Refined Isogeometric Analysis for fluid mechanics and electromagnetism

Daniel Garcia[a,*], David Pardo[b,a,c], Victor M. Calo[d,e]

[a] *Basque Center for Applied Mathematics, (BCAM), Bilbao, Spain.*
[b] *Department of Applied Mathematics, Statistics, and Operational Research, University of the Basque Country UPV/EHU, Leioa, Spain*
[c] *Ikerbasque (Basque Foundation for Sciences), Bilbao, Spain.*
[d] *Department of Applied Geology, Western Australian School of Mines, Curtin University, Perth, Australia.*
[e] *Mineral Resources, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Perth, Australia.*

Abstract

Starting from a highly continuous Isogeometric analysis, we introduce hyperplanes that partition the domain into subdomains and reduce the continuity of the discretization spaces along those hyperplanes. As the continuity is reduced, the number of degrees of freedom in the system grows. The resulting discretization spaces are finer than standard maximal continuity IGA spaces. Despite the increase in the number of degrees of freedom, these finer spaces entail faster results with direct solvers than both traditional Finite Element Analysis (FEA) and Isogeometric Analysis (IGA) for meshes with a fixed number of elements. In this work, we analyze the impact of continuity reduction on the number of Floating Point Operations (FLOPs) and computational times required to solve fluid flow and electromagnetic problems with structured meshes and uniform polynomial orders. Theoretical estimates show that for sufficiently large grids, an optimal continuity reduction decreases the computational cost by a factor of $O(p^2)$. Numerical results confirm these theoretical estimates. In a 2D mesh with one million elements and polynomial order equal to five, the discretization including an optimal continuity pattern allows to solve the vector electric field, the scalar magnetic field, and the fluid flow problems an order of magnitude faster, than when using a highly continuous IGA discretization. 3D numerical results exhibit more moderate savings due to the limited mesh sizes considered in this work.

Keywords: Isogeometric Analysis (IGA), Finite Element Analysis (FEA), refined Isogeometric Analysis (rIGA), solver-based discretization, Multi-field problems, k-refinement.

1. Introduction

Isogeometric analysis (IGA) is a discretization method frequently employed to solve numerical problems governed by partial differential equations (PDEs) [1–3]. This method defines the geometry using conventional Computer-aided design (CAD) functions and, in particular, non-uniform rational B-spline (NURBS). These functions can represent complex geometries commonly found in engineering design and are capable of preserving exactly the geometry description under refinement. Moreover, IGA employs the functions that Computed-Aided Design (CAD) uses to build the discretization of the governing PDEs. Therefore, it is possible to build algebraic systems directly from the computational domain representation based on spline functions, which arise from CAD, avoiding the definition of a separate geometric description for the numerical analysis. This results in a reduction of the total analysis time [1].

The performance of direct solvers strongly depends upon the employed discretization. In particular, for IGA, the continuity of the solution spaces plays a significant role in the direct solvers performance. Highly-continuous spaces degrade their performance, increasing the solution times by a factor up to $O(p^3)$ with respect to traditional finite element analysis (FEA) per unknown, being p the polynomial order [4–6].

[*] Corresponding author
Email address: dgarcia@bcamath.org (Daniel Garcia)
URL: www.bcamath.org/dgarcia (Daniel Garcia)

As a remedy to the lack of performance exhibited by direct solvers when applied to IGA discretizations, in [7–9] we proposed the refined Isogeometric analysis method (rIGA). Starting from a highly continuous IGA discretization, rIGA introduces lower continuous hyperplanes, which act as separators in terms of the direct solver, to reduce the interconnection between the degrees of freedom (DoF) used in the mesh. By doing so, both the solution time and best approximation errors are simultaneously improved. In particular, rIGA delivers speed-up factors proportional to p^2 when solving scalar problems. For instance, in a 2D mesh with four million elements and $p = 5$, a Laplace linear system resulting from rIGA is solved 22 times faster than the one from highly continuous IGA. In a 3D mesh with one million elements and $p = 3$, the linear rIGA system is solved 15 times faster than the IGA one [7].

In this work, we apply rIGA to solve multi-field problems that possibly employ high-continuity discretizations of H^1, $\boldsymbol{H}$(curl), $\boldsymbol{H}$(div), and L^2 spaces and/or a combination of them (as described in [10]). We consider an electromagnetic problem and an incompressible fluid flow problem to numerically analyze the main features and limitations of rIGA in multi-fields examples. In the electromagnetic problem, we approximate the electric field resulting from a time-harmonic magnetic dipole source by using a spline-based generalization of the 2D Ndelec finite element spaces. In the incompressible Stoke flow problem in an enclosed domain, we use a spline-based generalization of the Raviar-Thomas finite element spaces to approximate the velocity field.

This paper is organized as follows: Section 2 introduces the main ideas of IGA when applied to multi-field problems, and the performance of direct solvers for IGA discretizations. Section 3 describes the refined isogeometric analysis (rIGA) proposed to solve multi-field problems. Section 4 details the implementation. Section 5 presents the numerical experiments. We discuss some features and limitations of the proposed method in Section 6.

2. Isogeometric Analysis discretizations for direct solvers

In order to approximate the solution of problems governed by partial differential equations (PDEs), we can use numerical methods as finite element analysis (FEA) and isogeometric analysis (IGA). These methods build discretizations of the problems using a variational formulation with trial and test functions defined as a linear combination of the basis functions. Depending on the PDEs that govern the problems, we select the discretization spaces. For instance, in FEA, we implement classical Lagrange spaces (gradient-conforming spaces) to study heat transfer problems, Nedelec (or Whitney) spaces (curl-conforming spaces) to study electromagnetics problems, and Taylor-Hood or Raviart-Thomas spaces (divergence-conforming spaces) for incompressible fluid flow problems (Figure 1).

In IGA, we may use spline generalizations of the FEA discretization spaces. These discretization spaces are globally continuous due to the high inter-element smoothness that the splines provide. Moreover, those spaces are structure-preserving since the gradient-, curl-, divergence-, and integral-conforming properties of the FEA spaces hold in the continuous setting.

We now give a brief introduction to B-spline based spaces and then we describe proper gradient-, curl-, divergence-, and integral-conforming IGA spaces.

2.1. B-spline spaces

A B-spline space consists of a set of all possible combinations of univariate (one-dimensional) B-splines functions arranged on knot vectors. The univariate B-spline functions are defined by the Cox-de Boor recursion formula [11, 12], and the knot vectors are defined as

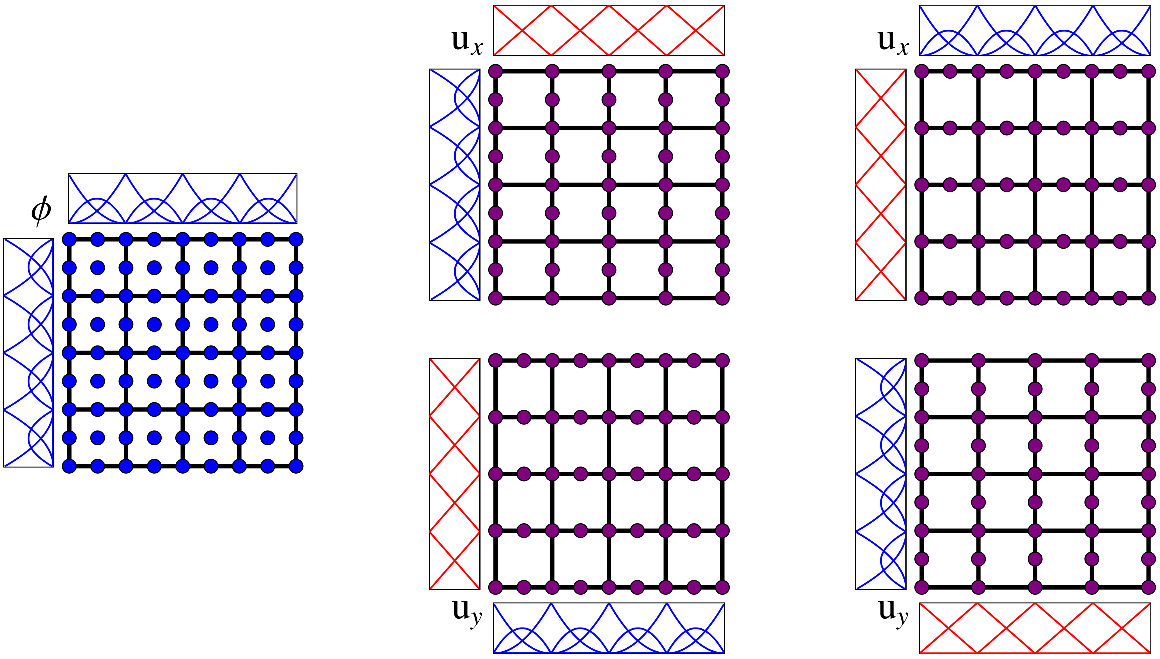
$$\varXi = (\xi_1, \ldots, \xi_{n_{bfx}+p_x+1}) \qquad \forall\, \xi_i \in \mathbb{R}, \tag{1}$$

where ξ_i is the i-th knot, p_x is the polynomial order, and $n_{bf} = n_{\text{elem}} + p_x$ is the number of basis functions, where $n_{\text{elem}} = \sqrt[d]{N_{\text{elem}}}$ is the number of elements in one spatial dimension, and d the dimension of the space.

A set of B-spline basis functions $\{\mathcal{N}_{1,p_x}, \ldots, \mathcal{N}_{n_{bfx},p_x}\}$ with support on a single knot vector (e.g., $\varXi$) defines a 1D space (with all the properties required for analysis purposes). This space is denoted as

$$\mathcal{S}^{p_x}_{k_x} := \text{span}\left\{\mathcal{N}_{i,p_x}\right\}_{i=1}^{n_{bfx}}, \tag{2}$$

where p_x and k_x indicates the polynomial degree and continuity of the functions that define the space.



(a) Bezier spaces: $\phi \in H^1(\Omega)$.

(b) Nedelec spaces: $\mathbf{u} = (\mathrm{u}_x, \mathrm{u}_y) \in \boldsymbol{H}(\mathrm{curl};\Omega)$.

(c) Raviart-Thomas spaces: $\mathbf{u} = (\mathrm{u}_x, \mathrm{u}_y) \in \boldsymbol{H}(\mathrm{div};\Omega)$.

Figure 1: Example of three types of FEA spaces for a 2D discretization with 4 × 4 elements and polynomial basis functions of order $p = 2$ (and $p = 1$). Red and violet circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton.

