Inf-sup stable non-conforming finite elements on tetrahedra with second and third order accuracy

Loïc Balazi, Grégoire Allaire, Pierre Jolivet, Pascal Omnes

Inf-sup stable non-conforming finite elements on tetrahedra with second and third order accuracy

Loïc Balazi[1,2], Grégoire Allaire[1], Pierre Jolivet[3], Pascal Omnes[2,4]

April 11, 2024

[1] CMAP, CNRS, École Polytechnique, Institut Polytechnique de Paris, 91120 Palaiseau, France.
[2] Université Paris-Saclay, CEA, Service de Génie Logiciel pour la Simulation, 91191, Gif-sur-Yvette, France.
[3] Sorbonne Université, CNRS, LIP6, 75005 Paris, France.
[4] Université Sorbonne Paris Nord, LAGA, CNRS UMR 7539, Institut Galilée, 99 Av. J.-B. Clément, 93430, Villetaneuse, France.

Abstract

We introduce a family of scalar non-conforming finite elements with second and third order accuracy with respect to the H^1-norm on tetrahedra. Their vector-valued versions generate, together with discontinuous pressure approximations of order one and two respectively, inf-sup stable finite element pairs with convergence order two and three for the Stokes problem in energy norm.

1 Introduction

In this paper, we consider the Stokes problem, which is a simple model for viscous incompressible flows and a first step towards more complex problems such as the Oseen problem or the Navier–Stokes problem. Let $\Omega \subset \mathbb{R}^3$ be a bounded connected domain. The steady-state Stokes problem is to find the velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^3$ and the pressure $p : \Omega \to \mathbb{R}$ solutions to:

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \nabla p &= \boldsymbol{f} \quad \text{in } \Omega, \\ \operatorname{div}\boldsymbol{u} &= 0 \quad \text{in } \Omega, \\ \boldsymbol{u} &= \boldsymbol{0} \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

with $\nu > 0$ the kinematic viscosity and $\boldsymbol{f}$ the load. Although, throughout this paper, we consider homogeneous boundary conditions, the generalization of the study for non-homogeneous Dirichlet boundary conditions is straightforward.

We denote by

$$(u, v) = \int_\Omega uv \mathrm{d}x$$

the scalar product in $L^2(\Omega)$. We use the same notation for vector-valued functions (which are denoted using boldface letters), i.e., the $L^2(\Omega)$ scalar product for vector-valued functions is given by:

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}x,$$

where $\boldsymbol{u} \cdot \boldsymbol{v}$ denotes the Euclidean scalar product in $\mathbb{R}^3$. Then, we define the classical velocity space $V = \left(H_0^1(\Omega)\right)^3$ and pressure space $M = L_0^2(\Omega) = \{q \in L^2(\Omega), \ s.t. \ \int_\Omega q = 0\}$ equipped

respectively with the $|\cdot|_1$ H^1 semi-norm and the $\|\cdot\|_0$ L^2 norm. We define the bilinear forms $a : V \times V \to \mathbb{R}$ and $b : M \times V \to \mathbb{R}$ by:

$$a(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{\nabla u}, \boldsymbol{\nabla v}) \qquad b(p, \boldsymbol{v}) = (p,\, \mathrm{div}\, \boldsymbol{v}).$$

Then, a weak formulation of the Stokes problem (1) reads as follows. Find $\boldsymbol{u} \in V$ and $p \in M$ such that:

$$\left\{ \begin{array}{rcll} a(\boldsymbol{u}, \boldsymbol{v}) + b(p, \boldsymbol{v}) & = & (\boldsymbol{f}, \boldsymbol{v}) & \forall \boldsymbol{v} \in V \\ b(\boldsymbol{u}, q) & = & 0 & \forall q \in M, \end{array} \right. \tag{2}$$

with $\boldsymbol{f} \in L^2(\Omega)^3$. It is well known [16] that system (2) has a unique solution due to the Ladyzhenskaya–Babuška–Brezzi (LBB) condition, also called inf-sup condition, namely there exists $\beta > 0$ such that:

$$\inf_{q \in M} \sup_{\boldsymbol{v} \in V} \frac{b(q, \boldsymbol{v})}{|\boldsymbol{v}|_1 \|q_h\|_0} \geq \beta.$$

Then, when the Stokes problem is discretized using finite elements, it is important that the discrete inf-sup condition is satisfied uniformly in the mesh size h. This condition imposes constraints on the choice of finite element pairs for velocity and pressure, as not every pair satisfies this requirement.

Although the present paper is only interested in the resolution of the Stokes problem by the finite element method, this work is motivated by the multiscale finite element method (MsFEM, see [11] for a general overview of this method). More precisely, in an MsFEM for the Stokes problem, one needs to solve so-called Stokes local problems with specific boundary conditions and source terms. For the class of MsFEM developed in [14], these local problems involve polynomial divergence and Lagrange multipliers, and therefore cannot be solved using classical pairs of finite elements. In two dimensions, the non-conforming finite elements introduced by [20] associated with discontinuous piecewise polynomial pressures of order n allow to solve these local problems. However, in three dimensions, no existing finite element pairs led to a discrete problem whose well-posedness could be proved, except the ones developed here. The combination of the new proposed finite elements and of the MsFEM for the Stokes system can be found in [5] and will be the subject of a separate publication.

Additionally, to our own interest, the design of inf-sup stable pairs for the Stokes equations, especially in three dimensions, is essential in many applications involving fluid mechanics (environmental flow, biological flow, in the energy sector, and others). A first class of such inf-sup stable methods is given by the conforming velocity / pressure pair $\mathbb{P}_{n+1} - \mathbb{P}_n$ (where $\mathbb{P}_{n+1}$ and $\mathbb{P}_n$ are respectively the spaces of polynomials of order less or equal to $n+1$ and n), called the Taylor–Hood method. Indeed, the classical Taylor–Hood method, obtained with $n = 1$ for tetrahedra in three dimensions, was first studied by R. Stenberg [26] who showed that it is sufficient that any element of the tetrahedral mesh has at least one vertex in the interior of Ω for this pair to be inf-sup stable for the Stokes problem. Then, D. Boffi [6] extended the Taylor–Hood method for higher-order polynomials for tetrahedra and proved that these pairs of finite elements satisfy the inf-sup condition for $n \geq 1$ under mesh restrictions less stringent than those in [26].

Another appealing approach to build inf-sup stable pairs for the Stokes problem is to consider non-conforming approximations. This class of approximations may have some advantages such as fewer geometric constraints on the mesh and in some cases fewer degrees of freedom and far fewer non-zero matrix entries than the corresponding conforming discretisations [19]. It also allows to use the inf-sup stable lowest order approximation ($\mathbb{P}_1$-$\mathbb{P}_0$) which is not possible with conforming elements. Thus, in [10], families of non-conforming finite elements are introduced: one of accuracy order one (the Crouzeix–Raviart finite element) on triangles and tetrahedra, and one of accuracy order three on triangles in two dimensions, which are inf-sup stable for the Stokes problem. Then, non-conforming finite elements have been developed for quadrilaterals

and hexahedra. Indeed, a constructive method for deriving finite elements of nodal type for rectangles and rectangle parallelepipeds was developed in [18]. More recently, [19] developed families of scalar non-conforming finite elements of arbitrary orders with optimal approximation properties on parallelograms and parallelepipeds. In two dimensions, [20] introduced a family of scalar non-conforming finite elements of accuracy order $n+1$ ($n \geq 0$) with respect to the H^1-norm on triangles, by enriching the local space of polynomials of order $n+1$ with a proper subspace of polynomials of order $n+2$. The authors of [20] showed that their vector-valued versions associated with discontinuous pressure approximations of polynomial order n form inf-sup stable finite element pairs of accuracy order $n+1$ for the Stokes problem. For $n=0$, their non-conforming finite element recovers the well-known Crouzeix–Raviart element. However, the generalization of these non-conforming finite elements to three dimensions on tetrahedra has not been studied. Based on the general definition of non-conforming finite elements in [10], the authors of [25] introduce a specific example of such a family on tetrahedra with $\mathbb{P}_2$ velocities. This finite element associated with a discontinuous pressure of polynomial order one, form an inf-sup stable finite element pair with quadratic accuracy for the Stokes problem.

More recently, two families of scalar non-conforming finite elements of accuracy order two and three with respect to the H^1-norm on tetrahedra were introduced in [9]. The authors showed that their vector-valued versions, associated with a discontinuous pressure approximation of polynomial order one and two, respectively, form inf-sup stable finite element pairs with accuracy order two and three for the Stokes problem.

In the present paper, inspired by the work of [20], we develop a new non-conforming family of finite elements of order two (quadratic velocity approximation error) and three (cubic velocity approximation error). The main difference with [9] is that we consider moments of order $n-1$ in the element while in [9] only moments of order zero in the element are considered. For the case $n=1$, the two methods are equivalent.

The content of this paper is the following. Section 2 recalls the discretisation setting for the Stokes problem. Section 3 is concerned with the definition of a family of scalar non-conforming finite elements, that are used to approximate the velocity space. Explicit bases of order two and three are provided. Section 4 is devoted to the study of the discrete inf-sup condition and the approximation properties of the finite element pair for the Stokes problem defined by the previous family of non-conforming finite elements for velocity and a piecewise discontinuous polynomial pressure of order n. To complete this study, we present an efficient preconditioning strategy for the resulting linear system in Section 5. In Section 6, we carry out numerical experiments allowing to assess the new non-conforming pairs of finite elements and we compare them with conforming pairs. Finally, Section 7 gives a comparison of the non-conforming and conforming pairs of finite elements in terms of degrees of freedom and computational times.

2 Discrete setting

Let $\Omega \subset \mathbb{R}^3$ be a polyhedral domain. We note $\{\mathcal{T}_h\}$ a family of triangulations of Ω parameterized by a positive parameter h which tends to 0. Each triangulation $\mathcal{T}_h$ consists of a finite number of tetrahedra K such that $\overline{\Omega} = \cup_{K\in\mathcal{T}_h}\overline{K}$. Let $h_K = \mathrm{diam}(K)$ and $h := \max\ h_k$. We assume that the triangulations are conformal in the sense that the intersection of the closures of two different cells K is either empty, a common vertex, a common edge, or a common face. Besides, we assume that the triangulations are shape regular, i.e., there exists a positive constant C independent of the mesh parameter h such that:

$$\frac{h_K}{\sigma_K} \leq C \qquad \forall K \in \mathcal{T}_h,\ h > 0,$$

where σ_K denotes the maximum diameter of a ball which can be inscribed in $\overline{K}$.

We denote by $\mathcal{F}_h$ the set of all faces F of $\mathcal{T}_h$. The set of inner faces will be denoted by $\mathcal{F}_h^i$ and the set of boundary faces by $\mathcal{F}_h^b$. The area of F and the volume of K are denoted respectively by $|F|$ and $|K|$. We note $\boldsymbol{n}_K$ the outer unit normal vector on ∂K. For any face F, we note $\boldsymbol{n}_F$ a fixed unit normal vector to F. If $F \in \mathcal{F}_h^b$, then $\boldsymbol{n}_F$ coincides with the outer normal vector to $\partial\Omega$.

For any integer n and any integer $1 \leq l \leq d$ (with d the space dimension, here $d = 3$), we denote by $\mathbb{P}_n^l$ the linear space spanned by l-variate polynomial functions of total degree at most n. The dimension of $\mathbb{P}_n^l$ is

$$N_n^l := \dim(\mathbb{P}_n^l) = \binom{n+l}{n}.$$

For any $K \in \mathcal{T}_h$, we denote by $\mathbb{P}_n^3(K)$ the restriction to K of polynomials in $\mathbb{P}_n^3$. For any $F \in \mathcal{F}_h$, we denote by $\mathbb{P}_n^2(F)$ the restriction to F of $\mathbb{P}_n^2$. For the sake of simplicity, we denote $\mathbb{P}_n^3(K)$ and $\mathbb{P}_n^2(F)$ respectively by $\mathbb{P}_n(K)$ and $\mathbb{P}_n(F)$.

We define the H^1 broken space by:

$$H^{1,h}(\Omega) := \{v \in L^2(\Omega),\ s.t.\ v|_K \in H^1(K),\ \forall K \in \mathcal{T}_h\}.$$

For $v \in H^{1,h}(\Omega)$, we define the jump across a face $F \in \mathcal{F}_h^i$ by:

$$[v]_F = (v|_K)|_F - (v|_{\tilde{K}})|_F,$$

where K and $\tilde{K}$ are the two cells which are adjacent to the face F oriented such that the unit normal vector $\boldsymbol{n}_F$ points into $\tilde{K}$. For a domain $D \subset \mathbb{R}^3$, we use the Sobolev spaces $H^m(D)$, $H_0^m(D)$, and $L^2(D) = H^0(D)$ for non-negative integer m. The norms and semi-norms in the scalar and vector-valued versions of $H^m(D)$ are denoted by $\|\cdot\|_{m,D}$ and $|\cdot|_{m,D}$ respectively. For any $\boldsymbol{v}$ in the broken space $\left(H^{1,h}(\Omega)\right)^3$, an analogue of its semi-norm $|\cdot|_1$ is defined, by:

$$|\boldsymbol{v}|_{1,h} := \left(\sum_{K \in \mathcal{T}_h} |\boldsymbol{v}|_{1,K}^2\right)^{1/2},$$

where $|\boldsymbol{v}|_{1,K} = \|\boldsymbol{\nabla v}\|_{0,K}$. Let $\boldsymbol{\nabla}_h \boldsymbol{v}$ and $\operatorname{div}_h \boldsymbol{v}$ denote the piecewise gradient and piecewise divergence of $\boldsymbol{v} \in H^{1,h}(\Omega)^3$, defined by:

$$(\boldsymbol{\nabla}_h\ \boldsymbol{v})|_K = \boldsymbol{\nabla}(\boldsymbol{v}|_K), \qquad (\operatorname{div}_h \boldsymbol{v})|_K = \operatorname{div}(\boldsymbol{v}|_K).$$

Let $V_h \subset L^2(\Omega)^3$ be a non-conforming finite element space and $M_h \subset M$. Let $\boldsymbol{f}$ be in $L^2(\Omega)^3$. Then, the discrete Stokes problem reads as follow. Find $\boldsymbol{u_h} \in V_h$ and $p_h \in M_h$ such that:

$$\left\{\begin{array}{rcll} a_h(\boldsymbol{u_h}, \boldsymbol{v_h}) + b_h(p_h, \boldsymbol{v_h}) & = & (\boldsymbol{f}, \boldsymbol{v_h}) & \forall \boldsymbol{v_h} \in V_h \\ b_h(q_h, \boldsymbol{u_h}) & = & 0 & \forall q_h \in M_h, \end{array}\right. \tag{3}$$

where $a_h(\cdot,\cdot) : V_h \times V_h \to \mathbb{R}$ and $b_h(\cdot,\cdot) : M_h \times V_h \to \mathbb{R}$ are respectively the discrete versions of the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$, defined by:

$$a_h(\boldsymbol{u_h}, \boldsymbol{v_h}) = (\boldsymbol{\nabla}_h \boldsymbol{u_h}, \boldsymbol{\nabla}_h \boldsymbol{v_h}) \qquad b_h(p_h, \boldsymbol{v_h}) = (p_h, \operatorname{div}_h \boldsymbol{v_h}). \tag{4}$$

It is well known that the discrete problem (4) is well posed under the following conditions: a_h and b_h are continuous, a_h is symmetric and coercive and the spaces V_h and M_h satisfy the discrete Ladyzhenskaya–Babuška–Brezzi (LBB) condition (or the discrete inf-sup condition, or are said to be inf-sup stable), namely there exists $\beta^\star > 0$, independent of h, such that

$$\inf_{q_h \in M_h} \sup_{\boldsymbol{v_h} \in V_h} \frac{b_h(q_h, \boldsymbol{v_h})}{|\boldsymbol{v_h}|_1 \|q_h\|_0} \geq \beta^\star.$$

3 A family of scalar non-conforming finite elements

In this section, we introduce a family of scalar non-conforming finite elements and a set of degrees of freedom.

