Acta Polytechnica **61**(SI):33–48, 2021
DOI:10.14311/AP.2021.61.0033

NUMERICAL SIMULATION OF LID-DRIVEN CAVITY FLOW BY ISOGEOMETRIC ANALYSIS

Bohumír Bastl*, Marek Brandner, Jiří Egermaier, Hana Horníková, Kristýna Michálková, Eva Turnerová

University of West Bohemia, Faculty of Applied Sciences, Department of Mathematics and European Centres of Excellence New Technologies for the Information Society, Univerzitní 8, 301 00, Plzeň, Czech Republic

* corresponding author: bastl@kma.zcu.cz

Abstract. In this paper, we present numerical results obtained by an in-house incompressible fluid flow solver based on isogeometric analysis (IgA) for the standard benchmark problem for incompressible fluid flow simulation – lid-driven cavity flow. The steady Navier-Stokes equations are solved in their velocity-pressure formulation and we consider only inf-sup stable pairs of B-spline discretization spaces. The main aim of the paper is to compare the results from our IgA-based flow solver with the results obtained by a standard package based on finite element method with respect to degrees of freedom and stability of the solution. Further, the effectiveness of the recently introduced rIgA method for the steady Navier-Stokes equations is studied.

The authors dedicate the paper to Professor K. Kozel on the occasion of his 80th birthday.

Keywords: Isogeometric analysis, fluid flow simulation, B-splines, Navier-Stokes equations.

1. Introduction

The fluid flow simulation is one of the fundamental problems solved in engineering practice. During the history, many different numerical methods were proposed to solve this extremely challenging problem, ranging from Finite Difference Methods (FDM), through Finite Volume Methods (FVM) to Finite Element Methods (FEM) with many variants of these methods (see e.g. [1–3]). Relatively recently, Hughes et al. (see [4, 5]) introduced a powerful numerical method for solving partial differential equations called *isogeometric analysis* (IgA). This method has many features common with FEM and the main motivation is to bridge the gap between geometric modelling (Computer-Aided Design – CAD) and numerical simulation (Finite Element Analysis – FEA). Optimally, one would like to use directly the description of an object in CAD system also for numerical simulations. However, this is not possible with the standard approaches (FDM, FVM, FEM) since they require the construction of some computational mesh (typically, triangular/quadrilateral meshes for 2D problems and tetrahedral/hexahedral for 3D problems) which is, moreover, only an approximation of the original computational domain.

The isogeometric analysis is based on an isoparametric framework, i.e., the same basis functions are used for representation of the geometry and for the solution space. Currently, the de facto industrial standard for representation of complex objects in CAD systems are Non-Uniform Rational B-spline (NURBS) objects (see [6]). This is the reason why isogeometric analysis was formulated specifically for this class of objects. They allow for the exact representation of a wide variety of complex shapes in a compact form, i.e., by a small number of control points. Even though the NURBS description of an object in CAD system can be used directly for numerical simulations only in case when IgA-BEM (the isogeometric formulation of Boundary Element Method, see e.g. [7]) is used, the exact NURBS representation is still beneficial. Many methods providing geometrically exact bivariate/trivariate NURBS representation of planar/spatial domains, respectively, from its boundary were proposed in the last years (see e.g. [8–13]). Such NURBS representations then serve as computational meshes used in solvers based on isogeometric analysis. Other advantages of the isogeometric analysis include e.g. higher accuracy of the solution with respect to a degree of freedom ([14]) or favourableness of higher continuity of NURBS basis functions for solving higher order PDEs ([15]). From a practical point of view, a typical implementation of IgA-based solver automatically allows for basis functions of the geometry and the solution space of any degree and continuity, without any additional effort needed for implementing the solver. This is typically not the case for codes based on FEM, where in p-version of FEM each degree of the basis functions of the solution space needs to be added/implemented individually. Because of many advantages of the IgA, the method has been successfully applied in various practical problems, like linear elasticity, structural vibrations, phase transition phenomena, fluid flow simulation, plate and shell analysis, heat transfer analysis, shape optimization, etc., see e.g. [15–33].

On the other hand, several disadvantages of the isogeometric analysis were identified during the extensive research in the last years. For example, the properties of NURBS objects induce that true local refinement is not possible for this class of objects. This drawback can be overcome by various generalizations of NURBS objects, like T-splines, THB-splines or LR-splines (see e.g. [34]), which were recently studied in the context of isogeometric analysis. Among other drawbacks belong e.g. time-consuming matrix assembly in the construction of the matrix of linear systems arising in Galerkin isogeometric methods (see e.g. [35–37]), degraded performance of direct solvers because of wide support of B-spline/NURBS basis functions (see [38]) or the necessity to modify existing well-known stabilization techniques to suppress spurious oscillations.

In this paper, we focus on one of the standard benchmark problems for incompressible fluid flow simulation – lid-driven cavity flow. This problem is frequently studied in the literature. A well-done overview of the results for different Reynolds numbers obtained by different methods can be found e.g. in the recent paper [39], but even this extensive list is not complete. As it can be seen from the summary of results for this problem in [39], FDM is quite frequently used for obtaining very accurate results for this problem. To the best of our knowledge, there exist also several papers which solve the lid-driven cavity problem by isogeometric analysis. The paper [15] applies isogeometric analysis to solve the fourth order stream function formulation of the Navier-Stokes equations and presents the results for Reynolds numbers up to 5000. Further, the paper [40] uses divergence-conforming B-splines which may be interpreted as smooth generalizations of Raviart-Thomas elements and presents the results for Reynolds numbers up to 1000. Finally, the paper [33] is based on velocity-pressure formulation of steady and unsteady Navier-Stokes equations, uses LBB compatible B-spline spaces which may be regarded as smooth generalizations of Taylor-Hood pairs of finite element spaces and presents the results for Reynolds numbers up to 1000.

The main goal of this paper is to extend the study done in [33] for the lid-driven cavity problem to higher Reynolds number, but in a slightly different manner. We want to compare the solutions obtained by our inhouse fluid flow solver based on isogeometric analysis (see [30, 31]) for different B-spline discretization spaces with solutions obtained by finite element method with respect to degrees of freedom and stability of the solution. Further, we want to study the potential “stabilization effect” of high continuity B-spline discretization spaces (see [41]), i.e., if high continuity B-spline discretization spaces provide solutions with less oscillations than standard finite element spaces and B-spline discretization spaces of low continuity (FEM-like) for the similar number of degrees of freedom. Moreover, we want to investigate if this potential stabilization effect of the high continuity B-spline discretizations dominates the effect of the SUPG stabilization for the low continuity B-spline discretizations. Finally, inspired by the recently introduced rIgA method (see [42, 43]), we want to explore the effectiveness of rIgA method for solving the Navier-Stokes equations, which has not yet been studied.

The paper is organized as follows. Section 2 introduces the steady Navier-Stokes equations, their weak formulation and Galerkin discretization. In Section 3, we overview the fundamentals of B-spline objects and present the main ideas of isogeometric analysis. Section 4 is devoted to a very brief overview of Streamline Upwind/Petrov-Galerkin method used for stabilization of a numerical solution of convection-dominated problems typically for lower degree discretization spaces. In Section 5, we formulate the benchmark problem studied in the paper – the lid-driven cavity flow – and present the reference solutions used for comparison with our results. Section 6 then demonstrates the main results of the paper, i.e., the results obtained by our in-house fluid flow solver based on isogeometric analysis and their comparison with reference solutions and other solutions obtained by a standard FEM package. In Section 7, we conclude the paper.

2. Navier-Stokes equations

The mathematical model of incompressible viscous Newtonian flow is based on the incompressible Navier–Stokes equations (NSE). Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, d being the number of spatial dimensions, with the boundary $\partial\Omega$ consisting of two complementary parts, Dirichlet $\partial\Omega_D$ and Neumann $\partial\Omega_N$. The steady-state incompressible Navier–Stokes problem is given as a system of $d+1$ differential equations together with boundary conditions

$$\begin{array}{rcll} -\nu\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &=& \mathbf{0} & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} &=& 0 & \text{in } \Omega, \\ \mathbf{u} &=& \mathbf{g}_D & \text{on } \partial\Omega_D, \\ \nu\frac{\partial\mathbf{u}}{\partial\mathbf{n}} - \mathbf{n}p &=& \mathbf{0} & \text{on } \partial\Omega_N, \end{array} \tag{1}$$

where $\mathbf{u}$ is the flow velocity, p is the kinematic pressure, ν is the kinematic viscosity and $\mathbf{g}_D$ is a given function. If the velocity is specified everywhere on the boundary, the pressure solution is only unique up to a hydrostatic constant.

The laminar/turbulent behaviour of the fluid can be predicted by the dimensionless Reynolds number generally defined as the ratio of inertial forces to viscous forces

$$Re = \frac{\hat{u}L}{\nu}, \tag{2}$$

where $\hat{u}$ is the mean velocity of the fluid and L is the characteristic linear dimension.

