mathrm{d}\Omega, \qquad (\mathbf{P}_1^p)_{(i,j)} := \int_\Omega \Phi_{i,p}\Phi_{j,1}\,\mathrm{d}\Omega. \tag{17}$$

The restriction operator $\mathcal{I}_p^1 : \mathcal{V}_{h,p} \to \mathcal{V}_{h,1}$ is defined by

$$\mathcal{I}_p^1(\mathbf{v}_p) = (\mathbf{M}_1)^{-1}\mathbf{P}_p^1\,\mathbf{v}_p. \tag{18}$$

To prevent the explicit solution of a linear system of equations for each projection step, the consistent mass matrix $\mathbf{M}$ in both transfer operators is replaced by its lumped counterpart $\mathbf{M}^L$ by applying row-sum lumping:

$$\mathbf{M}^L_{(i,i)} = \sum_{j=1}^{N_{\mathrm{dof}}} \mathbf{M}_{(i,j)}. \tag{19}$$

Note that, row-sum lumping can be applied within the variational formulation, due to the partition of unity and non-negativity of the B-spline basis functions. Alternatively, the mass matrix could be inverted by exploiting the tensor product structure, cf. [13]. Numerical results, not presented in this paper, indicate that the use of a lumped mass matrix does not influence the overall convergence or accuracy of the p-multigrid method. Note that this choice of prolongation and restriction operator leads to a non-symmetric multigrid operator even for symmetric PDE problems. As a consequence, the resulting p-multigrid method can only be applied as a preconditioner to a Krylov

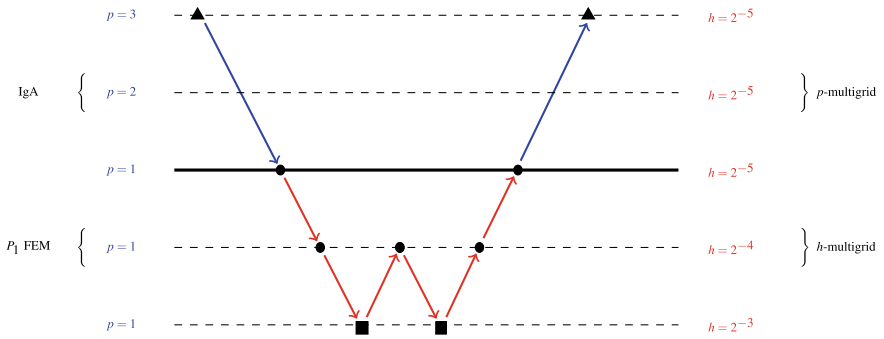


Fig. 2 Illustration of the p-multigrid method [37]. At $p = 1$, Gauss-Seidel is always adopted as a smoother (●), whereas at the high-order level different smoothers can be applied (▲). At the coarsest level, a direct solver is applied to solve the residual Eq. (■)

solver suited for non-symmetric matrices like the stabilized Bi-Conjugate Gradient (Bi-CGSTAB) method. Choosing the prolongation and restriction operator as the transpose of each other would restore symmetry. However, numerical experiments, not presented in this paper, indicate that his choice leads to a less robust p-multigrid method [37].

Different choices can be made with respect to the smoother. The use of Gauss-Seidel or (damped) Jacobi as a smoother at level p leads to convergence rates that depend significantly on the approximation order p [35]. Alternative smoothers have been developed in recent year to overcome this shortcoming [10, 18, 27, 32]. In particular, the use of ILUT factorizations has shown to be very effective. However, the costs of setting up the smoother (i.e. performing the ILUT factorization) are relatively high compared to other smoothers. A computationally efficient alternative consists in using a *block* ILUT smoother, which lends itself to parallelization. Therefore, we consider a *block* ILUT smoother in this paper and compare it to a global ILUT smoother.

4 (Block) ILUT Smoother

The global ILUT smoother is based on an Incomplete LU factorization with dual Threshold strategy [28]. The factorization is based on two parameters, the fill factor f and threshold τ that specifies that all smaller entries of the factorization are dropped. Furthermore, the maximum number of nonzero entries in each row is f times the average number of nonzero entries in a row of the original operator. Throughout this paper, we choose $f = 1$ and $\tau = 10^{-13}$ so that the number of nonzero entries in the ILUT factorization is comparable to the number of nonzero entries of the original operator $\mathbf{A}_{h,p}$.

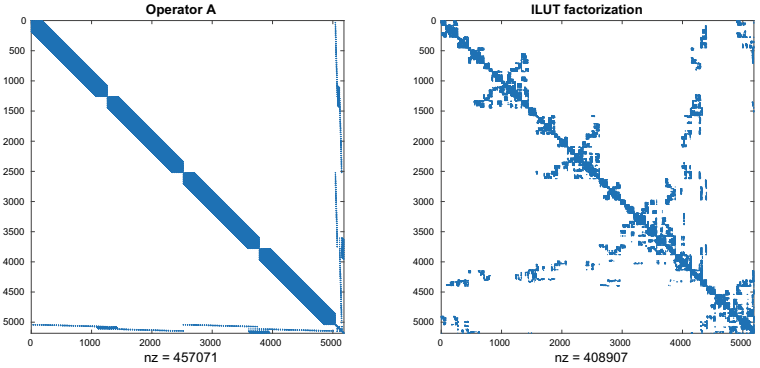


Fig. 3 Sparsity pattern of $\mathbf{A}_{h,p}$ (left) and $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ (right) for $p = 4$, $h = 2^{-5}$ and 4 patches. The number of nonzero (nz) entries for both operators is provided as well

A typical sparsity pattern of the system matrix for a multipatch geometry is shown in Fig. 3 (left). The sparsity pattern of the resulting (global) ILUT factorization plotted as $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ is shown in Fig. 3 (right). Here, we consider a discretization with $p = 4$ and $h = 2^{-5}$, consisting of 4 patches. Since an approximate minimum degree (AMD) ordering [1] is applied during the ILUT factorization to reduce the fill-in, sparsity patterns differ significantly. Note that, indeed, $\mathbf{A}_{h,p}$ and $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ have a comparable number of nonzero entries.

