# Efficient $p$-multigrid based solvers for isogeometric analysis on multipatch geometries

Roel Tielen, Matthias Möller and Kees Vuik

**Abstract** Isogeometric Analysis can be considered as the natural extension of the Finite Element Method (FEM) to higher-order spline based discretizations simplifying the treatment of complex geometries with curved boundaries. Finding a solution of the resulting linear systems of equations efficiently remains, however, a challenging task. Recently, $p$-multigrid methods have been considered [26], in which a multigrid hierarchy is constructed based on different approximation orders $p$ instead of mesh widths $h$ as it would be the case in classical $h$-multigrid schemes [13]. The use of an Incomplete LU-factorization as a smoother within the $p$-multigrid method has shown to lead to convergence rates independent of both $h$ and $p$ for single patch geometries [27]. In this paper, the focus lies on the application of the aforementioned $p$-multigrid method on multipatch geometries having a $C^0$-continuous coupling between the patches. The use of ILUT as a smoother within $p$-multigrid methods leads to convergence rates that are essentially independent of $h$ and $p$, but depend mildly on the number of patches.

## 1 Introduction

Isogeometric Analysis (IgA) [15] can be considered as the natural extension of the Finite Element Method (FEM) to higher-order spline based discretizations simplifying the treatment of complex geometries with curved boundaries. However, solving the resulting linear systems arising in IgA efficiently is considered a challenging task, especially for higher-order discretizations. The exponential increase of the condition numbers of the mass and stiffness matrices with the approximation order $p$, make the use of (standard) iterative solvers inefficient. The wider support of the basis functions and, consequently, increasing bandwidth of the matrices for larger values of $p$ make the use of direct solvers on the other hand also not straightforward.

Roel Tielen
Delft University of Technology, Van Mourik Broekmanweg 6, e-mail: r.p.w.m.tielen@tudelft.nl

The use of established solution techniques from FEM in IgA has been an active field of research. For example, $h$-multigrid methods have been investigated, as they are considered among the most efficient solvers in Finite Element Methods for elliptic problems. The use of standard smoothers like (damped) Jacobi or Gauss-Seidel leads, however, to convergence rates which deteriorate drastically for increasing values of $p$ [10], caused by very small eigenvalues associated with high-frequency eigenvectors [8]. Non-classical smoothers have been developed to solve this problem leading to (geometric) multigrid methods which are robust in both $h$ and $p$ [13, 20].

An alternative solution strategy are $p$-multigrid methods. In contrast to $h$-multigrid methods, where each level of the constructed hierarchy is obtained by refining the mesh, in $p$-multigrid methods each level represents a different approximation order. $p$-Multigrid methods are widely used within the Discontinuous Galerkin framework [9, 17, 18, 21], where $p = 0$ is used on the coarsest hierarchy level.

Some research has been performed for continuous Galerkin methods [12] as well, where the coarse grid correction was obtained at level $p = 1$. Throughout this paper, the coarse grid is also obtained at level $p = 1$. Here, B-spline basis functions coincide with piecewise-linear ($p = 1$) Lagrange basis functions, enabling the use of well known solution techniques for standard FEM.

Recently, the authors developed an efficient $p$-multigrid method for IgA discretizations that makes use of an Incomplete LU factorization based on a dual treshold strategy (ILUT) [22] as a smoother. This approach was shown to result in a $p$-multigrid method with essentially $h$- and $p$-independent convergence rates [27] in contrast to the use of Gauss-Seidel as a smoother.

In this paper, the focus lies on the extension of $p$-multigrid based methods on multipatch geometries, giving rise to (reduced) $C^0$-continuity between individual patches. The spectral properties of the $p$-multigrid method are analysed and numerical results are presented for different two-dimensional benchmarks. The use of ILUT as a smoother leads to a $p$-multigrid method that shows essentially $h$- and $p$-independent convergence rates on multipatch geometries. Furthermore, the number of iterations needed to achieve convergence depends only mildly on the number of patches.

This paper is organised as follows. The model problem and spatial discretization are briefly considered in Section 2. Section 3 presents the $p$-multigrid method together with the adopted ILUT smoother in more detail. In Section 4, a spectral analysis is performed and discussed. Numerical results for the considered benchmarks are presented in Section 5. Conclusions are finally drawn in Section 6.

## 2 Model Problem

As a model problem to describe the spatial discetisation, Poisson's equation is considered:

$$-\Delta u = f, \quad \text{on } \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a connected, Lipschitz domain, $f \in L^2(\Omega)$ and $u = 0$ on the boundary $\partial\Omega$. Let $\mathcal{V} = H_0^1(\Omega)$ denote the subspace of the Sobolev space $H^1(\Omega)$ that contains all functions that vanish on the boundary $\partial\Omega$. By multiplying both sides of (1) with an arbitrary test function $v \in \mathcal{V}$ and applying integration by parts in the left side, the following variational form of (1) is obtained:

Find $u \in \mathcal{V}$ such that

$$a(\nabla u, \nabla v) = (f, v) \quad \forall v \in \mathcal{V}, \tag{2}$$

where

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}\Omega \quad \text{and} \quad (f, v) = \int_\Omega f v \, \mathrm{d}\Omega. \tag{3}$$

A bijective geometry function $\mathbf{F}$ is then defined to parameterize the physical domain $\Omega$:

$$\mathbf{F} : \Omega_0 \to \Omega, \qquad \mathbf{F}(\boldsymbol{\xi}) = \mathbf{x}, \tag{4}$$

where $\boldsymbol{\xi} = (\xi, \eta)$ and $\mathbf{x} = (x, y)$ denote the coordinates in the parametric and physical domain, respectively. The geometry function $\mathbf{F}$ describes an invertible mapping connecting the parameter domain $\Omega_0 \subset \mathbb{R}^2$ with the physical domain $\Omega$. In case $\Omega$ cannot be described by a single geometry function, the physical domain is divided in a collection of non-overlapping subdomains $\Omega^{(k)}$ such that

$$\Omega = \bigcup_{k=1}^K \Omega^{(k)}. \tag{5}$$

A geometry function $\mathbf{F}^{(k)}$ is then defined to parameterize each subdomain $\Omega^{(k)}$:

$$\mathbf{F}^{(k)} : \Omega_0 \to \Omega^{(k)}, \qquad \mathbf{F}^{(k)}(\boldsymbol{\xi}) = \mathbf{x}. \tag{6}$$