In 2D, we define the spaces by a set of bivariate B-spline basis functions. These functions are constructed as the tensor product of two univariate B-splines basis functions with support on different knot vectors and are defined as

$$\mathbf{N}^{p_x,p_y}_{i_x,i_y} = \mathcal{N}^{p_x}_{i_x}|_{\Xi} \otimes \mathcal{N}^{p_y}_{i_y}|_{\Psi}, \quad i_x = 1,\ldots,n_{bf_x}, \quad i_y = 1,\ldots,n_{bf_y}, \tag{3}$$

where $\mathcal{N}^{p_x}_{i_x}|_{\Xi}$ and $\mathcal{N}^{p_y}_{i_y}|_{\Psi}$ are the B-spline functions with support on the knot vectors $\Xi = (\xi_i,\ldots,\xi_{n_{bf_x}+p_x+1})$ and $\Psi = (\psi_i,\ldots,\psi_{n_{bf_y}+p_y+1})$, respectively. We denote the 2D space as

$$\mathcal{S}^{p_x,p_y}_{k_x,k_y} := \operatorname{span}\left\{\mathbf{N}^{p_x,p_y}_{i_x,i_y}\right\}^{n_{bf_x},n_{bf_y}}_{i_x,i_y=1}. \tag{4}$$

The 3D spaces are defined by a set of trivariate B-splines basis functions. Using three different sets of B-splines functions with support on the knot vectors $\Xi = (\xi_i,\ldots,\xi_{n_{bf_x}+p_x+1})$, $\Psi = (\psi_i,\ldots,\psi_{n_{bf_y}+p_y+1})$, and $\Phi = (\varphi_i,\ldots,\varphi_{n_{bf_z}+p_z+1})$, we define the trivariate B-splines basis functions as

$$\mathbf{N}^{p_x,p_y,p_z}_{i_x,i_y,i_z} = \mathcal{N}^{p_x}_{i_x}|_{\Xi} \otimes \mathcal{N}^{p_y}_{i_y}|_{\Psi} \otimes \mathcal{N}^{p_y}_{i_z}|_{\Phi}, \quad i_x = 1,\ldots,n_{bf_x}, \quad i_y = 1,\ldots,n_{bf_y}, \quad i_z = 1,\ldots,n_{bf_z}, \tag{5}$$

where $\mathcal{N}^{p_x}_{i_x}|_{\Xi}$, $\mathcal{N}^{p_y}_{i_y}|_{\Psi}$, and $\mathcal{N}^{p_y}_{i_z}|_{\Phi}$ are the B-spline functions with support on the Ξ, Ψ, and Φ knot vectors, respectively. We denote the 3D space as

$$\mathcal{S}^{p_x,p_y,p_z}_{k_x,k_y,k_z} := \operatorname{span}\left\{\mathbf{N}^{p_x,p_y,p_z}_{i_x,i_y,i_z}\right\}^{n_{bf_x},n_{bf_y},n_{bf_z}}_{i_x,i_y,i_z=1}. \tag{6}$$

2.2. B-spline discrete spaces

Next, we present the smooth generalization of the gradient (grad), divergence (div), curl, and integral (int) conforming spaces on which we base our study. We define the parametric spaces, and then we introduce an appropriate transformation to obtain the discrete spaces on the physical domain.

2.2.1. *H^1 conforming discrete spaces*

We now describe the smooth gradient-conforming IGA discrete spaces (Figure 2). They coincide with the classical linear Lagrange FEA spaces for polynomial degree equal one. However, for a higher polynomial degree, the support of the basis functions increases with the continuity of the space. This results in smoother spaces than their FEA counterparts.

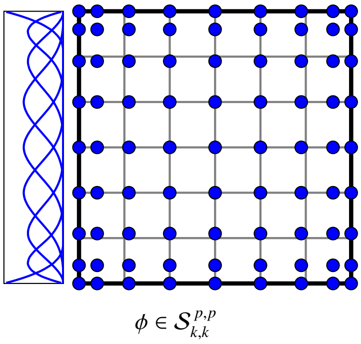


Figure 2: Illustration of a H^1 conforming IGA space for a 2D discretization with 6×6 elements, polynomial order $p = 3$, and continuity degree $k = 2$. Blue circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton.

The gradient-conforming IGA discrete spaces in a parametric domain $\hat{\Omega} \subset \mathbb{R}$ are given by

$$\hat{Q}_h^{\mathrm{grad}}(\hat{\Omega}) = \begin{cases} \mathcal{S}_{k,\,k}^{p,\,p} & \text{2D} \\ \mathcal{S}_{k,\,k,\,k}^{p,\,p,\,p} & \text{3D}, \end{cases} \tag{7}$$

where $\mathcal{S}$ with indexes refers to a B-spline space, p is the polynomial degree, k indicates the basis functions continuity.

To move the space definition from the parametric to a physical domain Ω, we use the following pullback mapping

$$\iota_\phi^{\mathrm{grad}}(\theta) = \theta \circ \mathbf{F}, \qquad \theta \in H^1(\Omega) \tag{8}$$

where ι^{grad} is a gradient-preserving transformation and $\mathbf{F}$ is the geometric mapping from the parametric to the physical domain. Thus, the discrete space in the physical domain is

$$Q_h^{\mathrm{grad}} := \left\{\phi \in H^1(\Omega) : \iota_\phi^{\mathrm{grad}}(\theta) \in \hat{Q}_h^{\mathrm{grad}}(\hat{\Omega})\right\}. \tag{9}$$

2.2.2. *$\boldsymbol{H}$(curl) conforming discrete space*

The curl-conforming IGA discrete spaces consist of a spline generalization of the Nedelec FEA spaces (Figure 3). A set of those discrete spaces is defined in a parametric domain $\hat{\Omega} \subset \mathbb{R}^{d=2,3}$ as follows

$$\hat{\mathcal{V}}_h^{\mathrm{curl}}(\hat{\Omega}) = \begin{cases} \mathcal{S}_{k-1,\,k}^{p-1,\,p} \times \mathcal{S}_{k,\,k-1}^{p,\,p-1} & \text{2D} \\ \mathcal{S}_{k-1,\,k,\,k}^{p-1,\,p,\,p} \times \mathcal{S}_{k,\,k-1,\,k}^{p,\,p-1,\,p} \times \mathcal{S}_{k,\,k,\,k-1}^{p,\,p,\,p-1} & \text{3D}, \end{cases} \tag{10}$$

where p is the polynomial degree, k indicates the basis functions continuity, and $\mathcal{S}$ with indexes is a spline space.

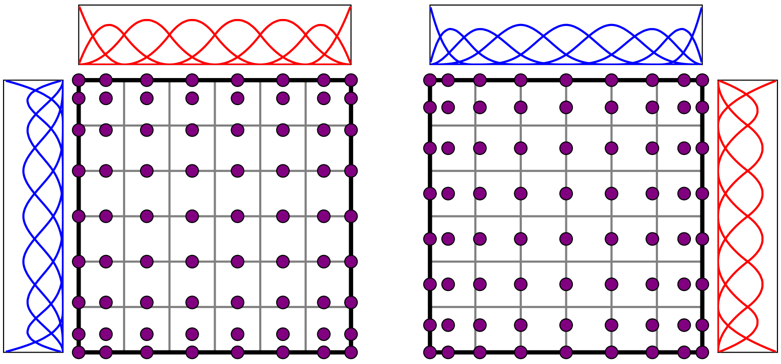
We use a curl-preserving map (that corresponds to the inverse of the covariant Piola transformation) to define the discrete space on the physical domain. This transformation is

$$\iota_{\mathbf{u}}^{\mathrm{curl}}(\mathbf{v}) = (D\mathbf{F})^T(\mathbf{v} \circ \mathbf{F}), \quad \mathbf{v} \in \boldsymbol{H}(\mathrm{curl};\Omega) \cap \boldsymbol{H}^1(\Omega) \tag{11}$$

where $\mathbf{F}$ is the geometric mapping from the parametric domain to the physical domain, and $D\mathbf{F}$ is the gradient of the mapping. Then, the set of discrete spaces in the physical domain is defined as

$$\mathcal{V}_h^{\mathrm{curl}}(\Omega) := \left\{\mathbf{v} \in \boldsymbol{H}(\mathrm{curl};\Omega) \cap \boldsymbol{H}^1(\Omega) : \iota_{\mathbf{u}}^{\mathrm{curl}}(\mathbf{v}) \in \hat{\mathcal{V}}_h^{\mathrm{curl}}(\hat{\Omega})\right\}. \tag{12}$$

This set of curl-conforming spaces was introduced by Buffa et al. in [13] to obtain regular discrete solutions of Maxwell's equations.

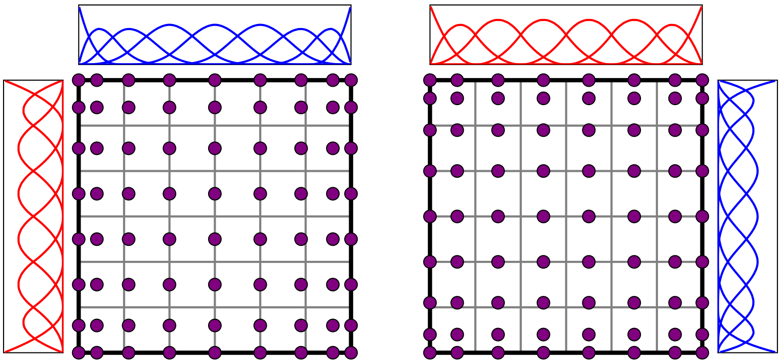


$\mathrm{u}_x \in \mathcal{S}^{p-1,p}_{k-1,k}$ $\mathrm{u}_y \in \mathcal{S}^{p,p-1}_{k,k-1}$

Figure 3: Example of highly-continuous curl-conforming spaces for a 2D discretization with 6 × 6 elements, polynomial order $p = 3$, and continuity degree $k = 2$. Violet circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton.

2.2.3. *$\boldsymbol{H}$(div) conforming discrete spaces*

The set of divergence-conforming IGA discrete spaces we consider here consists of a spline generalization of the Raviart-Thomas FEA spaces (Figure 4).



$\mathrm{u}_x \in \mathcal{S}^{p,p-1}_{k,k-1}$ $\mathrm{u}_y \in \mathcal{S}^{p-1,p}_{k-1,k}$

Figure 4: Example of highly-continuous div-conforming spaces for a 2D discretization with 6 × 6 elements, polynomial order $p = 3$, and continuity degree $k = 2$. Violet and red circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton.

These spaces are defined in the parametric domain ($\hat{\Omega} \subset \mathbb{R}^{d=2,3}$) as follows

$$\hat{\mathcal{V}}^{\mathrm{div}}_h(\hat{\Omega}) = \begin{cases} \mathcal{S}^{p,\,p-1}_{k,\,k-1} \times \mathcal{S}^{p-1,\,p}_{k-1,\,k} & \text{2D} \\ \mathcal{S}^{p,\,p-1,\,p-1}_{k,\,k-1,\,k-1} \times \mathcal{S}^{p-1,\,p,\,p-1}_{k-1,\,k,\,k-1} \times \mathcal{S}^{p-1,\,p-1,\,p}_{k-1,\,k-1,\,k} & \text{3D,} \end{cases} \tag{13}$$

where p is the polynomial degree, k indicates the basis functions continuity, and $\mathcal{S}$ with indexes is a spline space.