3.1 Definition of the degrees of freedom

In the vein of [20], our aim is to define a scalar non-conforming finite element space of accuracy order $n+1$, $n \geq 0$ such that its vector-valued version $V_{h,n+1}$ is rich enough to satisfy the inf-sup condition when associated to the pressure approximation space $M_{h,n}$ made up of globally discontinuous piecewise $\mathbb{P}_n(K)$ pressures, with vanishing mean-value on Ω. This global pressure approximation space is also denoted $\mathbb{P}_n^{\mathrm{dc}}$ in what follows. The scalar (local) degrees of freedom are defined for any $v \in H^1(K)$ as:

$$N_j^{F_\alpha}(v) = \int_{F_\alpha} v L_j^{F_\alpha} \qquad j \geq 0,\ \alpha = 1,2,3,4 \tag{5}$$

$$N_j^{K}(v) = \int_{K} v M_j^{K} \qquad j \geq 0, \tag{6}$$

where $(L_j^{F_\alpha})_j$ define an arbitrary basis of $\mathbb{P}_n(F_\alpha)$ of dimension $(n+1)(n+2)/2$, and $(M_j^K)_j$ define an arbitrary basis of $\mathbb{P}_{n-1}(K)$ of dimension $n(n+1)(n+2)/6$. Thus, for each element $K \in \mathcal{T}_h$ and any integer $n \geq 0$, we define the set of degrees of freedom $\mathcal{N}_{n+1}(K)$ by:

$$\begin{aligned}\mathcal{N}_{n+1}(K) := &\left\{ N_j^{F_\alpha} \qquad 1 \leq j \leq \frac{(n+1)(n+2)}{2},\ \alpha = 1,2,3,4 \right\} \\ &\cup \left\{ N_j^{K} \qquad 1 \leq j \leq \frac{n(n+1)(n+2)}{6} \right\}.\end{aligned} \tag{7}$$

Remark 3.1. *In the $n = 0$ case, Eq. (5) reduces to the integral of v on each of the 4 faces of the tetrahedron while Eq. (6) completely disappears. Since for $\mathbb{P}_1$ functions, the integral over a face is simply the area of the face multiplied by the value at the barycenter of that face, the degrees of freedom in the $n = 0$ case can be assimilated to those of the standard $\mathbb{P}_1$ non-conforming (Crouzeix-Raviart) finite element space. Then, the vector version of this finite element space associated to the $\mathbb{P}_0$ pressure space is known to satisfy the discrete inf-sup property and to provide a first-order accurate approximation of the velocity in the H^1 norm and of the pressure.*

3.2 Definition of the finite element basis

Following [20], for $n \geq 1$, the idea is to enrich the local space $\mathbb{P}_{n+1}(K)$ with a proper subspace of $\mathbb{P}_{n+2}(K)$, denoted by $\Sigma_{n+2}(K) \subset \mathbb{P}_{n+2}(K)$, such that $\Sigma_{n+2}(K)$ has a trivial intersection with $\mathbb{P}_{n+1}(K)$. Thus, the enriched space $V_{n+1}(K)$ will be given by:

$$V_{n+1}(K) = \mathbb{P}_{n+1}(K) \oplus \Sigma_{n+2}(K).$$

The first requirement is to ensure that:

$$\dim(V_{n+1}) = \operatorname{card}(\mathcal{N}_{n+1}).$$

Given that:

$$\dim(\mathbb{P}_{n+1}) = \frac{(n+2)(n+3)(n+4)}{6},$$

we should have:

$$\begin{aligned}\dim(\Sigma_{n+2}) &= \dim(V_{n+1}) - \dim(\mathbb{P}_{n+1}) \\ &= \frac{n(n+1)(n+2)}{6} + 4\frac{(n+1)(n+2)}{2} - \frac{(n+2)(n+3)(n+4)}{6} \\ &= n(n+2),\end{aligned}$$

and we have to ensure that:

$$\mathbb{P}_{n+1}(K) \cap \Sigma_{n+2}(K) = \{0\}.$$

Consequently, to build the space V_{n+1}, i.e., to complete the space $\mathbb{P}_{n+1}(K)$, we are looking for a proper subspace $\Sigma_{n+2}(K)$ of $\mathbb{P}_{n+2}$ of dimension $n(n+2)$.

Remark 3.2. *By noting that:*

$$\dim(\mathbb{P}_{n+2} \setminus \mathbb{P}_{n+1}) = \frac{(n+3)(n+4)}{2},$$

we remark that, for $n > 5$,

$$\frac{(n+3)(n+4)}{2} - n(n+2) < 0.$$

Therefore, the proposed approach would work in theory at most for $n \leq 5$. However, in standard applications and for our purposes, considering accuracy order up to three is enough.

Now, we recall the notion of unisolvence. A finite element is defined by a set $(K, V(K), \mathcal{N}(K))$ where $V(K)$ is a space of functions of finite dimension M, and $\mathcal{N}(K)$ is a space of M independent linear forms $(\psi_i)_{i=1,\dots,M}$. The *unisolvence* property reads as follows.

Definition 3.3. *The set $(K,\ V(K),\ \mathcal{N}(K))$ is said to be unisolvent if and only if the application*

$$\begin{array}{rcl} V(K) & \to & \mathbb{R}^M \\ v & \mapsto & (\psi_1(v), \dots, \psi_M(v)) \end{array}$$

is an isomorphism.

We can state the following result:

Lemma 3.4. *Let K be a tetrahedron with barycentric coordinates $\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4$. Then, the finite element spaces:*

$$V_2 = \mathbb{P}_2 + \text{span}\{\lambda_1\lambda_2^2,\ \lambda_1\lambda_3^2,\ \lambda_2\lambda_3^2\},$$

and

$$V_3 = \mathbb{P}_3 + \text{span}\{\lambda_1^3\lambda_2,\ \lambda_2^3\lambda_3,\ \lambda_3^3\lambda_4,\ \lambda_4^3\lambda_1,\ \lambda_2^3\lambda_1,\ \lambda_1^3\lambda_4,\ \lambda_4^3\lambda_3,\ \lambda_3^3\lambda_2\},$$

are unisolvent with respect to the sets of degrees of freedom $\mathcal{N}_2$ and $\mathcal{N}_3$, respectively, described in Eq. (7).

Proof. We propose a numerical proof of the unisolvence of the finite element spaces V_2 and V_3 using Definition 3.3. On the reference tetrahedron and with the basis of $\mathbb{P}_{n+1}$ given by Lemma A.1 and the additional basis functions given in Lemma 3.4, we build $(\Phi_j)_j$ a basis of V_{n+1}. Then, we assemble the square matrix $\mathcal{M}_{i,j} = N_i(\Phi_j)$ for $i = 1, \dots, \text{card}(\mathcal{N}_{n+1})$ and $j = 1, \dots, \dim(V_{n+1})$, with $N_i \in \mathcal{N}_{n+1}$. It should be noted that the two resulting matrices are of small size (13×13 for V_2 and 28×28 for V_3). Since the matrices contain small coefficients, the value of their determinant can be non relevant (near to zero) to check for inversibility. Instead, we compute the conditioning of these matrices (in this paper, we use the NumPy linear algebra functions [1]). In the end, we find that the conditioning of matrix $\mathcal{M}$ has a value around 188 for the case $n = 1$ and around 2307 for the case $n = 2$, which are clearly different from infinite and allows to conclude on the inversibility of $\mathcal{M}$. It should be noted that a rigorous proof could be done using interval arithmetic, with software such as Xcas [21]. For the unisolvence of the finite element space V_2, a proof can be found in [9]. □

In Lemma 3.4, we have proposed a basis for V_2 and one for V_3. In Appendix A we explain how these bases have been found and how to find additional bases.

Remark 3.5. *For the case $n = 1$, we recover the finite element proposed by [9]. However, for the case $n = 2$, the proposed finite element space differs from that proposed in [9], since in their work the authors consider only moments of order zero in the element, while in our work, we consider moments of order $n-1$ in the element (i.e., moment of order one for the case $n = 2$).*

Remark 3.6. *The finite element spaces V_2 and V_3 described in Lemma 3.4 are not unique. We propose a numerical strategy in Appendix A to find relevant bases of functions. It has to be noted that none of these bases are symmetric, that is to say, the finite element space depends on the face numbering. However, we can suppose that in non-structured meshes, the non-symmetry of the global finite element space will have little influence.*

Summarizing, for any integer $n \geq 0$, the finite element we consider is given by the set $(K,\ V_{n+1}(K),\ \mathcal{N}_{n+1}(K))$ provided that $V_{n+1}(K)$ is unisolvent with respect to the set $\mathcal{N}_{n+1}(K)$ (which is known for $n = 0$ and which we have proved for $n = 1, 2$). Then, we define the global, vector-valued finite element space $V_{h,n+1}$ by:

$$V_{h,n+1} = \left\{ \begin{array}{c} \boldsymbol{v} \in \left(H^{1,h}(\Omega)\right)^3 \ :\ \boldsymbol{v}|_K \in \left(V_{n+1}(K)\right)^3,\ \forall K \in \mathcal{T}_h \\ \displaystyle\int_F q[\boldsymbol{v}]_F = 0,\ \forall F \in \mathcal{F}_h^i,\ q \in \mathbb{P}_n(F) \\ \displaystyle\int_F q\boldsymbol{v} = 0,\ \forall E \in \mathcal{F}_h^b,\ q \in \mathbb{P}_n(F) \end{array} \right\}. \tag{8}$$

4 Approximation properties and the discrete inf-sup condition

In this section, the order n is fixed. We show that the non-conforming finite element space $V_{h,n+1}$ (under Assumption 4.1) together with the $\mathbb{P}_n^{\mathrm{dc}}$ space for the pressure fulfil the discrete inf-sup condition and have an accuracy of order $n + 1$.

Assumption 4.1. *In this section, we assume that such a space $V_{n+1}(K)$, unisolvent with respect to the set of degrees of freedom $\mathcal{N}_{n+1}(K)$, exists for $n \leq 5$.*

In what follows, we introduce the local (scalar-valued) interpolation operators. We define the finite element interpolators as:

$$I_h^K : H^1(K) \to V_{n+1}(K) \qquad J_h^K : L^2(K) \to \mathbb{P}_n(K).$$

Using the set of degrees of freedom from $\mathcal{N}_{n+1}(K)$, we define $I_h^K v$ for $v \in H^1(K)$ by:

$$N_i(I_h^K v) = N_i(v) \qquad \forall N_i \in \mathcal{N}_{n+1}(K). \tag{9}$$

Then, $I_h^K v$ can be written as:

$$I_h^K v = \sum_{j=1}^{\mathrm{card}(\mathcal{N}_{n+1})} N_j(v)\varphi_j,$$

where $\{\varphi_j\}$ is the dual basis for $\mathcal{N}_{n+1}(K)$ i.e. $N_i(\varphi_j) = \delta_{ij}\ \forall (i,j)$. Next, we define $J_h^K p \in \mathbb{P}_n(K)$ for $p \in L^2(K)$ by the $L^2(K)$-projection in $\mathbb{P}_n(K)$:

$$(J_h^K p, q)_K = (p, q)_K \qquad \forall q \in \mathbb{P}_n(K).$$

It can be seen that the interpolation operator I_h^K preserves $\mathbb{P}_{n+1}(K)$. Indeed, for $p \in V_{n+1}$, p and $I_h^K p$ have the same degrees of freedom and then by unisolvence $I_h^K p = p$. In particular, this holds for $\mathbb{P}_{n+1}$ which is included in V_{n+1}. From its definition, J_h^K preserves $\mathbb{P}_n(K)$ evidently.

Moreover, it can be seen from the definitions of I_h^K and its vector-valued version $\boldsymbol{I}_h^K$ and the definition of $V_{h,n+1}$ in Eq. (8) on the one hand, and from the definition of J_h^K on the other hand, that these local interpolators generate global interpolation operators $\boldsymbol{I}_h : V \to V_{h,n+1}$ and $J_h : M \to M_{h,n} := \mathbb{P}_n^{\mathrm{dc}} \bigcap L_0^2(\Omega)$.

4.1 The discrete inf-sup condition

In this section, we show that the discrete version of the inf-sup (LBB) condition, that we recall below, is valid uniformly in h, for the finite element pair $V_{h,n+1} - M_{h,n}$, i.e.,

$$\exists \beta_n^\star > 0 \text{ such that } \forall h \qquad \inf_{q_h \in M_{h,n}} \sup_{\boldsymbol{v_h} \in V_{h,n+1}} \frac{b_h(q_h, \boldsymbol{v_h})}{|\boldsymbol{v_h}|_{1,h} \|q_h\|_0} \geq \beta_n^\star, \tag{10}$$

where the bilinear form b_h is defined in Eq. (4).