We refer to $\Omega$ as a multipatch geometry consisting of $K$ patches. Throughout this paper, the tensor product of one-dimensional B-spline basis functions $\phi_{i_x,p}(\xi)$ and $\phi_{i_y,q}(\eta)$ of order $p$ and $q$, respectively, with maximum continuity are adopted for the spatial discetisation:

$$\Phi_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) := \phi_{i_x,p}(\xi)\phi_{i_y,q}(\eta), \qquad \mathbf{i} = (i_x, i_y), \ \mathbf{p} = (p, q). \tag{7}$$

Here, $\mathbf{i}$ and $\mathbf{p}$ are multi indices, with $i_x = 1, \ldots, n_x$ and $i_y = 1, \ldots, n_y$ denoting the one-dimensional basis functions in the $x$ and $y$-dimension, respectively. Furthermore, $i = i_x n_x + i_y n_y$ assigns a unique index to each pair of one-dimensional basis functions, where $i = 1, \ldots N_{\mathrm{dof}}$. The spline space $\mathcal{V}_{h,p}$ can then be written, using the inverse of the geometry mapping $\mathbf{F}^{-1}$ as pull-back operator, as follows:

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

Here, $N_{\text{dof}}$ denotes the number of degrees of freedom, or equivalently, the number of tensor-product basis functions. The Galerkin formulation of (2) becomes:

Find $u_{h,p} \in \mathcal{V}_{h,p}$ such that

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

or, equivalently:

$$\mathbf{A}_{h,p}\mathbf{u}_{h,p} = \mathbf{f}_{h,p}. \tag{10}$$

Here, $\mathbf{A}_{h,p}$ denotes the stiffness matrix resulting from the discretization of the left-hand side with the tensor-product of B-spline basis functions of order $p$ and knot span size $h$. To assess the quality of the $p$-multigrid method throughout this paper, the following benchmarks are considered:

**Benchmark 1**. Here, Poisson's equation is considered on the unit square, i.e. $\Omega = [0,1]^2$. The right-hand side is chosen such that the exact solution $u$ is given by:

$$u(x,y) = \sin(\pi x)\sin(\pi y).$$

**Benchmark 2**. Let $\Omega$ be the quarter annulus with an inner and outer radius of 1 and 2, respectively. Again, Poisson's equation is considered, where the exact solution $u$ is given by

$$u(x,y) = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2,$$

**Benchmark 3**. Let $\Omega = \{[-1,1] \times [-1,1]\} \setminus \{[0,1] \times [0,1]\}$ be an L-shaped domain. As with the other benchmarks, 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 arctangent 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 applied. Note that the geometry of each benchmark can be described by a single patch. The multipatch geometries considered throughout this paper are obtained by splitting the single patch uniformly in both directions.

## 3 p-Multigrid method

Multigrid methods solve linear systems of equations by defining a hierarchy of discretizations. At each level of the hierarchy, a basic iterative method, like Gauss-

Seidel or (damped) Jacobi, is then applied as a smoother. On the coarsest level, a correction is determined by solving the residual equation. With $p$-multigrid methods, a sequence of spaces $\mathcal{V}_{h,1}, \ldots, \mathcal{V}_{h,p}$ is obtained by applying refinement in $p$. The coarse grid correction is then determined at level $p = 1$. Since basis functions with maximal continuity are considered, the spaces in the hierarchy are not nested. For $p$-multigrid, the two-grid correction scheme consists of the following steps [26, 27]:

1. Apply a fixed number $\nu_1$ of presmoothing steps to update the initial guess $\mathbf{u}_{h,p}^{(0)}$:
$$\mathbf{u}_{h,p}^{(0,m)} = \mathbf{u}_{h,p}^{(0,m-1)} + \mathcal{S}_{h,p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(0,m-1)}\right), \quad m = 1, \ldots, \nu_1. \quad (11)$$

   Here, $\mathcal{S}$ is a smoothing operator applied to the high-order problem.

2. Project the residual at level $p$ onto $\mathcal{V}_{h,p-1}$ using the restriction operator $\mathcal{I}_p^{p-1}$:
$$\mathbf{r}_{h,p-1} = \mathcal{I}_p^{p-1}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(0,\nu_1)}\right). \quad (12)$$

3. Determine the coarse grid error, by solving the residual equation at level $p - 1$:
$$\mathbf{A}_{h,p-1}\mathbf{e}_{h,p-1} = \mathbf{r}_{h,p-1}. \quad (13)$$

4. Use the prolongation operator $\mathcal{I}_{p-1}^p$ to project the error $\mathbf{e}_{h,p-1}$ onto the space $\mathcal{V}_{h,p}$ and update $\mathbf{u}_{h,p}^{(0,\nu_1)}$:
$$\mathbf{u}_{h,p}^{(0,\nu_1)} := \mathbf{u}_{h,p}^{(0,\nu_1)} + \mathcal{I}_{p-1}^p\left(\mathbf{e}_{h,p-1}\right). \quad (14)$$

5. Apply $\nu_2$ postsmoothing steps to obtain $\mathbf{u}_{h,p}^{(0,\nu_1+\nu_2)} =: \mathbf{u}_{h,p}^{(1)}$:
$$\mathbf{u}_{h,p}^{(0,\nu_1+m)} = \mathbf{u}_{h,p}^{(0,\nu_1+m-1)} + \mathcal{S}_{h,p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}_{h,p}^{(0,\nu_1+m-1)}\right), \quad m = 1, \ldots, \nu_2. \quad (15)$$

The residual equation can be solved recursively by applying the same two-grid correction scheme until level $p = 1$, which results in a V-cycle. Different cycle types can be applied, however, as shown in Figure 1.
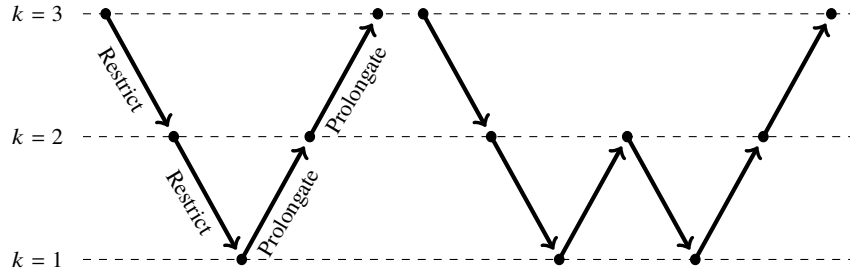