2.1. Galerkin discretization

In order to derive the weak formulation of the problem (1), we define velocity solution space V and test function space V_0 as follows

$$\begin{array}{rcl} V &=& \{\mathbf{u} \in H^1(\Omega)^d \,|\, \mathbf{u} = \mathbf{g}_D \text{ on } \partial\Omega_D\}, \\ V_0 &=& \{\mathbf{v} \in H^1(\Omega)^d \,|\, \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}, \end{array} \tag{3}$$

where the boundary equalities are understood in the sense of traces and we assume that $\mathbf{g}_D \in H^{1/2}(\partial\Omega)^d$. By multiplying the first line of (1) by a test function $\mathbf{v} \in V_0$ and the second line by a test function $q \in L^2(\Omega)$, integrating over Ω and using Green's theorem, we obtain the weak formulation: find $\mathbf{u} = (u_1, \dots, u_d) \in V$ and $p \in L^2(\Omega)$ such that

$$\begin{aligned} \nu \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} + \int_\Omega (\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{v} - \int_\Omega p\nabla\cdot\mathbf{v} = 0, \\ \int_\Omega q\nabla\cdot\mathbf{u} = 0 \end{aligned} \tag{4}$$

for all $\mathbf{v} = (v_1, \dots, v_d) \in V_0$ and $q \in L^2(\Omega)$, where

$$\begin{aligned} \nabla\mathbf{u} : \nabla\mathbf{v} &= \sum_{i=1}^{d} \nabla u_i \cdot \nabla v_i = \sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j}, \\ (\mathbf{u}\cdot\nabla\mathbf{u})\cdot\mathbf{v} &= \sum_{i=1}^{d}\sum_{j=1}^{d} u_j \frac{\partial u_i}{\partial x_j} v_i. \end{aligned}$$

The idea of Galerkin discretization method is to define finite dimensional subspaces $V^h \subset V$, $V_0^h \subset V_0$, $W^h \subset L^2(\Omega)$ together with their basis functions and solve the problem projected into these subspaces. Then, the discrete weak solutions $\mathbf{u}_h \in V^h$ and $p_h \in W^h$ are searched such that

$$\begin{aligned} \nu \int_\Omega \nabla\mathbf{u}_h : \nabla\mathbf{v}_h + \int_\Omega (\mathbf{u}_h\cdot\nabla\mathbf{u}_h)\cdot\mathbf{v}_h - \\ - \int_\Omega p_h\nabla\cdot\mathbf{v}_h &= 0, \\ \int_\Omega q_h\nabla\cdot\mathbf{u}_h &= 0 \end{aligned} \tag{5}$$

for all $\mathbf{v}_h \in V_0^h$ and $q_h \in W^h$.

Because of the non-linearity in the convective term, the problem needs to be linearized and solved iteratively. Using Picard's iteration method, we search $\mathbf{u}_h^{n+1} \in V^h$ and $p_h^{n+1} \in W^h$ as a solution of the following Oseen problem

$$\begin{aligned} \nu \int_\Omega \nabla\mathbf{u}_h^{n+1} : \nabla\mathbf{v}_h + \int_\Omega (\mathbf{u}_h^n\cdot\nabla\mathbf{u}_h^{n+1})\cdot\mathbf{v}_h - \\ - \int_\Omega p_h^{n+1}\nabla\cdot\mathbf{v}_h &= 0, \\ \int_\Omega q_h\nabla\cdot\mathbf{u}_h^{n+1} &= 0 \end{aligned} \tag{6}$$

for all $\mathbf{v}_h \in V_0^h$ and $q_h \in W^h$. The initial guess $\mathbf{u}^0$ is often chosen to be zero, which means solving the Stokes equations in the first Picard iteration.

Particular Galerkin-based discretization methods are then defined by the choice of the subspaces and their basis functions. Considering the basis of V_0^h formed by vector functions

$$\big(\underbrace{Q_i^u(\mathbf{x}), 0, \dots, 0}_{d}\big), \dots, \big(\underbrace{0, \dots, 0, Q_i^u(\mathbf{x})}_{d}\big),\ i = 1, \dots, n_u$$

and the basis of W^h formed by functions $Q_i^p(\mathbf{x})$, $i = 1, \dots, n_p$, we obtain after discretization a nonsymmetric saddle-point linear system of the form

$$\begin{bmatrix} \mathbf{F} & -\mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \tag{7}$$

where $\mathbf{F} \in \mathbb{R}^{d\cdot n_u \times d\cdot n_u}$ is block diagonal with the diagonal blocks consisting of the discretization of the viscous term and the linearized convective term, i.e.,

$$\mathbf{F} = \mathrm{diag}\Big(\underbrace{\mathbf{A} + \mathbf{N}(\mathbf{u}_h^n), \dots, \mathbf{A} + \mathbf{N}(\mathbf{u}_h^n)}_{d}\Big)$$

and

$$\begin{aligned} \mathbf{A} = [A_{ij}] &= \left[\nu \int_\Omega \nabla Q_i^u(\mathbf{x}) \cdot \nabla Q_j^u(\mathbf{x}) \mathrm{d}\Omega\right], \\ \mathbf{N}(\mathbf{u}) = [N_{ij}(\mathbf{u})] &= \left[\int_\Omega Q_i^u(\mathbf{x})\big(\mathbf{u}\cdot\nabla Q_j^u(\mathbf{x})\big)\mathrm{d}\Omega\right], \end{aligned}$$

$\mathbf{B}^T \in \mathbb{R}^{d\cdot n_u \times n_p}$ and $\mathbf{B} \in \mathbb{R}^{n_p \times d\cdot n_u}$ consist of d blocks, i.e.,

$$\mathbf{B} = \Big[\mathbf{B}_1, \dots, \mathbf{B}_d\Big]$$

and

$$\mathbf{B}_m = [B_{mij}] = \left[\int_\Omega Q_i^p(\mathbf{x})\big(\nabla Q_j^u(\mathbf{x})\cdot\mathbf{e}_m\big)\mathrm{d}\Omega\right].$$

As the right-hand side of (4) is equal to zero, $\mathbf{f}$ and $\mathbf{g}$ only reflect the given boundary conditions.

2.2. LBB condition

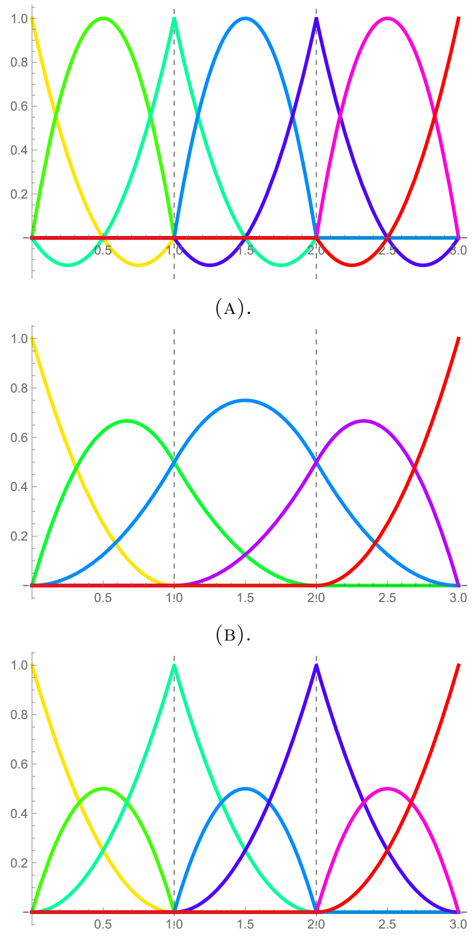
To get a stable Galerkin discretization of a saddle-point problem, the so-called LBB (or inf-sup) condition (see [44])

$$\inf_{q\in W^h\setminus\{0\}} \sup_{\mathbf{v}\in V^h\setminus\{\mathbf{0}\}} \frac{\int_\Omega q\nabla\cdot\mathbf{v}}{||\mathbf{v}||_{H^1(\Omega)^d}||q||_{L^2(\Omega)}} \geq \beta > 0 \tag{8}$$

has to be satisfied, where β is a constant independent of the mesh. Here, we consider only the case of so-called inf-sup stable combinations of velocity and pressure spaces ensuring that the inf-sup condition is satisfied. In standard finite elements, there are several known types of stable elements. One of the most popular is a Taylor–Hood element, where basis functions of degree k are used to approximate the pressure and basis functions of degree $k+1$ are used for the velocities.

3. Isogeometric analysis

In this section, we present fundamentals of isogeometric analysis with a special focus on the problem solved in this paper – fluid flow simulation in a lid-driven cavity.



(A).

(B).

(C).

Figure 1. Comparison of 1D quadratic discretization bases on a "mesh" with three elements: (A) finite element basis, (B) isogeometric basis with C^1 interelement continuity, (C) isogeometric basis with C^0 interelement continuity.

B-spline and NURBS objects are described using a B-spline basis. Its construction is based on a so-called knot vector $\Xi = (\xi_1, \xi_2, \ldots, \xi_{n+k+1})$, where $\xi_i \le \xi_{i+1}$ are the knots, k is the basis degree and n is the number of basis functions. Note that the knots can be multiple. The i-th B-spline basis function of degree k, $N_{i,k}(\xi)$, is defined recursively by Cox-de Boor formula as follows:

$$\begin{aligned}
&\text{for } k = 0: \\
&N_{i,0}(\xi) \quad = \quad \begin{cases} 1, & \xi_i \le \xi < \xi_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \\
&\text{for } k > 0: \\
&N_{i,k}(\xi) \quad = \quad s_{i,k}(\xi) N_{i,k-1}(\xi) + \\
&\qquad\qquad\quad + \big(1 - s_{i+1,k}(\xi)\big) N_{i+1,k-1}(\xi), \qquad (9) \\
&\text{where} \\
&s_{i,k}(\xi) \quad := \quad \begin{cases} \dfrac{\xi - \xi_i}{\xi_{i+k} - \xi_i}, & \xi_i < \xi_{i+k}, \\ \text{arbitrary}, & \text{otherwise}. \end{cases}
\end{aligned}$$

Note that $N_{i,k-1}(\xi) = 0$ if $\xi_i = \xi_{i+k}$.

One of the fundamental properties of B-spline basis functions is that they are C^{k-m_i} continuous at the knot ξ_i, where m_i is the multiplicity of the knot ξ_i. Other properties include non-negativity, partition of unity and local support.

Generally, IgA-based computations are based on B-spline/NURBS surfaces (in 2D) or B-spline/NURBS volumes (in 3D). Moreover, as it is well-known that the tensor-product structure of B-spline/NURBS objects prevents the possibility to describe an arbitrary computational domain by one B-spline/NURBS object. This is the reason why so-called multi-patch B-spline/NURBS meshes are typically used for the description of computational domains in isogeometric analysis. Our in-house solver used for obtaining results in Section 6 works with general, free-form multi-patch 2D and 3D B-spline domains.

However, in this paper we focus on a geometrically very simple domain – cavity – which allows (under some other conditions mentioned bellow) to use only a so-called *B-spline function*.