Theorem 4.2. *The finite element pair $V_{h,n+1} - M_{h,n}$ satisfies the discrete inf-sup stability condition* (10)*.*

In order to prove Theorem 4.2, we recall Fortin's lemma [15].

Lemma 4.3 (Fortin's lemma)**.** *If the continuous inf-sup condition holds with the constant β and if there exists a linear operator $\Pi_h : V \to V_{h,n+1}$ such that:*

$$\begin{aligned} b_h(q_h, \Pi_h \boldsymbol{u}) &= b(q_h, \boldsymbol{u}), \ \forall q_h \in M_{h,n}, \forall \boldsymbol{u} \in V \\ |\Pi_h \boldsymbol{u}|_{1,h} &\leq C |\boldsymbol{u}|_1, \forall \boldsymbol{u} \in V, \end{aligned}$$

then the discrete inf-sup condition holds with $\beta_n^\star \geq \beta \, |||\Pi_h|||^{-1}$, where $|||\cdot|||$ denotes the operator norm.

Below, we prove Theorem 4.2.

Proof. Let $\boldsymbol{I}_h$ be the interpolator defined above, then we remark that for any $q_h \in M_h$ and for any $\boldsymbol{v} \in V$, we have:

$$\begin{aligned} b_h(q_h, \boldsymbol{I}_h \boldsymbol{v}) &= (\operatorname{div}_h \boldsymbol{I}_h \boldsymbol{v}, q_h) \\ &= \sum_{K \in \mathcal{T}_h} (\operatorname{div}_h \boldsymbol{I}_h^K \boldsymbol{v}, q_h)_K \\ &= \sum_{K \in \mathcal{T}_h} \left(-(\boldsymbol{I}_h^K \boldsymbol{v}, \boldsymbol{\nabla} q_h)_K + (\boldsymbol{I}_h \boldsymbol{v} \cdot \boldsymbol{n}_K, q_h)_{\partial K} \right), \end{aligned}$$

by integration by parts. Now, since $\boldsymbol{\nabla} q_h|_K \in (\mathbb{P}_{n-1}(K))^3$ and from the definition of the degrees of freedom (6) and the interpolation properties (9) we get $(\boldsymbol{I}_h^K \boldsymbol{v}, \boldsymbol{\nabla} q_h)_K = (\boldsymbol{v}, \boldsymbol{\nabla} q_h)_K$. In the same way, since $q_h|_{F^\alpha} \in \mathbb{P}_n(K)$ and from the definition of the degrees of freedom (5) and the interpolation properties (9), we get that $(\boldsymbol{I}_h \boldsymbol{v} \cdot \boldsymbol{n}_K, q_h)_{\partial K} = (\boldsymbol{v} \cdot \boldsymbol{n}_K, q_h)_{\partial K}$. So that:

$$\begin{aligned} b_h(q_h, \boldsymbol{I}_h \boldsymbol{v}) &= \sum_{K \in \mathcal{T}_h} (-(\boldsymbol{v}, \boldsymbol{\nabla} q_h)_K + (\boldsymbol{v} \cdot \boldsymbol{n}_K, q_h)_{\partial K}) \\ &= \sum_{K \in \mathcal{T}_h} (\operatorname{div} \boldsymbol{v}, q_h)_K \\ &= (\operatorname{div} \boldsymbol{v}, q_h) = b(q_h, \boldsymbol{v}). \end{aligned}$$

Besides, from Lemma 4.4 (see Section 4.2) with $k = 0$, it follows that:

$$|\boldsymbol{I}_h \boldsymbol{u}|_{1,h} \leq C |\boldsymbol{u}|_1 \qquad \forall \boldsymbol{u} \in V,$$

with a constant C that does not depend on the mesh. Finally, we conclude thanks to Fortin's lemma, that the pair of finite elements $V_{h,n+1} - M_{h,n}$ satisfies the discrete inf-sup condition for the Stokes problem. □

4.2 Approximation property and error analysis

In this section, we derive an error estimate for the new non-conforming family of finite elements. We follow the same procedure as developed in [20]. First, we introduce Lemma 4.4.

Lemma 4.4. *The local interpolation operators* $I_h^K : H^1(K) \to V_{n+1}(K)$ *and* $J_h^K : L^2(K) \to \mathbb{P}_n(K)$ *satisfy for all* $K \in \mathcal{T}_h$*:*

$$|u - I_h^K u|_{1,K} \leq Ch^k |u|_{k+1,K} \qquad \forall u \in H^{k+1}(K),\ k = 0, \ldots, n+1 \tag{11}$$

$$\|p - J_h^K p\|_{0,K} \leq Ch^{n+1} |p|_{n+1,K} \qquad \forall p \in H^{n+1}(K). \tag{12}$$

with C *a constant that does not depend on the mesh.*

Proof. First, let K be an arbitrary element and $\widehat{K}$ the reference element and let us denote by:

$$\begin{array}{rccl} F: & \widehat{K} & \longrightarrow & K \\ & \widehat{x} & \longmapsto & x = T_K(\widehat{x}) = A_K \widehat{x} + b_K, \end{array}$$

the affine invertible mapping such that $K = F(\widehat{K})$. Using the standard scaling properties of the reference transformation and its inverse [13, chapter 11.1 - 11.2], it is sufficient to show the following estimates on the reference tetrahedron $\widehat{K}$:

$$|\widehat{u} - I_h^{\widehat{K}} \widehat{u}|_{1,\widehat{K}} \leq \widehat{C} |\hat{u}|_{k+1,\widehat{K}} \qquad \forall \widehat{u} \in H^{k+1}(\widehat{K}) \qquad k = 0, \ldots, n+1$$

$$\|\widehat{p} - J_h^{\widehat{K}} \widehat{p}\|_{0,\widehat{K}} \leq \widehat{C} |\widehat{p}|_{n+1,\widehat{K}} \qquad \forall \widehat{p} \in H^{n+1}(\widehat{K}),$$

where $I_h^{\widehat{K}}$ and $J_h^{\widehat{K}}$ denote the corresponding interpolation operators on the reference tetrahedron. In order to apply the Bramble–Hilbert Lemma [13, Lemma 11.9], we first need to show the continuity of the interpolation operators. From the continuity of the trace operator $\hat{u} \mapsto \hat{u}|_{\widehat{F}}$, we get for the degrees of freedom associated to the faces:

$$|N_j^{\widehat{F_\alpha}}(\widehat{u})| \leq \widehat{C} \|\widehat{u}\|_{0,\widehat{F_\alpha}} \leq \widehat{C} \|\widehat{u}\|_{1,\widehat{K}}.$$

The degrees of freedom associated to the element satisfy:

$$|N_j^{\widehat{K}}(\widehat{u})| \leq \widehat{C} \|\widehat{u}\|_{0,\widehat{K}} \leq \widehat{C} \|\widehat{u}\|_{1,\widehat{K}},$$

which leads to:

$$|\widehat{u} - I_h^{\widehat{K}} \widehat{u}|_{1,\widehat{K}} \leq C \|\widehat{u}\|_{1,\widehat{K}} \leq C \|\widehat{u}\|_{k+1,\widehat{K}}. \tag{13}$$

We conclude that the mapping $\widehat{u} \mapsto (\widehat{u} - I_h^{\widehat{K}} \widehat{u})$ is continuous on the spaces $H^{k+1}(\widehat{K})$ for $k = 0, \ldots, n+1$. Besides, this mapping vanishes on the subspace $\mathbb{P}_k$. Thus, from [13, Theorem 11.13] based on the Bramble–Hilbert Lemma, estimate (13) becomes:

$$|\widehat{u} - I_h^{\widehat{K}} \widehat{u}|_{1,\widehat{K}} \leq \widehat{C} |\widehat{u}|_{k+1,\widehat{K}}.$$

Then, using the standard scaling properties, we obtain (11). We prove estimate (12) using the same procedure. □

It is clear that the properties of the scalar local interpolation operator I_h^K carry over to its vector-valued version $\boldsymbol{I}_h^K$. Furthermore, for the generated global interpolation operators $\boldsymbol{I}_h : V \to V_{h,n+1}$ and $J_h : M \to M_{h,n}$, we have the following result.

Lemma 4.5. *The global interpolation operators $\boldsymbol{I}_h$ and J_h satisfy*

$$|\boldsymbol{u} - \boldsymbol{I}_h\boldsymbol{u}|_{1,h} \leq Ch^{n+1}|\boldsymbol{u}|_{n+2} \qquad \forall \boldsymbol{u} \in V \cap H^{n+2}(\Omega)$$

$$\|p - J_h p\|_0 \leq Ch^{n+1}|p|_{n+1} \qquad \forall p \in M \cap H^{n+1}(\Omega).$$

This leads to the error estimate for the solution of (3)

Theorem 4.6. *Let $(\boldsymbol{u}, p)$ be the solution to the Stokes system* (1) *and assume that:*

$$\boldsymbol{u} \in V \cap H^{n+2}(\Omega)^3, \ p \in H^{n+1}(\Omega).$$

Then, system (3) *has a unique solution $(\boldsymbol{u_h}, p_h)$ and the following error estimate holds:*

$$|\boldsymbol{u} - \boldsymbol{u_h}|_{1,h} + \|p - p_h\|_0 \leq Ch^{n+1}\left(|\boldsymbol{u}|_{n+2} + |p|_{n+1}\right), \tag{14}$$

with C a constant independent of the mesh size h.

Proof. Given that the approximation property (Lemma 4.5), the discrete inf-sup condition (10) and the patch test, which follows from the definition of V_h by (8), are satisfied, then we have the classical error estimate (14) that hold for non-conforming finite elements (see for example [10, Theorem 3] or [16, Theorem 2.6]). □

Remark 4.7. *To establish the result given by Theorem 4.6, we only use the $\mathbb{P}_{n+1}$ part of V_{n+1} and not the subspace Σ_{n+2}. One could imagine that taking into account Σ_{n+2} could improve the error bound of Theorem 4.6. However, as shown in the numerical results, the presence of the subspace Σ_{n+2} does not improve the order of convergence but only the accuracy level. This observation does not encourage us to investigate more about the consideration of the subspace Σ_{n+2} in the error estimate.*

We have shown, that the pair of finite elements $V_{h,n+1} - M_{h,n}$ is inf-sup stable for the Stokes equations and fulfils the approximation properties under Assumption 4.1. Thus, these properties are satisfied in particular for the non-conforming finite elements $V_{h,2} - M_{h,1}$ and $V_{h,3} - M_{h,2}$.

5 Efficient preconditioning strategy for the Stokes equations

Below, we present an efficient preconditioning strategy to solve the discrete Stokes problem in three dimensions. The solver is developed in FreeFEM [17] interfaced with PETSc [4]. The aim of this section is to show that standard preconditioners that are known to perform well for conforming finite elements may handle just as well the proposed non-conforming finite elements. This is an appealing feature, since it means there is no need to derive a new preconditioning strategy, which can be challenging for high-dimensional and ill-posed linear systems.

5.1 Stokes problem in matrix form

It is well-known that the discretisation of the Stokes problem (1) leads to a linear system featuring a two-by-two block matrix. Let m and n be two integers with $m < n$. Let A be a sparse $n \times n$ matrix, B be a sparse $n \times m$ full-rank matrix of constraints, and C an $m \times m$ matrix (in particular $C = 0$ is allowed). From this, a saddle point matrix is built as:

$$\mathcal{A} = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

We also define:

$$b = \begin{pmatrix} f \\ g \end{pmatrix},$$

where f is the source term and g is the value of the divergence (g is usually equal to 0 for the Stokes problem since we consider a divergence-free velocity). The objective is thus to efficiently solve the following linear system:

$$\mathcal{A}x = b, \tag{15}$$

where x is the vector of unknowns. In the case of the Stokes system, $x = \begin{pmatrix} u \\ p \end{pmatrix}$, with u the velocity unknowns and p the pressure unknowns. Below, we introduce a preconditioning strategy based on a decomposition of the problem by blocks with respect to the physics. This strategy is not new [12, 7], but our goal here is to make the paper self-contained. It is based on the Schur complement, which is an appealing approach for two-by-two blocks, as in the problem considered.

5.2 Fieldsplit preconditioning based on the Schur Complement

The Schur complement is defined as:

$$S = C - B^T A^{-1} B.$$

For the sake of simplicity, from now on we consider $C = 0$. In our implementation, we consider a lower Schur factorization, in which the action of the lower Schur factorization is defined by:

$$M_p = \begin{pmatrix} A & 0 \\ B^T & S \end{pmatrix}. \tag{16}$$

Remark 5.1. *Calculating explicitly M_p^{-1} leads to:*

$$M_p^{-1} = \begin{pmatrix} A^{-1} & 0 \\ -S^{-1}B^T A^{-1} & S^{-1} \end{pmatrix}.$$

It is then easy to show that M_p is a good left preconditioner for Eq. (15). However, applying this strategy requires to solve exactly systems of the form $Sz = w$, which may be very expensive. That is why we use M_p as a right preconditioner with not fully converged solutions of systems of the form $Sz = w$, taking advantage of the Flexible Generalized Minimal Residual method (FGMRES [24]) as the outer Krylov method which allows for inexact inner solvers at each outer iteration.

For applying this preconditioning strategy, one needs to solve systems of the form $M_p y = z$. These systems will themselves be preconditioned with approximate inverses of A and $\widehat{S}$ defined by $\widehat{S} = -B^T \mathrm{diag}(A)^{-1} B$ (where $\mathrm{diag}(A)$ stands for the square diagonal matrix made of the diagonal entries of A), since $\widehat{S}$ can be computed explicitly while S cannot because it is dense and not stored explicitly in practice.

In Section 5.3.1, we present a preconditioner to solve linear systems involving matrix A. In Section 5.3.2, we present a preconditioner to solve linear systems involving matrix S. These preconditioners are called respectively A_p and S_p.

5.3 Approximations of the physical blocks

5.3.1 Solution of linear systems involving matrix A

The preconditioner A_p is defined in a domain-decomposition approach using the restricted additive Schwarz method (RAS, see [8]) with an algebraic overlap:

$$A_p^{-1} = \sum_{j=1}^{N} \tilde{R}_j^T (R_j A R_j^T)^{-1} R_j,$$

where N is the number of blocks (or number of subdomains), which is by default the same as the number of MPI processes, R_j is the velocity restriction operator that returns a local vector from a global vector, and $\tilde{R}_j$ is the same operator, but scaled by 0 on the overlap, of the jth block. By default, this preconditioner has one layer of overlap.