An efficient implementation of ILUT is available in the Eigen library [14] based on [28, 29]. Once the factorization is obtained when setting up the multigrid solver, a single smoothing step at multigrid cycle i is applied as follows:

$$\mathbf{u}_{h,p}^{(i,n+1)} = \mathbf{u}_{h,p}^{(i,n)} + (\mathbf{L}_{h,p}\mathbf{U}_{h,p})^{-1}(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(i,n)}), \tag{20}$$

$$= \mathbf{u}_{h,p}^{(i,n)} + \mathbf{U}_{h,p}^{-1}\mathbf{L}_{h,p}^{-1}(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(i,n)}), \tag{21}$$

where the two matrix inversions in Eq. (21) amount to forward and backward substitutions.

In the following, we consider a block ILUT smoother as an alternative. This smoother is based on the observation that, in case of a multipatch geometry consisting of K patches, the resulting system matrix can be written as follows (see Fig. 1):

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & & \mathbf{0} & \mathbf{A}_{\Gamma 1} \\ & \ddots & & \vdots \\ \mathbf{0} & & \mathbf{A}_{KK} & \mathbf{A}_{\Gamma K} \\ \mathbf{A}_{1\Gamma} & \cdots & \mathbf{A}_{K\Gamma} & \mathbf{A}_{\Gamma\Gamma} \end{bmatrix} \tag{22}$$

Note that, in the remainder of this section, we drop the h and p as subscripts to improve readability.

In [26] a preconditioner $\mathscr{P}$ is defined consisting of a coarse part $\mathscr{P}_C$ and fine part $\mathscr{P}_F$ in a multiplicative way. That is, the iteration matrix of the resulting preconditioner is obtained by multiplying the iteration matrix of both preconditioners:

$$\begin{aligned} \mathbf{I} - \mathscr{P}\mathbf{A} &= (\mathbf{I} - \mathscr{P}_C\mathbf{A})(\mathbf{I} - \mathscr{P}_F\mathbf{A}), && (23)\\ &= \mathbf{I} - \mathscr{P}_F\mathbf{A} - \mathscr{P}_C\mathbf{A} + \mathscr{P}_F\mathbf{A}\mathscr{P}_C\mathbf{A} && (24) \end{aligned}$$

and hence:

$$\mathscr{P} = \mathscr{P}_F + \mathscr{P}_C - \mathscr{P}_F\mathbf{A}\mathscr{P}_C. \tag{25}$$

The coarse part consists of a subdomain deflation approach [25] and takes care of the low frequency components. However, it is known from multigrid that the coarse grid correction decreases the low frequency components of the error. We therefore only adopt the fine part of the preconditioner (i.e. $\mathscr{P}_F$) in our block ILUT smoother and ignore $\mathscr{P}_C$. Hence, we have $\mathscr{P} = \mathscr{P}_F$ and will use the linear iteration with respect to $\mathscr{P}$ as our smoother.

Based on the considerations above, the smoother $\mathscr{P}$ is then defined by observing that $\mathbf{A}$ can be written as follows:

$$\mathbf{A} = \mathbf{L}\mathbf{U} = \begin{bmatrix} \mathbf{L}_1 & & & \\ & \ddots & & \\ & & \mathbf{L}_K & \\ \mathbf{B}_1 & \cdots & \mathbf{B}_K & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 & & & \mathbf{C}_1 \\ & \ddots & & \vdots \\ & & \mathbf{U}_K & \mathbf{C}_K \\ & & & \mathbf{S} \end{bmatrix}. \tag{26}$$

Here, $\mathbf{A}_{ii} = \mathbf{L}_i\mathbf{U}_i$ is based on a complete LU factorization. Furthermore, we have $\mathbf{B}_i = \mathbf{A}_{i\Gamma}\mathbf{U}_i^{-1}$ and $\mathbf{C}_i = \mathbf{L}_i^{-1}\mathbf{A}_{\Gamma i}$. Finally, we have $\mathbf{S} = \mathbf{A}_{\Gamma\Gamma} - \sum_{i=1}^{K} \mathbf{B}_i\mathbf{C}_i$. These choices indeed imply that $\mathbf{A} = \mathbf{L}\mathbf{U}$.