**Fig. 1** Description of a V-cycle and W-cycle.

**Prolongation and restriction**

The prolongation and restriction operators transfer both coarse grid corrections and residuals between different levels of the hierarchy. The prolongation and restriction operator adopted in this paper are based on the $L_2$ projection [4, 5, 24]. The prolongation operator $\mathcal{I}_{k-1}^k : \mathcal{V}_{h,k-1} \rightarrow \mathcal{V}_{h,k}$ can be derived from the following variational form

$$(\mathcal{I}_{k-1}^k(u_{h,k-1}), v_{h,k}) = (u_{h,k-1}, v_{h,k}) \quad \forall u_{h,k-1} \in \mathcal{V}_{k-1}, \quad \forall v_{h,k} \in \mathcal{V}_k \qquad (16)$$

and is therefore given by

$$\mathcal{I}_{k-1}^k(\mathbf{v}_{k-1}) = (\mathbf{M}_k)^{-1} \mathbf{P}_{k-1}^k \, \mathbf{v}_{k-1}, \qquad (17)$$

where the mass matrix $\mathbf{M}_k$ and the transfer matrix $\mathbf{P}_{k-1}^k$ are defined, respectively, as follows:

$$(\mathbf{M}_k)_{(i,j)} := \int_\Omega \Phi_{\mathbf{i},\mathbf{k}} \Phi_{\mathbf{j},\mathbf{k}} \; \mathrm{d}\Omega, \qquad (\mathbf{P}_{k-1}^k)_{(i,j)} := \int_\Omega \Phi_{\mathbf{i},\mathbf{k}} \Phi_{\mathbf{j},\mathbf{k-1}} \; \mathrm{d}\Omega. \qquad (18)$$

The restriction operator $\mathcal{I}_k^{k-1} : \mathcal{V}_{h,k} \rightarrow \mathcal{V}_{h,k-1}$ is given by the Hilbert adjoint of the prolongation operator and defined by

$$\mathcal{I}_k^{k-1}(\mathbf{v}_k) = (\mathbf{M}_{k-1})^{-1} \mathbf{P}_k^{k-1} \, \mathbf{v}_k. \qquad (19)$$

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

$$\mathbf{M}_{(i,i)}^L = \sum_{j=1}^{N_{\mathrm{dof}}} \mathbf{M}_{(i,j)}. \qquad (20)$$

**Smoother**

In this paper, an Incomplete LU factorization with a dual treshold strategy (ILUT) [22] is adopted as a smoother. The ILUT factorization is completely determined by a tolerance $\tau$ and fillfactor $f$. All matrix entries in the factorization smaller (in absolute value) than the tolerance multiplied by the average magnitude of all entries in the current row are dropped. Furthermore, only the average number of non-zeros in each row of the original operator $\mathbf{A}_{h,p}$ multiplied with the fillfactor are kept in each row.
Throughout this paper, a fillfactor of 1 is adopted and the dropping tolerance $\tau$ equals $10^{-12}$. As a consequence, the number of non-zero entries of the factorization is similar to the number of non-zeros of the original operator. An efficient implementation of

an ILUT factorization is available in the Eigen library [11] based on [23]. Once the factorization is obtained, a single smoothing step is applied as follows:

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

$$\mathbf{u}_{h,p}^{(n+1)} = \mathbf{u}_{h,p}^{(n)} + \mathbf{e}_{h,p}^{(n)}. \tag{22}$$

**Coarse grid operator**

The system operator $\mathbf{A}_{h,p}$ is needed at each level of the hierarchy to apply the smoothing steps or solve the residual equation at level $p = 1$. The operators at the coarser levels can be obtained by rediscretizing the bilinear form in (9) with lower-order spline basis functions or by applying a Galerkin projection:

$$\mathbf{A}_{h,k-1}^{G} = \mathcal{I}_{k}^{k-1} \, \mathbf{A}_{h,k} \, \mathcal{I}_{k-1}^{k}. \tag{23}$$

Since the condition number of the resulting coarse grid operator $\mathbf{A}_{h,1}$ is significantly lower when using rediscretizing [27], this approach is adopted throughout the rest of this paper.

## 4 Spectral Analysis

To investigate the performance of the $p$-multigrid method on multipatch geometries, the spectrum of the iteration matrix is determined. The iteration matrix describes the effect of a single multigrid cycle on $\mathbf{u}_{h,p}$ and can be used to obtain the asymptotic convergence rate of the $p$-multigrid method. For all benchmarks introduced in Section 2, results are presented considering a different number of patches.

The asymptotic convergence rate of a multigrid method is determined by the spectral radius of the corresponding iteration matrix. This matrix can be obtained explicitly by considering $-\Delta u = 0$ with homogeneous Dirichtlet boundary conditions. By applying a single iteration of the $p$-multigrid method using the $i^{\text{th}}$ unit vector as initial guess, one obtains the $i^{\text{th}}$ column of the iteration matrix [28].

The spectra obtained for the first two benchmarks are shown in Figure 2 for a different number of patches. The multipatch geometries are obtained by splitting the single patch uniformly in both directions, leading to 4 or 16 patches with a $C^0$-continuous coupling at the interfaces. For the single patch, all eigenvalues of the iteration matrix are clustered around the origin. For the multipatch geometries, some eigenvalues are slightly further from the origin. Table 1, showing the spectral radius of the iteration matrix for different values of $h$ and $p$ for the first benchmark, confirms this observation. The spectral radii are larger for all numerical experiments when the number of patches is increased, but still relatively low. Furthermore, since the spectral radii remain almost constant for higher values of $p$, the $p$-multigrid method is expected to show (essentially) $p$-independent convergence rates.

**Table 1** Spectral radius $\rho$ for the first benchmark for different values of $h$ and $p$ for a different number of patches.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 0.012 | 0.130 | 0.200 | $h = 2^{-4}$ | 0.003 | 0.064 | 0.156 | $h = 2^{-4}$ | 0.004 | 0.012 | 0.083 |
| $h = 2^{-5}$ | 0.021 | 0.129 | 0.177 | $h = 2^{-5}$ | 0.013 | 0.132 | 0.214 | $h = 2^{-5}$ | 0.014 | 0.091 | 0.168 |
| $h = 2^{-6}$ | 0.021 | 0.131 | 0.133 | $h = 2^{-6}$ | 0.015 | 0.143 | 0.187 | $h = 2^{-6}$ | 0.031 | 0.140 | 0.223 |

The obtained spectral radii for the second benchmark for different values of $h$ and $p$ can be found in Table 2. Again, the multipatch geometries consist of 4 and 16 patches. For all configurations, the spectral radius for a single patch geometry is lower compared to the spectral radius obtained for the multipatch geometries. As a consequence, the $p$-multigrid is expected to show slower convergence behaviour for multipatch geometries. On the other hand, the asymptotic convergence rates for the multipatch geometries are almost independent of $p$ and still relatively low. For a single configuration the resulting $p$-multigrid method is instable, which is reflected by a spectral radius larger then 1. The obtained spectral radii for the third benchmark are presented in Table 3. As with the other benchmarks, the spectral radii remain almost constant for higher values of $p$, implying (essentially) $p$-independent convergence rates.