Remark 5.2. *In general, larger overlaps lead to faster convergence up to a certain point where increasing the overlap does not further improve the convergence rate. Unfortunately, larger overlap implies greater communication and computation requirements.*

In each block, MUMPS [2] is used to compute the action of local inverses using an LU factorization.

Remark 5.3. *Boundary conditions are applied using a symmetric method, based on row and column eliminations, which allows to keep matrix A symmetric. Since it is also symmetric definite positive, MUMPS could be used to compute a Cholesky factorization instead. In theory, this choice would lead to better performance, however, our numerical experiments show that LU factorizations give better performance (less undesirable numerical pivoting).*

5.3.2 Solution of linear systems involving matrix S

The effectiveness of Schur complement-based factorization depends on the availability of a good preconditioner S_p for the Schur complement matrix. Determining a good preconditioner for the Schur complement is often a difficult task since we hardly know the structure of the Schur complement (we recall that it is not assembled explicitly). This is why we replace in the preconditioner formula S by its approximation $\widehat{S}$. The preconditioner S_p is defined in a domain-decomposition approach by using the block Jacobi method:

$$S_p^{-1} = \sum_{j=1}^{N} R_j^T (R_j \widehat{S} R_j^T)^{-1} R_j,$$

where N is still the number of blocks (or number of subdomains), which is by default the same as the number of MPI processes, and R_j (resp. R_j^T) is the pressure restriction (resp. pressure extension by 0) operator of the jth block. It has to be noted that the operator R_j is not the same as in Section 5.3.1 since here we are dealing with pressure unknowns and not velocity unknowns, and we do not consider overlap compared to the one layer of overlap in Section 5.3.1. This method does not require any communication as each block of the preconditioner acts only on unknowns internal to each MPI process. In each of the blocks, the solution is computed using the sparse direct solver MUMPS, which is internally computing an LU factorization.

5.4 Final structure of the solver

Now, we summarize the proposed method for solving iteratively Eq. (15) and detail associated PETSc instructions for the sake of reproducibility. We use the Flexible Generalized Minimal Residual algorithm (FGMRES, `-ksp_type fgmres`), which iteratively solves Eq. (15) using Eq. (16) as an outer right-preconditioner. The linear solves involving M_p require inner velocity and pressure solves, with matrices A and S respectively, which are themselves solved iteratively using the preconditioners A_p and S_p defined in Section 5.3.1 and Section 5.3.2, respectively. The inner Krylov solver for both physical blocks is GMRES, performing at most 5 iterations (`-fieldsplit_ksp_max_it 5`), which justifies the use of the flexible GMRES algorithm as an outer Krylov solver. Concerning some other solver settings:

- the relative tolerance of the outer Krylov method is set to 10^{-10} (`-ksp_rtol 1.0E-10`, ratio according to the norm of the initial unpreconditioned residual);

- for the setup of the velocity preconditioner, MUMPS parameter ICNTL(35) is set to 2 (`-fieldsplit_0_sub_mat_mumps_icntl_35 2`), which allows to activate its block low-rank (BLR) feature [3] during both the factorization and solution phases. This allows for memory gains by storing low-rank factors. Finally, the accuracy of the low-rank approximation is itself controlled by MUMPS parameter CNTL(7), which is set to 10^{-6} (`-fieldsplit_0_sub_mat_mumps_cntl_7 1.0E-6`).

6 Numerical experiments

The finite element spaces V_2 and V_3, presented in Lemma 3.4, have been implemented in the open source finite element software FreeFEM [17]. Therein, they are referenced respectively under the name $P2pnc3d$[1] and $P3pnc3d$[2]. In this section, we present the convergence orders obtained with these new finite elements for two different test cases, in order to verify the theoretical analysis. The obtained results are compared with those obtained with classical Taylor–Hood finite elements of order two and three. The accuracy of these methods both in term of mesh size h and number of unknowns is also compared. For all the following cases, the computational domain is the cube $\Omega = [0,1]^3$ and the viscosity ν is set to 1.

6.1 Case #1: Stokes with non-homogeneous Dirichlet boundary conditions

First, we consider the Stokes test case proposed in [22, Guide to Stokes]. We consider the Stokes problem (1) with non-homogeneous Dirichlet boundary conditions $\boldsymbol{g}$ on all the faces of the cube. The load function $\boldsymbol{f}$ and the velocity Dirichlet boundary conditions $\boldsymbol{g}$ are chosen such that the exact solution is:

$$\begin{cases} u_x &= 2\sin(\pi x) + \sin(\pi y) + \sin(\pi z), \\ u_y &= -\pi\cos(\pi x)y, \\ u_z &= -\pi\cos(\pi x)z, \\ p &= \sin(2\pi x) + \sin(2\pi y) + \sin(2\pi z). \end{cases}$$

[1] `https://github.com/FreeFem/FreeFem-sources/blob/4307d439ca8313cd8fda1c6ce34384e096efea4a/plugin/seq/Element_P2pnc_3d.cpp`

[2] `https://github.com/FreeFem/FreeFem-sources/blob/4307d439ca8313cd8fda1c6ce34384e096efea4a/plugin/seq/Element_P3pnc_3d.cpp`

6.1.1 Relative errors according to the element size h

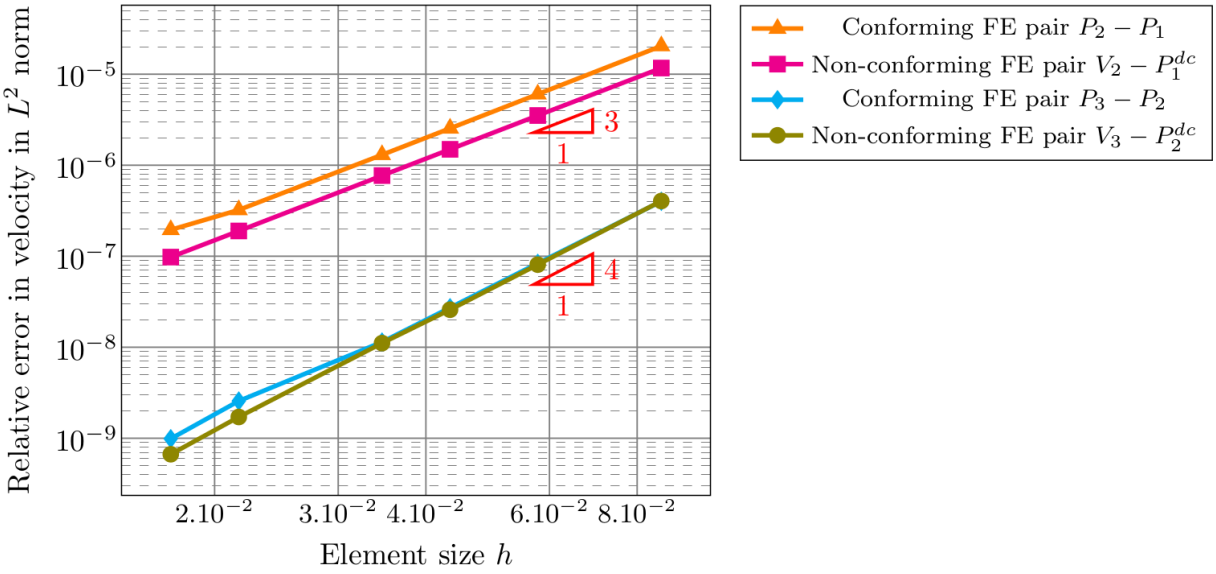


Figure 1: Relative error in L^2 norm between computed and exact velocities for case #1 according to the mesh size h.

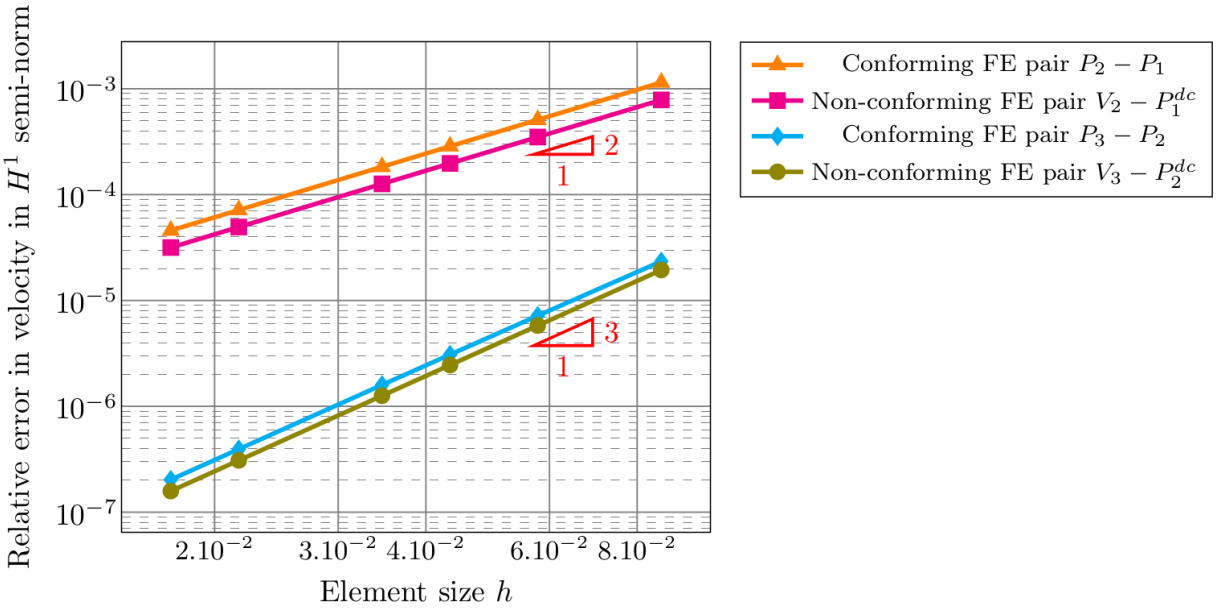


Figure 2: Relative error in H^1 semi-norm between computed and exact velocities for case #1 according to the mesh size h.

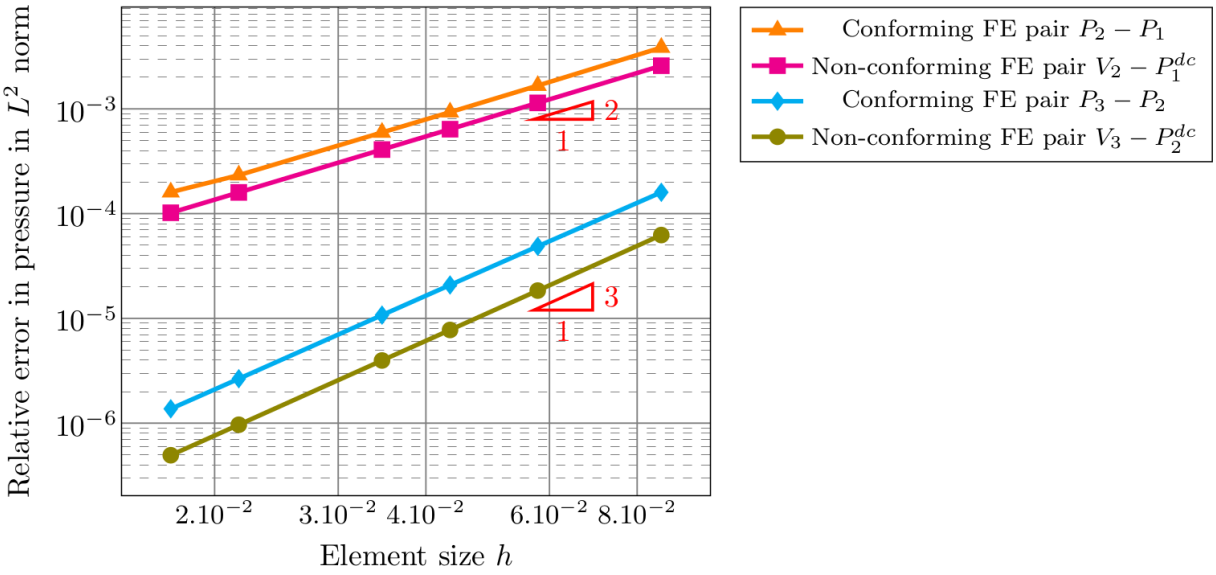


Figure 3: Relative error in L^2 norm between computed and exact pressures for case #1 according to the mesh size h.

Figures 1 to 3 display orders of convergence for different pairs of finite elements with respect to the mesh size h. It can be seen from these figures that the theoretical orders of convergence (see Theorem 4.6) are obtained. It should be noted that, for a given order n, the same orders of convergence are obtained for both the non-conforming and conforming pairs of finite elements. It seems that the addition of some basis functions to the space $\mathbb{P}_{n+1}$ does not improve the order of convergence. Similar behaviors can be seen for non-conforming finite elements of order n in two dimensions (see for example [20]). However, the non-conforming finite element pairs have a better level of accuracy compared to the conforming ones (i.e., the constant C in the error estimate is improved). This observation is more obvious for the relative pressure error in the L^2 norm.