Let us consider a two-dimensional domain $\Omega = \langle 0, 1\rangle \times \langle 0, 1\rangle \subset \mathbb{R}^2$ representing the cavity (see Sections 5 and 6) and let us consider a B-spline function $f(\xi, \psi)$ of degree k and order of continuity r in both parameters ξ and ψ in the form

$$\begin{aligned}
f(\xi, \psi) &= \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} N_{i,k}(\xi) N_{j,k}(\psi) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} c_{i,j} Q_{i,j}(\xi, \psi),
\end{aligned} \qquad (10)$$

where $c_{i,j}$ are coefficients of the B-spline function, $N_{i,k}(\xi)$ and $N_{j,k}(\psi)$ are B-spline basis functions corresponding to knot vectors $\Xi = (\xi_1, \xi_2, \ldots, \xi_{n+k+1})$, $\Psi = (\psi_1, \psi_2, \ldots, \psi_{m+k+1})$, respectively, and the product of two univariate B-spline basis functions $N_{i,k}(\xi) N_{j,k}(\psi)$ is denoted as one bivariate function $Q_{i,j}(\xi, \psi)$. As in our experiments we use open knot vectors in the form

$$(\underbrace{0, \ldots, 0}_{k+1}, t_{k+2}, \ldots, t_n, \underbrace{1, \ldots, 1}_{k+1}),$$

where k is the degree of basis functions, the parametric domain of the B-spline function f and the computational domain Ω coincide (both are represented by the unit square). Moreover, various degrees and continuities of the B-spline computational meshes used in experiments in Section 6 are obtained by degree elevation and knot insertion (see [6]) starting from bilinear B-spline description of Ω. This means that for all used B-spline computational meshes a point (ξ, ψ) in the parametric domain coincides with the point (x, y) in the physical domain. Altogether, the above mentioned facts allow us to limit to B-spline functions, instead of more general B-spline surfaces.

Let us denote the tensor product B-spline space spanned by the basis functions $Q_{i,j}(\xi, \psi)$ as

$$\mathcal{S}_r^k = \operatorname{span}\{Q_{i,j}\}_{i=1,j=1}^{n,m} \qquad (11)$$

and the Cartesian product of these tensor product B-spline spaces as

$$\boldsymbol{\mathcal{S}}_r^k = \mathcal{S}_r^k \times \mathcal{S}_r^k.$$

For simplicity of notation, let us re-index the coefficients and the basis functions with one index ℓ such that

$$f(\xi,\psi) = \sum_{\ell=1}^{N} c_\ell Q_\ell(\xi,\psi), \quad N = n \cdot m. \tag{12}$$

A B-spline mesh used in isogeometric analysis is already defined by the knot vectors Ξ, Ψ. If the subsequent knots are different from each other, i.e. $\xi_i \neq \xi_{i+1}$ and $\psi_j \neq \psi_{j+1}$, then $[\xi_i, \xi_{i+1}] \times [\psi_j, \psi_{j+1}]$ defines an element of the computational mesh.

Let us now define the discrete spaces V^h, V_0^h, W^h as subspaces of V, V_0, W generated by the basis functions defining the geometry of Ω. The velocity and pressure spaces must satisfy the condition (8). The inf-sup stability of some combinations of the discrete solution spaces in isogeometric analysis is addressed e.g. in [45, 46]. As examples of stable pairs of B-spline discretization spaces, let us mention Taylor–Hood (TH) pairs of B-spline spaces (an analogue of the Taylor–Hood pairs of finite element spaces) which can be defined by taking the pressure basis functions equal to the geometry basis functions (i.e., $Q_\ell^p(\xi,\psi) = Q_\ell$, $\forall \ell$) and the velocity basis functions $\{Q_j^u(\xi,\psi)\}$ as the B-spline basis functions obtained by degree elevation of the pressure basis functions $\{Q_\ell^p(\xi,\psi)\}$. Hence, the velocity basis functions are of bi-degree $(k+1, k+1)$, both velocity and pressure bases are defined on the same mesh and have the same order of continuity. This yields the following definition

$$\begin{aligned} V^h &= \{\mathbf{u} : \mathbf{u} \in \boldsymbol{\mathcal{S}}_r^{k+1} \wedge \mathbf{u} = \mathbf{g}_D \text{ on } \partial\Omega_D\}, \\ V_0^h &= \{\mathbf{v} : \mathbf{v} \in \boldsymbol{\mathcal{S}}_r^{k+1} \wedge \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}, \\ W^h &= \{q : q \in \mathcal{S}_r^k\}. \end{aligned} \tag{13}$$

The discrete velocity $\mathbf{u}_h \in V^h$ and pressure $p_h \in W^h$ can be written as linear combinations of the basis functions, i.e.,

$$\mathbf{u}_h = \sum_{i=1}^{N_f^u} \mathbf{u}_i Q_i^u(x,y) + \sum_{i=N_f^u+1}^{N^u} \mathbf{u}_i^* Q_i^u(x,y), \tag{14}$$

$$p_h = \sum_{j=1}^{N^p} p_j Q_j^p(x,y), \tag{15}$$

where

$$Q_i^u(x,y) = Q_i^u(\xi,\psi), \forall i, \quad Q_j^p(x,y) = Q_j^p(\xi,\psi), \forall j$$

(specifically in this case, see (10) and the discussion below) and $\mathbf{u}_i \in \mathbb{R}^2$ and $p_j \in \mathbb{R}$ are unknown coefficients, N^u, N^p are number of velocity and pressure basis functions, respectively, and N_f^u is the number of "free" velocity basis functions whose coefficients are not fixed due to Dirichlet boundary conditions. The coefficients $\mathbf{u}_i^*$ come from a B-spline representation of the Dirichlet data at the boundary, which is assumed as known.

After substituting (14) and (15) into the discrete linearized weak formulation, we obtain a linear system of the form (7) for the unknown coefficients $\mathbf{u}_i$ and p_j. Similarly to the finite element method, the system matrix is sparse thanks to local supports of the B-spline basis functions. For degree k equal to 1 the B-spline basis is the same as the corresponding finite element basis. For higher degrees, however, they differ from each other. For illustration, see the comparison of quadratic FEM and IgA bases on a 1D "mesh" consisting of three equal elements in Figure 1. Figure 1(a) shows the standard quadratic FEM basis and Figure 1(b) shows a C^1 continuous quadratic B-spline basis for the open knot vector $\Xi = (0,0,0,1,2,3,3,3)$. Obviously, we have less basis functions for the same mesh in isogeometric analysis, they are all pointwise non-negative and generally non-interpolatory, i.e. their coefficients do not correspond to nodal values of the IgA solution. Moreover, their values form partition of unity for arbitrary $\xi \in (\xi_1, \xi_{n+k+1})$ and the support of most of the B-spline basis functions (except the ones close to the boundary) is $k+1$ knot spans. Both FEM and IgA bases result in the same bandwidth, since the maximum number of basis functions that a given basis function shares support with (including itself) is $2k+1$ in both cases. However, the FEM matrix is sparser since there are basis functions with supports containing only one element, thus sharing support with only $k+1$ basis functions. From the matrix structure point of view, a B-spline analogue of the FEM basis is the B-spline basis with C^0 interelement continuity displayed in Figure 1(c), which is obtained for the knot vector $\Xi = (0,0,0,1,1,2,2,3,3,3)$.

4. Streamline Upwind/Petrov-Galerkin (SUPG) stabilization

Since the isogeometric analysis is continuous Galerkin-based method, the numerical solution of convection dominated problems (for high Reynolds numbers) is usually polluted by spurious (unphysical) oscillations, which cause a loss of accuracy and stability. Stabilization techniques are investigated which improve the stability, without degrading accuracy. There are many linear and nonlinear stabilization techniques for convection-dominated problems. Their overview can be found, e.g., in [47]. In our case, we focused only on the SUPG method. The SUPG method proposed in [48] has become the most popular stabilization method. The weighted element residual is added to the Galerkin problem (5), i.e.,

$$\sum_{i_e=1}^{n_{el}} \int_{Q_{i_e}} \tau_S R(\mathbf{u}_h)\mathbf{u}_h \cdot \nabla \mathbf{v}_h, \tag{16}$$

where

$$R(\mathbf{u}_h) \quad = \quad \mathbf{u}_h \cdot \nabla \mathbf{u}_h - \nu \Delta \mathbf{u}_h + \nabla p_h \qquad (17)$$

is the element residual, n_{el} is the number of elements Q_{i_e}, and τ_S is a non-negative SUPG stabilization parameter (cf. e.g. [49, 50])

$$\tau_S = \left(\left(\frac{2 \deg(\mathbf{u}_h) \, \|\mathbf{u}_h\|}{h} \right)^2 + 9 \left(\frac{4\nu}{h^2} \right)^2 \right)^{-\frac{1}{2}}, \quad (18)$$

where $\deg(\mathbf{u}_h)$ is the degree of the velocity basis functions and h is the characteristic size of the element. Alternatives to the stabilization parameter τ_S are discussed e.g. in [51]. Note that the second derivatives are not zero in the interior of the elements if higher order B-spline basis functions are used.

5. Lid-Driven Cavity Flow

In this section, we introduce the standard benchmark problem for incompressible fluid flow simulation – lid-driven cavity flow – for which we present the results of our numerical experiments in Section 6. One of the reasons for this choice is a wide range of comparative data available in the literature.

A thorough overview of results for lid-driven cavity flow is done in [39]. The authors summarize the papers dealing with this problem starting from 1966 up to the most recent results in 2014. For comparisons, the results presented in [52] are frequently used, where the authors used FDM and solved the problem for Reynolds numbers up to 10000. Other papers, which are usually used for comparisons are e.g. [53, 54].

For comparison with our results we chose two references. The first one is the above mentioned recent paper [39], where the authors solve lid-driven cavity flow by FVM for Reynolds numbers up to 5000. The results can be considered as very accurate because a very fine mesh with 1301×1301 grid points for the case of Reynolds number equal to 5000 was used (compare with [52], where FDM with 257×257 grid points was used). The results presented in [55] is another source to which we compare our data. To the best of our knowledge, these results are unpublished in a journal. Here, the author uses FDM to solve the problem on several very fine graded grids ranging from 1201×1201 to 1661×1661 grid points. Then, Richardson extrapolation is used for improving the solutions' accuracy and the extrapolated result is projected onto the grid of 1201×1201 points. Comparing these results with [39], they are also considered as very accurate. Figure 2 shows the comparison of the results presented in [39] and [55] where u_x- and u_y-velocity profiles along vertical/horizontal lines through the geometric center of the cavity are compared.