In this case, we use a divergence-preserving map (that corresponds to the inverse of the contravariant Piola transformation) to define the discrete spaces on the physical domain. This transformations is

$$\iota^{\mathrm{div}}_{\mathbf{u}}(\mathbf{v}) = \det(D\mathbf{F})(D\mathbf{F})^{-1}(\mathbf{v} \circ \mathbf{F}), \qquad \mathbf{v} \in \boldsymbol{H}(\mathrm{div};\Omega) \cap \boldsymbol{H}^1(\Omega), \tag{14}$$

where $\mathbf{F}$ is the geometric mapping from the parametric domain to the physical domain, $D\mathbf{F}$ is the gradient of the mapping, and $\det(D\mathbf{F})$ is its determinant. Then, the set of discrete spaces in the physical domain is defined as

$$\mathcal{V}^{\mathrm{div}}_h(\Omega) := \left\{\mathbf{v} \in \boldsymbol{H}\,(\mathrm{div};\Omega) \cap \boldsymbol{H}^1\,(\Omega) : \iota^{\mathrm{div}}_{\mathbf{u}}(\mathbf{v}) \in \hat{\mathcal{V}}^{\mathrm{div}}_h(\hat{\Omega})\right\}. \tag{15}$$

Buffa et al. [10] introduced this set of divergence-conforming spaces to approximate the solution of Stokes problems. Later, Evans et al. [14–16] implemented this discretization framework for the Generalized Stokes and Navier-Stokes equations. Sarmiento et al. [17–19] used these spaces to analyze the performance of solvers and implemented these spaces in a high-performance framework for isogeometric analysis [20]. This structure preserving discretization allowed the detailed analysis of the interfacial interactions between complex fluids in [21].

2.2.4. L^2 conforming discrete spaces

We illustrate the integral-conforming IGA discrete spaces in Figure 5. These spaces are high-continuity discretization spaces with inter-element regularity $C^{(k-1)} : (p-1) \geq (k-1) \geq 0$.

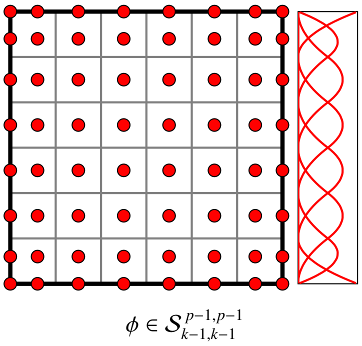


Figure 5: Illustration of a L^2 conforming IGA space for a 2D discretization with 6×6 elements, polynomial order of degree 2, and continuity degree one, considering that $p = 3$ and $k = 2$. Red circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton.

The L^2 conforming discrete spaces satisfy the following relationship in 2D and 3D, respectively.

$$\mathcal{S}^{p-1,\,p-1}_{k-1,\,k-1} \equiv \left\{ \nabla v : v \in \mathcal{S}^{p,\,p}_{k,\,k} \right\} \text{ in 2D,} \qquad \mathcal{S}^{p-1,\,p-1,\,p-1}_{k-1,\,k-1,\,k-1} \equiv \left\{ \nabla v : v \in \mathcal{S}^{p,\,p,\,p}_{k,\,k,\,k} \right\} \text{ in 3D,}$$

where $\mathcal{S}^{p,\,p}_{k,\,k}$ and $\mathcal{S}^{p,\,p,\,p}_{k,\,k,\,k}$ are smooth grad-conforming discrete spaces.

We define the integral-conforming IGA discrete spaces in the parametric domain ($\hat{\Omega} \subset \mathbb{R}$) as follows

$$\hat{Q}^{\text{int}}_h(\hat{\Omega}) = \begin{cases} \mathcal{S}^{p-1,\,p-1}_{k-1,\,k-1} & \text{2D} \\ \mathcal{S}^{p-1,\,p-1,\,p-1}_{k-1,\,k-1,\,k-1} & \text{3D,} \end{cases} \tag{16}$$

where $p-1$ is the polynomial degree, $k-1$ indicates the basis functions continuity, and $\mathcal{S}$ with indexes is a spline space.

In this case, we use an integral-preserving transformation to define the discrete space on the physical domain. The transformation is

$$\iota^{\text{int}}_{\phi}(q) = \det(D\mathbf{F})(q \circ \mathbf{F}), \qquad q \in L^2(\Omega) \tag{17}$$

where $\mathbf{F}$ is the geometric mapping from the parametric domain to the physical domain, $D\mathbf{F}$ is the gradient of the mapping, and $\det(D\mathbf{F})$ is its determinant. Thus, the discrete space in the physical domain is defined as

$$Q^{\text{int}}_h := \left\{ q \in L^2(\Omega) : \iota^{int}_{\phi}(q) \in \hat{Q}^{\text{int}}_h(\hat{\Omega}) \right\}. \tag{18}$$

2.3. Computational complexity

We consider a direct multifrontal method to solve the algebraic systems that numerically represent the problems governed by PDEs. The multifrontal solver partitions the domain into macro-elements interconnected by separators. In multi-field cases, the size of the macro-elements and separators is equal to the sum of the unknowns that each of those subdomains contains in each field. For instance, when solving an electromagnetic problem on a 2D square domain, the subdomains contain the unknowns of both components of the vectorial electric/magnetic field. In scalar problems (e.g., heat transfer), the subdomains contain the unknowns of the scalar-valued field. The elimination of the degrees of freedom on the separators dominates the cost of the LU factorization [4, 5].

2.3.1. H^1 conforming

Figure 6 illustrates the vertical and horizontal separators that partition a 2D square mesh composed by 6 × 6 elements into four subdomains, and shows the size of the vertical separator (number of degrees of freedom that it contains) when using a H^1-conforming IGA space.

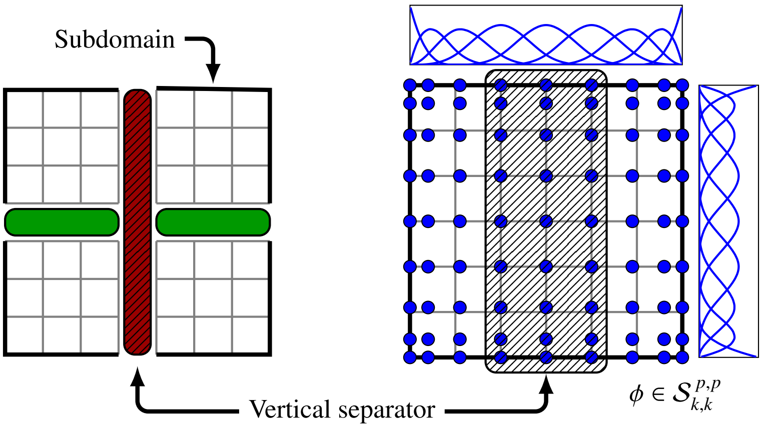


Figure 6: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The H^1 conforming IGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$.

In 2D, the size of the separators at the first and second partition (vertical and horizontal cut) is given by

$$\begin{aligned} q_{sep}|_y &= \mathcal{O}\left(n_\phi|_y(k+1)\right), \qquad \text{(vertical separator)} \\ q_{sep}|_x &= \mathcal{O}\left(n_\phi|_x(k+1)\right), \qquad \text{(horizontal separators)} \end{aligned} \tag{19}$$

where $n_\phi|_y = (n_{\text{elem}} + p)$ and $n_\phi|_x \approx 0.5(n_{\text{elem}} + p)$ are the lengths of the separators, and $(k+1)$ represents the separators thickness. Therefore,

$$q_{sep}|_y = \mathcal{O}\left((n_{\text{elem}} + p)(k+1)\right), \qquad q_{sep}|_x = \mathcal{O}\left(0.5(n_{\text{elem}} + p)(k+1)\right). \tag{20}$$

The size of the separators in *all* partition levels can be computed based on the previous mathematical expressions (20) by following a recursive partition of the domain. Then, the size of the separators at the i-th partition level is

$$q_{sep}|_y = \mathcal{O}\left(2^{-(i-1)}\left((n_{\text{elem}} + p)(k+1)\right)\right), \qquad q_{sep}|_x = \mathcal{O}\left(2^{-i}\left((n_{\text{elem}} + p)(k+1)\right)\right), \tag{21}$$

and the cost to solve the LU factorization in 2D is given by

$$\begin{aligned} \theta^{H^1}_{\text{IGA}}|_{2\text{D}} &= \mathcal{O}\left(\sum_{i=1}^{\ell} 2^{2(i-1)}\left(2^{-(i-1)}\left((n_{\text{elem}} + p)(k+1)\right)\right)^3 + 2^{2(i-1)+1}\left(2^{-i}\left((n_{\text{elem}} + p)(k+1)\right)\right)^3\right), \\ &= \mathcal{O}\left(\left((n_{\text{elem}} + p)(k+1)\right)^3\right), \end{aligned} \tag{22}$$

where ℓ is the number of partition levels. We follow the same procedure that we use to estimate the cost of LU factorization in 2D to build the estimate for the 3D case. The cost in this case is

$$\theta^{H^1}_{\text{IGA}}|_{3\text{D}} = \mathcal{O}\left(\left((n_{\text{elem}} + p)^2(k+1)\right)^3\right). \tag{23}$$

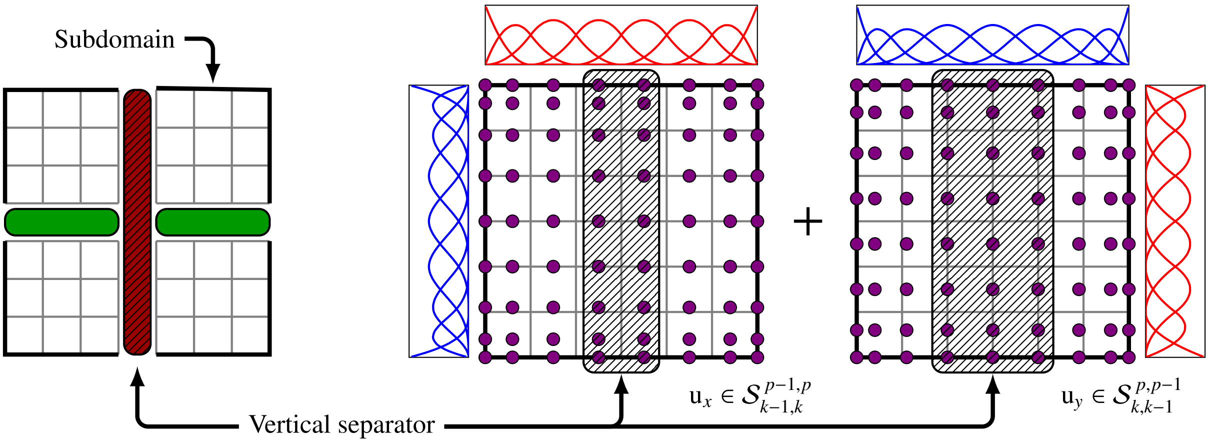

Subdomain
+
$u_x \in \mathcal{S}^{p-1,p}_{k-1,k}$
$u_y \in \mathcal{S}^{p,p-1}_{k,k-1}$
Vertical separator

Figure 7: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The $\boldsymbol{H}$(curl) conforming IGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$.

2.3.2. *$\boldsymbol{H}$(curl) conforming*

Figure 7 illustrates the vertical and horizontal separators that partition a 2D square mesh composed by 6 × 6 elements into four subdomains, and shows the size of the vertical separator for the case in which we use curl-conforming IGA spaces.