A smoother can now be obtained by replacing the LU factorizations by ILUT factorizations:

$$\mathbf{A} \approx \tilde{\mathbf{L}}\tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\mathbf{L}}_1 & & & \\ & \ddots & & \\ & & \tilde{\mathbf{L}}_K & \\ \tilde{\mathbf{B}}_1 & \cdots & \tilde{\mathbf{B}}_K & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}}_1 & & & \tilde{\mathbf{C}}_1 \\ & \ddots & & \vdots \\ & & \tilde{\mathbf{U}}_K & \tilde{\mathbf{C}}_K \\ & & & \tilde{\mathbf{S}} \end{bmatrix}. \tag{27}$$

Here, $\mathbf{A}_{ii} \overset{(!)}{\approx} \tilde{\mathbf{L}}_i\tilde{\mathbf{U}}_i$ and all other blocks ($\tilde{\mathbf{B}}_i = \mathbf{A}_{i\Gamma}\tilde{\mathbf{U}}_i^{-1}$, $\tilde{\mathbf{C}}_i = \tilde{\mathbf{L}}_i^{-1}\mathbf{A}_{\Gamma i}$ and $\tilde{\mathbf{S}}$) are based on $\tilde{\mathbf{L}}_i$ and $\tilde{\mathbf{U}}_i$. Figure 4 depicts the sparsity pattern of the operator $\mathbf{A}_{h,p}$ and the block ILUT factorization $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ for $p = 4$, $h = 2^{-4}$ and 4 patches. Note that, compared to the global ILUT factorization, the factorizations are obtained within each block. As a consequence, all non-zero entries of the global matrix $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ stay within these blocks.

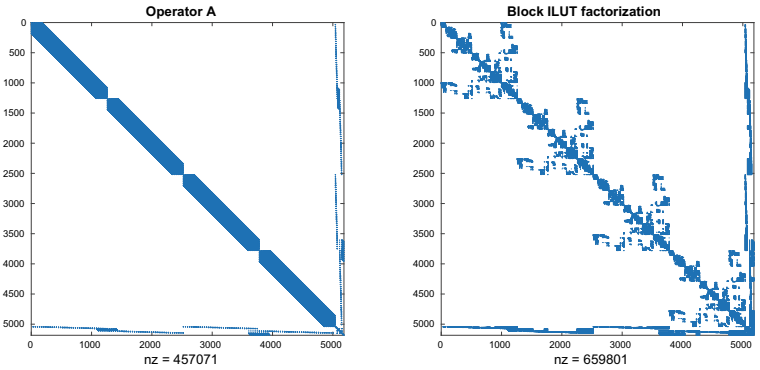


Fig. 4 Sparsity pattern of $\mathbf{A}_{h,p}$ (left) and $\mathbf{L}_{h,p} + \mathbf{U}_{h,p}$ (right) for $p = 4$, $h = 2^{-5}$ and 4 patches. The number of nonzero (nz) entries for both operators is provided as well

Computational Costs

In this subsection, the computational costs of the p-multigrid method adopting both smoothers are discussed. First, we consider the assembly costs of the considered operators after which both set-up and application costs of the block ILUT and global ILUT smoother are presented.

Assuming an element-based assembly loop with standard Gauss-quadrature, the assembly costs at level p for the stiffness matrix and transfer matrix are $\mathcal{O}(N_{\text{dof}}\, p^{3d})$ floating point operations (flops) [9]. Note that the stiffness matrix has to be assembled both at the high-order level and level $p = 1$. Furthermore, the (variationally) lumped mass matrices have to be assembled for the prolongation and restriction operator, at the cost of $\mathcal{O}(N_{\text{dof}})$ flops. It was shown in [37] that the assembly costs form a significant part of the total computational costs within the p-multigrid method. Alternative (and more efficient) assembly techniques exist [2, 8, 24], but will not be explored in this paper.

At the low-order level, Gauss-Seidel is applied as a smoother, which costs $\mathcal{O}(N_{\text{dof}})$ flops per smoothing step and has no set-up costs. The costs of setting up the block ILUT and global ILUT smoother at the high-order level differ significantly. The global ILUT factorization of $\mathbf{A}_{h,p}$ costs $\mathcal{O}(N_{\text{dof}}\, p^{2d})$ flops, provided that $f = 1$ and $\tau = 10^{-13}$ (i.e. no fill-in is allowed). Under these assumptions, applying the global ILUT smoother costs $\mathcal{O}(N_{\text{dof}}\, p^{d})$ flops. Setting up the block ILUT smoother consists of different steps. Here, we assume that the total number of degrees of freedom N_{dof} is given by $N_{\text{dof}} = K\, N_{\text{patch}} + N_{\text{interface}}$, where N_{patch} is the number of degrees of freedom associated to the interior of a single patch and $N_{\text{interface}}$ denotes the total number of degrees of freedom associated to the interface. In general, N_{patch} is significantly higher compared to $N_{\text{interface}}$.

First, an ILUT factorization is determined for all patches, which costs $\mathcal{O}(N_{\text{patch}}\, p^{2d})$ for each patch. The matrices $\tilde{\mathbf{B}}_i$, $\tilde{\mathbf{C}}_i$ $(i = 1, \ldots, K)$ are obtained by solving the following systems, respectively, with multiple matrices:

$$\mathbf{U}_i^\top \mathbf{B}_i^\top = \mathbf{A}_{i\Gamma}^\top, \qquad \mathbf{L}_i \mathbf{C}_i = \mathbf{A}_{\Gamma i}$$

Note that these solves only require forward substitutions and can therefore be performed efficiently and in parallel at the cost of $\mathcal{O}(N_{\text{patch}} p^d)$. Finally, $\tilde{\mathbf{S}}$ can be determined at the costs of $\mathcal{O}(N_{\text{interface}}^3)$, assuming the worst-case scenario of a full matrix $\mathbf{S}$. Applying the block ILUT smoother $\mathcal{O}(N_{\text{patch}} p^d)$ flops, assuming the smoother is applied in parallel.