The reason why we use also reference [55] is that other available sources usually present velocity profiles only in the above mentioned two cross-sections through the computational domain – u_x-velocity profiles along a vertical line through the geometric center of the cavity and u_y-velocity profiles along a horizontal line through the geometric center of the cavity. However, [55] presents velocity profiles also for some other horizontal and vertical cross-sections through the computational domain, which we will also use for the presentation of our results and for comparison in Section 6.

The problem description is shown in Figure 3. Lid-driven cavity flow represents a flow inside a square container which is covered by a lid that slides sideways. From the existing literature, it is commonly accepted that there exists a steady solution of the problem for Reynolds numbers up to 8000 (in 2D, see e.g. [56, 57]). In our study presented in this paper, we use the kinematic viscosity $\nu = 0.0002$ which implies Reynolds number equal to 5000.

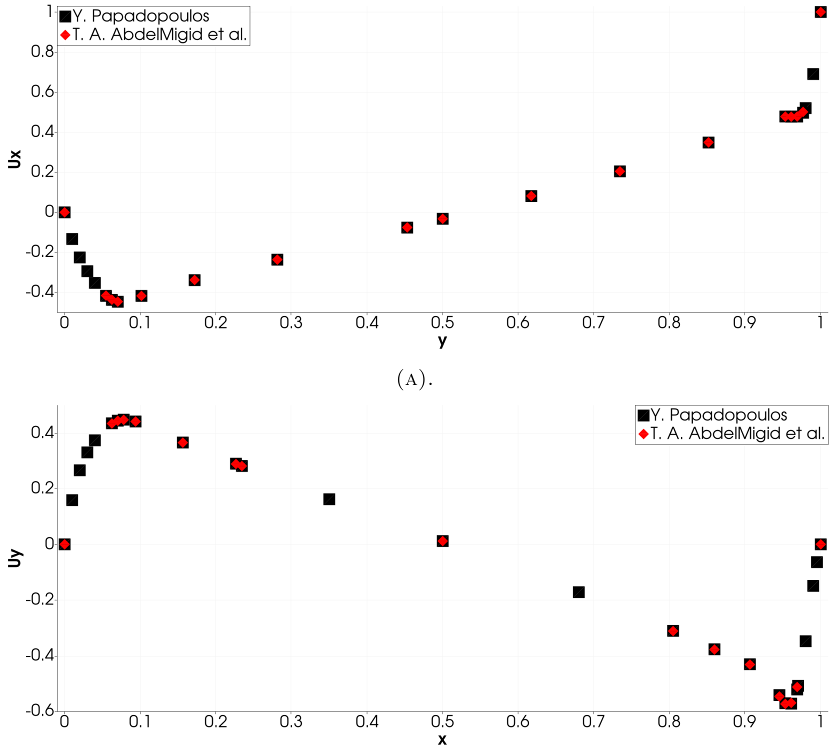
Generally, homogeneous Dirichlet boundary condition for velocity is set on the left, right and bottom boundaries of the computational domain and constant velocity $\mathbf{u} = (1, 0)$ is prescribed on the lid part (top) of the boundary. However, since $\mathbf{g}_D$ is a discontinuous function in this case, it does not lie in $H^{1/2}(\partial\Omega)^d$ and thus the velocity solution cannot be searched in the space V defined in (3), see e.g. [58] for more details on this issue. For discretizations of such singular problem, slower convergence can be expected when refining the mesh.

In order to ensure well-posedness of the problem, a so-called regularization is often used, which means that the discontinuous function $\mathbf{g}_D$ is replaced by some suitable function from $H^{1/2}(\partial\Omega)^d$. For instance, we can define a function $\mathbf{g}_D^\epsilon$ such that its first component changes linearly from 0 to 1 on a part of the boundary of width ϵ near the upper corners as in [59].

However, in this paper, we discretize the original singular problem, i.e., no regularization is used here. Thus, we can compare our IgA solutions with the results available in the references mentioned above, where the singular lid-driven cavity is also considered. In fact, after IgA discretization, the finite dimensional problem is solved with a continuous boundary condition $\mathbf{g}_{Dh}$ defined as a B-spline approximation of $\mathbf{g}_D$ on the given mesh.

In principle, there are two ways to approximate the discontinuous boundary condition, called "leaky" and "non-leaky" cavity. The leaky cavity approach considers the upper corners as part of the lid and the first component of $\mathbf{g}_{Dh}$ goes from 0 to 1 on some part of the vertical walls (dependent on the mesh) near the upper corners. We consider the non-leaky cavity approach, where, on the contrary, the upper corners are considered as part of the vertical walls.

As Dirichlet boundary conditions for velocity are imposed on the whole boundary of the computational domain, the pressure is determined up to an arbitrary constant. Thus, to have the discrete pressure field uniquely defined, we fix the value of pressure to zero at the bottom left corner.



(A).

(B).

Figure 2. Comparison of reference solutions (Y. Papadopoulos stands for [55], T. A. AbdelMigid et al. for [39]): u_x-velocity profile along a vertical line at $x = 0.5$ (A), u_y-velocity profile along a horizontal line at $y = 0.5$ (B).

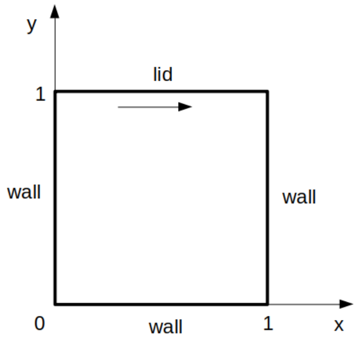


Figure 3. Lid-driven cavity flow – the problem description.

	$\mathcal{S}_0^2 \times \mathcal{S}_0^1$	$\mathcal{S}_0^4 \times \mathcal{S}_0^3$	$\mathcal{S}_2^4 \times \mathcal{S}_2^3$
32 × 32	9026	41666	9674
64 × 64	36482	167298	37770
128 × 128	146690	670466	149258

Table 1. Degrees of freedom (DOFs) for simulations on different meshes in the in-house IgA-based flow solver.

6. Results

In this section, we present the numerical results obtained by our in-house isogeometric fluid flow solver for the model problem, lid-driven cavity flow, and compare them with reference solutions and with the numerical results obtained by standard package FEniCS (see [60]), which is based on the finite element method. Let us note that the simplicity of this benchmark example is in favor of FEM because it is possible to represent the computational domain exactly also by finite element meshes. On more complicated domains, where finite element meshes only approximate the domain boundary, isogeometric analysis have an advantage of exact representation of the computational domain by a B-spline/NURBS mesh.

Our in-house isogeometric fluid flow solver is implemented in C++ in the framework of G+Smo (Geometry and Simulation modules) library (see [61]). For all the simulations performed with our solver we used the kinematic viscosity $\nu = 0.0002$ yielding $Re = 5000$ and we compare them with the results obtained by FEniCS and with the results of the reference solutions presented in [39] or [55] for the same Reynolds number. Specifically, the results from [39] are used for the comparison of u_x- and u_y-velocity profiles along cross-sections of the computational domain at $x = 0.5$ and $y = 0.5$, respectively, and the results from [55] for comparison of velocity profiles along all other cross-sections of the computational domain.

We tested several different inf-sup stable pairs of B-spline discretization spaces, i.e.,

- $\mathcal{S}_0^2 \times \mathcal{S}_0^1$ with linear basis functions for pressure and quadratic for velocity, which is close to the standard FEM,

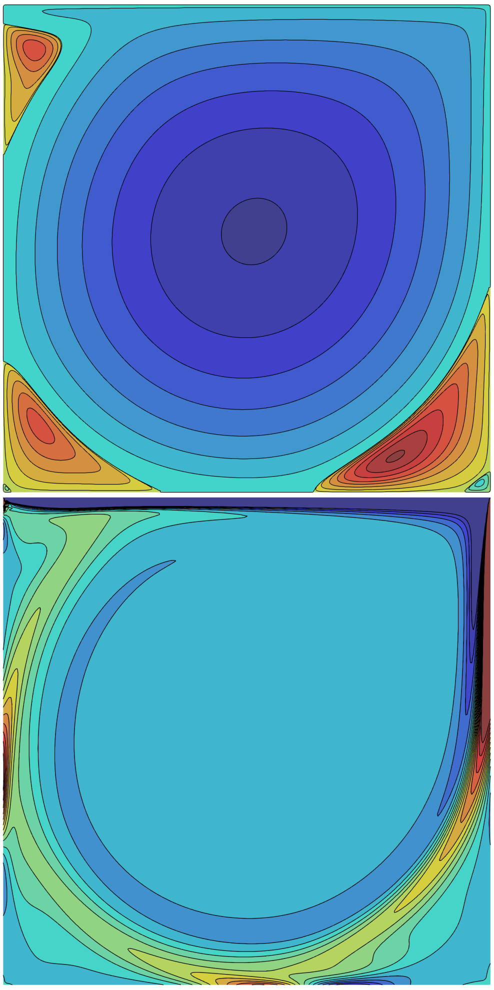


Figure 4. Stream function (top) and vorticity (bottom) profiles obtained for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ on the mesh composed of 128 × 128 elements. The contour ranges for stream function and vorticity are $\psi_{iso} \in [-0.12, 0.003]$ and $\omega_{iso} \in [-10, 20]$, respectively.

- $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ with cubic basis functions for pressure and quartic for velocity, with C^0 continuity across the elements, and
- $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ with cubic basis functions for pressure and quartic for velocity, with C^2 continuity across the elements.

For all these pairs of B-spline discretization spaces, we run the simulations on three different uniform B-spline meshes obtained in consecutive uniform mesh refinement levels, i.e., on B-spline meshes composed of 32 × 32, 64 × 64 and 128 × 128 B-spline elements. The corresponding numbers of degrees of freedom for all these simulations are demonstrated in Table 1.