6.1.2 Relative errors according to the number of unknowns

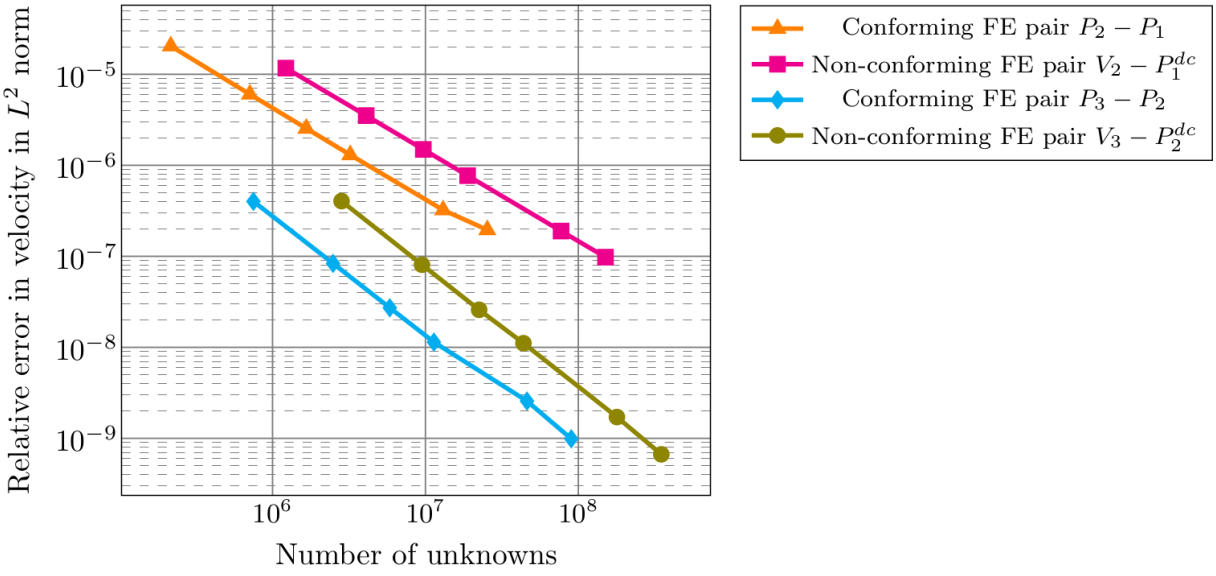


Figure 4: Relative error in L^2 norm between computed and exact velocities for case #1 according to the number of unknowns.

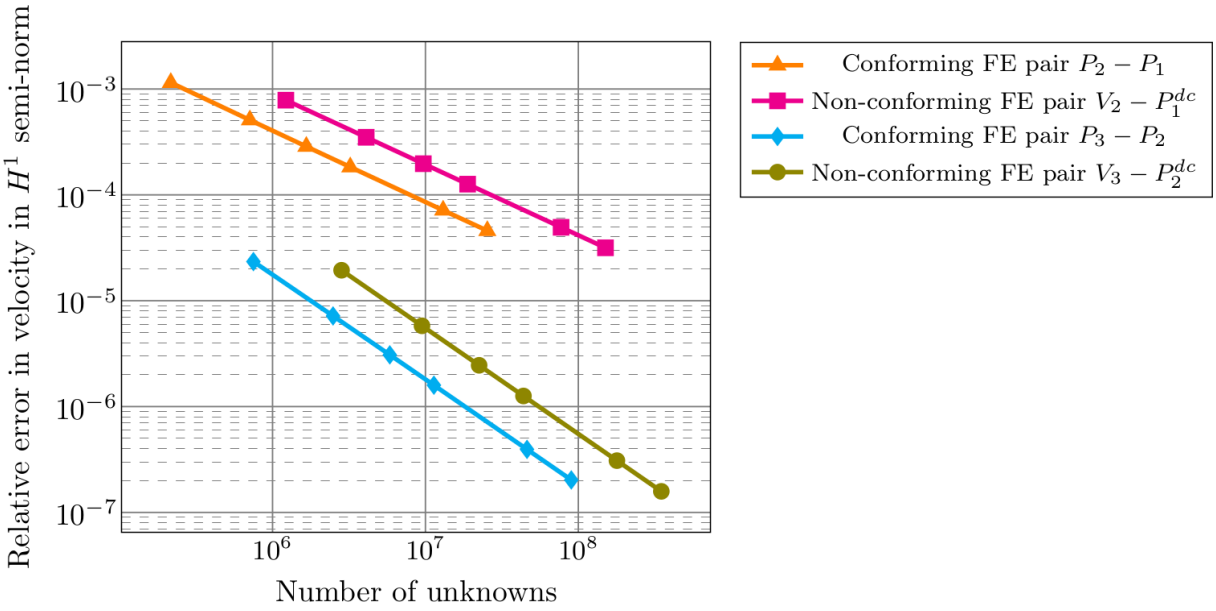


Figure 5: Relative error in H^1 semi-norm between computed and exact velocities for case #1 according to the number of unknowns.

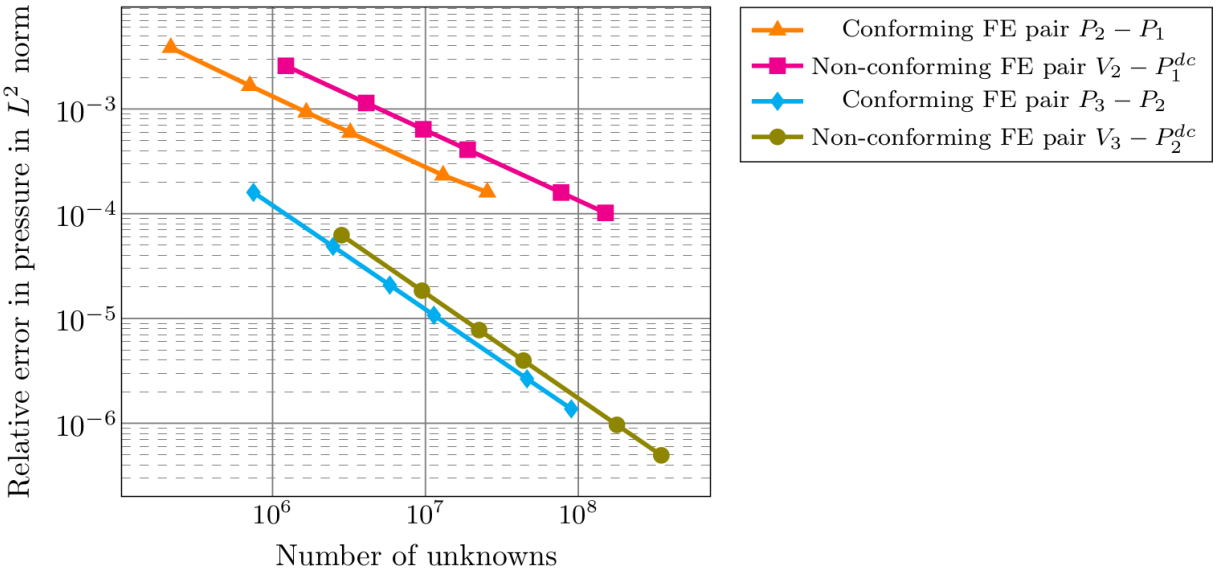


Figure 6: Relative error in L^2 norm between computed and exact pressures for case #1 according to the number of unknowns.

Figures 4 to 6 display the relative errors with respect to the number of degrees of freedom. These figures illustrate the well-known fact that, for regular solutions, it is better to choose higher order polynomials than to refine the mesh to decrease the errors since a given accuracy is achieved with fewer degrees of freedom.

6.2 Case #2: Stokes with partly Neumann boundary conditions

Second, we perform the test case proposed in [9]. The Stokes problem Eq. (1) is solved with a different boundary condition. More precisely, a homogeneous Neumann boundary condition is imposed on $\Gamma_N = \{z = 1\} \cap \partial\Omega$ and a homogeneous Dirichlet boundary condition is imposed on $\Gamma_D = \partial\Omega \setminus \Gamma_N$. The load function $\boldsymbol{f}$ is derived from the following exact solution:

$$\boldsymbol{u} = \operatorname{curl} \begin{pmatrix} y^2(1-y)^2x(1-x)z^2(1-z)^3 \\ x^2(1-x)^2y(1-y)z^2(1-z)^3 \\ 0 \end{pmatrix} \qquad p = (x - \frac{1}{2})(y - \frac{1}{2})(1-z).$$

One can indeed check that the respective boundary conditions are homogeneous on Γ_N and Γ_D.

6.2.1 Relative errors according to the element size h

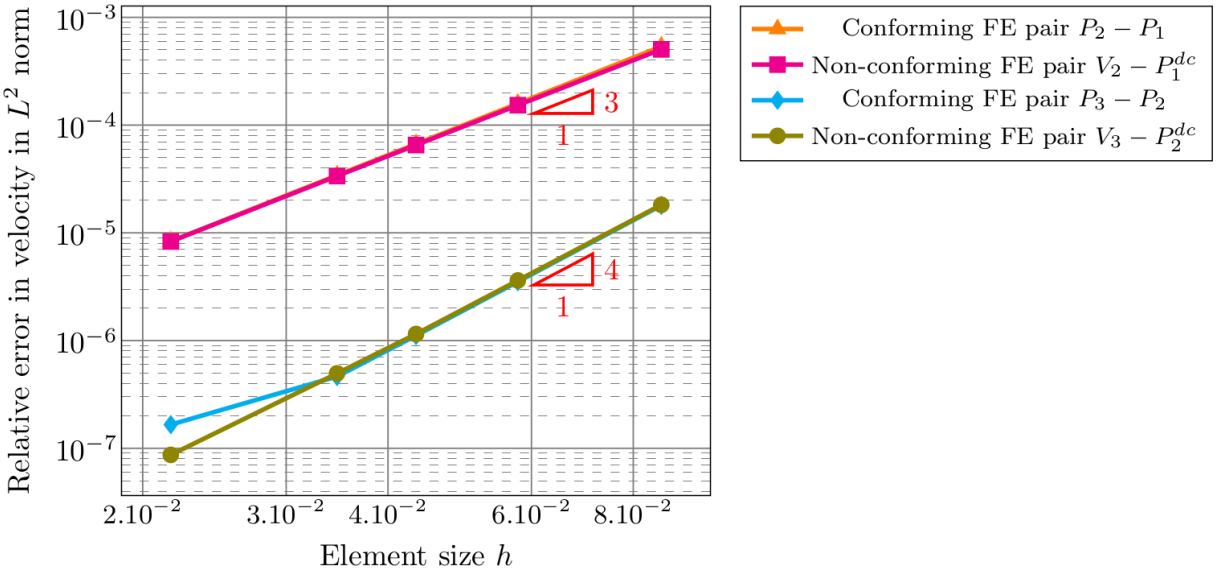


Figure 7: Relative error in L^2 norm between computed and exact velocities for case #2 according to the mesh size h.

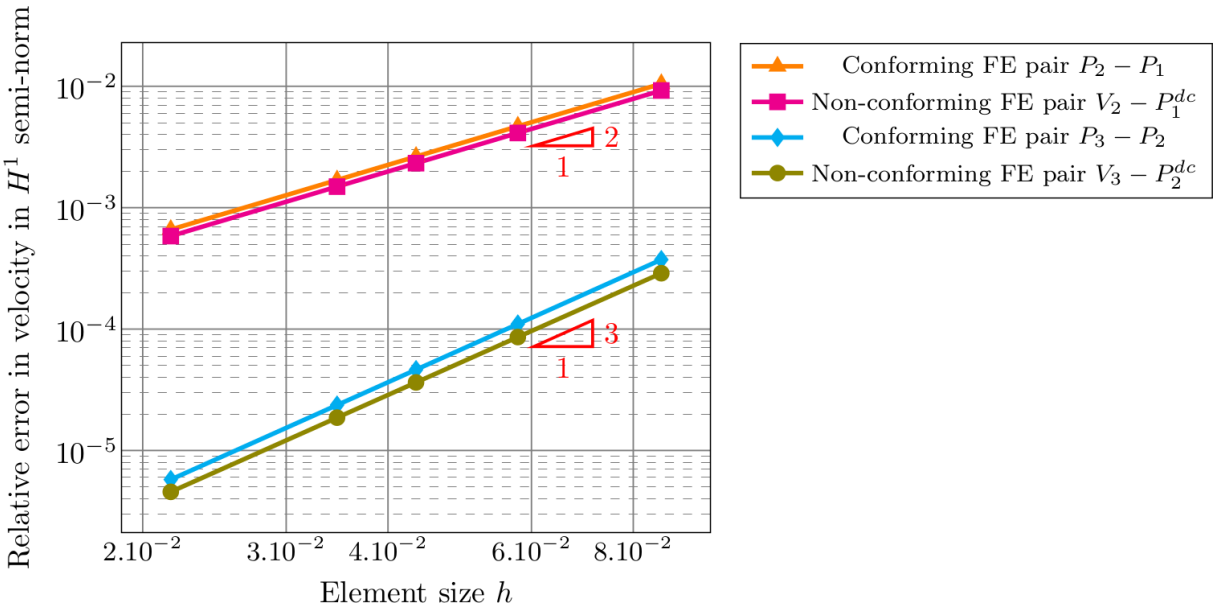


Figure 8: Relative error in H^1 semi-norm between computed and exact velocities for case #2 according to the mesh size h.

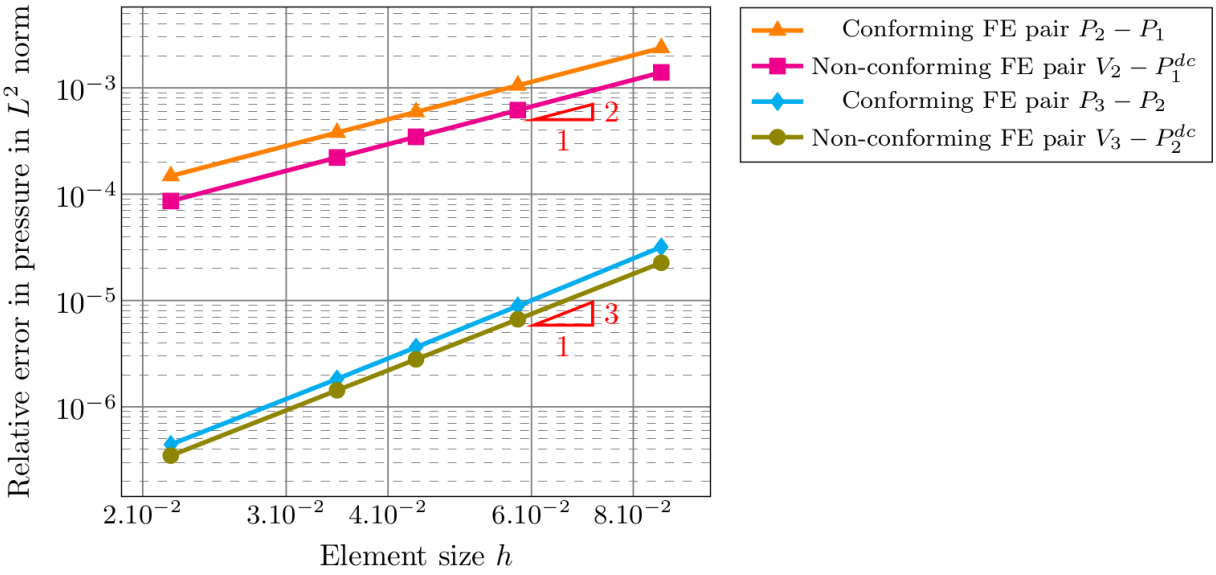


Figure 9: Relative error in L^2 norm between computed and exact pressures for case #2 according to the mesh size h.