**Table 2** Spectral radius $\rho$ for the second benchmark for different values of $h$ and $p$ for a different number of patches.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 0.014 | 0.049 | 0.141 | $h = 2^{-4}$ | 0.003 | 0.013 | 0.073 | $h = 2^{-4}$ | 0.003 | 0.007 | 1.312 |
| $h = 2^{-5}$ | 0.039 | 0.093 | 0.155 | $h = 2^{-5}$ | 0.020 | 0.073 | 0.155 | $h = 2^{-5}$ | 0.029 | 0.035 | 0.090 |
| $h = 2^{-6}$ | 0.057 | 0.103 | 0.116 | $h = 2^{-6}$ | 0.024 | 0.124 | 0.169 | $h = 2^{-6}$ | 0.023 | 0.114 | 0.174 |

**Table 3** Spectral radius $\rho$ for the third benchmark for different values of $h$ and $p$ for a different number of patches.

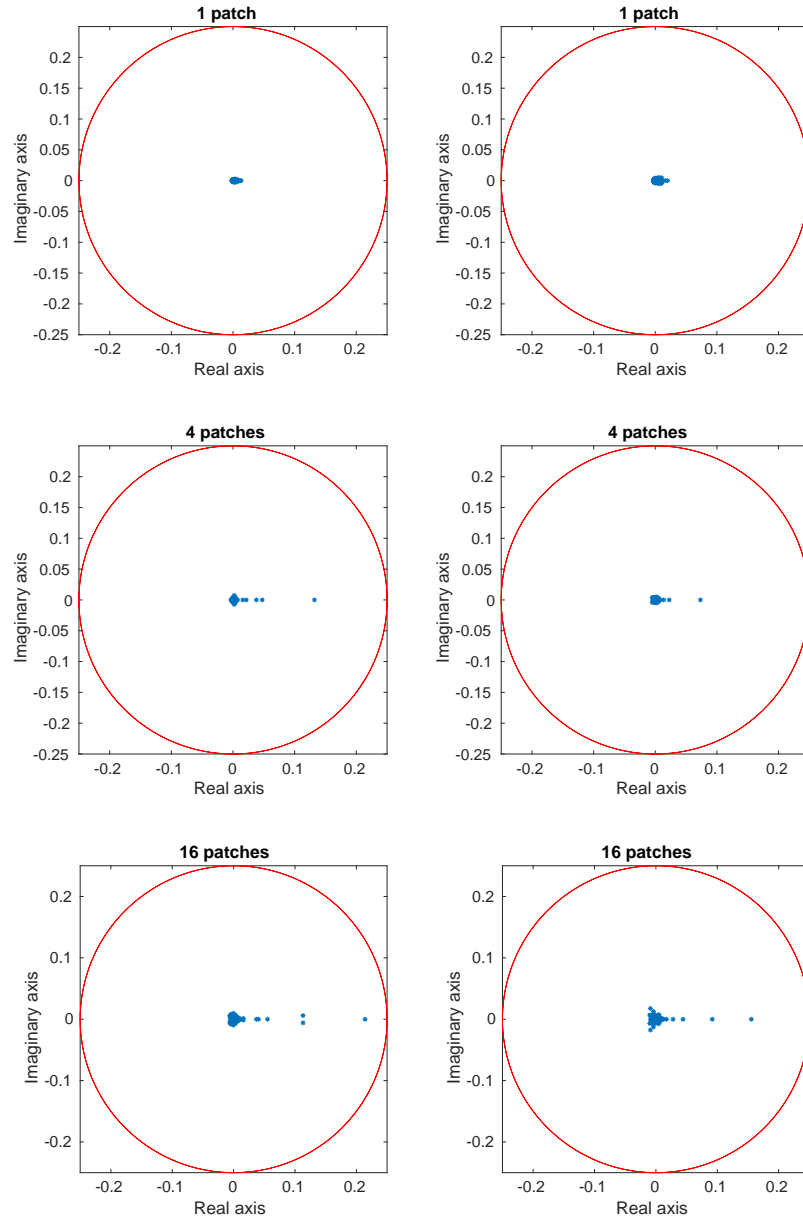| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 0.004 | 0.038 | 0.080 | $h = 2^{-4}$ | 0.001 | 0.002 | 0.013 | $h = 2^{-4}$ | $2.6 \cdot 10^{-5}$ | $6.5 \cdot 10^{-5}$ | 0.003 |
| $h = 2^{-5}$ | 0.012 | 0.082 | 0.129 | $h = 2^{-5}$ | 0.007 | 0.035 | 0.080 | $h = 2^{-5}$ | 0.002 | 0.005 | 0.020 |
| $h = 2^{-6}$ | 0.016 | 0.089 | 0.127 | $h = 2^{-6}$ | 0.010 | 0.091 | 0.159 | $h = 2^{-6}$ | 0.005 | 0.059 | 0.118 |

**Fig. 2** Spectra of the iteration matrix (with $p = 3$ and $h = 2^{-5}$) for the first (left) and second (right) benchmark for a different number of patches.

## 5 Numerical Results

The proposed $p$-multigrid method is applied in this Section as a stand-alone solver and as a preconditioner within a Biconjugate gradient stabilized (BiCGSTAB) method. Results are obtained using different numbers of patches. Furthermore, results are compared when using ILUT as a solver. Finally, different coarsening strategies (i.e. coarsening in $h$, $p$ or both) are compared with respect to the number of iterations and computational time.

The initial guess $\mathbf{u}_{h,p}^{(0)}$ is chosen randomly for all experiments, where each entry is sampled from a uniform distribution on the interval $[-1, 1]$ using the same seed. The method is considered to be converged if

$$\frac{||\mathbf{r}_{h,p}^{(n)}||}{||\mathbf{r}_{h,p}^{(0)}||} < 10^{-8}, \tag{24}$$

where $\mathbf{r}_{h,p}^{(n)}$ denotes the residual after iteration $n$. The solution of the residual equation is obtained at level $p = 1$ by means of a Conjugate Gradient solver with a relatively high tolerance ($\epsilon = 10^{-4}$). The same number of pre- and postsmoothing steps are applied for all experiments ($\nu = \nu_1 = \nu_2$) and boundary conditions are imposed using Nitsche's method [19].