Similarly, we tested different finite element discretization spaces in FEniCS, i.e.,

- P_2/P_1 with linear basis functions for pressure and quadratic for velocity, and
- P_4/P_3 with cubic basis functions for pressure and quartic for velocity.

	P_2/P_1	P_4/P_3
32 × 32	9539	42691
64 × 64	37507	169347
128 × 128	148739	674563
256 × 256	592387	2692611

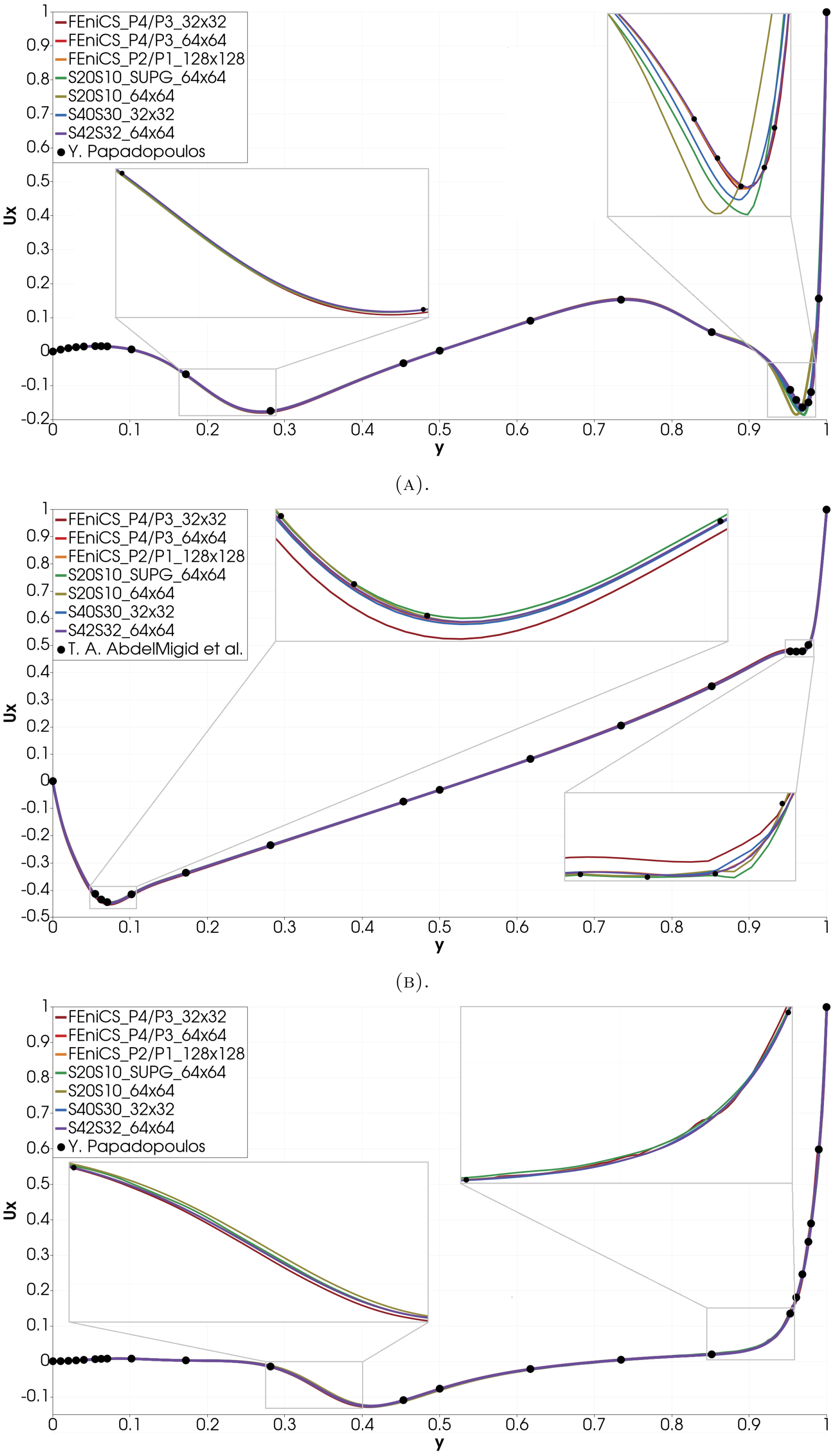
Table 2. Degrees of freedom (DOFs) for simulations on different meshes in FEniCS.

Let us emphasize that FEniCS is based on triangular meshes. The advantage of FEniCS is that it allows to choose higher degree finite element discretization spaces which can be compared with higher degree B-spline discretization spaces used in our isogeometric fluid flow solver. Further, for these pairs of finite elements discretization spaces, we run the simulations on four different meshes obtained in consecutive refinement levels. The triangular meshes are composed of 32 × 32, 64 × 64, 128 × 128 and 256 × 256 quads divided into two triangles by a diagonal. The corresponding numbers of degrees of freedom for all these simulations are demonstrated in Table 2.

In Figure 4, we present stream function and vorticity profiles which were obtained for B-spline discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (see [33] for details on stream function and vorticity computation). We do not present stream function and vorticity profiles for other cases in this study because the figures are very similar and presenting these figures would not add value. Let us only note that visually both profiles match the corresponding profiles available in the literature very well (see e.g. [54]).

Further, we present the comparison of u_x-velocity profiles along vertical cross-sections of the computational domain at $x = 0.1$, $x = 0.5$, $x = 0.95$ and u_y-velocity profiles along horizontal cross-sections of the computational domain at $y = 0.1$, $y = 0.5$, $y = 0.99$ obtained for selected discretization spaces in study in Figures 5 and 6. Let us emphasize that except the velocity profiles along cross-sections at $x = 0.5$, $y = 0.5$, which are quite commonly presented in the literature, we demonstrate the results also for other cross-sections. The reason is that we want to compare the solutions obtained for different discretization spaces more thoroughly in the whole domain and the cross-sections near the boundary of the computational domain intersect the problematic parts of the solution.

From Figures 5 and 6, we can conclude that most of the solutions obtained for selected discretization space pairs and selected meshes match the reference solution well, or even very well. The best solutions (the closest to the reference solution) are provided by discretization space pairs and meshes $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (64×64), P_4/P_3 (64 × 64) and P_2/P_1 (128 × 128). However, let us emphasize that the latter two cases have about four times more degrees of freedom (DOFs) than the former one which indicates the advantage of B-spline discretization spaces and isogeometric analysis in this

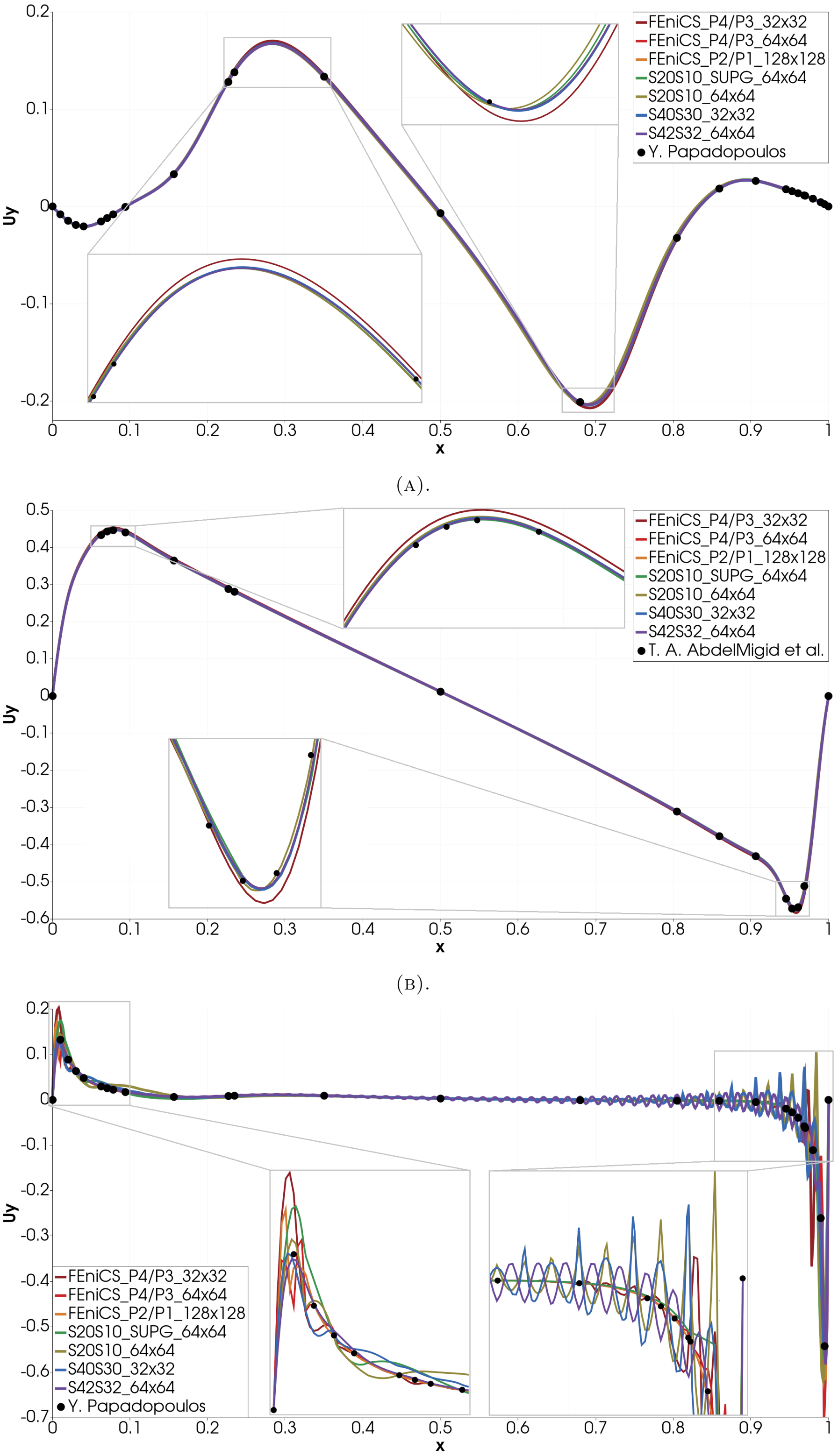


(a).