The analysis of the computational costs show that the efficiency of the block ILUT smoother depends heavily on an efficient parallel implementation of this smoother. This paper, however, places the focus on the analysis of the above block ILUT smoother when applied to multipatch IgA discretizations and not on its efficient parallel implementation. Future research will focus on a parallel and distributed implementation of the considered block ILUT smoother.

5 Numerical Results

In this section, we apply both the global ILUT and the block ILUT smoother within a p-multigrid method for a variety of benchmarks. In particular, the spectral properties of the resulting p-multigrid are analyzed for different values of the mesh width h, the approximation order p and a different number of patches. Furthermore, the number of iterations to reach convergence are determined adopting both smoothers for the benchmarks described in Sect. 5.1.

5.1 Benchmarks

To analyze the block ILUT smoother throughout this paper, we consider the following two dimensional benchmarks:

Benchmark 1. Let Ω be the unit square, i.e. $\Omega = [0, 1]^2$, so that the geometry function is the identity and a simple shrinking of Ω_0 to the subdomain $\Omega^{(k)}$ in case of a multipatch discretization. The CDR equation is considered, where the exact solution u is given by $u(x, y) = \sin(\pi x)\sin(\pi y)$ and the coefficients are chosen as follows:

$$\mathbf{D} = \begin{bmatrix} 1.2 & -0.7 \\ -0.4 & 0.9 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix}, \qquad R = 0.3.$$

Benchmark 2. Let Ω be the quarter annulus with an inner and outer radius of 1 and 2, respectively. Here, Poisson's equation is considered, where the exact solution u is given by $u(x, y) = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2$:

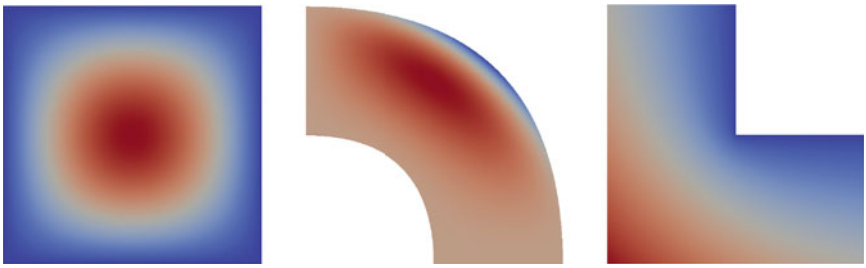


Fig. 5 Exact solutions on the three domains of the considered benchmarks

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad R = 0.$$

Benchmark 3. Let $\Omega = \{[-1, 1] \times [-1, 1]\} \backslash \{[0, 1] \times [0, 1]\}$ be an L-shaped domain. As with the previous benchmark, Poisson's equation is considered, where the exact solution u is given by

$$u(x, y) = \begin{cases} \sqrt[3]{x^2 + y^2} \sin\left(\frac{2\text{atan2}(y,x) - \pi}{3}\right) & \text{if } y > 0 \\ \sqrt[3]{x^2 + y^2} \sin\left(\frac{2\text{atan2}(y,x) + 3\pi}{3}\right) & \text{if } y < 0 \end{cases},$$

where atan2 is the 2-argument arc tangent function. The right-hand side is chosen according to the exact solution.

For the first two benchmarks, homogeneous Dirichlet boundary conditions are applied on the entire boundary $\partial\Omega$, while for the third benchmark inhomogeneous Dirichlet boundary conditions are used given by the exact solution. Figure 5 shows the exact solution for all three benchmarks. All three geometries can be described by a single patch which is converted into a multipatch form by splitting Ω uniformly in both directions. As a consequence, the number of patches K is given by 4^S, where S denotes the number of splittings.

5.2 *Spectral Analysis*

For any multigrid method, the asymptotic convergence rate is determined by the spectral radius (the largest eigenvalue in absolute value) of the iteration matrix. This matrix can easily be obtained numerically following the procedure described in [39]. Table 1 shows the spectral radius obtained for the first benchmark for different values of p and K with the global ILUT and block ILUT smoother for $h = 2^{-5}$. As the spectral radii are obtained numerically, we use a relatively coarse grid.

For both smoothers, the spectral radius increases when the number of patches is increased. For higher values of p, the spectral radius decreases when block ILUT

Table 1 Asymptotic convergence rate of the p-multigrid adopting a global ILUT and block ILUT smoother for different values of the approximation order p and number of patches K

	$p = 2$		$p = 3$		$p = 4$	
# patches K	Global	Block	Global	Block	Global	Block
4	0.094	0.015	0.090	0.011	0.062	0.005
16	0.156	0.019	0.187	0.015	0.146	0.005
64	0.275	0.039	0.374	0.067	**3.659**	0.058

is applied as a smoother. With global ILUT, the dependence on p is more erratic. In particular, the resulting p-multigrid method is diverging for one of the configurations, indicated by a spectral radius larger than 1. Note that, for all configurations, the spectral radius obtained with the block ILUT smoother is significantly lower compared to the one obtained with the global ILUT smoother. As a consequence, the p-multigrid method (using block ILUT as a smoother) is expected to show superior convergence behaviour.

Figure 6 shows the spectrum of the iteration matrix for the p-multigrid method ($h = 2^{-5}$) adopting the global and block ILUT smoother. For the global ILUT smoother, the increased spectral radius for a higher number of patches can be related to single eigenvalues positioned further away from the origin. As a consequence, the use of an outer Krylov method is expected to mitigate this dependency on the number of patches. This has been shown and verified numerically in literature for the global ILUT solver, see [36]. Recall that our specific choice of the prolongation and restriction operator leads to a non-symmetric p-multigrid method, making the use of a Krylov method suited for non-symmetric systems, like Bi-CGSTAB, necessary. For the block ILUT smoother, the spectra are more clustered around the origin, resulting in a lower spectral radius. Furthermore, the dependency on the number of patches is hardly visible in the spectra, although it is visible in Table 1. Therefore, the use of an outer Krylov solver is not expected to significantly decrease the dependency on the number of patches when block ILUT is applied as a smoother.