The number of V-cycles needed to reach convergence for different values of $h$ and $p$ for a different number of patches for the first benchmark are shown in Table 4. Here, the number of smoothing steps at each level equals 2. Results have been obtained considering 1, 4 and 16 patches, where the multipatch is based on splitting a single patch uniformly in both directions. In general, the $p$-multigrid method shows convergence rates which are essentially independent of both $h$ and $p$. However, an increase of the number of patches leads to an increase in the number of V-cycles needed to achieve convergence. Note that this increase is relatively low, especially for smaller values of $h$.

**Table 4** Number of V-cycles needed for achieving convergence for the first benchmark with $p$-multigrid, $\nu = 2$.

| $p = 2$ | # patches 1 | 4 | 16 | $p = 3$ | # patches 1 | 4 | 16 | $p = 4$ | # patches 1 | 4 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-4}$ | 3 | 5 | 7 | $h = 2^{-4}$ | 2 | 3 | 5 | $h = 2^{-4}$ | 2 | 2 | 4 |
| $h = 2^{-5}$ | 3 | 6 | 7 | $h = 2^{-5}$ | 3 | 5 | 7 | $h = 2^{-5}$ | 2 | 4 | 5 |
| $h = 2^{-6}$ | 3 | 5 | 6 | $h = 2^{-6}$ | 3 | 6 | 7 | $h = 2^{-6}$ | 3 | 5 | 7 |
| $h = 2^{-7}$ | 3 | 5 | 5 | $h = 2^{-7}$ | 3 | 5 | 5 | $h = 2^{-7}$ | 2 | 5 | 6 |

Table 5 shows the number of V-cycles needed when the number of smoothing steps is doubled. Hence, $\nu = 4$ for all numerical experiments. Doubling the number of smoothing steps at each level, slightly decreases the number of V-cycles. However, since the number of V-cycles is already relatively low, the reduction is limited.

**Table 5** Number of V-cycles needed for reaching convergence for the first benchmark with $p$-multigrid, $\nu = 4$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 2 | 4 | 5 | $h = 2^{-4}$ | 1 | 2 | 3 | $h = 2^{-4}$ | 1 | 2 | 2 |
| $h = 2^{-5}$ | 3 | 5 | 6 | $h = 2^{-5}$ | 2 | 4 | 5 | $h = 2^{-5}$ | 2 | 3 | 3 |
| $h = 2^{-6}$ | 2 | 5 | 6 | $h = 2^{-6}$ | 2 | 5 | 6 | $h = 2^{-6}$ | 2 | 4 | 5 |
| $h = 2^{-7}$ | 2 | 4 | 4 | $h = 2^{-7}$ | 2 | 4 | 5 | $h = 2^{-7}$ | 2 | 4 | 5 |

The number of V-cycles needed to achieve convergence for the second benchmark are presented in Table 6. Compared to the first benchmark, convergence is obtained in the same or even a lower number of V-cycles. Furthermore, only a small increase of the number of V-cycles needed is observed when the number of patches increases. For one configuration, however, the $p$-multigrid method diverges, as expected based on the spectral analysis; see Table 2.

**Table 6** Number of V-cycles needed to achieve convergence for the second benchmark with $p$-multigrid, $\nu = 2$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 2 | 3 | 5 | $h = 2^{-4}$ | 2 | 2 | 3 | $h = 2^{-4}$ | 1 | 2 | – |
| $h = 2^{-5}$ | 3 | 4 | 5 | $h = 2^{-5}$ | 2 | 3 | 4 | $h = 2^{-5}$ | 2 | 2 | 3 |
| $h = 2^{-6}$ | 3 | 4 | 5 | $h = 2^{-6}$ | 2 | 4 | 5 | $h = 2^{-6}$ | 2 | 4 | 5 |
| $h = 2^{-7}$ | 3 | 4 | 4 | $h = 2^{-7}$ | 2 | 4 | 5 | $h = 2^{-7}$ | 2 | 4 | 5 |

Table 7 presents the number of V-cycles needed to achieve convergence for the third benchmark. Again, convergence is reached in a relatively low number of V-cycles. Furthermore, no instabilities are observed for the considered configurations. The increase in the number of V-cycles needed to reach convergence when the number of patches is increased is only mild.

**Table 7** Number of V-cycles needed to achieve convergence for the third benchmark with $p$-multigrid, $\nu = 2$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 3 | 4 | 5 | $h = 2^{-4}$ | 2 | 2 | 3 | $h = 2^{-4}$ | 2 | 2 | 3 |
| $h = 2^{-5}$ | 3 | 5 | 6 | $h = 2^{-5}$ | 2 | 4 | 5 | $h = 2^{-5}$ | 2 | 3 | 3 |
| $h = 2^{-6}$ | 3 | 4 | 5 | $h = 2^{-6}$ | 2 | 5 | 6 | $h = 2^{-6}$ | 2 | 4 | 5 |
| $h = 2^{-7}$ | 3 | 4 | 5 | $h = 2^{-7}$ | 2 | 4 | 5 | $h = 2^{-7}$ | 2 | 4 | 5 |

Alternatively, the $p$-multigrid method is applied as a preconditioner within a Biconjugate gradient stabilized (BiCGSTAB) method. At the preconditioning phase of every iteration of the BiCGSTAB method, a single V-cycle is applied. Again, a tolerance of $\epsilon = 10^{-8}$ is adopted as a stopping criterium for the BiCGSTAB solver.

Results obtained for the first benchmark are presented in Table 8. Compared to the use of $p$-multigrid as a stand-alone solver, the number of iterations needed on a single patch geometry is smaller for all configurations. In case of a multipatch geometry, however, the number of iterations needed reduces even more when a BiCGSTAB method is adopted. Hence, the difference in BiCGSTAB iterations for single patch and multipatch geometries becomes even smaller.

**Table 8** Number of iterations needed to reach convergence for the first benchmark with preconditioned BiCGSTAB, $\nu = 2$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 2 | 2 | 2 | $h = 2^{-4}$ | 1 | 2 | 2 | $h = 2^{-4}$ | 1 | 1 | 2 |
| $h = 2^{-5}$ | 2 | 2 | 3 | $h = 2^{-5}$ | 1 | 2 | 2 | $h = 2^{-5}$ | 1 | 2 | 2 |
| $h = 2^{-6}$ | 2 | 2 | 3 | $h = 2^{-6}$ | 2 | 2 | 3 | $h = 2^{-6}$ | 1 | 2 | 2 |
| $h = 2^{-7}$ | 2 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 |

For the second and third benchmark, results are presented in Table 9 and 10, respectively. For the single patch geometry, the number of iterations with the BiCGSTAB method is again smaller compared to the number of V-cycles for the $p$-multigrid method for almost all configurations. A slightly larger reduction in the number of iterations can be observed for some numerical experiments in case of a multipatch geometry. Note that BiCGSTAB restores stability for the setting in which the $p$-multigrid algorithm separately is unstable; see Table 2.