The size of the separators depends on the number of fields used to build the curl-conforming discrete spaces. In appendix A, we estimate the size of the separators for any partition level. The cost to solve the LU factorization in 2D is then

$$\theta^{\boldsymbol{H}(\mathrm{curl})}_{\mathrm{IGA}}|_{\mathrm{2D}} = O\left(\left((n_{\mathrm{elem}} + p)(2k+1)\right)^3\right), \tag{24}$$

and in 3D, the cost is

$$\theta^{\boldsymbol{H}(\mathrm{curl})}_{\mathrm{IGA}}|_{\mathrm{3D}} = O\left(\left((n_{\mathrm{elem}} + p)^2(3k+2)\right)^3\right). \tag{25}$$

2.3.3. *$\boldsymbol{H}$(div) conforming*

Figure 8 illustrates the vertical and horizontal separators that partition a 2D square mesh composed by 6 × 6 elements into four subdomains, and shows the size of the vertical separator for the case in which we use divergence-conforming IGA spaces.

These separators contain unknowns of the fields corresponding to the $\boldsymbol{H}$(div) conforming spaces. We present the size of the separators for this case in appendix B. In 2D, the cost to solve the LU factorization discretized using these spaces is

$$\theta^{\boldsymbol{H}(\mathrm{div})}_{\mathrm{IGA}}|_{\mathrm{2D}} = O\left(\left((n_{\mathrm{elem}} + p)(2k+1)\right)^3\right). \tag{26}$$

In 3D, the cost is

$$\theta^{\boldsymbol{H}(\mathrm{div})}_{\mathrm{IGA}}|_{\mathrm{3D}} = O\left(\left((n_{\mathrm{elem}} + p)^2(3k+1)\right)^3\right). \tag{27}$$

2.3.4. *L^2 conforming*

Figure 9 illustrates the vertical and horizontal separators that partition a 2D square mesh composed by 6 × 6 elements into four subdomains, and shows the size of the vertical separator when using an integral-conforming IGA space.

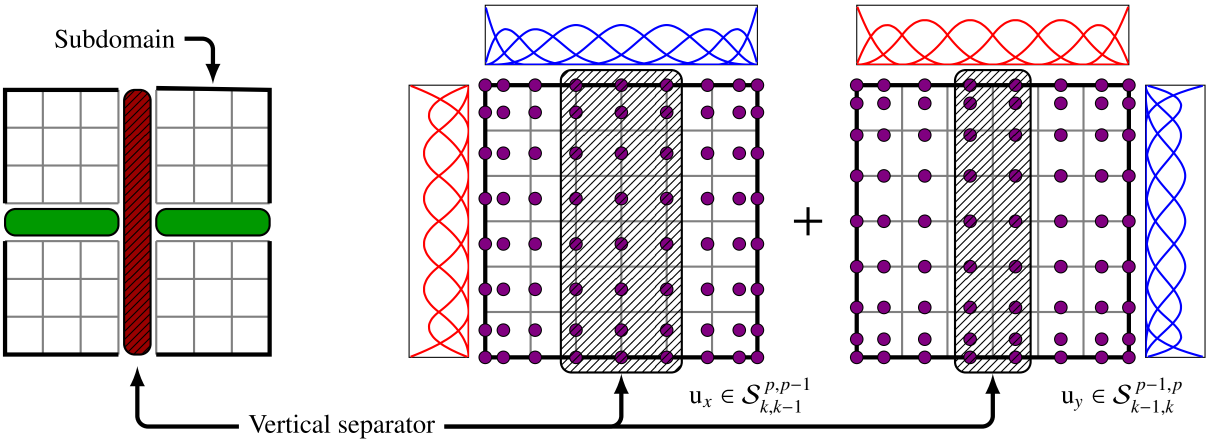

Subdomain
+
$u_x \in \mathcal{S}^{p,p-1}_{k,k-1}$
$u_y \in \mathcal{S}^{p-1,p}_{k-1,k}$
Vertical separator

Figure 8: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The $\boldsymbol{H}$(div) conforming IGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$.

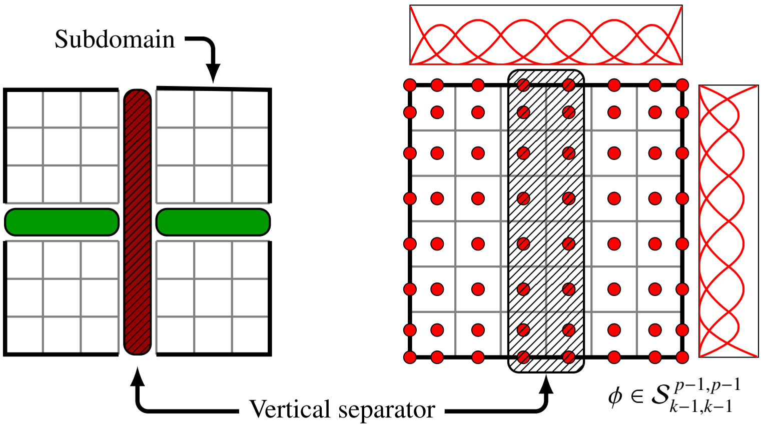


Figure 9: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The L^2 conforming IGA space has a polynomial order of degree 2, and continuity degree one, considering that $p = 3$ and $k = 2$.

These separators contain field unknowns corresponding to the L^2 conforming discrete IGA space. In appendix C, we present the size of the separators for this case. The cost to solve the LU factorization discretized using this spaces in 2D is

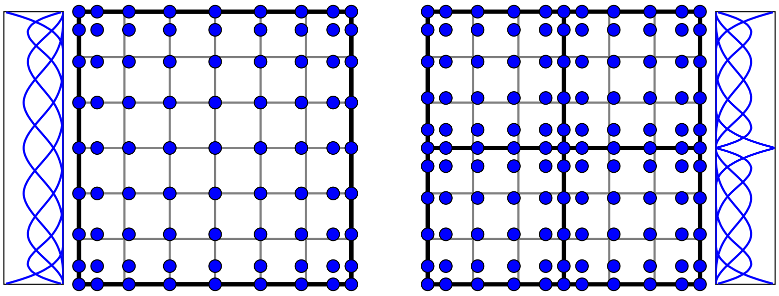
$$\theta^{L^2}_{\text{IGA}}|_{\text{2D}} = \mathcal{O}\left(\left((n_{\text{elem}} + p)(k)\right)^3\right). \tag{28}$$

In 3D, the cost is

$$\theta^{L^2}_{\text{IGA}}|_{\text{3D}} = \mathcal{O}\left(\left((n_{\text{elem}} + p)^2(k)\right)^3\right). \tag{29}$$

3. Refined Isogeometric Analysis discretizations for direct solvers

The refined isogeometric analysis (rIGA) is a discretization technique first presented by Garcia et al. in [7]. This strategy optimizes the direct solver performance while using highly continuous discretizations and preserves the optimal convergence order of the method with respect to a fixed number of elements in the domain. rIGA reduces the continuity over specific inter-element boundaries that correspond with the location of the separators at different partition levels. Figure 10 illustrates a classical C^{p-1} IGA discretization and a rIGA discretization based on a gradient-conforming space on a single field.


(a) C^{p-1} IGA

(b) rIGA

Figure 10: Illustration of an IGA and a rIGA space. We show the spaces of a 2D discretization with a mesh size of 6 × 6 elements and basis functions of order $p = 3$. Blue circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton. Bold lines represent C^0 continuity.

For multi-field problems discretized using gradient- or integral-conforming spaces, rIGA uses C^0-hyperplanes to reduce the continuity over certain inter-element boundaries. Specifically, we reduce the continuity to zero across the interface between the subdomains (macro-elements) that result from the recursive partitioning of the mesh performed during the elimination process. For problems discretized using curl- or divergence-conforming spaces, rIGA uses a combination of C^1- and C^0-hyperplanes to reduce the continuity in k degrees across the interface between the subdomains resulting from the recursive partition. 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 (decreases the separator thickness), which improves the performance of the factorization.

For scalar problems, the best discretization provided by rIGA reduces the solution time and the memory requirements and improves the best approximation error with respect to those on maximum continuity (C^{p-1}) IGA. In particular, the solution via a direct solver becomes faster by a factor of approximately p^2 when compared to IGA, and the gains are even larger when compared to FEA, particularly in 3D (see [7]). Moreover, by using an alternative version of rIGA introduced in [8], the speed-up to solve numerical problems is approximately 25% faster.

3.1. *Computational complexity*

In rIGA, the computational cost of the LU factorization consists of the cost of eliminating the degrees of freedom on the separators and on the macro-elements (subdomains that result from the recursive partitioning of the mesh using low continuity separators) [7–9].

3.1.1. *H^1 conforming*

The continuity reduction modifies the separators size, reducing their thickness and increasing their length. Figure 11 illustrates the vertical separator that interconnects the two subdomains that result from the first partition level once we reduce the continuity over the inter-element boundaries that correspond with the location of the separators at the first and second partition levels.

Considering that the separators correspond to C^0-hyperplanes, the size of the vertical separator at the first partition (vertical cut) and the horizontal separators at the second partition (horizontal cut) is given by

$$\begin{aligned} q_{sep}|_y &= O\left(n_\phi|_y(1)\right), && \text{(vertical separator)} \\ q_{sep}|_x &= O\left(n_\phi|_x(1)\right). && \text{(horizontal separators)} \end{aligned} \tag{30}$$

The length of these separators is $n_\phi|_y = (n_{\text{elem}} + p + (2^i - 1)(k))$ and $n_\phi|_x \approx 0.5(n_{\text{elem}} + p + (2^i - 1)(k))$. The additional term (last term inside the parenthesis) corresponds to the number of unknowns added to the separators length due to

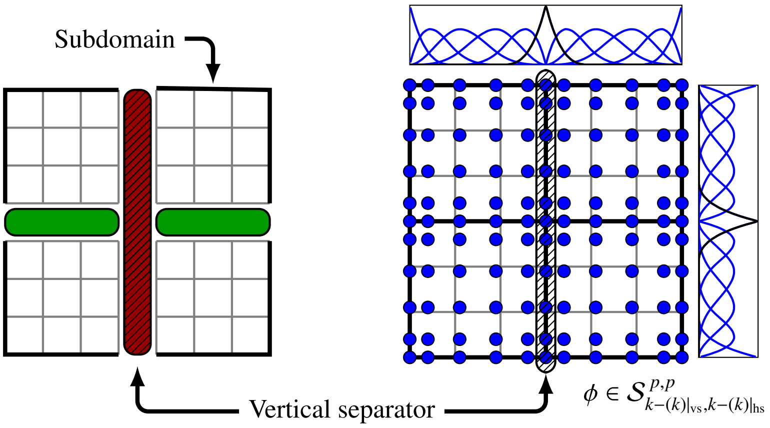

Subdomain
Vertical separator
$\phi \in \mathcal{S}^{p,p}_{k-(k)|_{\mathrm{vs}},k-(k)|_{\mathrm{hs}}}$

Figure 11: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The H^1 conforming rIGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$. The continuity is reduced by k degrees across the macro-element interfaces where the vertical (vs) and horizontal (hs) separators are located on the mesh.

the continuity reduction at i partition levels. Therefore,

$$q_{sep}|_y = \mathcal{O}\Big(\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big), \qquad q_{sep}|_x = \mathcal{O}\Big(0.5\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big), \tag{31}$$

and the size of the separators at the i-th partition level is given by

$$q_{sep}|_y \ = \mathcal{O}\Big(2^{-(i-1)}\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big), \qquad q_{sep}|_x = \mathcal{O}\Big(2^{-i}\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big). \tag{32}$$

The cost to eliminate the degrees of freedom in *all* the separators is

$$\begin{aligned}\theta^{H^1}_{sep}|_{2\text{D}} &= \mathcal{O}\left(\sum_{i=1}^{\ell} 2^{2(i-1)}\Big(2^{-(i-1)}\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big)^3 + 2^{2(i-1)+1}\Big(2^{-i}\big(n_{\text{elem}} + p + (2^i - 1)(k)\big)\Big)^3\right) \\ &= \mathcal{O}\Big(\big(n_{\text{elem}} + p + (2^\ell - 1)(k)\big)^3\Big),\end{aligned} \tag{33}$$

where ℓ is the number of partition levels that involve continuity reduction.