5.3 Iteration Numbers

Based on the spectral analysis in the previous subsection, the use of the block ILUT smoother is expected to show improved iteration numbers compared to the p-multigrid method equipped with the global ILUT smoother. In this section, the number of p-multigrid cycles needed to achieve convergence is determined using both smoothers. As a stopping criterion, a reduction of the relative residual by 10^8 is chosen. We consider a random vector as an initial guess, where each entry is sampled from a uniform distribution on the interval $[-1, 1]$ using the same seed.

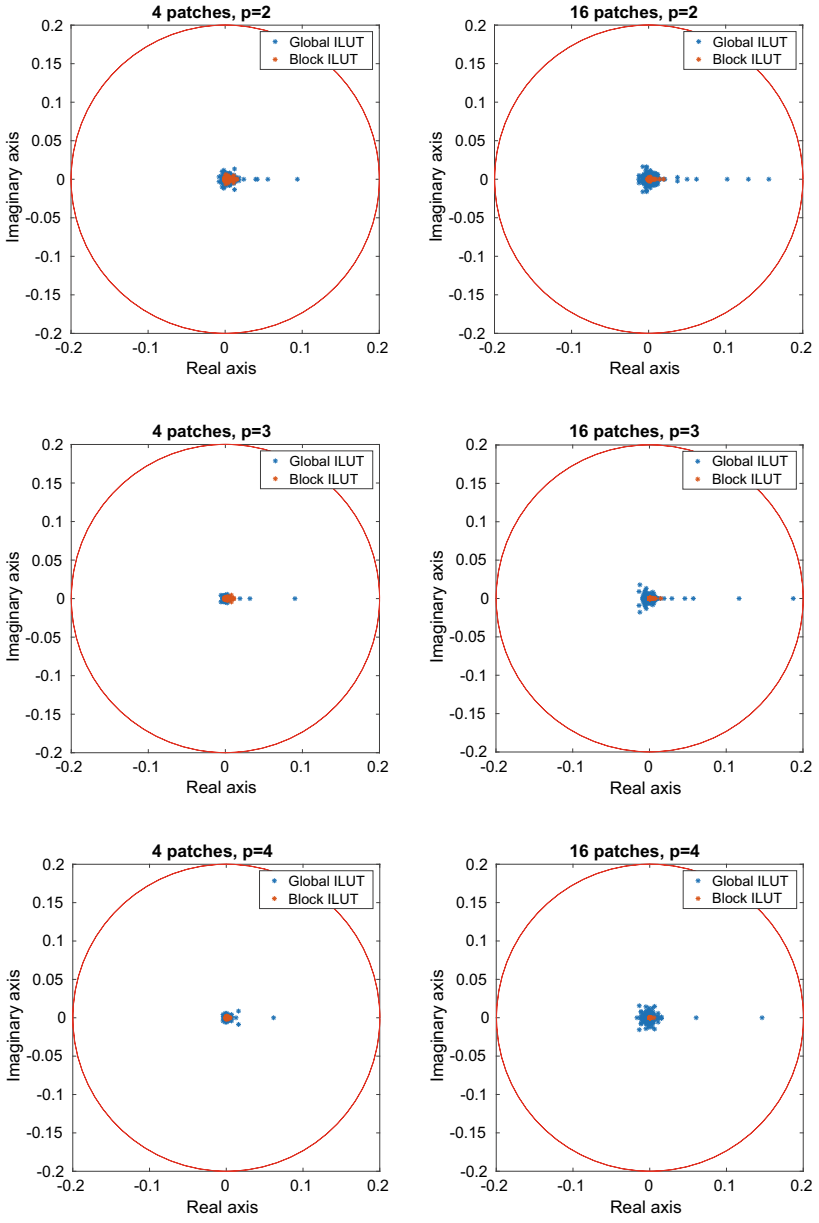


Fig. 6 Spectra of the iteration matrix for the first benchmark obtained with p-multigrid ($h = 2^{-5}$). Here, global ILUT and block ILUT are applied as a smoother

Table 2 shows the number of multigrid cycles needed to achieve convergence for the considered benchmarks with block ILUT as a smoother. The number of iterations needed to achieve convergence with block ILUT as a smoother is, in general, independent of h. For two configurations, however, the p-multigrid method is diverging (indicated by $-$). A small dependence on the number of patches is visible on the coarse mesh, but this dependency becomes negligible for smaller values of h. For a fixed number of patches and higher values of p, the number of iterations slightly decreases. This behavior has been observed before in literature (See [37]) and is related to the fact that the fill-in allowed increases with p. It should be noted, however, that the extra fill-in significantly increases the set-up times of the smoother for higher values of p.

The number of p-multigrid cycles needed to reach convergence with the global ILUT smoother are shown in Table 3. In general, the iteration numbers are (more or less) independent of h and p. However, a dependency on the number of patches can be observed, in particular for coarser meshes. Furthermore, the number of iterations needed to reach convergence is higher for all configurations compared to the use of block ILUT as a smoother.