6.2.2 Relative errors according to the number of unknowns

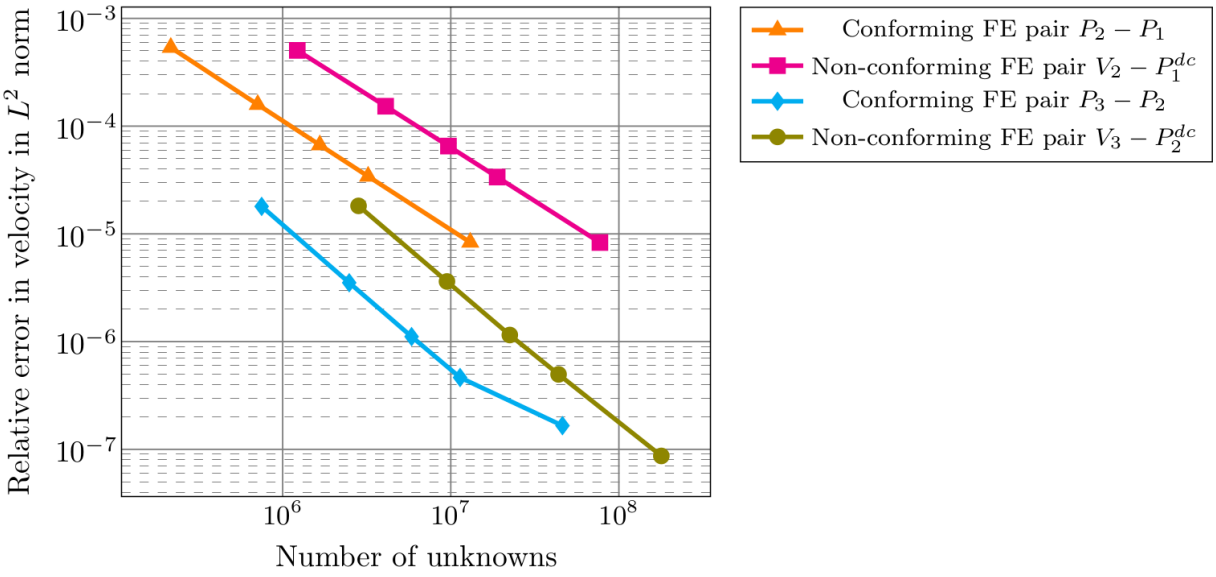


Figure 10: Relative error in L^2 norm between computed and exact velocities for case #2 according to the number of unknowns.

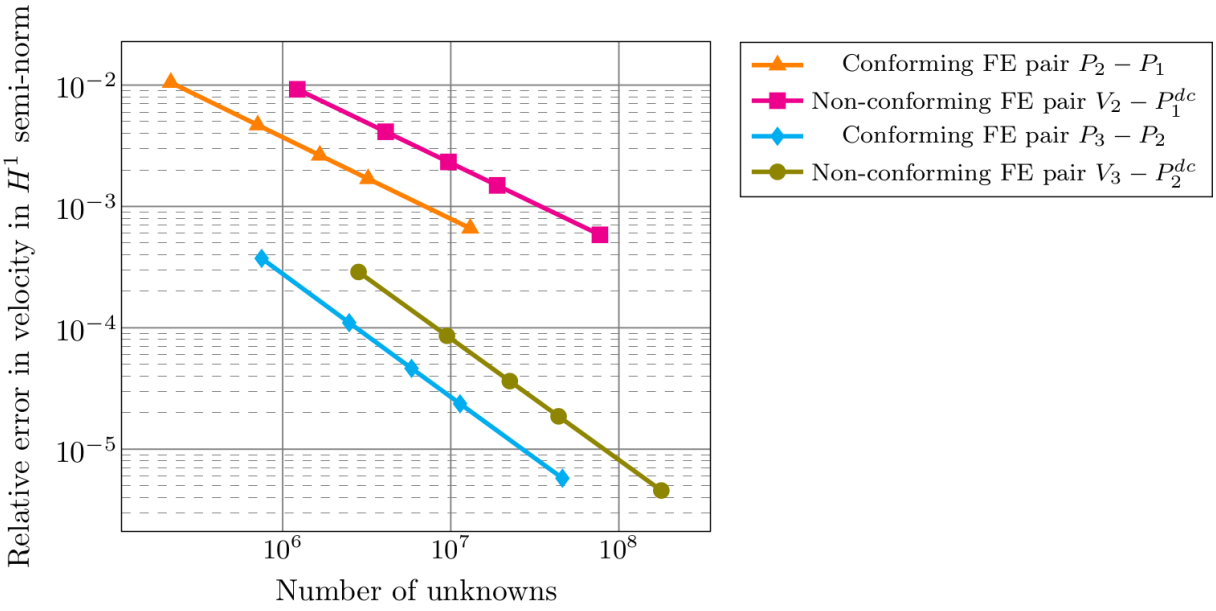


Figure 11: Relative error in H^1 semi-norm between computed and exact velocities for case #2 according to the number of unknowns.

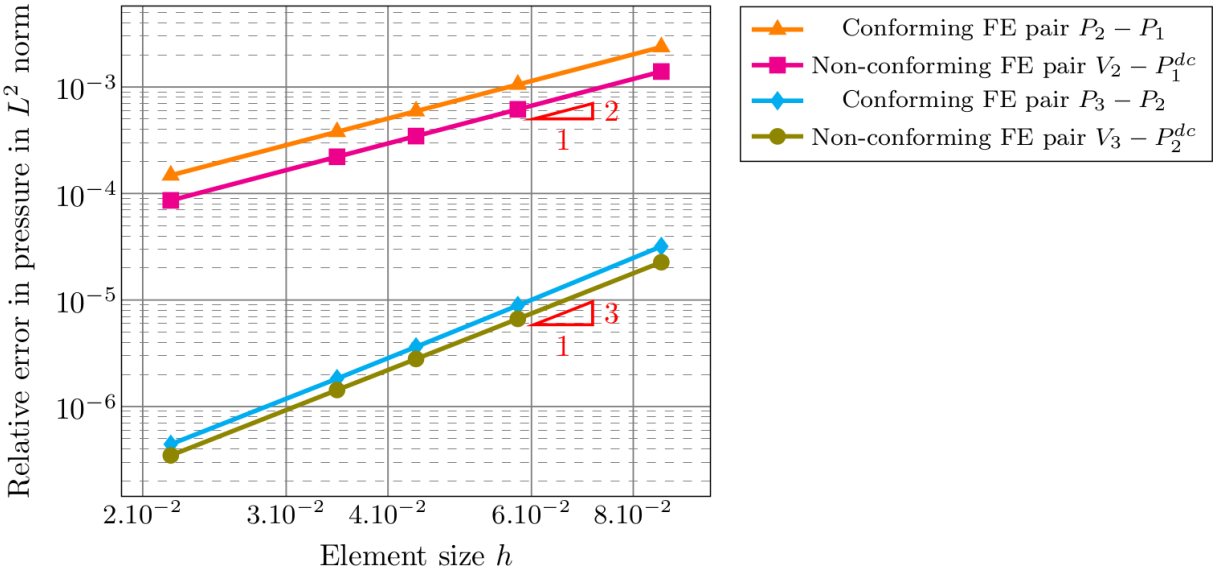


Figure 12: Relative error in L^2 norm between computed and exact pressures for case #2 according to the number of unknowns.

The cases presented above allow to assess our family of finite elements with Neumann boundary conditions. From Figs. 7 to 9 and from Figs. 10 to 12, similar observations can be done as in the case with Dirichlet boundary conditions (see Section 6.1).

7 Discussion about the number of degrees of freedom and the computational times

In this section, we compare the non-conforming pairs and the conforming ones both in terms of asymptotic number of degrees of freedom and in terms of computational times. We show

that the non-conforming discretisations have asymptotically more degrees of freedom than the conforming ones, leading to longer computational times. However, in return, the computational times per degree of freedom are lower for the non-conforming discretisations.

7.1 Discussion about the number of degrees of freedom

We consider a family of three-dimensional triangulations $(\mathcal{T}_{h_n})_{n\in\mathbb{N}}$, obtained from an initial mesh $\mathcal{T}_{h_0}$ in which a regular partition is recursively applied. The triangulation $\mathcal{T}_{h_n}$ is made of N_n nodes, E_n edges, F_n faces, and T_n tetrahedra. As noted in [23], the asymptotic behavior (as n goes to infinity) of the adjacency relations between the topological elements (nodes, edges, faces, and tetrahedra) in the mesh depends on the particular partition considered. First, we recall the distribution of the degrees of freedom of different finite elements according to the topological elements. Then, we make explicit the asymptotic number of degrees of freedom for different pairs of finite elements on four different partitions.

7.1.1 Distribution of the degrees of freedom

In Tables 1 to 3 the distribution of the degrees of freedom for different finite elements according to topological elements is recalled. This will be needed to calculate the number of the degrees of freedom for each conforming pair $\mathbb{P}_{n+1} - \mathbb{P}_n$ and of the proposed non-conforming pairs $V_2 - \mathbb{P}_1^{\mathrm{dc}}$ and $V_3 - \mathbb{P}_2^{\mathrm{dc}}$. In Table 2, for discontinuous finite elements, we consider that all the degrees of freedom are located in the tetrahedra since these degrees of freedom are not shared with neighboring elements.

	N_n	E_n	F_n	T_n
$\mathbb{P}_1$	1	0	0	0
$\mathbb{P}_2$	1	1	0	0
$\mathbb{P}_3$	1	2	0	4

Table 1: Number of unknowns of scalar conforming finite elements

	N_n	E_n	F_n	T_n
$\mathbb{P}_1^{\mathrm{dc}}$	0	0	0	4
$\mathbb{P}_2^{\mathrm{dc}}$	0	0	0	10

Table 2: Number of unknowns of scalar fully discontinuous finite elements

	N_n	E_n	F_n	T_n
V_2	0	0	3	1
V_3	0	0	6	4

Table 3: Number of unknowns of scalar non-conforming finite elements

7.1.2 Computation of the number of degrees of freedom

Case of non-conforming pairs As noted in [23], the ratio T_n/F_n does not depend on the choice of a given partition. Indeed, for any partition $T_n/F_n = \frac{1}{2}$ asymptotically. Given that for the non-conforming pairs, the degrees of freedom are located on faces and in tetrahedra, the number of unknowns of these pairs does not depend on the partition considered. Thus, an easy computation, using data of Tables 2 and 3 (considering three components for the velocity field and one for the pressure field) leads to:

Finite element pairs	Asymptotic number of unknowns
$V_2 - \mathbb{P}_1^{\text{dc}}$	$25T_n$
$V_3 - \mathbb{P}_2^{\text{dc}}$	$58T_n$

Table 4: Number of unknowns of non-conforming pairs

Case of conforming pairs Contrary to non-conforming pairs, for conforming pairs some unknowns are located on edges and vertices. As noted in [23], the relations between the number of edges (as well as the number of vertices) and the number of tetrahedra depend on the partition considered. Below, for the sake of completeness we give the definitions of common $3D$ partitions as presented in [23] (for more details the reader can refer to [23]).

Definition 7.1 ($3D$-Bey partition)**.** *For any tetrahedron K, the $3D$-Bey partition is defined by dividing K into eight sub-tetrahedra by cutting off the four corners by the midpoints of the edges, and the remaining octahedron is subdivided further into four tetrahedra by one of the three possible interior diagonals.*

Definition 7.2 ($8T-LE$ partition)**.** *For any tetrahedron K, the $8T-LE$ partition is defined by dividing K into eight sub-tetrahedra by performing the $4T-LE$ partition of the faces, and then subdividing the interior of the tetrahedron in a manner consistent with the performed division in the 2-skeleton.*

Definition 7.3 ($3D$ barycentric partition)**.** *For any tetrahedron K, the barycentric partition is defined by inserting a new node P at the barycenter of K, putting new nodes at the barycenters of the faces of K, and putting new nodes at the midpoints of the edges of K. Then, on each face of K a barycentric triangular partition is performed. Finally, the node P is joined with all the vertices of K, and with all the new nodes introduced.*

Definition 7.4 ($4T$ partition)**.** *For any tetrahedron K, the $4T$ barycentric partition is defined by inserting a new node P at some interior point of K (for example at the barycenter of K) and joining P with the all the vertices of K.*

Then, for each of these partitions, the following relations hold asymptotically:

	$3D$-Bey	$8T-LE$	$3D$ barycentric	$4T$
N_n/T_n	1/6	1/6	2/11	1/3
E_n/T_n	7/6	7/6	13/11	4/3
F_n/T_n	2	2	2	2

Table 5: Relations between topological elements for different partitions

Using the relations in Table 5 with the data of Table 1 and considering three components for the velocity field and one for the pressure field, we get the asymptotic number of degrees of freedom for different pairs of finite elements given in Table 6.