**Table 9** Number of iterations needed for reaching convergence for the second benchmark with preconditioned BiCGSTAB, $\nu = 2$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 1 | 2 | 2 | $h = 2^{-4}$ | 1 | 1 | 2 | $h = 2^{-4}$ | 1 | 1 | 3 |
| $h = 2^{-5}$ | 2 | 2 | 2 | $h = 2^{-5}$ | 1 | 1 | 2 | $h = 2^{-5}$ | 1 | 1 | 2 |
| $h = 2^{-6}$ | 2 | 2 | 2 | $h = 2^{-6}$ | 1 | 2 | 2 | $h = 2^{-6}$ | 1 | 2 | 2 |
| $h = 2^{-7}$ | 2 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 |

**Table 10** Number of iterations needed for reaching convergence for the third benchmark with preconditioned BiCGSTAB, $\nu = 2$.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 1 | 2 | 2 | $h = 2^{-4}$ | 1 | 1 | 2 | $h = 2^{-4}$ | 1 | 1 | 1 |
| $h = 2^{-5}$ | 2 | 2 | 3 | $h = 2^{-5}$ | 1 | 2 | 2 | $h = 2^{-5}$ | 1 | 1 | 2 |
| $h = 2^{-6}$ | 2 | 2 | 3 | $h = 2^{-6}$ | 1 | 2 | 3 | $h = 2^{-6}$ | 1 | 2 | 2 |
| $h = 2^{-7}$ | 2 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 | $h = 2^{-7}$ | 1 | 2 | 2 |

As discussed in Section 3, different cycle types can be adopted. The use of a W-cycle instead of the V-cycle leads to the same number of cycles needed for all numerical experiments. Considering the higher computational costs for a single W-cycle, a V-cycle is adopted throughout the rest of this paper.

### ILUT as a solver

The obtained results are compared to using ILUT as a stand-alone solver. In this way, the effectiveness of the coarse grid correction within the $p$-multigrid method can be investigated. Table 11 shows the number of iterations needed to achieve convergence with ILUT as a solver. For all numerical experiments, the number of iterations needed with ILUT is significantly higher compared to the number of V-cycles needed with $p$-multigrid (see Table 4 for comparison). Furthermore, the number of iterations needed with ILUT as a solver becomes $h$-dependent, leading to a high number of iterations on finer meshes. As shown in Table 12 and Table 13, the same observations can be made for the second and third benchmark, respectively. These results indicate that the coarse grid correction is necessary to obtain a low number of iterations until convergence in reached.

**Table 11** Number of iterations needed for achieving convergence for the first benchmark with ILUT as solver.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 25 | 40 | 55 | $h = 2^{-4}$ | 12 | 18 | 29 | $h = 2^{-4}$ | 7 | 10 | 18 |
| $h = 2^{-5}$ | 96 | 125 | 148 | $h = 2^{-5}$ | 50 | 53 | 67 | $h = 2^{-5}$ | 22 | 27 | 35 |
| $h = 2^{-6}$ | 352 | 397 | 437 | $h = 2^{-6}$ | 171 | 182 | 199 | $h = 2^{-6}$ | 80 | 86 | 106 |
| $h = 2^{-7}$ | 1280 | 1356 | 1440 | $h = 2^{-7}$ | 609 | 623 | 664 | $h = 2^{-7}$ | 288 | 307 | 324 |

**Table 12** Number of iterations needed to obtain convergence for the second benchmark with ILUT as solver.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 14 | 17 | 28 | $h = 2^{-4}$ | 6 | 9 | 15 | $h = 2^{-4}$ | 4 | 6 | – |
| $h = 2^{-5}$ | 56 | 55 | 63 | $h = 2^{-5}$ | 23 | 21 | 30 | $h = 2^{-5}$ | 12 | 13 | 16 |
| $h = 2^{-6}$ | 194 | 219 | 217 | $h = 2^{-6}$ | 76 | 84 | 83 | $h = 2^{-6}$ | 37 | 39 | 41 |
| $h = 2^{-7}$ | 716 | 710 | 700 | $h = 2^{-7}$ | 251 | 276 | 301 | $h = 2^{-7}$ | 131 | 138 | 137 |

**Table 13** Number of iterations needed to obtain convergence for the third benchmark with ILUT as solver.

| $p = 2$ | # patches | | | $p = 3$ | # patches | | | $p = 4$ | # patches | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 4 | 16 | | 1 | 4 | 16 | | 1 | 4 | 16 |
| $h = 2^{-4}$ | 33 | 26 | 32 | $h = 2^{-4}$ | 13 | 10 | 16 | $h = 2^{-4}$ | 8 | 7 | 12 |
| $h = 2^{-5}$ | 126 | 90 | 88 | $h = 2^{-5}$ | 45 | 25 | 33 | $h = 2^{-5}$ | 23 | 14 | 17 |
| $h = 2^{-6}$ | 469 | 290 | 288 | $h = 2^{-6}$ | 168 | 88 | 100 | $h = 2^{-6}$ | 80 | 40 | 47 |
| $h = 2^{-7}$ | 1667 | 1046 | 1050 | $h = 2^{-7}$ | 596 | 320 | 332 | $h = 2^{-7}$ | 283 | 150 | 146 |

## Comparison $h$- and $hp$-multigrid

In the previous subsection, it was shown that a coarse grid correction is necessary to obtain an efficient solution method. To determine the quality of the coarse grid correction with $p$-multigrid in more detail, results are compared with $h$- and $hp$-multigrid methods. In these methods, only the way in which the hierarchy is constructed differs. For the $h$-multigrid method, coarsening in $h$ is applied, while for the $hp$-multigrid method coarsening in $h$ and $p$ is applied simultaneously. All other components (i.e. smoothing, prolongation and restriction) are identical. It should be noted that, since coarsening in $h$ leads to a nested hierarchy of discretizations, a canonical prolongation/restriction operator could be defined for the $h$-multigrid method. These transfer operators are, however, not taken into account in this paper. Results obtained for the benchmarks with the different coarsening strategies on a multipatch geometry for different values of $h$ and $p$ are shown in Table 14, 15 and 16, respectively.