We estimate the cost to eliminate the remaining degrees of freedom, located within the macro-elements, assuming that those are C^{p-1} systems. Then in 2D, the size of the macro-elements is

$$n_{\text{m-e}} = \Big(2^{-\ell}\big(n_{\text{elem}} + p + (2^\ell - 1)(k)\big)\Big)^2, \tag{34}$$

and the cost to eliminate the degrees of freedom in *all* macro-elements is

$$\theta^{H^1}_{\text{m-e}}|_{2\text{D}} = 2^{2\ell}\mathcal{O}\Big(2^{-3\ell}\left(\big(n_{\text{elem}} + p + (2^\ell - 1)(k)\big)(k+1)\right)^3\Big), \tag{35}$$

where $2^{2\ell}$ is the number of macro-elements in a 2D mesh.

The total cost of the LU factorization when using rIGA to build a gradient-conforming discretization in 2D is

$$\begin{aligned}\theta^{H^1}_{\text{rIGA}}|_{2\text{D}} &= \theta^{H^1}_{\text{m-e}}|_{2\text{D}} + \theta^{H^1}_{sep}|_{2\text{D}} \\ \theta^{H^1}_{\text{rIGA}}|_{2\text{D}} &= 2^{-\ell}\mathcal{O}\Big(\big((n_{\text{elem}} + p + (2^\ell - 1)(k))(k+1)\big)^3\Big) + \mathcal{O}\Big(\big(n_{\text{elem}} + p + (2^\ell - 1)(k)\big)^3\Big),\end{aligned} \tag{36}$$

and the cost in 3D is

$$\begin{aligned}\theta^{H^1}_{\text{rIGA}}|_{3\text{D}} &= \theta^{H^1}_{\text{m-e}}|_{3\text{D}} + \theta^{H^1}_{sep}|_{3\text{D}} \\ \theta^{H^1}_{\text{rIGA}}|_{3\text{D}} &= 2^{-3\ell}\mathcal{O}\Big(\big((n_{\text{elem}} + p + (2^\ell - 1)(k))^2(k+1)\big)^3\Big) + \mathcal{O}\Big(\ \big((n_{\text{elem}} + p + (2^\ell - 1)(k))^2\big)^3\ \Big).\end{aligned} \tag{37}$$

3.1.2. $\boldsymbol{H}$(*curl*) *conforming*

The size of a separator is equal to the sum of the unknowns it contains in each of the components of the vectorial field corresponding to the set of curl-conforming discrete spaces (Figure 12).

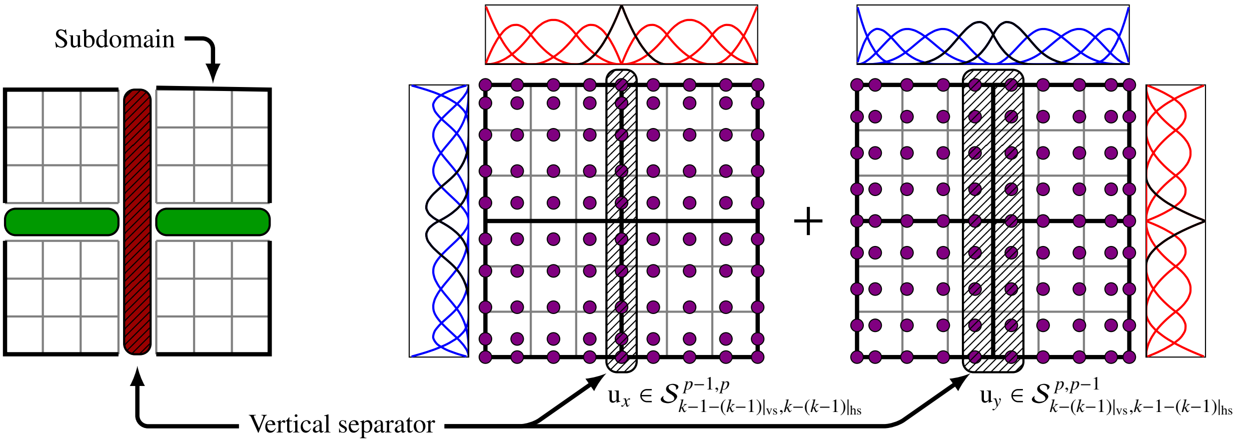

Subdomain
Vertical separator
+
$\mathrm{u}_x \in \mathcal{S}^{p-1,p}_{k-1-(k-1)|_{\mathrm{vs}},k-(k-1)|_{\mathrm{hs}}}$
$\mathrm{u}_y \in \mathcal{S}^{p,p-1}_{k-(k-1)|_{\mathrm{vs}},k-1-(k-1)|_{\mathrm{hs}}}$

Figure 12: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The $\boldsymbol{H}$(curl) conforming rIGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$. The continuity is reduced by $k - 1$ degrees across the macro-element interfaces where the vertical (vs) and horizontal (hs) separators are located on the mesh.

The vertical separators correspond to C^0- and C^1-hyperplanes, while the horizontal separators correspond to C^1- and C^0-hyperplanes in the fields u_x and u_y, respectively. Appendix D presents the separators size.

The cost to eliminate the degrees of freedom in *all* the separators is

$$\theta_{sep}^{\boldsymbol{H}(\mathrm{curl})}|_{\mathrm{2D}} = \mathcal{O}\left(\left(n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1)\right)^3\right), \tag{38}$$

where ℓ is the number of partition levels that involve continuity reduction. Further, and considering the macro-elements' size presented in Appendix D, the cost to eliminate the degrees of freedom on the macro-elements becomes

$$\theta_{\text{m-e}}^{\boldsymbol{H}(\mathrm{curl})}|_{\mathrm{2D}} = 2^{2\ell}\mathcal{O}\left(2^{-3\ell}\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))(2k+1)\right)^3\right). \tag{39}$$

The total cost of the LU factorization when using rIGA to build a curl-conforming solution is equal to the sum of the cost to eliminate the degrees of freedom on both separators and the macro-elements. In 2D, this cost is

$$\theta_{\mathrm{rIGA}}^{\boldsymbol{H}(\mathrm{curl})}|_{\mathrm{2D}} = 2^{-\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))(2k+1)\right)^3\right) + \mathcal{O}\left(\left(n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1)\right)^3\right). \tag{40}$$

The cost in 3D is

$$\theta_{\mathrm{rIGA}}^{\boldsymbol{H}(\mathrm{curl})}|_{\mathrm{3D}} = 2^{-3\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))^2(3k+2)\right)^3\right) + \mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))^2\right)^3\right). \tag{41}$$

3.1.3. $\boldsymbol{H}$(*div*) *conforming*

The separators contain unknowns of each of the components of the vectorial field corresponding to the divergence-conforming space. Figure 13 illustrates the size of the vertical separator that divides the domain into two subdomains at the first partition level.

In appendix E, we present the separators size for *all* partition levels that involve continuity reduction. The cost to eliminate the separators degrees of freedom is given by

$$\theta_{sep}^{\boldsymbol{H}(\mathrm{div})}|_{\mathrm{2D}} = \mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))\right)^3\right), \tag{42}$$

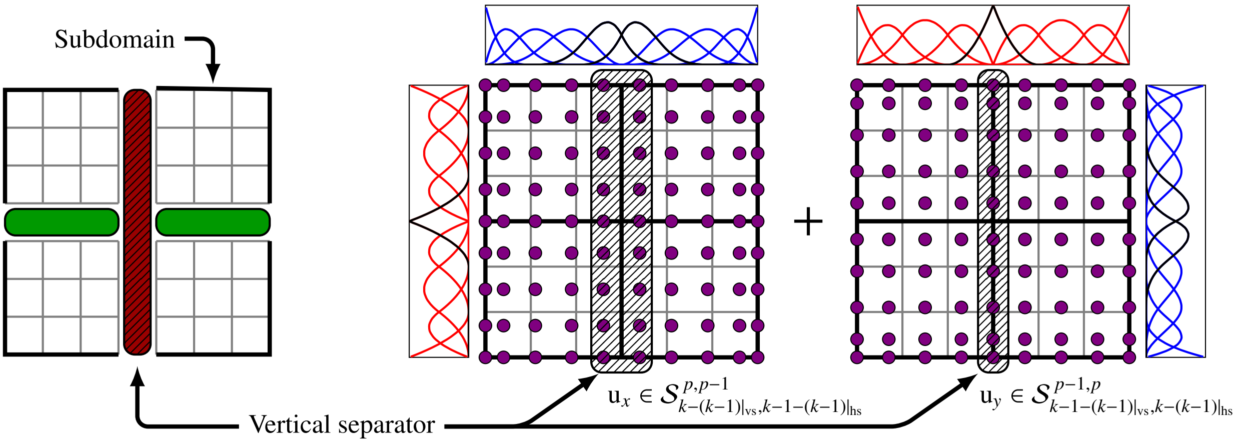

Subdomain
+
$\mathrm{u}_x \in \mathcal{S}^{p,p-1}_{k-(k-1)|_{\mathrm{vs}},k-1-(k-1)|_{\mathrm{hs}}}$
$\mathrm{u}_y \in \mathcal{S}^{p-1,p}_{k-1-(k-1)|_{\mathrm{vs}},k-(k-1)|_{\mathrm{hs}}}$
Vertical separator

Figure 13: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The $\boldsymbol{H}(\mathrm{div})$ conforming rIGA space has a polynomial order $p = 3$ and a continuity degree $k = 2$. The continuity is reduced by k degrees across the macro-element interfaces where the vertical (vs) and horizontal (hs) separators are located on the mesh.

where ℓ is the number of partition levels that involve continuity reduction.

The cost to eliminate the degrees of freedom on the macro-elements is computed based on the macro-elements size presented in appendix E. The cost to eliminate the degrees of freedom in *all* macro-elements is

$$\theta^{\boldsymbol{H}(\mathrm{div})}_{\text{m-e}}|_{2\mathrm{D}} = 2^{2\ell}\mathcal{O}\left(2^{-3\ell}\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))(2k+1)\right)^3\right). \tag{43}$$

Then, the total cost of the LU factorization when using rIGA to build a discretization using $\boldsymbol{H}(\mathrm{div})$ conforming spaces in 2D is

$$\theta^{\boldsymbol{H}(\mathrm{div})}_{\mathrm{rIGA}}|_{2\mathrm{D}} = 2^{-\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))(2k+1)\right)^3\right) + \mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))\right)^3\right). \tag{44}$$

In 3D, the LU factorizaton cost is

$$\theta^{\boldsymbol{H}(\mathrm{div})}_{\mathrm{rIGA}}|_{3\mathrm{D}} = 2^{-3\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))^2(3k+1)\right)^3\right) + \mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))^2\right)^3\right). \tag{45}$$

3.1.4. L^2 conforming

Figure 14 illustrates the size of the vertical separator that divides the domain into two subdomains at the first partition level. In this case, the sepators contain unknowns of one scalar field that corresponds to an integral-conforming space.