Finite element pairs	$3D$-Bey	$8T-LE$	$3D$ barycentric	$4T$
$\mathbb{P}_2 - \mathbb{P}_1$	$4.16T_n$	$4.16T_n$	$4.27T_n$	$5.33T_n$
$\mathbb{P}_3 - \mathbb{P}_2$	$20.83T_n$	$20.83T_n$	$21T_n$	$22.66T_n$

Table 6: Number of unknowns of different conforming pairs according to the partition considered

Thus, for the non-conforming $V_2 - \mathbb{P}_1^{\text{dc}}$ pair, we asymptotically have $25T_n$ degrees of freedom. In comparison, for the conforming $\mathbb{P}_2 - \mathbb{P}_1$ pair, we have asymptotically around $5T_n$ degrees of freedom (approximately four times less than the non-conforming case). For the non-conforming

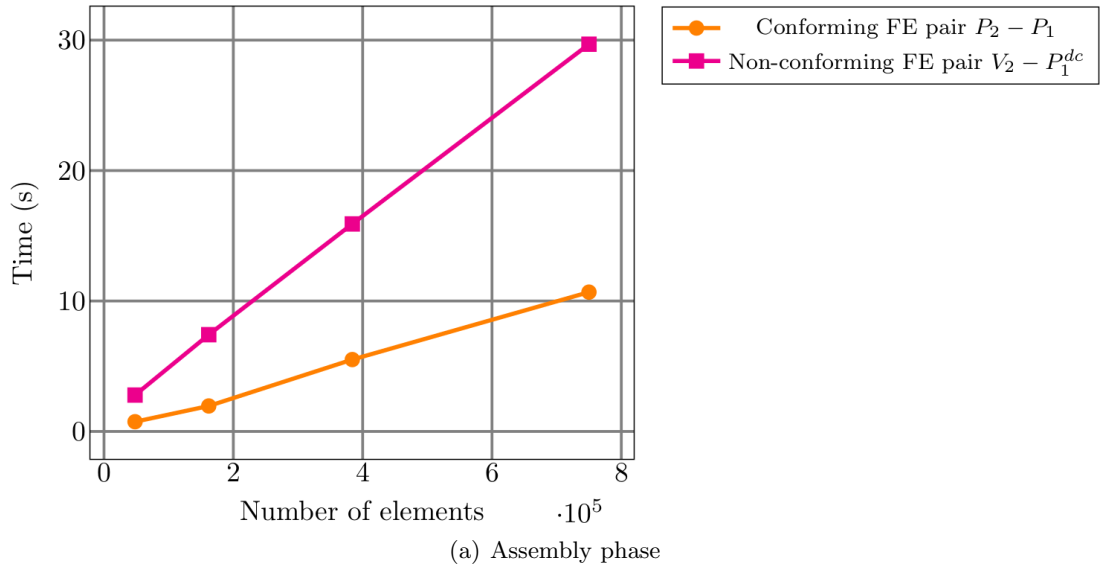
$V_3-\mathbb{P}_2^{\mathrm{dc}}$ pair, we asymptotically have $58T_n$ degrees of freedom. In comparison, for the conforming $\mathbb{P}_3-\mathbb{P}_2$ pair, we have asymptotically around $22T_n$ degrees of freedom (approximately 2.2 times less than the non-conforming case). In view of these results, it is clear that using conforming methods for high-order is more advantageous.

7.2 Discussion about the computational times

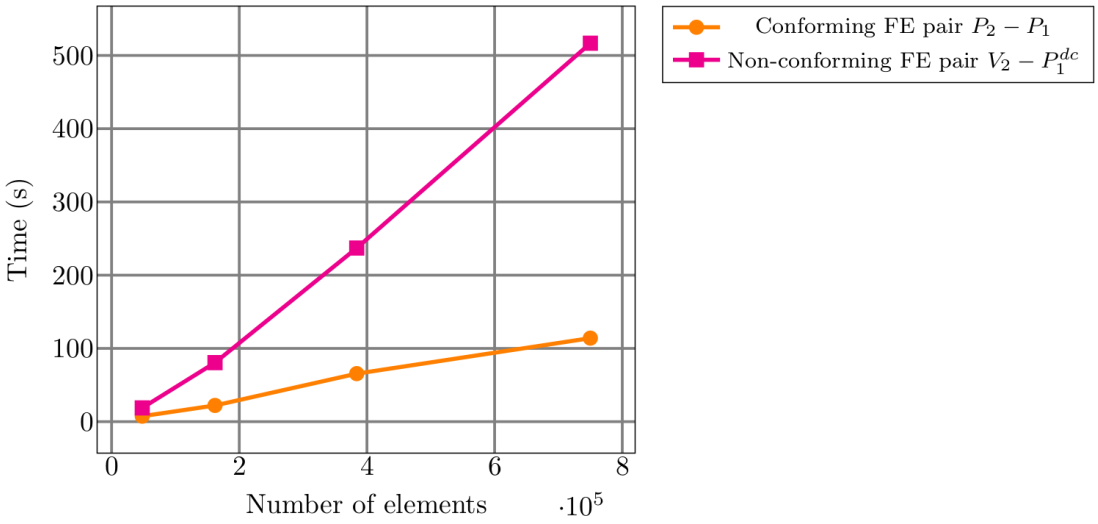
In this section, we consider case #1 (Stokes with non-homogeneous Dirichlet boundary conditions) seen in Section 6. We compare the computational times needed by the non-conforming $V_2-\mathbb{P}_1^{\mathrm{dc}}$ finite element pair with that needed by the conforming $\mathbb{P}_2-\mathbb{P}_1$ finite element pair. To do this, we consider the following computational steps:

- the assembly phase, during which the stiffness matrix and the right-hand side are assembled in FreeFEM;
- the solution phase, during which the preconditioner is setup and the linear system is solved algebraically.

The computations are run on ORCUS a cluster from CEA Saclay hosted at TGCC. It is composed of AMD EPYC 7281 CPUs, clocked at 2.3 GHz. For each test, the average number of outer iterations is about 40. Deriving a more efficient preconditioner to decrease the number of outer iterations is out of the scope of this paper.



(a) Assembly phase



(b) Solution phase

Figure 13: Comparison of the computational times of the non-conforming FE pair $V_2 - \mathbb{P}_1^{\text{dc}}$ and the conforming FE pair $\mathbb{P}_2 - \mathbb{P}_1$ for different mesh sizes computed on 64 processes.

First, in Fig. 13, we compare different computational times needed by the two pairs of finite elements with different mesh sizes computed on 64 processes. As expected, the computational times needed by the non-conforming pair of finite elements are higher than those needed by the conforming one, since it has more degrees of freedom. Besides, we clearly see that the different computational times increase linearly with the mesh size (and thus with the number of degrees of freedom), a clear manifestation of the good scaling of the implemented solver. In Table 8, we compute the ratio of the computational times needed by the two finite element pairs for different mesh sizes presented in Table 7.

Mesh ID	1	2	3	4	5	6
# of elements ($6\times$)	20^3	30^3	40^3	50^3	80^3	100^3

Table 7: Different considered meshes

Mesh ID	# of processes	÷ # of unknowns	÷ assembly phase	÷ solution phase	÷ total computational time
1	64	5.6	3.7	2.5	2.6
2		5.8	3.8	3.6	3.6
3		5.8	2.9	3.6	3.5
4		5.8	2.8	4.5	4.3
5	512	5.9	2.9	6.0	5.0
6		5.9	2.8	7.6	5.6

Table 8: Ratio (represented by the symbol ÷) between different computational times obtained with the conforming $\mathbb{P}_2 - \mathbb{P}_1$ finite element pair and the non-conforming $V_2 - \mathbb{P}_1^{\mathrm{dc}}$ finite element pair

From Table 8, we see that the ratio of the computational times needed with the two finite element pairs is lower than the ratio of the number of degrees of freedom of the two finite element pairs. Indeed, although the non-conforming finite element pair has about six times more degrees of freedom than the conforming one, the computational time ranges between 2.6 and 5.6 times higher.

Remark 7.5. *Similar comparisons can be done between the conforming $\mathbb{P}_3 - \mathbb{P}_2$ finite element pair and the non-conforming $V_3 - \mathbb{P}_2^{dc}$ finite element pair. The same conclusions apply.*

8 Conclusion

In this article, two inf-sup stable non-conforming discretisations with accuracy order two and three on tetrahedra were developed for the Stokes problem in three dimensions. From a given set of degrees of freedom, we propose a strategy to find unisolvent finite element families with respect to those degrees of freedom. Our family of finite elements fulfils the consistency, approximability and stability conditions, guaranteeing optimal orders of convergence which are indeed observed in the numerical results. We recall that this family of finite elements has been designed for solving more complex problems (for example with polynomial divergence and Lagrange multipliers) such as the local problems involved in MsFEM [14] (further details can be found in [5]). Given the theoretical and numerical results presented in this paper, the two new non-conforming finite elements proposed meet the requirements of these complex problems. However, for the solution of the classical Stokes problem, the non-conforming discretisations are more expensive than the conforming ones both in terms of number of degrees of freedom and computational times, for ultimately comparable errors.

Acknowledgements

The authors would like to thank Prof. Frédéric Hecht for his help with the implementation of the two finite elements in FreeFEM.

A Strategy to find the unisolvent basis functions

We propose to find numerically the additional functions of $\mathbb{P}_{n+2}$ needed to complete the space $\mathbb{P}_{n+1}$ while ensuring that the generated space V_{n+1} is unisolvent (in the sense of Definition 3.3) with respect to $\mathcal{N}_{n+1}$, the set of degrees of freedom. First, we introduce the following lemma.

Lemma A.1. *Let K be a tetrahedron of vertices A, B, C, D, and λ_A, λ_B, λ_C, λ_D, the associated barycentric coordinates. Then, the family $\left\{(\lambda_A^\alpha \lambda_B^\beta \lambda_C^\gamma \lambda_D^\delta)$ such that $(\alpha, \beta, \gamma, \delta) \in [\![0, n]\!]^4$ and $\alpha + \beta + \gamma + \delta = n\right\}$ is a basis of the space $\mathbb{P}_n(K)$.*

Proof. The proof consists of two steps. In the first one, we show that the family $\left\{(\lambda_A^\alpha \lambda_B^\beta \lambda_C^\gamma \lambda_D^\delta) \text{ such that } (\alpha,\beta,\gamma,\delta) \in [\![0,n]\!]^4 \text{ and } \alpha+\beta+\gamma+\delta = n\right\}$ is linearly independent. In the second one, we show that this family has the same dimension as the space $\mathbb{P}_n(K)$.

Step 1 Let us call (ABC) the face delimited by vertices A, B and C, and (AB) the edge delimited by vertices A and B. We define similarly the other faces and edges of tetrahedron K. We shall work with restrictions of λ_S (with $S \in \{A,B,C\}$) to (ABC) and (AB) and, for the sake of simplicity, we shall still denote these restrictions λ_S instead of $\lambda_S|_{(AB)}$ or $\lambda_S|_{(ABC)}$.

Let us show that the family $(\lambda_A^{\alpha_i}\lambda_B^{\beta_i}\lambda_C^{\gamma_i}\lambda_D^{\delta_i})$ with $\alpha_i+\beta_i+\gamma_i+\delta_i = n$ is linearly independent. We define the polynomial P as:

$$P = \sum_i a_i \lambda_A^{\alpha_i}\lambda_B^{\beta_i}\lambda_C^{\gamma_i}\lambda_D^{\delta_i}$$

with some real scalars a_i. The objective is to show that $P=0$ implies that all coefficients a_i are equal to 0. Below, we give the general procedure to show this implication.

First, we consider the restriction of P on the edge (AB), i.e., on the edge on which $\lambda_C = 0$ and $\lambda_D = 0$. Consequently, P is reduced to:

$$P|_{(AB)} = \sum_i a_i \lambda_A^{\alpha_i}\lambda_B^{\beta_i} = 0 \tag{17}$$

with $\alpha_i+\beta_i = n$. Now, given that on (AB), λ_A is the first-order polynomial that is equal to 1 in A and 0 in B, and that λ_B verifies $\lambda_B = 1-\lambda_A$, the family $(\lambda_A^{\alpha_i}\lambda_B^{\beta_i})_i$ is a family of Bernstein polynomials which is known to be a basis of the space $\mathbb{P}_n(AB)$, polynomials of order n on the edge (AB). Consequently, all the coefficients a_i in (17) are null. We repeat the same procedure on all the other edges of K, which allows to show that all the coefficients a_i associated with monomials involving at most two barycentric coordinates are null.

At this stage, if $n \leq 2$, the proof is over and the property is proved. If $n \geq 3$, it remains in P only monomials involving at least three barycentric coordinates. We consider the restriction of P to the face (ABC), i.e., to the face on which $\lambda_D = 0$. Consequently, P is reduced to:

$$P|_{(ABC)} = \sum_i a_i \lambda_A^{\alpha_i}\lambda_B^{\beta_i}\lambda_C^{\gamma_i} = 0$$

with $\alpha_i+\beta_i+\gamma_i = n$ and $\alpha_i \neq 0$, $\beta_i \neq 0$ and $\gamma_i \neq 0$. Consequently, it is possible to factorize P by the term $\lambda_A\lambda_B\lambda_C$ which leads to:

$$P|_{(ABC)} = \lambda_A\lambda_B\lambda_C \sum_i a_i \lambda_A^{\alpha'_i}\lambda_B^{\beta'_i}\lambda_C^{\gamma'_i} = 0$$

with $\alpha'_i+\beta'_i+\gamma'_i = n-3$. The polynomial P vanishing on (ABC) and $\lambda_A\lambda_B\lambda_C$ not being the null polynomial on (ABC), it follows that:

$$Q|_{(ABC)} := \sum_i a_i \lambda_A^{\alpha'_i}\lambda_B^{\beta'_i}\lambda_C^{\gamma'_i} = 0.$$

It should be noted that in Q some α'_i, β'_i or γ'_i are equal to zero. Consequently, it is possible to restrict Q to edges of (ABC) to show that the coefficients associated with monomials involving at most two barycentric coordinates are null. Afterwards, if necessary, it is once again possible to factorize Q by $\lambda_A\lambda_B\lambda_C$ and we repeat the procedure until we have shown that all the coefficients a_i in the polynomial Q are null. We repeat the same procedure on all the other faces of K, which allows to show that all the coefficients a_i associated with monomials involving at most three barycentric coordinates are null.

At this stage, if $n \leq 3$, the proof is over and the property is proved. If $n \geq 4$, it remains in P only monomials involving exactly four barycentric coordinates, i.e.:

$$P = \sum_i a_i \lambda_A^{\alpha_i} \lambda_B^{\beta_i} \lambda_C^{\gamma_i} \lambda_D^{\delta_i} = 0$$

with $\alpha_i \neq 0$, $\beta_i \neq 0$, $\gamma_i \neq 0$ and $\delta_i \neq 0$ and $\alpha_i + \beta_i + \gamma_i + \delta_i = n$. Consequently, it is possible to factorize P by $\lambda_A \lambda_B \lambda_C \lambda_D$ which leads to:

$$P = \lambda_A \lambda_B \lambda_C \lambda_D \sum_i a_i \lambda_A^{\alpha'_i} \lambda_B^{\beta'_i} \lambda_C^{\gamma'_i} \lambda_D^{\delta'_i} = 0$$

with $\alpha'_i + \beta'_i + \gamma'_i + \delta'_i = n - 4$. The polynomial P vanishing on K and $\lambda_A \lambda_B \lambda_C \lambda_D$ not being the null polynomial on K, it follows that:

$$Q := \sum_i a_i \lambda_A^{\alpha'_i} \lambda_B^{\beta'_i} \lambda_C^{\gamma'_i} \lambda_D^{\delta'_i} = 0.$$

It should be noted that in Q some α'_i, β'_i, γ'_i or δ'_i are equal to zero. Consequently, we can repeat the above procedure (restriction of Q to faces, then to edges) to show that all coefficients a_i in Q are null. We have thus shown that all coefficients a_i associated with monomials involving four barycentric coordinates are null.