As the number of p-multigrid cycles when adopting (block) ILUT as a smoother is very low, the use of (block) ILUT as a stand-alone solver has been investigated as well. Table 4 shows the number of iterations needed for the second benchmark when both block ILUT and global ILUT are applied as a solver. For all configurations, the number of iterations is significantly higher compared to the number of p-multigrid cycles (see Tables 2 and 3). Furthermore, a dependency on the mesh width h can be

Table 2 Number of p-multigrid V-cycles needed to achieve convergence using block ILUT as a smoother

	$p=2$			$p=3$			$p=4$			$p=5$		
	# patches			# patches			# patches			# patches		
	4	16	64	4	16	64	4	16	64	4	16	64
(a) CDR-equation on the unit square												
$h=2^{-5}$	4	4	7	3	3	5	2	3	5	2	2	4
$h=2^{-6}$	4	4	5	3	3	4	3	3	4	3	3	3
$h=2^{-7}$	4	4	4	3	3	3	3	3	3	4	3	3
(b) Poisson's equation on a quarter annulus												
$h=2^{-5}$	3	4	4	3	3	4	2	2	4	2	2	–
$h=2^{-6}$	3	3	4	3	3	4	3	3	3	3	3	3
$h=2^{-7}$	3	3	3	3	3	3	3	3	3	–	6	3
(c) Poisson's equation on an L-shaped domain												
$h=2^{-5}$	3	3	4	2	3	4	2	2	3	2	2	2
$h=2^{-6}$	3	3	3	3	3	3	2	2	3	2	2	2
$h=2^{-7}$	3	3	3	2	3	3	2	2	3	2	2	3

Table 3 Number of p-multigrid cycles needed to achieve convergence using global ILUT as a smoother

	$p=2$			$p=3$			$p=4$			$p=5$		
	# patches			# patches			# patches			# patches		
	4	16	64	4	16	64	4	16	64	4	16	64
(a) CDR-equation on the unit square												
$h=2^{-5}$	6	8	11	6	9	15	6	8	15	5	7	14
$h=2^{-6}$	6	7	8	6	8	10	7	9	13	7	8	13
$h=2^{-7}$	6	6	7	6	7	8	7	7	10	6	8	12
(b) Poisson's equation on a quarter annulus												
$h=2^{-5}$	5	7	9	5	7	11	4	6	–	4	6	–
$h=2^{-6}$	5	5	7	5	7	10	6	7	11	5	7	10
$h=2^{-7}$	5	5	5	5	6	8	5	6	10	5	7	11
(c) Poisson's equation on an L-shaped domain												
$h=2^{-5}$	6	7	11	5	8	13	5	6	11	4	5	–
$h=2^{-6}$	6	6	8	6	8	10	5	8	12	5	7	10
$h=2^{-7}$	7	7	8	6	7	8	5	6	10	5	7	12

Table 4 Number of iterations needed to achieve convergence for Poisson's equation on a quarter annulus. Here, block ILUT and global ILUT are used as a solver

	$p=2$			$p=3$			$p=4$			$p=5$		
	# patches			# patches			# patches			# patches		
	4	16	64	4	16	64	4	16	64	4	16	64
(a) Block ILUT												
$h=2^{-5}$	28	22	16	14	10	14	8	6	10	4	4	–
$h=2^{-6}$	112	90	68	48	40	26	24	18	12	14	10	8
$h=2^{-7}$	406	386	312	178	162	130	86	64	54	48	38	26
(b) Global ILUT												
$h=2^{-5}$	56	64	100	22	30	56	14	16	–	8	12	–
$h=2^{-6}$	220	218	234	84	84	100	40	42	56	24	26	36
$h=2^{-7}$	710	700	782	276	302	296	138	138	148	80	80	76

observed, making the use of (block) ILUT as a solver even less efficient for smaller values of h.

Alternatively, a single p-multigrid cycle can be applied as a preconditioner within a Bi-CGSTAB method. Table 5 shows the number of Bi-CGSTAB iterations needed to achieve convergence for the second benchmark, Poisson's equation on the quarter annulus. Compared to the use of p-multigrid as a solver, the number of Bi-CGSTAB iterations is only slightly lower when block ILUT is applied as a smoother. However, for the global ILUT smoother, a significant reduction of the iteration numbers can be

Table 5 Number of Bi-CGSTAB iterations needed to achieve convergence for the second benchmark. Here, a single p-multigrid cycle is applied as a preconditioner using block ILUT and global ILUT as smoother

	$p=2$			$p=3$			$p=4$			$p=5$		
	# patches			# patches			# patches			# patches		
	4	16	64	4	16	64	4	16	64	4	16	64
(a) Block ILUT smoother												
$h=2^{-5}$	2	2	2	2	2	2	1	1	2	1	1	50
$h=2^{-6}$	2	2	2	2	2	2	2	2	2	2	2	2
$h=2^{-7}$	2	2	2	2	2	2	2	2	2	34	3	2
(b) Global ILUT smoother												
$h=2^{-5}$	2	3	3	2	3	4	2	3	13	2	3	353
$h=2^{-6}$	2	2	3	3	3	4	2	3	4	2	3	3
$h=2^{-7}$	2	2	3	2	3	3	2	3	3	2	3	4

achieved, as expected from the spectral analysis. Note that the use of a Krylov solver restores stability for configurations that diverged before, but lead to a relatively high number of iterations.

6 Application: Yeti Footprint

In the previous section, we applied the proposed block ILUT smoother on benchmarks defined on geometries consisting of a different number of patches. In general, the application of both smoothers within a p-multigrid method resulted in iteration numbers independent of h and p. A small dependency on the number of patches was noticeable, however, in particular on coarser meshes.