Appendix F shows the size of the separators for any partition level that involve continuity reduction when using an integral-conforming rIGA space, and the cost to eliminate the degrees of freedom in *all* separators is

$$\theta^{L^2}_{sep}|_{2\mathrm{D}} = \mathcal{O}\left(\left(n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1)\right)^3\right), \tag{46}$$

where ℓ is the number of partition levels that involve continuity reduction.

Considering the macro-elements size presented in appendix F, the cost to eliminate the degrees of freedom in *all* macro-elements is

$$\theta^{L^2}_{\text{m-e}}|_{2\mathrm{D}} = 2^{2\ell}\mathcal{O}\left(2^{-3\ell}\left((n_{\mathrm{elem}} + p + (2^\ell - 1)(k-1))(k)\right)^3\right), \tag{47}$$

where $2^{2\ell}$ is the number of macro-elements in a 2D mesh.

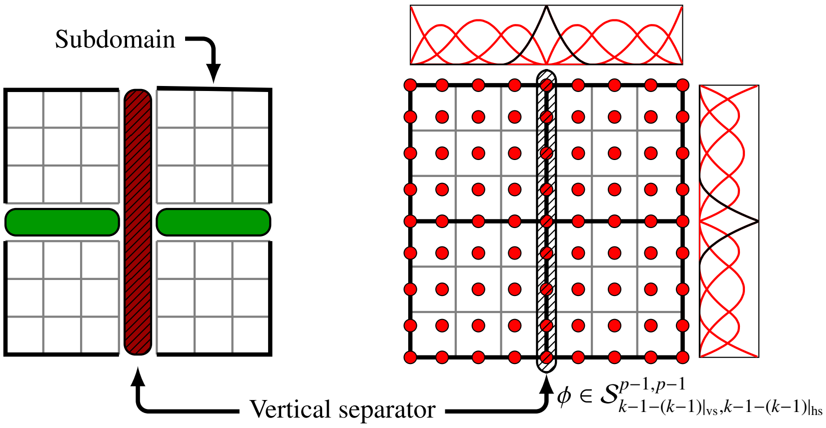

Subdomain
Vertical separator
$\phi \in \mathcal{S}^{p-1,p-1}_{k-1-(k-1)|_{\mathrm{vs}},k-1-(k-1)|_{\mathrm{hs}}}$

Figure 14: Illustration of the size of the vertical separator that splits a 2D mesh of 6 × 6 elements into two symmetric subdomains. The L^2 conforming rIGA space has a polynomial order of degree 2, and continuity degree one, considering that $p = 3$ and $k = 2$. The continuity is reduced by k degrees across the macro-element interfaces where the vertical (vs) and horizontal (hs) separators are located on the mesh.

The total cost of the LU factorization when using rIGA to build an integral-conforming discretization in 2D is

$$\theta^{L^2}_{\mathrm{rIGA}}|_{2\mathrm{D}} = 2^{-\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^{\ell} - 1)(k-1))(k)\right)^3\right) + \mathcal{O}\left(\left(n_{\mathrm{elem}} + p + (2^{\ell} - 1)(k-1)\right)^3\right), \tag{48}$$

and the cost in 3D is

$$\theta^{L^2}_{\mathrm{rIGA}}|_{3\mathrm{D}} = 2^{-3\ell}\mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^{\ell} - 1)(k-1))^2(k)\right)^3\right) + \mathcal{O}\left(\left((n_{\mathrm{elem}} + p + (2^{\ell} - 1)(k-1))^2\right)^3\right). \tag{49}$$

3.1.5. *Theoretical reduction factors*

Table 1 shows the theoretical reduction factors obtained when using rIGA method to discretize the problems as opposed to highly-continuous (C^{p-1}) IGA. Once we achieve the asymptotic regime (large enough mesh sizes), the optimal reduction factor delivers a reduction factor in the number of FLOPs of approximately p^2.

4. Implementation details

We implement the model problem using PetIGA-MF, a multi-field extension of PetIGA that we describe in [19]. This framework allows the use of a different space for each field of interest. For instance, we can use $\boldsymbol{H}$(curl) discretizations to solve electromagnetics problems or $\boldsymbol{H}$(div) spaces to solve incompressible flows. PetIGA-MF generalizes the implementation of PetIGA [20] which is based on PETSc [22, 23] and uses data management libraries to pack the data of the multiple fields in a single object, thus simplifying the discretization construction.

We solve the system that results from the discretization of the problem using the sequential version of the multi-frontal solver MUMPS [24, 25] with the automatic choice of partitioning technique, resulting in METIS [26] algorithm for all the studied cases.

The standard PETSc installation uses by default 32-bit indices. This configuration fails once the system matrix of the problem contains more than $2^{31} - 1$ (approximately 1.6e+9) non-zero entries on a single process. This limit is independent of the available physical memory of the machine. By setting up PETSc with 64-bit indices, the limit in the number of non-zero entries grows. Nevertheless, since MUMPS does not support 64-bit indices, we cannot use this package to perform the matrix factorization. PARDISO [27, 28] is a sparse direct solver package that supports 64-bit indices. This package uses a combination of supernode techniques [29] to perform the matrix factorizations. Assuming that the performance of PARDISO is approximately the same as that of MUMPS, we use this package to provide the computational cost for the cases we cannot solve using PETSc+MUMPS.

We report the global FLOPs and computational times (in seconds). This allows us to analyze the impact of the local reduction of continuity in the computational cost. For each of the mesh sizes and polynomial degrees that we use

Discretization	2D					3D				
	n_{elem}	Polynomial degree (p)				n_{elem}	Polynomial degree (p)			
		3	4	5	6		3	4	5	6
Gradient-conforming (H^1)	512	10.74	13.38	13.79	13.11	64	11.48	15.30	16.26	15.46
	1024	13.69	19.74	22.51	22.79	128	14.68	23.50	31.98	37.80
	2048	16.13	26.00	33.22	36.64	256	18.74	35.76	54.60	71.69
Curl-conforming ($\boldsymbol{H}$(curl))	512	3.50	6.67	9.19	10.51	64	3.01	5.57	7.92	9.49
	1024	3.81	7.84	11.68	14.76	128	3.41	7.09	11.17	14.70
	2048	3.98	9.00	14.81	19.99	256	3.65	8.07	13.50	19.89
Divergence-conforming ($\boldsymbol{H}$(div))	512	3.50	6.67	9.19	10.51	64	3.86	7.64	10.77	12.4
	1024	3.81	7.84	11.68	14.76	128	4.38	9.74	15.23	19.25
	2048	3.98	9.00	14.81	19.99	256	4.69	11.1	20.26	30.98
Integral-conforming (L^2)	512	5.54	10.74	13.38	13.79	64	5.52	11.48	15.30	16.26
	1024	6.05	13.69	19.74	22.51	128	6.29	14.68	23.50	31.98
	2048	6.60	16.13	26.00	33.22	256	6.73	18.74	35.76	54.60

Table 1: Theoretical reduction factors obtained when using rIGA method to build the solution spaces. These factors correspond to the ratio between the C^{p-1} IGA and the optimal rIGA solutions.

to discretize the model problem, we consider a range of cases with a particular number of partition levels that locally reduce the continuity.

All computational tests are solved sequentially in TACC systems. We use the Stampede supercomputer to solve the Stoke flow problems and Stampede2 supercomputer for the electromagnetic problems. In Stampede, we use nodes outfitted with 2.7 GHz cores and 1TB of memory. In Stampede2, we use nodes equipped with turbo boost 2.1 GHz cores (up to peak 3.7 GHz depending on instruction set and number of active cores) and 192GB of memory.

5. Numerical Results

We analyze the performance of the direct solvers when using rIGA discretizations to solve multi-field problems. For this, we consider an electromagnetic problem solved using $\boldsymbol{H}$(curl) and H^1 conforming spaces, and an incompressible fluid flow problem solved with a set of $\boldsymbol{H}$(div) × L^2 conforming spaces.

5.1. $\boldsymbol{H}$(curl) and H^1 conforming spaces

We first study the impact of the continuity reduction on $\boldsymbol{H}$(curl) and H^1 conforming spaces. The model problems use in this case are based on the time-harmonic Maxwell and Helmholtz equations, in 2D. The Maxwell problem formulation is

$$\begin{cases} \text{Find } \mathbf{E}\,,\ \text{with } \mathbf{E} : \Omega \to \mathbb{R}^d,\ \text{such that:} & \\ \quad \nabla \times (\tilde{\mu}^{-1} \nabla \times \mathbf{E}) + \tilde{\sigma}\,\mathbf{E} = -\nabla \times \mathbf{M} & \text{in } \Omega, \\ \qquad\qquad\qquad\qquad \mathbf{E} \times \mathbf{n} = \mathbf{0} & \text{in } \partial\Omega, \end{cases} \tag{50}$$

where Ω is the region of study, $\mathbf{E}$ is the electric field, $\mathbf{M} = (0, 0, \mathrm{M}_H)$ is the time-harmonic magnetic dipole source, $\mathbf{n}$ is the outward unit normal vector to the domain boundary $\partial\Omega$, $\tilde{\mu} = i\omega\mu$, and $\tilde{\sigma} = \sigma + i\omega\varepsilon$. Additionally, ε is the electric permittivity, σ is the electrical conductivity, μ is the magnetic permeability, and ω is the angular frequency.