(b).

(c).

Figure 5. Comparison u_x-velocity profiles for vertical cross-sections of the computational domain at $x = 0.1$ (A), $x = 0.5$ (B), $x = 0.95$ (C) obtained for different discretization spaces, where S20S10 stands for $\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$, S40S30 for $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ and S42S32 for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$.



(A).

(B).

(C).

Figure 6. Comparison u_y-velocity profiles for horizontal cross-sections of the computational domain at $y = 0.1$ (A), $y = 0.5$ (B), $y = 0.99$ (C) obtained for different discretization spaces, where S20S10 stands for $\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$, S40S30 for $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ and S42S32 for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$.

	u_x at $x = 0.1$	u_x at $x = 0.5$	u_y at $y = 0.1$	u_y at $y = 0.5$
$\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$ (64×64)	$2.1 \cdot 10^{-2}$	$2.3 \cdot 10^{-4}$	$4.1 \cdot 10^{-5}$	$2.9 \cdot 10^{-4}$
$\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$ SUPG (64×64)	$6.8 \cdot 10^{-3}$	$3.9 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$6.5 \cdot 10^{-5}$
$\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ (32×32)	$2.4 \cdot 10^{-3}$	$3.4 \cdot 10^{-5}$	$6.9 \cdot 10^{-7}$	$1.2 \cdot 10^{-5}$
$\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (64×64)	$5.1 \cdot 10^{-6}$	$3.8 \cdot 10^{-5}$	$6.5 \cdot 10^{-8}$	$2.4 \cdot 10^{-6}$
P_2/P_1 (64×64)	$2.0 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$8.3 \cdot 10^{-4}$	$4.9 \cdot 10^{-4}$
P_2/P_1 (128×128)	$2.5 \cdot 10^{-5}$	$1.2 \cdot 10^{-6}$	$9.0 \cdot 10^{-8}$	$1.3 \cdot 10^{-6}$
P_4/P_3 (32×32)	$1.2 \cdot 10^{-4}$	$4.1 \cdot 10^{-4}$	$2.7 \cdot 10^{-5}$	$6.4 \cdot 10^{-4}$
P_4/P_3 (64×64)	$3.1 \cdot 10^{-6}$	$1.7 \cdot 10^{-6}$	$8.8 \cdot 10^{-8}$	$1.9 \cdot 10^{-6}$

Table 3. Residual sum of squares obtained for studied discretization space pairs and the corresponding reference solution at selected cross-sections through the computational domain.

studied case. To have a fair comparison of higher degree finite element discretization space pairs with B-spline discretization space pairs, $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (64×64) has to be compared with P_4/P_3 (32×32) because they have the similar number of DOFs (see Table 1 and 2). Looking at these two cases, P_4/P_3 (32×32) gives a significantly worse solution (more distant from the reference solution) than $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (64×64). Further, $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ (32×32), which is similar to the case P_4/P_3 (32×32) regarding the number of DOFs, degree of discretization spaces and continuity of the basis functions, behaves better than P_4/P_3 (32×32) but worse than $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ (64×64), which has also the similar number of DOFs. To quantify these conclusions, we also computed residual sum of squares (RSS) of the obtained solutions and the reference solutions. For selected cross-sections, the results of RSS are demonstrated in Table 3 and they confirm the above mentioned conclusions.

	Initial mesh of $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$	
Macro-elements size	64×64	128×128
128×128	–	149258
64×64	37770	151850
32×32	39082	157106
16×16	41778	167906
8×8	47458	190658
4×4	59970	240770
2×2	89602	359426
1×1	167298	670466

Table 4. Degrees of freedom (DOFs) for simulations in an in-house IgA-based flow solver with rIgA discretization space pairs arising from $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$.

The only exception of the discussion presented in the previous paragraph is the u_y-velocity profile along a horizontal cross-section at $y = 0.99$, see Figure 6(C). Near the upper right corner of the computational domain (right part of Figure 6(C)), the solutions obtained for B-spline discretization space pairs strongly oscillate. As the above mentioned cross-section is very close to the lid part of the boundary, these oscillations are caused by a sudden change in the boundary condition near the upper corner of the computational domain and also by the fluid flow striking the wall of the cavity. Because of that, oscillations can be expected in this part of the computational domain. One can see that oscillations of the solution for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ are smaller in amplitude than for $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$, but on a wider domain (which is caused by a wider support of B-spline basis functions of high continuity). Note that this behaviour can be also improved and the oscillations reduced by considering finer computational meshes (at least 128×128 for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$), by local refinement near the walls of the cavity or by stabilization of the numerical solution by SUPG method. It is also interesting that the FEM solutions obtained by FEniCS are less oscillative for comparable discretization spaces. On the other hand, near the upper left corner of the computational domain (left part of Figure 6(C)) the behaviour of the solutions significantly differs from the behaviour near the upper right corner. The solutions obtained by FEniCS oscillate. The solutions obtained for low continuity B-spline discretization space pairs ($\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$, $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$) also oscillate but less than FEniCS solutions. Even stabilization of the numerical solution by SUPG method does not help to completely eliminate the oscillations in the case of B-spline discretization space pair $\boldsymbol{\mathcal{S}}_0^2 \times \mathcal{S}_0^1$. The best solution in this part is the solution obtained for B-spline discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ which is without any oscillations and very close to the reference solution. Note that oscillations in the solution in this part of the computational domain are mainly caused by a sudden change in the boundary condition. The above mentioned observations indicate a small stabilization effect of B-spline discretization space pairs of higher continuity.

Further, motivated by the recent results on the so-called rIgA (refined Isogeometric Analysis, see [42, 43]) we performed several experiments whether this approach can speed up the time-consuming computations also in the case of the Navier-Stokes equations. The main idea of rIgA approach lies in reduction of the continuity of highly continuous isogeometric discretization spaces over specific inter-element boundaries (C^0 separators). The continuity reduction weakens the interconnection between the subdomains since it reduces the number of basis functions with support over the corresponding inter-element boundary, which improves the performance of the factorization using a multifrontal direct solver (see [62]).

In our experiments, we start with the high continuity isogeometric discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ and successively increase the number of C^0 separators (and concurrently decrease the macro-element size). Insertion of C^0 separator means inserting a multiple knot into the knot vector determining isogeometric discretization space which leads to the increase of the number of basis functions, and also degrees of freedom. Table 4 shows the numbers of degrees of freedom for rIgA discretization space pairs with decreasing macro-elements size arising from $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ considered on two different meshes 64 × 64 and 128 × 128. Note that macro-elements size 1 × 1 corresponds to discretization space pair $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ used above.

Figure 7 displays speed-up in solving the linear system (by a direct solver in PARDISO library) obtained from discretization of the steady Navier-Stokes equations for lid-driven cavity problem in one Picard's iteration. This speed-up is plotted with respect to the macro-elements size and the values on the y-axis correspond to the ratio of time obtained for standard high continuity isogeometric discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ and time for rIgA discretization for the particular macro-elements size. All times are averages of 100 Picard's iterations. For example, in the case of the mesh of 128 × 128 elements solving the linear system obtained for macro-elements size 4 × 4 is 3.76× faster than solving the linear system without any C^0 separators. Similarly, Figure 8 shows the reduction in the numbers of floating point operations needed for factorization during solving the linear system obtained by rIgA method for various macro-elements sizes by a direct solver (in PARDISO library) with respect to high continuity isogeometric discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$.

It can be seen that our results for the steady Navier-Stokes equations are in a good agreement with results presented in [43] for the Stokes equations and they confirm the positive effect of rIgA for solving linear systems also in the case of the steady Navier-Stokes equations. Speed-up indicated by the reduction of floating point operations is not fully reached, as one can see by comparing Figure 7 and Figure 8, but this is also in an agreement with the results presented in [43]. Further, Figures 7 and 8 indicate that one can expect higher speed-up for finer meshes. The reason for this lies in the fact that we are not in a fully asymptotic regime with our computational grids where theoretical speed-up in FLOPs reaches $\mathcal{O}(p^2)$ for the Stokes problem (see [43]). However, larger grid sizes are too time-consuming because we solve non-linear problem with many iterations required for convergence of the steady Navier-Stokes equations.

As an assembly of matrices is considered as one of the major drawbacks of IgA frequently studied in the literature (see e.g. [35–37]), one could ask if increasing the number of degrees of freedom by insertion of C^0 separators leads to more time-consuming assembly of matrices for the resulting linear system. If matrices are assembled element-wise, which is still frequently done in IgA-based solvers and which is also done in our case, and if an assembly of matrices only requires evaluation of B-spline basis functions and their derivatives for computation of local element matrices and assembling of local matrices to a global matrix (as in the case of mass matrix or stiffness matrix), the cost of an assembly of matrices in high continuity IgA and rIgA is identical. The reason lies in the fact that the number of basis functions per element is the same for high continuity IgA and rIgA. Moreover, during assembling of local matrices to the global matrix, the local matrices highly overlap in high continuity IgA and overlapping of the local matrices reduces in rIgA with decreasing macro-element size. This means that assembly time for rIGA with various macro-elements sizes is at most the same as for high continuity IgA.

Figure 9 shows speed-up in assembly of matrices for the linear system in one Picard's iteration of the steady Navier-Stokes equations discretized by rIgA method with various macro-elements sizes. It is seen that for small macro-elements sizes the time needed for assembly of matrices slightly increases. The reasons lies in the fact that we construct the velocity solution in each Picard's iteration which is used in the construction of the matrix $\mathbf{N}(\mathbf{u})$ (see (7)). As the construction of the velocity solution depends on the number of degrees of freedom, which rapidly grows for smaller macro-element sizes (see Table 4), this leads to a small increase in the computational cost of an assembly of matrices. However, this behaviour does not significantly affect the overall effectiveness of the rIgA method.