In this section, the block ILUT and global ILUT smoother are applied on a more challenging geometry. We consider Poisson's equation on a two dimensional Yeti footprint, where the exact solution is given by $u(x,t) = \sin(5\pi x)\sin(5\pi y)$:

$$\begin{aligned} -\Delta u &= 50\pi^2 \sin(5\pi x)\sin(5\pi y), && (x,y) \in \Omega, \\ u &= 0, && (x,y) \in \partial\Omega. \end{aligned}$$

This benchmark has been considered in literature in the context of parallel multigrid methods for IgA, see [17]. Figure 7 shows the Yeti footprint multipatch geometry (left) and its exact solution (right) consisting of 21 patches.

Table 6 shows the number of multigrid cycles needed to reach convergence for the p-multigrid method adopting both smoothers and the aforementioned convergence criterion. For all configurations, the use of block ILUT leads to a lower number of cycles compared to the use of global ILUT. Furthermore, the number of iterations

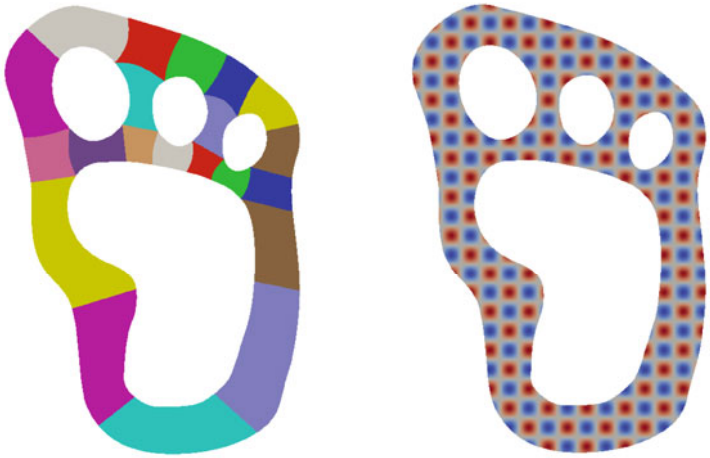


Fig. 7 Underlying multipatch geometry (left) and the exact solution (right) of the Yeti footprint

Table 6 Number of multigrid cycles needed to reach convergence for the Yeti footprint for a p-multigrid method with global and block ILUT as a smoother

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	Global	Block	Global	Block	Global	Block	Global	Block
$h = 2^{-3}$	5	4	4	2	4	2	4	2
$h = 2^{-4}$	8	4	5	3	5	3	4	2
$h = 2^{-5}$	8	4	6	3	5	3	5	3

needed to reach convergence is, in general, lower for higher values of the approximation order p. Results are, to a large extent, comparable with the ones presented in the previous section. A small h-dependence can be observed, however, for smaller values of p when applying the global ILUT smoother.

7 Conclusions

In this paper, we presented a block ILUT smoother for p-multigrid methods in Isogeometric Analysis for multipatch geometries. The block ILUT smoother is based on a preconditioner, which has been developed in the context of domain decomposition methods [26]. The key idea is to make use of the block structure of the resulting system matrix when setting up the smoother. As a consequence, the ILUT factorizations are only obtained for smaller blocks, where the number of blocks depends on the number of patches describing the geometry.

A spectral analysis has shown that this smoother is more effective when applied within a p-multigrid method compared to a global ILUT smoother. Numerical results, obtained for a variety of two dimensional benchmarks, indeed showed that the number of iterations needed to achieve convergence is lower for all considered configurations when adopting the block ILUT smoother. The presented block ILUT smoother lends itself for a parallel implementation, which will be the focus of future research. In particular, focus will lie on the comparison with the global ILUT smoother in terms of computational efficiency.

References

1. Amestoy, P.R., Davis, T.A., Duff, I.S.: An approximate minimum degree ordering algorithm. SIAM J. Matrix Anal. Appl. **17**, 886–905 (1996)
2. Antolin, P., Buffa, A., Calabro, F., Martinelli, M., Sangalli, G.: Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. Comput. Methods Appl. Mech. Eng. **285**, 817–828 (2015)
3. Beirao da Veiga, L., Cho, D., Pavarino, L.F., Scacchi, S.: Overlapping Schwarz methods for isogeometric analysis. SIAM J. Numer. Anal. **50**, 1394–1416 (2012)
4. de Boor, C.: A Practical Guide to Splines, 1st edn. Springer, New York (1978)
5. Brandt, A.: Multi-level adaptive solutions to boundary-value problems. Math. Comput. **31**, 333–390 (1977)
6. Brenner, S.C., Scott, L.R.: The Mathematical Theory of Finite Element Methods. Texts in Applied Mathematics. Springer, New York (1994)
7. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, 2nd edn. SIAM, Philadelphia (2000)
8. Calabro, F., Sangalli, G., Tani, M.: Fast formation of isogeometric Galerkin matrices by weighted quadrature. Comput. Methods Appl. Mech. Eng. **316**, 606–622 (2017)
9. Collier, N., Dalcin, L., Pardo, D., Calo, V.M.: The cost of continuity: performance of iterative solvers on Isogeometric Finite Elements. SIAM J. Sci. Comput. **35**, 767–784 (2013)
10. Donatelli, M., Garoni, C., Manni, C., Capizzano, S., Speleers, H.: Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. SIAM J. Numer. Anal. **55**, 31–62 (2017)
11. Fidkowski, K.J., Oliver, T.A., LU, J., Darmofal, D.L.: p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. J. Comput. Phys. **207**, 92–113 (2005)
12. Gahalaut, K.P.S., Kraus, J.K., Tomar, S.K.: Multigrid methods for isogeometric discretizations. Comput. Methods Appl. Mech. Eng. **253**, 413–425 (2013)
13. Gao, L., Calo, V.: Fast isogeometric solvers for explicit dynamics. Comput. Methods Appl. Mech. Eng. **274**, 19–41 (2014)
14. Guennebaud, G., Jacob, B.: Eigen v3 (2010). http://eigen.tuxfamily.org
15. Hackbush, W.: Multi-Grid Methods and Applications. Springer, Berlin (1985)
16. Helenbrook, B., Mavriplis, D., Atkins, H.: Analysis of p-multigrid for continuous and discontinuous finite element discretizations. In: 16th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences (2003)
17. Hofer, C., Takacs, S.: A parallel multigrid solver for multi-patch isogeometric analysis. In: Apel, T., Langer, U., Meyer, A., Steinbach, O., (eds.), Advanced Finite Element Methods with Applications. FEM 2017. Lecture Notes in Computational Science and Engineering, vol. 128, pp. 205–219. Springer, Cham (2019)
18. Hofreither, C., Takacs, S., Zulehner, W.: A robust multigrid method for isogeometric analysis in two dimensions using boundary correction. Comput. Methods Appl. Mech. Eng. **316**, 22–42 (2017)