At the end of this general procedure, we have showed that all the coefficients a_i in P are null. Therefore, the set $\left\{(\lambda_A^{\alpha_i} \lambda_B^{\beta_i} \lambda_C^{\gamma_i} \lambda_D^{\delta_i}) \text{ such that } (\alpha_i, \beta_i, \gamma_i, \delta_i) \in [\![0, n]\!]^4 \text{ and } \alpha_i + \beta_i + \gamma_i + \delta_i = n\right\}$ is linearly independent.

Step 2 In the second step, let us consider the following set:

$$\mathcal{E}_n^p = \left\{ (\alpha_1, \ldots, \alpha_p) \in [\![0, n]\!]^p \text{ such that } \sum_{i=1}^{p} \alpha_i = n \right\}.$$

A basis of $\mathbb{P}_n^d$ (polynomials of order n with d variables) is given by $\left(\prod_{i=1}^{d} x_i^{\alpha_i} \right)$ with $\sum_{i=1}^{d} \alpha_i \leq n$. Here, the family we consider is $\left(\prod_{i=1}^{d+1} \lambda_i^{\alpha_i} \right)$ with $\sum_{i=1}^{d+1} \alpha_i = n$. By noting that $\sum_{i=1}^{d} \alpha_i = n - \alpha_{d+1}$ and $\sum_{i=1}^{d} \alpha_i \leq n$ are equivalent since $0 \leq \alpha_{d+1} \leq n$, then it is clear that:

$$\text{card}(\mathcal{E}_n^{d+1}) = \dim(\mathbb{P}_n^d) = \binom{n+d}{n}. \tag{18}$$

And it particular:

$$\text{card}(\mathcal{E}_n^4) = \frac{(n+1)(n+2)(n+3)}{6} = \dim(\mathbb{P}_n^3). \tag{19}$$

Below, we prove the last equality of (18). First, we note that

$$\text{card}(\mathcal{E}_n^p) = \sum_{i=0}^{n} \text{card}(\mathcal{E}_i^{p-1}).$$

By noting that:

$$\text{card}(\mathcal{E}_n^1) = 1,$$

then, it follows that:

$$\begin{aligned}\mathrm{card}(\mathcal{E}_n^2) &= n+1,\\ \mathrm{card}(\mathcal{E}_n^3) &= \frac{(n+1)(n+2)}{2},\\ \mathrm{card}(\mathcal{E}_n^4) &= \frac{(n+1)(n+2)(n+3)}{6}.\end{aligned}$$

Consequently, $\mathrm{card}(\mathcal{E}_n^4)$ is exactly the dimension of $\mathbb{P}_n^3$. More generally we show that:

$$\mathrm{card}(\mathcal{E}_n^{d+1}) = \binom{n+d}{n} = \frac{(n+d)!}{n!\,d!} = \dim(\mathbb{P}_n^d).$$

Combining Step 1 and Step 2, in particular equality (19), allows to conclude the proof of the lemma. □

Based on this result, we explain below the main steps of the algorithm that finds a basis of the proposed space V_{n+1}. Since V_{n+1} is the direct sum of $\mathbb{P}_{n+1}$ with a subspace of $\mathbb{P}_{n+2}$, the basis of V_{n+1} is composed of the basis of $\mathbb{P}_{n+1}$ with $n(n+2)$ suitable additional functions selected in the basis of $\mathbb{P}_{n+2}$.

1. **Step 1**: In order to complete the space $\mathbb{P}_{n+1}$, we generate all functions of degree $n+2$ that can be written as $\lambda_1^i\lambda_2^j\lambda_3^k\lambda_4^l$ with $i+j+k+l = n+2$ (see Lemma A.1). We store them in the list $AllFunctionsInSigma$.

2. **Step 2**: For all possible combinations of $n(n+2)$ functions among the list $AllFunctionsSigma$ (for all possible Σ_{n+2}):

   - **Step 2.1**: We generate the corresponding candidate space V_{n+1} by concatenating a basis of $\mathbb{P}_{n+1}$ with the considered basis of Σ_{n+2}.
   - **Step 2.2**: We test the unisolvence of the considered candidate basis of V_{n+1}. To verify the unisolvence of the set of functions in the candidate V_{n+1} according to the set of degrees of freedom, we assemble the matrix $\mathcal{M}$ defined by its coefficients as follows

     $$\mathcal{M}_{ij} = N_i(\varphi_j) \qquad i,\ j \geq 0,$$

     for N_i in $\mathcal{N}_{n+1}$ and $\varphi_j \in V_{n+1}$. Then, we compute the rank of matrix $\mathcal{M}$ (in this work, we use the NumPy linear algebra function `matrix_rank` [1]). If matrix $\mathcal{M}$ is of full rank, i.e., invertible, the set of functions (φ_j) is unisolvent with respect to the set of degrees of freedom $\mathcal{N}_{n+1}$ (see Definition 3.3).

Results for the case $n = 1$ We discuss the output of the algorithm for the case $n = 1$.

- At the first step of the algorithm, we generate a list of 20 functions of degree three.
- Then, by considering 3 functions among 20, we obtain 1140 potential spaces Σ_3.
- By testing the unisolvence of the resulting space V_2, we find that 128 spaces Σ_3 are suitable to complete $\mathbb{P}_2$. These subspaces are presented in the appendix.

Results for the case $n = 2$ We discuss about the output of the algorithm for the case $n = 2$.

- At the first step of the algorithm, we generate a list of 35 functions of degree 4.
- Then, by considering 8 functions among 35, we obtain 23,535,820 potential spaces Σ_4.
- By testing the unisolvence of the resulting space V_3, we find that 2,368,236 spaces Σ_4 are suitable to complete $\mathbb{P}_3$.

Conjecture for the case $n = 3$ For the case $n = 3$, we follow the same procedure. We generate first the list of the 56 basis functions of $\mathbb{P}_5(K)$. However, at this stage by considering 15 functions among the 56 functions to form the subspace Σ_4, the number of possible combinations is higher than 16,000 billion. Obviously, we cannot test the unisolvence of all resulting space V_4. We test randomly millions of possible space V_4, but as of today, no suitable space Σ_5 has been found. The existence (or not) of such bases is still an open question.

B Suitable bases of Σ_3

In this section, we present the output of the procedure described in Appendix A for the case $n = 1$, i.e., all the bases of the space Σ_3 that can be used to enrich $\mathbb{P}_2$ to build the space V_2.

1. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2)$
2. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1)$
3. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
4. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
5. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
6. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
7. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1)$
8. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2)$
9. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
10. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
11. $(\lambda_3^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
12. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1)$
13. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2)$
14. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_2^2)$
15. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
16. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
17. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2)$
18. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_2^2)$
19. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1)$
20. $(\lambda_3^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
21. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
22. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
23. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
24. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
25. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
26. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
27. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
28. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
29. $(\lambda_3^1\lambda_4^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
30. $(\lambda_3^1\lambda_4^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
31. $(\lambda_3^1\lambda_4^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
32. $(\lambda_3^1\lambda_4^2,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
33. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2)$
34. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1)$
35. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
36. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
37. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
38. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
39. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1)$
40. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2)$
41. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
42. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
43. $(\lambda_3^2\lambda_4^1,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
44. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1)$
45. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2)$
46. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_2^2)$
47. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
48. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
49. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2)$
50. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_2^2)$
51. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1)$
52. $(\lambda_3^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
53. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
54. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
55. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
56. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
57. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
58. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
59. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
60. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
61. $(\lambda_3^2\lambda_4^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
62. $(\lambda_3^2\lambda_4^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
63. $(\lambda_3^2\lambda_4^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
64. $(\lambda_3^2\lambda_4^1,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
65. $(\lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2)$
66. $(\lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_3^2)$
67. $(\lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
68. $(\lambda_2^1\lambda_4^2,\ \lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_3^1)$
69. $(\lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2)$
70. $(\lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_3^2)$
71. $(\lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1)$
72. $(\lambda_2^1\lambda_4^2,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_3^1)$
73. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
74. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
75. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
76. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
77. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
78. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
79. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
80. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
81. $(\lambda_2^1\lambda_4^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
82. $(\lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
83. $(\lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
84. $(\lambda_2^1\lambda_4^2,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
85. $(\lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_4^2)$
86. $(\lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2)$
87. $(\lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_4^1)$
88. $(\lambda_2^1\lambda_3^2,\ \lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
89. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
90. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
91. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
92. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
93. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
94. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
95. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
96. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
97. $(\lambda_2^1\lambda_3^2,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
98. $(\lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
99. $(\lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
100. $(\lambda_2^1\lambda_3^2,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
101. $(\lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2)$
102. $(\lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_3^2)$
103. $(\lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1)$
104. $(\lambda_2^2\lambda_4^1,\ \lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_3^1)$
105. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
106. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
107. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
108. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$

109. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
110. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
111. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
112. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
113. $(\lambda_2^2\lambda_4^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
114. $(\lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
115. $(\lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
116. $(\lambda_2^2\lambda_4^1,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$
117. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_3^2)$
118. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^1\lambda_2^2)$
119. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_3^1)$
120. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_4^2,\ \lambda_1^2\lambda_2^1)$
121. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^1\lambda_2^2)$
122. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_4^1)$
123. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_3^2,\ \lambda_1^2\lambda_2^1)$
124. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_4^1)$
125. $(\lambda_2^2\lambda_3^1,\ \lambda_1^1\lambda_2^2,\ \lambda_1^2\lambda_3^1)$
126. $(\lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_3^1)$
127. $(\lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_4^1,\ \lambda_1^2\lambda_2^1)$
128. $(\lambda_2^2\lambda_3^1,\ \lambda_1^2\lambda_3^1,\ \lambda_1^2\lambda_2^1)$

References

[1] numpy.linalg.matrix_rank — NumPy v1.26 Manual.

[2] P. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[3] P. R. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary. Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures. *ACM Transactions on Mathematical Software*, 45(1):1–26, Mar. 2019.

[4] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc Web page. `https://petsc.org/`, 2023.

[5] L. Balazi. *Development of a multiscale finite element method for incompressible flows in heterogeneous media*. PhD thesis, Institut Polytechnique de Paris, 2024 (in preparation).

[6] D. Boffi. Three-Dimensional Finite Element Methods for the Stokes Problem. *SIAM Journal on Numerical Analysis*, 34(2):664–670, Apr. 1997.

[7] J. Brown, M. G. Knepley, D. A. May, L. C. McInnes, and B. Smith. Composable linear solvers for multiphysics. In *2012 11th International Symposium on Parallel and Distributed Computing*, pages 55–62, 2012.

[8] X.-C. Cai and M. Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[9] W. Chen, J. Hu, and M. Zhang. Nonconforming finite element methods of order two and order three for the Stokes flow in three dimensions, Dec. 2022. arXiv:2212.11748 [cs, math].

[10] M. Crouzeix and P.-A. Raviart. Conforming and nonconforming finite element methods for solving the stationary Stokes equations I. *Revue française d'automatique informatique recherche opérationnelle. Mathématique*, 7(R3):33–75, 1973.

[11] Y. Efendiev and T. Y. Hou. *Multiscale finite element methods. Theory and applications.*, volume 4 of *Surv. Tutor. Appl. Math. Sci.* New York, NY: Springer, 2009.

[12] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers. With applications in incompressible fluid dynamics.* Numer. Math. Sci. Comput. Oxford: Oxford University Press, 2nd ed. edition, 2014.

[13] A. Ern and J.-L. Guermond. *Finite elements. 1: Approximation and interpolation.* Number Volume 72 in Texts in applied mathematics. Springer, Cham, Switzerland, 2021.

[14] Q. Feng, G. Allaire, and P. Omnes. Enriched Nonconforming Multiscale Finite Element Method for Stokes Flows in Heterogeneous Media Based on High-order Weighting Functions. *Multiscale Modeling & Simulation*, pages 462–492, Mar. 2022. Publisher: Society for Industrial and Applied Mathematics.

[15] M. Fortin. An analysis of the convergence of mixed finite element methods. *RAIRO. Analyse numérique*, 11(4):341–354, 1977.

[16] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations. Theory and algorithms.*, volume 5 of *Springer Ser. Comput. Math.* Springer, Cham, (Extended version of the 1979 publ.) edition, 1986.

[17] F. Hecht. New development in FreeFEM++. *J. Numer. Math.*, 20(3-4):251–265, 2012.

[18] J.-P. Hennart, J. Jaffre, and J. E. Roberts. A constructive method for deriving finite elements of nodal type. *Numerische Mathematik*, 53(6):701–738, Nov. 1988.

[19] G. Matthies. Inf-sup stable nonconforming finite elements of higher order on quadrilaterals and hexahedra. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(5):855–874, Sept. 2007. Number: 5 Publisher: EDP Sciences.

[20] G. Matthies and L. Tobiska. Inf-sup stable non-conforming finite elements of arbitrary order on triangles. *Numerische Mathematik*, 102(2):293–309, Dec. 2005.

[21] B. Parisse and R. D. Graeve. Giac/Xcas, logiciel libre de calcul formel.

[22] PETSc. Guide to the Stokes Equations using Finite Elements — PETSc v3.19.1-422-gcb867a2c0e7 documentation.

[23] A. Plaza and M. Rivara. Average adjacencies for tetrahedral skeleton-regular partitions. *Journal of Computational and Applied Mathematics*, 177(1):141–158, May 2005.

[24] Y. Saad. A Flexible Inner-Outer Preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[25] S. Sauter and C. Torres. On the inf-sup stability of Crouzeix–Raviart Stokes elements in 3D. *Mathematics of Computation*, 92(341):1033–1059, May 2023.

[26] R. Stenberg. On some three-dimensional finite elements for incompressible media. *Computer Methods in Applied Mechanics and Engineering*, 63(3):261–269, Aug. 1987.