Finally, our experience with solving the Navier-Stokes and RANS (Reynolds-Averaged Navier-Stokes) equations by IgA have shown that decreased continuity to only C^0 inside the computational domain can lead to spurious oscillations along this C^0 separator, especially for high Reynolds number fluid flow simulations in complex computational domains. This can be considered as a disadvantage of rIgA method, specifically in its usage for fluid flow simulation.

7. Conclusions

In this paper, we have presented numerical results obtained by our in-house incompressible fluid flow solver based on isogeometric analysis and we have compared our results with the results obtained by a standard package based on finite element method FEniCS and with the selected reference solutions. For this comparison, we have chosen the standard benchmark problem for incompressible fluid flow simulation – lid-driven cavity flow.

Our experiments on uniform meshes have shown that even for coarse grids and low numbers of degrees

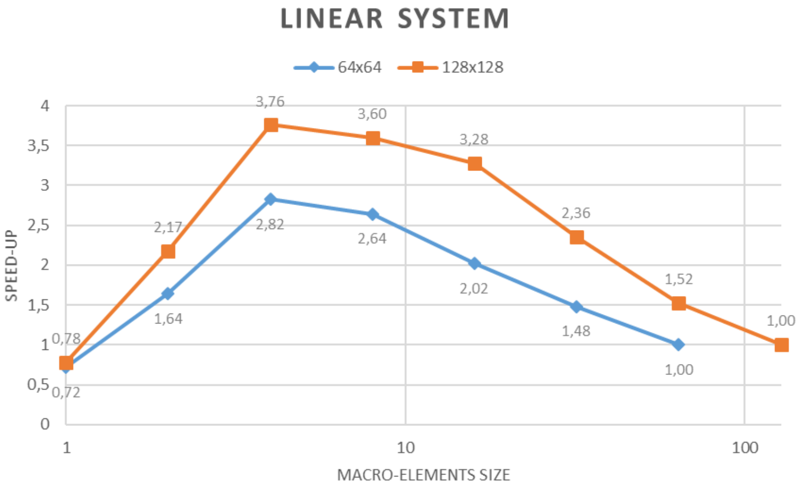


Figure 7. Time to solve the linear system arising from rIgA discretizations of the Navier-Stokes equation with a direct solver (PARDISO library) presented as speed-up gained by decreasing the macro-elements size.

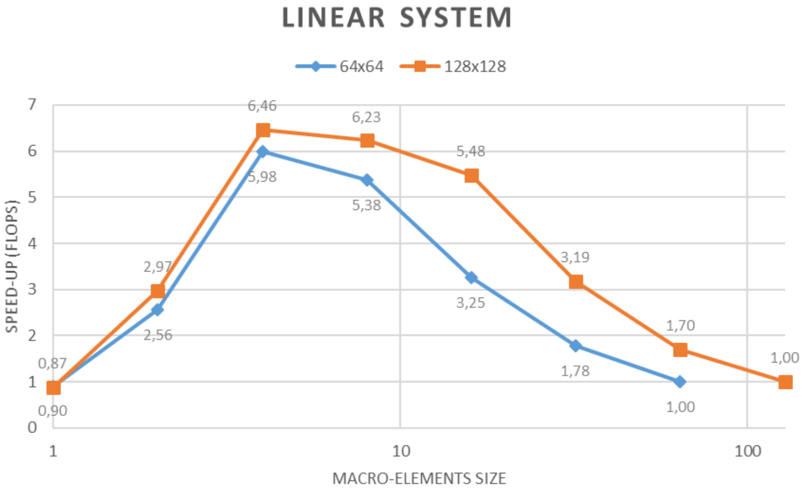


Figure 8. Floating point operations needed for factorization during solving the linear system arising from rIgA discretizations of the Navier-Stokes equation with a direct solver (PARDISO library) presented as speed-up (reduction of FLOPs) gained by decreasing the macro-elements size.

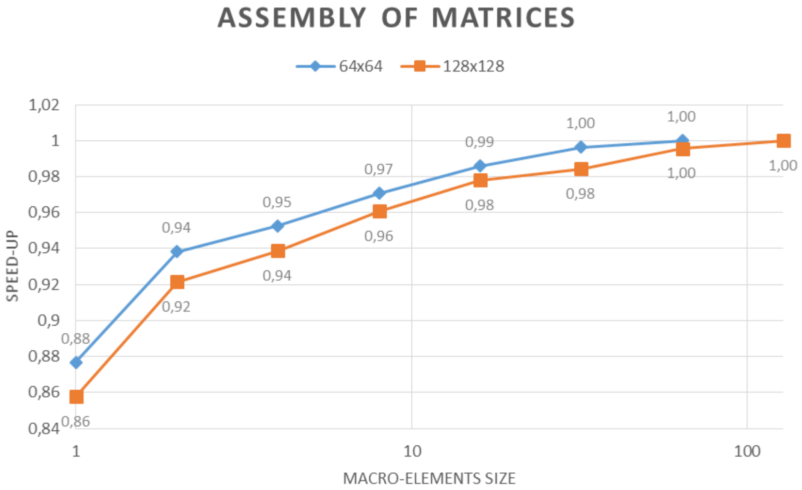


Figure 9. Time needed for assembly of matrices in one Picard's iteration in rIgA discretizations of the Navier-Stokes equation presented as speed-up gained by decreasing the macro-elements size.

of freedom, isogeometric analysis based on inf-sup stable B-spline discretization space pairs of higher degree and continuity can provide results which match reference solutions available in the literature very well. Further, comparing these results with results obtained by FEniCS (based on inf-sup stable finite element discretization space pairs), isogeometric analysis has produced the solutions with higher accuracy for meshes with the approximately same number of degrees of freedom. As the best case of the tested cases, we have identified B-spline discretization space pair $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ which produces an accurate solution even for a low number of degrees of freedom (about 40000 DOFs) and which corresponds to the solutions in FEniCS on the meshes with four times more degrees of freedom. Moreover, the solutions obtained for $\boldsymbol{\mathcal{S}}_2^4 \times \mathcal{S}_2^3$ are superior to the solutions obtained for $\boldsymbol{\mathcal{S}}_0^4 \times \mathcal{S}_0^3$ for the similar number of degrees of freedom, which confirms the superiority of B-spline discretization spaces of high continuity. Further, our experiments have indicated a small stabilization effect of high continuity B-spline discretization space pairs. However, this effect is much smaller than the effect of stabilization by SUPG method. Finally, we have shown numerically that the idea of rIgA method is also valid for the steady Navier-Stokes equations, i.e., inserting suitable C^0 separators helps to decrease the time needed for solving linear system using a direct solver.

Acknowledgements

This work was supported by the Czech Science Foundation (GA ČR) grant No. 19-04006S. We thank to all referees for their valuable comments which helped us to improve the paper.

References

[1] H. Versteeg, W. Malalasekera. *An Introduction to Computational Fluid Dynamics - Second Edition*. Prentice Hall, 2007.

[2] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2003. DOI:10.1137/1.9780898718003.

[3] H. C. Elman, D. Silvester, A. J. Wathen. *Finite Elements and Fast Iterative Solvers With Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2014. DOI:10.1093/acprof:oso/9780199678792.001.0001.

[4] T. Hughes, J. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* **194**(39–41):4135–4195, 2005. DOI:10.1016/j.cma.2004.10.008.

[5] J. Cottrell, T. Hughes, Y. Bazilevs. Isogeometric analysis: Toward integration of CAD and FEA. *John Wiley & Sons, Ltd* 2009.

[6] L. Piegl, W. Tiller. *The NURBS book*. Springer Verlag, 1997.

[7] A. Falini, C. Giannelli, T. Kanduč, et al. An adaptive IgA-BEM with hierarchical B-splines based on quasi-interpolation quadrature schemes. *International Journal for Numerical Methods in Engineering* **117**(10):1038–1058, 2018. DOI:10.1002/nme.5990.

[8] G. Farin, D. Hansford. Discrete Coons patches. *Computer Aided Geometric Design* **16**:691–700, 1999. DOI:10.1016/s0167-8396(99)00031-x.

[9] J. Gravesen, A. Evgrafov, D. Nguyen, P. Nørtoft. Planar parametrization in isogeometric analysis. In *International Conference on Mathematical Methods for Curves and Surfaces*, pp. 189–212. Springer, 2012. DOI:10.1007/978-3-642-54382-1_11.

[10] X. Nian, F. Chen. Planar domain parameterization for isogeometric analysis based on Teichmüller mapping. *Computer Methods in Applied Mechanics and Engineering* **311**:41–55, 2016. DOI:10.1016/j.cma.2016.07.035.

[11] A. Mantzaflaris, B. Jüttler, B. Khoromskij, U. Langer. Low rank tensor methods in Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* **316**:1062–1085, 2017. DOI:10.1016/j.cma.2016.11.013.

[12] M. Aigner, C. Heinrich, B. Jüttler, et al. Swept volume parameterization for isogeometric analysis. In *IMA International Conference on Mathematics of Surfaces*, pp. 19–44. Springer, 2009. DOI:10.1007/978-3-642-03596-8_2.

[13] P. Anděl, B. Bastl, K. Slabá. Parameterizations of generalized NURBS volumes of revolution. *Engineering Mechanics* **19**:293–306, 2012.

[14] T. Hughes, A. Reali, G. Sangalli. 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**:4104–4124, 2008. DOI:10.1016/j.cma.2008.04.006.

[15] A. Tagliabue, L. Dedè, A. Quarteroni. Isogeometric Analysis and error estimates for high order partial differential equations in fluid dynamics. *Computers & Fluids* **102**:277–303, 2014. DOI:10.1016/j.compfluid.2014.07.002.

[16] D. Benson, Y. Bazilevs, M. Hsu, T. Hughes. Isogeometric shell analysis: The Reissner-Mindlin shell. *Computer Methods in Applied Mechanics and Engineering* **199**(5):276 – 289, 2010. Computational Geometry and Analysis, DOI:10.1016/j.cma.2009.05.011.