19. Hofreither, C., Takacs, S.: Robust multigrid for isogeometric analysis based on stable splittings of spline spaces. SIAM J. Numer. Anal. **55**, 2004–2024 (2017)
20. Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
21. Hughes, T.J.R., Reali, A., Sangalli, G.: Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. Comput. Methods Appl. Mech. Eng. **197**(49–50), 4104–4124 (2008)
22. Luo, H., Baum, J.D., Löhner, R.: A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. J. Comput. Phys. **211**, 767–783 (2006)
23. Luo, H., Baum, J.D., Löhner, R.: Fast p-multigrid discontinuous Galerkin method for compressible flows at all speeds. AIAA J. **46**, 635–652 (2008)
24. Mantzaflaris, A., Jüttler, B., Khoromskij, B.N., Langer, U.: Low rank tensor methods in Galerkin-based isogeometric analysis. Comput. Methods Appl. Mech. Eng. **316**, 1062–1085 (2017)
25. Nicolaides, R.A.: Deflation of conjugate gradients with applications to boundary value problems. SIAM J. Numer. Anal. **24**, 355–365 (1987)
26. Nievinksi, I.C.L., Souza, M., Goldfeld, P., Augusto, D.A., Rogrigues, J.R.P., Carvalho, L.M.: Parallel implementation of a two-level algebraic ILU(k)-based domain decomposition preconditioner. Tendencias em Matematics Aplicada e Computatcional **19**, 59–77 (2018)
27. de la Riva, A., Rodrigo, C., Gaspar, F.: An efficient multigrid solver for isogeometric analysis (2018). arXiv:1806.05848v1
28. Saad, Y.: ILUT: a dual threshold incomplete LU factorization. Numer. Linear Algebra Appl. **1**, 387–402 (1994)
29. Saad, Y.: SPARSKIT: a basic tool kit for sparse matrix computations (1994)
30. Sampath, R.S., Biros, G.: A parallel geometric multigrid method for finite elements on octree meshes. SIAM J. Sci. Comput. **32**, 1361–1392 (2010)
31. van Slingerland, P., Vuik, C.: Fast linear solver for diffusion problems with applications to pressure computation in layered domains. Comput. Geosci. **18**, 343–356 (2014)
32. Sogn, J., Takacs, S.: Robust multigrid solvers for the biharmonic problem in isogeometric analysis. Comput. Methods Appl. Mech. Eng. **77**, 105–124 (2019)
33. Tang, J.M., Saad, Y.: Domain-decomposition-type methods for computing the diagonal of a matrix inverse. SIAM J. Sci. Comput. **33**, 2823–2847 (2011)
34. Sangalli, G., Tani, M.: Isogeometric preconditioners based on fast solvers for the Sylvester equation. SIAM J. Sci. Comput. **38**, 3644–3671 (2016)
35. Tielen, R., Möller, M., Vuik, C.: Efficient multigrid based solvers for Isogeometric Analysis. In: Proceedings of the 6th European Conference on Computational Mechanics and the 7th European Conference on Computational Fluid Dynamics, Glasgow (2018)
36. Tielen, R., Möller, M., Vuik, C.: Efficient p-multigrid solvers for multipatch geometries in Isogeometric Analysis. In: Proceedings of the 3rd Conference on Isogeometric Analysis and Applications, Delft, The Netherlands (2018)
37. Tielen, R., Möller, M., Göddeke, D., Vuik, C.: p-multigrid methods and their comparison to h-multigrid methods within isogeometric analysis. Comput. Methods Appl. Mech. Eng. **372**, 113347 (2020)
38. Tielen, R., Möller, M., Vuik, C.: A direct projection to low-order level for p-multigrid methods in isogeometric analysis. In: Proceedings of the 13th Conference on European Numerical Mathematics and Advanced Applications (ENUMATH). Egmond aan Zee, The Netherlands (2019)
39. Trottenberg, U., Oosterlee, C., Schüller, A.: Multigrid. Academic, New York (2001)