**Table 14** Comparison of $p$-multigrid with $h$- and $hp$-multigrid for the first benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 5 | 8 | 9 | $h = 2^{-3}$ | 3 | 4 | 4 | $h = 2^{-3}$ | 2 | 3 | 3 |
| $h = 2^{-4}$ | 6 | 18 | 20 | $h = 2^{-4}$ | 5 | 10 | 11 | $h = 2^{-4}$ | 4 | 6 | 6 |
| $h = 2^{-5}$ | 5 | 28 | 31 | $h = 2^{-5}$ | 6 | 25 | 31 | $h = 2^{-5}$ | 5 | 14 | 16 |
| $h = 2^{-6}$ | 5 | 32 | 35 | $h = 2^{-6}$ | 5 | 56 | 70 | $h = 2^{-6}$ | 5 | 36 | 48 |

For all benchmarks, the number of V-cycles needed with $p$-multigrid is significantly lower for all configurations compared to $h$- and $hp$-multigrid. Furthermore, the difference in the number of V-cycles increases when the knot span size is halved. In general, coarsening in $h$ is more efficient compared to coarsening both in $h$ and $p$. Results indicate that the coarsening in $p$ leads to the most effective coarse grid correction, resulting in the lowest number of V-cycles.

**Table 15** Comparison of $p$-multigrid with $h$- and $hp$-multigrid for the second benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 3 | 4 | 4 | $h = 2^{-3}$ | 2 | 2 | 3 | $h = 2^{-3}$ | 2 | 2 | 2 |
| $h = 2^{-4}$ | 4 | 10 | 11 | $h = 2^{-4}$ | 3 | 5 | 5 | $h = 2^{-4}$ | 2 | 3 | 3 |
| $h = 2^{-5}$ | 4 | 20 | 22 | $h = 2^{-5}$ | 4 | 13 | 16 | $h = 2^{-5}$ | 4 | 8 | 8 |
| $h = 2^{-6}$ | 4 | 26 | 27 | $h = 2^{-6}$ | 4 | 31 | 39 | $h = 2^{-6}$ | 4 | 19 | 23 |

**Table 16** Comparison of $p$-multigrid with $h$- and $hp$-multigrid for the third benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 4 | 6 | 6 | $h = 2^{-3}$ | 2 | 3 | 3 | $h = 2^{-3}$ | 2 | 2 | 2 |
| $h = 2^{-4}$ | 5 | 14 | 16 | $h = 2^{-4}$ | 4 | 6 | 6 | $h = 2^{-4}$ | 3 | 3 | 3 |
| $h = 2^{-5}$ | 4 | 23 | 25 | $h = 2^{-5}$ | 5 | 14 | 17 | $h = 2^{-5}$ | 4 | 8 | 8 |
| $h = 2^{-6}$ | 4 | 28 | 30 | $h = 2^{-6}$ | 4 | 34 | 46 | $h = 2^{-6}$ | 4 | 20 | 25 |

Besides the number of iterations, CPU times have been compared for the different multigrid methods. This comparison is in particular interesting, since the coarse grid correction is more expensive for $p$-multigrid methods compared to $h$- and $hp$-multigrid approaches. A serial implementation is considered on a Intel(R) Xeon(R) E5-2687W CPU (3.10 GHz). Table 17, 18 and 19 present the computational times (in seconds) for all benchmarks obtained with the different coarsening strategies.

**Table 17** Computational times (in seconds) with $p$, $h$- and $hp$-multigrid for the first benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 1.4 | 1.0 | 1.2 | $h = 2^{-3}$ | 1.4 | 1.4 | 1.4 | $h = 2^{-3}$ | 2.0 | 1.8 | 1.9 |
| $h = 2^{-4}$ | 4.2 | 4.9 | 4.5 | $h = 2^{-4}$ | 5.1 | 5.8 | 5.8 | $h = 2^{-4}$ | 7.4 | 7.9 | 6.8 |
| $h = 2^{-5}$ | 13.1 | 22.1 | 21.9 | $h = 2^{-5}$ | 21.9 | 38.7 | 40.7 | $h = 2^{-5}$ | 31.6 | 48.0 | 44.7 |
| $h = 2^{-6}$ | 62.0 | 126.3 | 127.7 | $h = 2^{-6}$ | 105.1 | 384.2 | 419.5 | $h = 2^{-6}$ | 169.1 | 508.8 | 542.4 |

On coarser grids, the computational times for all multigrid methods are comparable. For smaller values of $h$, however, the computational time needed with $p$-multigrid is significantly smaller compared to $h$- and $hp$-multigrid due to the considerable $h$-dependency of the latter two approaches. Furthermore, the computational time needed with $p$-multigrid scales (almost) linearly with the number of degrees of freedom. This holds for all benchmarks and all values of $p$ considered in this study.

**Table 18** Computational times (in seconds) with $p$, $h$- and $hp$-multigrid for the second benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 1.2 | 1.0 | 1.1 | $h = 2^{-3}$ | 1.6 | 1.5 | 1.7 | $h = 2^{-3}$ | 2.2 | 1.8 | 1.9 |
| $h = 2^{-4}$ | 4.4 | 4.9 | 4.8 | $h = 2^{-4}$ | 5.7 | 5.9 | 5.3 | $h = 2^{-4}$ | 7.3 | 7.7 | 6.9 |
| $h = 2^{-5}$ | 14.4 | 22.1 | 23.0 | $h = 2^{-5}$ | 22.1 | 33.7 | 33.5 | $h = 2^{-5}$ | 35.5 | 42.4 | 36.8 |
| $h = 2^{-6}$ | 64.2 | 142.3 | 136.7 | $h = 2^{-6}$ | 105.6 | 306.3 | 320.8 | $h = 2^{-6}$ | 175.8 | 382.6 | 376.6 |

**Table 19** Computational times (in seconds) with $p$, $h$- and $hp$-multigrid for the third benchmark on 4 patches, $\nu = 2$.

| $p = 2$ | $p$ | $h$ | $hp$ | $p = 3$ | $p$ | $h$ | $hp$ | $p = 4$ | $p$ | $h$ | $hp$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $h = 2^{-3}$ | 1.1 | 1.1 | 1.0 | $h = 2^{-3}$ | 1.3 | 1.3 | 1.6 | $h = 2^{-3}$ | 2.0 | 1.8 | 1.7 |
| $h = 2^{-4}$ | 4.2 | 4.8 | 3.9 | $h = 2^{-4}$ | 5.4 | 5.2 | 5.0 | $h = 2^{-4}$ | 7.1 | 6.4 | 6.1 |
| $h = 2^{-5}$ | 13.1 | 20.0 | 20.1 | $h = 2^{-5}$ | 20.5 | 27.3 | 27.9 | $h = 2^{-5}$ | 29.1 | 32.9 | 29.9 |
| $h = 2^{-6}$ | 59.8 | 120.1 | 119.7 | $h = 2^{-6}$ | 94.3 | 257.5 | 299.1 | $h = 2^{-6}$ | 149.0 | 307.6 | 308.8 |