A weak formulation of the problem is

$$\begin{cases} \text{Find } \mathbf{E}\,,\ \text{with } \mathbf{E} \in V_0^{\text{curl}},\ \text{such that:} \\ \quad \left(\nabla \times \mathbf{W}, \tilde{\mu}^{-1} \nabla \times \mathbf{E}\right)_\Omega + \left(\mathbf{W}, \tilde{\sigma}\,\mathbf{E}\right)_\Omega = -\left(\nabla \times \mathbf{W}, \mathbf{M}\right)_\Omega \\ \text{for all } \mathbf{W} \in V_0^{\text{curl}}, \end{cases} \tag{51}$$

where $V_0^{\text{curl}} = \{\mathbf{v} \in \boldsymbol{H}(\text{curl};\Omega) : \mathbf{v} \times \mathbf{n} = \mathbf{0} \text{ on } \partial\Omega\}$ corresponds to the trial and test spaces for the electric field, respectively. We denote by $(\cdot,\cdot)_\Omega$ the L^2 inner product on Ω. Now, we introduce a discrete formulation of the Maxwell test problem.

$$\begin{cases} \text{Find } \mathbf{E}\,,\ \text{with } \mathbf{E} \in \mathcal{V}_{0,h}^{\text{curl}},\ \text{such that:} \\ \qquad \left(\nabla \times \mathbf{W}, \tilde{\mu}^{-1} \nabla \times \mathbf{E}\right)_\Omega + \left(\mathbf{W}, \tilde{\sigma}\, \mathbf{E}\right)_\Omega = -\left(\nabla \times \mathbf{W}, \mathbf{M}\right)_\Omega \\ \text{for all } \mathbf{W} \in \mathcal{V}_{0,h}^{\text{curl}}, \end{cases} \tag{52}$$

where $\mathcal{V}_{0,h}^{\text{curl}}$ is computed using (12). In this case, the parametric space reads as

$$\hat{\mathcal{V}}_{0,h}^{\text{curl}} := \left\{\hat{\mathbf{v}} \in \hat{\mathcal{V}}_h^{\text{curl}}(\hat{\Omega}) : \hat{\mathbf{v}} \times \hat{\mathbf{n}} = 0 \text{ on } \partial\hat{\Omega}\right\}.$$

The strong formulation of the Helmholtz problem in terms of the only nonzero Magnetic field component is

$$\begin{cases} \text{Find H}\,,\ \text{with H} : \Omega \to \mathbb{R},\ \text{such that:} \\ \qquad -\tilde{\sigma}^{-1}\nabla^2 \text{H} + \tilde{\mu}\,\text{H} = -\tilde{\mu}\,\text{M}_H & \text{in } \Omega, \\ \qquad\qquad \text{H} = 0 & \text{in } \partial\Omega, \end{cases} \tag{53}$$

where Ω is the region of study, H is the only nonzero magnetic field component, and M_H is the only nonzero component of the time-harmonic magnetic dipole source. The weak formulation of the system problem is

$$\begin{cases} \text{Find H}\,,\ \text{with H} \in V_0^{H^1},\ \text{such that:} \\ \qquad \left(\nabla \text{F}, \tilde{\sigma}^{-1} \nabla \text{H}\right)_\Omega + \left(\text{F}, \tilde{\mu}\,\text{H}\right)_\Omega = -\left(\text{F}, \tilde{\mu}\,\text{M}_H\right)_\Omega \\ \text{for all F} \in V_0^{H^1}, \end{cases} \tag{54}$$

where $V_0^{H^1} = \{\text{v} \in H^1(\Omega) : \text{v} = 0 \text{ on } \partial\Omega\}$ corresponds to the trial and test spaces for the magnetic field, respectively. The Galerkin formulation of the Helmholtz test problem is

$$\begin{cases} \text{Find H}\,,\ \text{with H} \in Q_{0,h}^{\text{grad}},\ \text{such that:} \\ \qquad \left(\nabla \times \text{F}, \tilde{\sigma}^{-1} \nabla \times \text{H}\right)_\Omega + \left(\text{F}, \tilde{\mu}\,\text{H}\right)_\Omega = -\left(\text{F}, \tilde{\mu}\,\text{M}_H\right)_\Omega \\ \text{for all F} \in Q_{0,h}^{\text{grad}}, \end{cases} \tag{55}$$

where $Q_{0,h}^{\text{grad}}$ is computed using (9). In this case, the parametric space reads as

$$\hat{Q}_{0,h}^{\text{grad}} := \left\{\hat{\text{v}} \in \hat{Q}_h^{\text{grad}}(\hat{\Omega}) : \hat{\text{v}} = 0 \text{ on } \partial\hat{\Omega}\right\}.$$

5.1.1. Electro-magnetics problem

The performance of the direct solvers is unrelated to the specific boundary conditions, domain size or physical properties implemented in the test problems. However, in order to obtain an adequate numerical solution on a lossy media ($\sigma > 0$) without boundary-induced errors, we employ a sufficiently large domain so the electric waves vanish before reaching the domain boundaries. In particular, we solve both the electric and magnetic problems over a domain $\Omega = (0, 10)^d$, being d the spatial dimension. We impose homogeneous Dirichlet boundary conditions along the entire domain boundary ($\mathbf{E} \times \mathbf{n} = \mathbf{0}$, $\text{H} = 0$), and a scalar magnetic source

$$\text{M}_H = \begin{cases} 10 & \|\mathbf{x} - \mathbf{x}_0\| <= r \\ 0 & \text{elsewhere}, \end{cases} \tag{56}$$

where r is the radius and $\mathbf{x}_0$ refers to the location of the magnetic source. We consider a magnetic permeability $\mu = 4\pi 10^{-7}$, a free-space electric permittivity $\varepsilon = 8.8510^{-12}$, and an angular frequency $\omega = 4\pi 10^{7}$.

We assume that the electrical conductivity σ is piecewise constant and that it is given by $\sigma = \sigma_j(\mathbf{x})\,\forall\,\mathbf{x} \in \Omega_j$ where $\Omega = \cup_{j=1}^{n}\Omega_j$ and $\cap_{j=1}^{n}\Omega_j = \emptyset$. Figure 15 illustrates the electrical conductivity coefficient distribution. We solve two

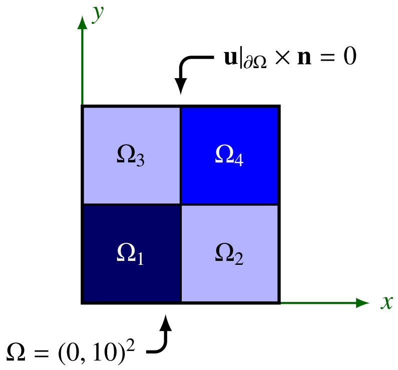


Figure 15: Electrical conductivity coefficient distribution

model problems. The first one involves a homogeneous electrical conductivity coefficient equal to one ($\sigma = 1$) over the entire domain, while in the second model, the electrical conductivity σ is given by

$$\sigma = \begin{cases} \sigma|_{\blacksquare} = 1 \\ \sigma|_{\square} = 10 \\ \sigma|_{\text{🟦}} = 0.1. \end{cases}$$

Figures 16 and 17 illustrate the electric field ($\mathbf{E}$) and the only nonzero magnetic field (H) resulting from imposing the magnetic source (M_H) of radius $r = 0.01$ at $\mathbf{x}_0 = [4.8, 4.8]$. We solve these 2D problems using a mesh size of 1024^2 elements, and polynomial degree $p = 4$.

To ensure that the numerical problems are converging to the correct solution, we compute the L^2-norm of $\nabla \times \mathrm{H} - \tilde{\sigma}\mathbf{E}$, being $\mathbf{E} = (\mathrm{E}_x, \mathrm{E}_y)$ and $\mathrm{H} = (\mathrm{H}_z)$ the only nonzero magnetic field. The numerical results in Table 2 exhibit a slow convergence as we increase the order of approximation. This is a consequence of the discontinuity (lack of regularity) of the source. To recover a fast convergence, we consider the following alternative magnetic source:

$$\mathrm{M}_H = b\, e^{\frac{-\|\mathbf{x} - \mathbf{x}_0\|^2}{2a}} \tag{57}$$

where $b = 50$ and $a = 0.001$. The above source is C^∞. Table 3 presents the L^2-norm of $\nabla \times \mathrm{H} - \tilde{\sigma}\mathbf{E}$. rIGA delivers L^2-norms values that are smaller than the ones obtained with IGA and larger than those from FEM, as expected. In addition, rIGA shows a similar (up to a constant) rate of convergence than IGA in terms of mesh size and polynomial degree. The poor convergence observed for the case with $\sigma = \sigma_j(\mathbf{x})$ is due to the discontinuities of σ.

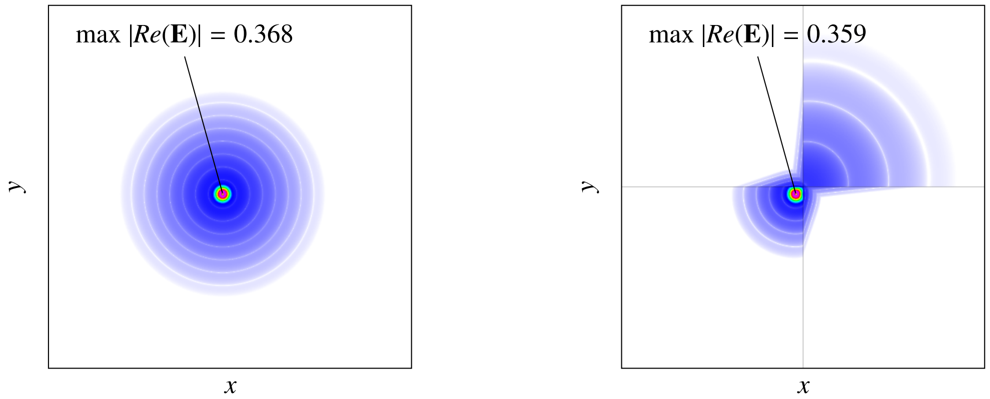
Figures 18 and 19 display the number of FLOPs required to solve the Maxwell and Helmholtz problem in 2D, respectively. The number of FLOPs is plotted with respect to the macro-elements size $s = n_{\text{elem}}/2^\ell$, being ℓ the number of partition levels in which we perform continuity reduction.

We use a constant factor approximately equal to 16 to fit the theoretical estimates (equations (24) and (40)) with the computed number of FLOPs required to solve the 2D Maxwell problem and the 2D Helmholtz problem. This constant includes the contribution of building the Schur complements and the total number of FLOPs that LAPACK performs to factorize the system. The constant factor is slightly dependent on the polynomial degree.

Tables 4 and 5 provide the number of FLOPs and computational times (in seconds) required to solve the 2D Maxwell and 2D Helmholtz equation using IGA and optimal rIGA discretizations, respectively.

The numerical results show that rIGA reduces the number of FLOPs required for solving both Maxwell and Helmholtz problems by a factor up to $O(p^2)$. For instance, when we solve the 2D Maxwell problem using a mesh size of 1024^2 elements and polynomial degree $p = 5$, the optimal rIGA case reports a reduction with respect to C^{p-1} IGA of approximately 12 times in terms of the number of FLOPs. When we solve the 2D Helmholtz problem using the same mesh size and polynomial degree, the reduction factor in the number of FLOPs is 31.

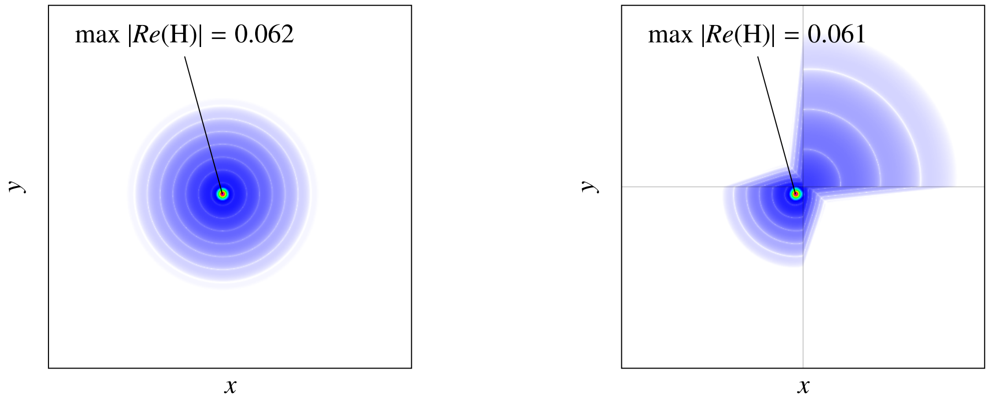
The theoretical estimates explain well the numerical results. In particular, the estimates predict the optimal rIGA case which delivers the maximum reduction cost for both Maxwell and Helmholtz problems.