[17] F. Auricchio, L. B. da Veiga, A. Buffa, et al. A fully “locking-free” isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer Methods in Applied Mechanics and Engineering* **197**(1):160 – 172, 2007. DOI:10.1016/j.cma.2007.07.005.

[18] A. Buffa, G. Sangalli, R. Vázquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering* **199**(17):1143 – 1152, 2010. DOI:10.1016/j.cma.2009.12.002.

[19] H. Gómez, V. M. Calo, Y. Bazilevs, T. J. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering* **197**(49):4333 – 4352, 2008. DOI:10.1016/j.cma.2008.05.003.

[20] X. Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering* **199**(29):2059 – 2071, 2010. doi:10.1016/j.cma.2010.03.005.

[21] N. D. Manh, A. Evgrafov, A. R. Gersborg, J. Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering* **200**(13):1343 – 1353, 2011. doi:10.1016/j.cma.2010.12.015.

[22] W. A. Wall, M. A. Frenzel, C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering* **197**(33):2976 – 2988, 2008. doi:10.1016/j.cma.2008.01.025.

[23] Y.-D. Seo, H.-J. Kim, S.-K. Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures* **47**(11):1618 – 1640, 2010. doi:10.1016/j.ijsolstr.2010.03.004.

[24] N. Liu, A. E. Jeffers. Isogeometric analysis of laminated composite and functionally graded sandwich plates based on a layerwise displacement theory. *Composite Structures* **176**:143 – 153, 2017. doi:10.1016/j.compstruct.2017.05.037.

[25] N. Liu, A. E. Jeffers. A geometrically exact isogeometric Kirchhoff plate: Feature-preserving automatic meshing and C^1 rational triangular Bézier spline discretizations. *International Journal for Numerical Methods in Engineering* **115**(3):395–409, 2018. `https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.5809` doi:10.1002/nme.5809.

[26] N. Liu, X. Ren, J. Lua. An isogeometric continuum shell element for modeling the nonlinear response of functionally graded material structures. *Composite Structures* **237**:111893, 2020. doi:10.1016/j.compstruct.2020.111893.

[27] N. Liu, A. E. Jeffers. Adaptive isogeometric analysis in structural frames using a layer-based discretization to model spread of plasticity. *Computers & Structures* **196**:1 – 11, 2018. doi:10.1016/j.compstruc.2017.10.016.

[28] N. Liu, P. A. Beata, A. E. Jeffers. A mixed isogeometric analysis and control volume approach for heat transfer analysis of nonuniformly heated plates. *Numerical Heat Transfer, Part B: Fundamentals* **75**(6):347–362, 2019. doi:10.1080/10407790.2019.1627801.

[29] N. Liu, A. E. Jeffers. Feature-preserving rational Bézier triangles for isogeometric analysis of higher-order gradient damage models. *Computer Methods in Applied Mechanics and Engineering* **357**:112585, 2019. doi:10.1016/j.cma.2019.112585.

[30] B. Bastl, M. Brandner, J. Egermaier, et al. IgA-based solver for turbulence modelling on multipatch geometries. *Advances in Engineering Software* **113**:7–18, 2017. doi:10.1016/j.advengsoft.2017.06.012.

[31] B. Bastl, M. Brandner, J. Egermaier, et al. Isogeometric analysis for turbulent flow. *Mathematics and Computers in Simulation* **145**:3–17, 2018. doi:10.1016/j.matcom.2016.05.010.

[32] Y. Bazilevs, V. Calo, J. Cottrell, et al. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer methods in applied mechanics and engineering* **197**(1-4):173–201, 2007. doi:10.1016/j.cma.2007.07.016.

[33] S. Hosseini, M. Möller, S. Turek. Isogeometric Analysis of the Navier-Stokes equations with Taylor-Hood B-spline elements. *Appl Math Comput* **267**:264 – 281, 2015. doi:10.1016/j.amc.2015.03.104.

[34] A. Falini, J. Špeh, B. Jüttler. Planar domain parameterization with THB-splines. *Computer Aided Geometric Design* **35–36**:95–108, 2015. doi:10.1016/j.cagd.2015.03.014.

[35] A. Mantzaflaris, B. Jüttler, B. N. Khoromskij, U. Langer. Matrix generation in isogeometric analysis by low rank tensor approximation. In *International Conference on Curves and Surfaces*, pp. 321–340. 2014. doi:10.1007/978-3-319-22804-4_24.

[36] M. Bartoň, V. M. Calo. Gauss-Galerkin quadrature rules for quadratic and cubic spline spaces and their application to isogeometric analysis. *Computer-Aided Design* **82**:57–67, 2017. doi:10.1016/j.cad.2016.07.003.

[37] R. R. Hiemstra, G. Sangalli, M. Tani, et al. Fast formation and assembly of finite element matrices with application to isogeometric linear elasticity. *Computer Methods in Applied Mechanics and Engineering* **355**:234–260, 2019. doi:10.1016/j.cma.2019.06.020.

[38] N. Collier, D. Pardo, L. Dalcin, et al. The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers. *Comput Methods Appl Mech Engrg* **213-216**:353–361, 2012. doi:10.1016/j.cma.2011.11.002.

[39] T. A. AbdelMigid, K. M. Saqr, M. A. Kotb, A. A. Aboelfarag. Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using gpu accelerated code. *Alexandria Engineering Journal* **56**:123–135, 2017. doi:10.1016/j.aej.2016.09.013.

[40] J. A. Evans, T. J. R. Hughes. Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations. *Math Models Methods Appl Sci* **23**(8):1421–1478, 2013. doi:10.21236/ada560496.

[41] V. John, L. Schumacher. A study of isogeometric analysis for scalar convection–diffusion equations. *Applied Mathematics Letters* **27**:43–48, 2014. doi:10.1016/j.aml.2013.08.004.

[42] D. Garcia, D. Pardo, L. Dalcin, et al. The value of continuity: Refined isogeometric analysis and fast direct solvers. *Computer Methods in Applied Mechanics and Engineering* **316**:586–605, 2017. doi:10.1016/j.cma.2016.08.017.

[43] D. Garcia, D. Pardo, V. M. Calo. Refined isogeometric analysis for fluid mechanics and electromagnetics. *Computer Methods in Applied Mechanics and Engineering* **356**:598–628, 2019. doi:10.1016/j.cma.2019.06.011.

[44] F. Brezzi, M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, 1991.

[45] A. Bressan, G. Sangalli. Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique. *IMA Journal of Numerical Analysis* **33**(2):629–651, 2012. DOI:10.1093/imanum/drr056.

[46] P. N. Nielsen, A. R. Gersborg, J. Gravesen, N. L. Pedersen. Discretizations in isogeometric analysis of Navier–Stokes flow. *Computer Methods in Applied Mechanics and Engineering* **200**(45):3242–3253, 2011. DOI:10.1016/j.cma.2011.06.007.

[47] V. John, P. Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I - A review. *Computer Methods in Applied Mechanics and Engineering* **196**(17):2197 – 2215, 2007. DOI:10.1016/j.cma.2006.11.013.

[48] A. N. Brooks, T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering* **32**(1-3):199–259, 1982. DOI:10.1016/0045-7825(82)90071-8.

[49] K. Takizawa, T. E. Tezduyar, Y. Otoguro. Stabilization and discontinuity-capturing parameters for space–time flow computations with finite element and isogeometric discretizations. *Computational Mechanics* **62**(5):1169–1186, 2018. DOI:10.1007/s00466-018-1557-x.

[50] Y. Chai, J. Ouyang. Appropriate stabilized Galerkin approaches for solving two-dimensional coupled Burgers' equations at high Reynolds numbers. *Computers & Mathematics with Applications* **79**(5):1287–1301, 2019. DOI:10.1016/j.camwa.2019.08.036.

[51] V. John, E. Schmeyer. Finite element methods for time-dependent convection–diffusion–reaction equations with small diffusion. *Computer methods in applied mechanics and engineering* **198**(3-4):475–494, 2008. DOI:10.1016/j.cma.2008.08.016.

[52] U. Ghia, K. Ghia, C. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* **48**(3):387–411, 1982. DOI:10.1016/0021-9991(82)90058-4.

[53] O. Botella, R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers & Fluids* **27**(4):421–433, 1998. DOI:10.1016/s0045-7930(98)00002-4.

[54] E. Erturk, T. C. Corke, C. Gökçöl. Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers. *International Journal for Numerical Methods in Fluids* **48**(7):747–774, 2005. DOI:10.1002/fld.953.

[55] Y. Papadopoulos. A driven cavity exploration. <https://www.acenumerics.com/the-cavity-sessions.html> [Online; accessed 30-Jan-2020].

[56] C.-H. Bruneau, M. Saad. The 2D lid-driven cavity problem revisited. *Computers & Fluids* **35**:326–348, 2006. DOI:10.1016/j.compfluid.2004.12.004.

[57] E. Erturk. Discussions on driven cavity flow. *International Journal for Numerical Methods in Fluids* **60**(3):275–294, 2009. DOI:10.1002/fld.1887.

[58] Z. Cai, Y. Wang. An error estimate for two-dimensional stokes driven cavity flow. *Mathematics of computation* **78**(266):771–787, 2009.

[59] P. Matuszyk, M. Paszyński. Fully automatic hp adaptive finite element method for the Stokes problem in two dimensions. *Computer Methods in Applied Mechanics and Engineering* **197**(51-52):4549–4558, 2008. DOI:10.1016/j.cma.2008.05.027.

[60] M. S. Alnaes, J. Blechta, J. Hake, et al. The FEniCS Project Version 1.5. *Archive of Numerical Software* **3**:9–23, 2015.

[61] B. Jüttler, U. Langer, A. Mantzaflaris, et al. Geometry + simulation modules: Implementing isogeometric analysis. In *PAMM – Proceedings of Applied Mathematics and Mechanics*, vol. 14, pp. 961–962. 2014. DOI:10.1002/pamm.201410461.

[62] I. S. Duff, J. K. Reid. The multifrontal solution of indefinite sparse symmetric linear. *ACM Trans Math Softw* **9**(3):302–325, 1983. DOI:10.1145/356044.356047.