## 6 Conclusion

In this paper, we have extended our $p$-multigrid solver for IgA discretizations using a Incomplete LU factorization [27] to multipatch geometries. An analysis of the spectrum of the iteration matrix shows that this $p$-multigrid method can be applied on multipatch geometries, with convergence rates essentially independent of the knot span size $h$ and approximation order $p$. Only a mild dependence of the convergence rate on the number of patches is observed. Numerical results, obtained for Poisson's equation on the unit square, the quarter annulus and an L-shaped domain, confirm this analysis. Furthermore, results show the necessity of the coarse grid correction within the $p$-multigrid method. Finally, different coarsening strategies have been compared, indicating that coarsening in $p$ is most effective compared to coarsing in $h$ or $h$ and $p$ simultaneously. Future research should focus on the application of the $p$-multigrid method on partial differential equations of higher-order, for example the biharmonic equation. Furthermore, the use of $p$-multigrid in a HPC framework can be investigated, in which block ILUT can be applied efficiently as a smoother on each multipatch separately.

## 7 Acknowledgements

## References

1. Bazilevs, Y., Calo, V.M., Zhang,Y. and Hughes, T.J.R.: Isogeometric Fluid-structure Interaction Analysis with Applications to Arterial Blood Flow. Computational Mechanics, **38**(4-5), 310–322 (2006)
2. Beirao da Veiga, L., Cho, D., Pavarino, L.F. and Scacchi, S.: Overlapping Schwarz methods for isogeometric analysis. SIAM Journal on Numerical Analysis, **50**(3), 1394–1416 (2012)
3. De Boor, C.: A practical guide to splines. 1st edition. Springer-Verlag, New York (1978)
4. Brenner, S.C. and Scott, L.R.: The mathematical theory of finite element methods, Texts Applied Mathematics, 15, Springer, New York (1994).
5. Briggs, W.L., Henson, V. E. and McCormick, S.F.: A Multigrid Tutorial 2nd edition, SIAM, Philadelphia, (2000)
6. Tielen, R., Wobbes, E., Möller, M. and Beuth, L.: A high order material point method. Proceedings of the 1st International Conference on the Material Point Method, 10-13 January 2017, Delft, The Netherlands, **175**, 265–272 (2017)
7. Cottrell, J.A., Reali, A., Bazilevs, Y. and Hughes, T.J.R.: Isogeometric analysis of structural vibrations. Computer Methods in Applied Mechanics and Engineering, **195**(41-43), 5257–5296 (2006)

8. Donatelli, M., Garoni, C., Manni,C., Capizzano, S. and Speleers,H.: Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. SIAM Journal on Numerical Analysis, **55**(1), 31–62 (2017)

9. Fidkowski, K.J., Oliver, T.A., Lu. J and Darmofal, D.L.: p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. Journal of Computational Physics, **207**(1), 92–113 (2005)

10. Gahalaut, K.P.S., Kraus, J.K. and Tomar,S.K.: Multigrid methods for isogeometric discretizations. Computer Methods in Applied Mechanics and Engineering, **253**, 413–425 (2013)

11. Guennebaud, G., Benoît, J. et al.: Eigen v3, http://eigen.tuxfamily.org, (2010)

12. Helenbrook, B., Mavriplis, D. and Atkins, H.: Analysis of p-Multigrid for Continuous and Discontinuous Finite Element Discretizations. 16th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences

13. Hofreither, C., Takacs, S. and Zulehner, W.: A robust multigrid method for Isogeometric Analysis in two dimensions using boundary correction. Computer Methods in Applied Mechanics and Engineering, **316**, 22–42 (2017)

14. Hofreither, C. and Zulehner, W.: Spectral Analysis of Geometric Multigrid Methods for Isogeometric Analysis. Numerical Methods and Applications, **8962**, 123–129 (2015)

15. Hughes, T.J.R., Cottrell J.A. and Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering. **194**, 4135–4195 (2005)

16. Hughes, T.J.R., Reali, A. and 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. Computer Methods in Applied Mechanics and Engineering, **197**(49-50), 4104–4124 (2008)

17. Luo, H., Baum, J.D. and Löhner, R.: A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. Journal of Computational Physics, **211**(2), 767–783 (2006)

18. Luo, H., Baum, J.D. and Löhner, R.: Fast p-Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds. AIAA Journal, **46**(3), 635–652 (2008)

19. Nitsche, J.: Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen die keinen Randbedingungen unterworfen sind. Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, **36**(1), 9–15 (1971)

20. de la Riva, A., Rodrigo, C. and Gaspar, F.: An efficient multigrid solver for isogeometric analysis. arXiv:1806.05848v1, 2018

21. van Slingerland, P. and Vuik, C.: Fast linear solver for diffusion problems with applications to pressure computation in layered domains. Computational Geosciences, **18**(3-4), 343–356 (2014)

22. Saad, Y.: ILUT: A dual threshold incomplete LU factorization. Numerical Linear Algebra with Applications, **1**(4), 387–402 (1994)

23. Saad, Y.: SPARSKIT: A basic tool kit for sparse matrix computations. (1994)

24. Sampath, R.S. and Biros, G.: A parallel geometric multigrid method for finite elements on octree meshes, SIAM Journal on Scientific Computing, **32**(3), 1361–1392 (2010)

25. Sangalli, G. and Tani,M.: Isogeometric preconditioners based on fast solvers for the Sylvester equation. SIAM Journal on Scientific Computing, **38**(6), 3644–3671 (2016)

26. Tielen, R., Möller, M. and Vuik, C.: Efficient multigrid based solvers for Isogeometric Analysis. Proceedings of the 6th European Conference on Computational Mechanics and the 7th European Conference on Computational Fluid Dynamics, Glasgow, UK, 2018

27. Tielen, R., Möller, M. and Vuik, C.: Efficient *p*-multigrid based solvers for Isogeometric Analysis. Submitted to Computer Methods in Applied Mechanics and Engineering, arXiv:1901.01685

28. Trottenberg, U., Oosterlee, C. and Schüller, A.: Multigrid, Academic Press (2001).

29. Wall, W.A., Frenzel, M.A. and Cyron, C.: Isogeometric structural shape optimization. Computer Methods in Applied Mechanics and Engineering, **197**(33-40), 2976–2988 (2008)