(a) Homogeneous electrical conductivity coefficient case ($\sigma = 1$)

(b) Discontinuous piecewise constant electrical conductivity coefficient case ($\sigma = \sigma_j(\mathbf{x})$)

Figure 16: Absolute value of the real part of the 2D electric field ($|Re(\mathbf{E})|$) distribution in logarithmic scale. Problem solution approximated over a mesh size of 1024^2 elements and a polynomial degree $p = 5$.



(a) Homogeneous electrical conductivity coefficient case ($\sigma = 1$)

(b) Discontinuous piecewise constant electrical conductivity coefficient case ($\sigma = \sigma_j(\mathbf{x})$)

Figure 17: Absolute value of the real part of the 2D magnetic field ($|Re(\mathrm{H})|$) distribution in logarithmic scale. Problem solution approximated over a mesh size of 1024^2 elements and a polynomial degree $p = 5$.

In terms of computational times, rIGA reports a maximum reduction factor with respect to C^{p-1} IGA of almost 11 times when solving the Maxwell problem, and of almost 20 times when solving the Helmholtz problem, both in 2D, with a mesh size of 1024^2 elements and polynomial degree $p = 5$. The optimal rIGA case required approximately 3 minutes to solve the Maxwell problem, while IGA required 30 minutes. For solving Helmholtz problem, we go from 5 minutes (with IGA) to 15 seconds (with rIGA).

Remark. *Focusing on the case with a smooth magnetic source* (57) *and considering that both Maxwell and Helmholtz problems are solved using a mesh size of* 1024^2 *elements and polynomial degree* $p = 5$*, the* L^2*-norm of* $\nabla \times H - \tilde{\sigma}\mathbf{E}$ *computed with optimal rIGA reports a reduction of* 10% *with respect to* C^{p-1} *IGA for* $\sigma = 1$*, while for* $\sigma = \sigma_j(\mathbf{x})$ *optimal rIGA reports a reduction factor with respect to* C^{p-1} *IGA of two order of magnitud.*

5.2. $\boldsymbol{H}(div) \times L^2$ *conforming spaces*

We now study the impact of the continuity reduction on a set of $\boldsymbol{H}(\mathrm{div}) \times L^2$ conforming spaces. The model problem consists of a linear system of PDEs for the conservation of linear momentum and mass. This boundary value

		$\sigma = 1$			$\sigma = \sigma_j(\mathbf{x})$		
Polynomial degree	Mesh size (N_{elem})	IGA	rIGA	FEM	IGA	rIGA	FEM
3	512^2	2.52e-3	2.52e-3	2.64e-4	2.79e-3	2.55e-3	2.87e-4
	1024^2	1.57e-4	1.53e-4	4.19e-5	3.54e-4	1.80e-4	6.22e-5
4	512^2	1.83e-3	1.83e-3	5.66e-5	2.16e-3	1.85e-3	1.16e-4
	1024^2	1.03e-4	9.86e-5	1.06e-5	3.28e-4	1.17e-4	1.82e-5
5	512^2	1.64e-3	1.58e-3	8.23e-5	1.99e-3	1.60e-3	1.03e-4
	1024^2	6.05e-5	5.42e-5	***	2.66e-4	7.04e-5	***
6	512^2	1.26e-3	1.24e-3	1.82e-5	1.62e-3	1.26e-3	1.07e-5
	1024^2	***	3.38e-5	***	***	4.80e-5	***

Table 2: L^2-norm of $\nabla \times \mathrm{H} - \tilde{\sigma}\mathbf{E}$. Case with the discontinuous magnetic source presented in (56). The asterisks reflect that the computation was not accomplished due to the lack of memory.

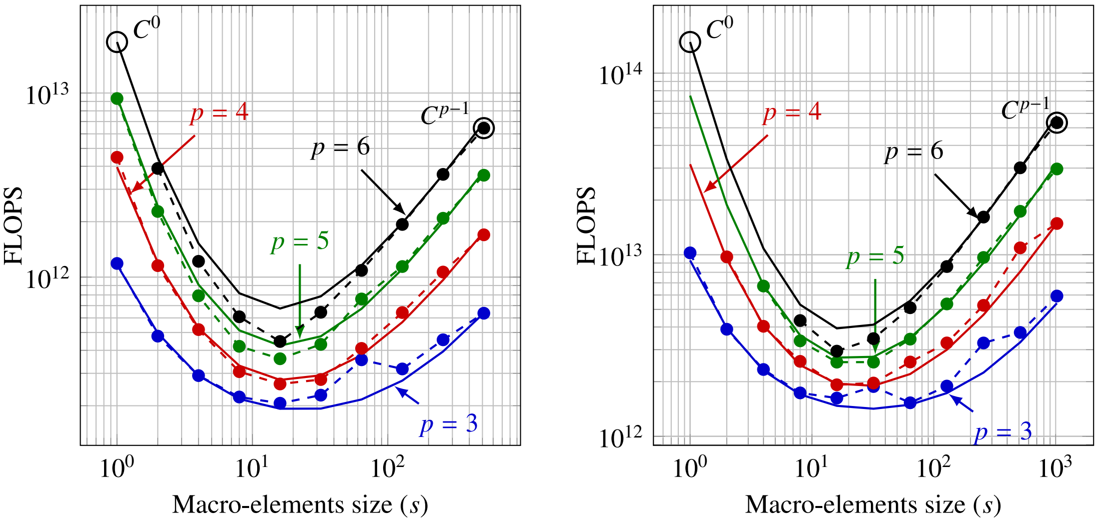
		$\sigma = 1$			$\sigma = \sigma_j(\mathbf{x})$		
Polynomial degree	Mesh size (N_{elem})	IGA	rIGA	FEM	IGA	rIGA	FEM
3	512^2	5.06e-04	4.60e-04	7.41e-05	1.62e-3	4.80e-4	8.97e-5
	1024^2	4.33e-06	4.26e-06	1.29e-06	5.98e-4	1.48e-5	9.78e-6
4	512^2	2.58e-05	1.70e-05	2.79e-07	1.70e-3	3.03e-5	9.58e-6
	1024^2	3.40e-08	2.94e-08	1.05e-09	8.32e-4	7.32e-6	5.17e-6
5	512^2	2.12e-06	1.41e-06	9.06e-10	1.32e-3	1.14e-5	6.35e-6
	1024^2	3.99e-10	3.68e-10	***	7.00e-4	5.55e-6	***
6	512^2	2.55e-07	1.37e-07	2.88e-12	1.57e-3	8.09e-6	4.71e-6
	1024^2	***	5.45e-12	***	***	4.48e-6	***

Table 3: L^2-norm of $\nabla \times \mathrm{H} - \tilde{\sigma}\mathbf{E}$. Case with the smooth magnetic source presented in (57). The asterisks reflect that the computation was not accomplished due to the lack of memory.

problem represents incompressible fluid flows in steady state. We formulate the problem as follows

$$\begin{cases} \text{Find } \{\mathbf{u}, p_e\}, \text{ with } \mathbf{u} : \Omega \to \mathbb{R}^d, \text{ and } p_e : \Omega \to \mathbb{R}, \text{ such that:} & \\ \qquad -\nabla \cdot \sigma(\mathbf{u}, p_e) = \mathbf{f} & \text{in } \Omega, \\ \qquad \nabla \cdot \mathbf{u} = \mathbf{0} & \text{in } \Omega, \\ \qquad \mathbf{u} = \mathbf{g} & \text{on } \partial\Omega, \end{cases} \tag{58}$$

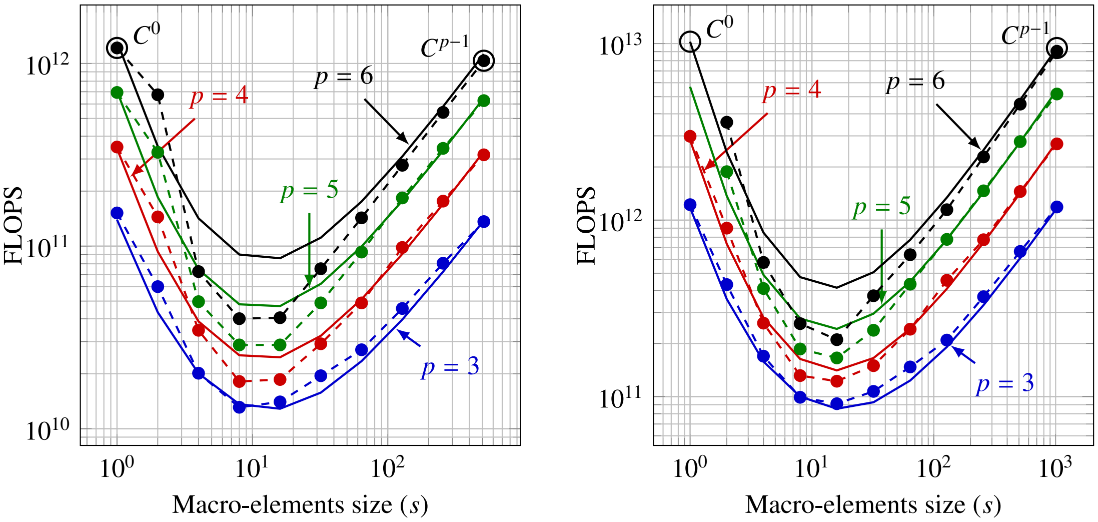
where $\Omega = (0,1)^d$ is the region occuped by the fluid, $\mathbf{u}$ is the fluid velocity field, p_e is the pressure field, $\mathbf{f}$ is the external volumetric forces acting on the fluid, and $\mathbf{g}$ is the fluid velocity at the boundary. Additionally, $\sigma(\mathbf{u}, p_e) = -p_e\mathbf{I} + 2\nu\nabla^s\mathbf{u}$ is the Cauchy stress tensor for incompressible Newtonian fluids, where $\mathbf{I}$ refers to the identity matrix, ν is the kinematic viscosity, and $\nabla^s\mathbf{u}$ is the symmetric part of the velocity gradient (strain rate).



(a) Mesh size of 512^2 elements

(b) Mesh size of 1024^2 elements

Figure 18: Number of FLOPs required to solve the 2D Maxwell equation with the multifrontal direct solver. The dashed lines with rounded markers correspond to the numerical results and the solid lines represent the theoretical estimates.



(a) Mesh size of 512^2 elements

(b) Mesh size of 1024^2 elements

Figure 19: Number of FLOPs required to solve the 2D Helmholtz problem with the multifrontal direct solver. The dashed lines with rounded markers correspond to the numerical results and the solid lines represent the theoretical estimates.

The weak formulation of the system is

$$\begin{cases} \text{Find } \{\mathbf{u}, p_e\}\,, \text{ with } \mathbf{u} \in V_g^{\boldsymbol{H}^1}, \text{ and } p_e \in Q_0, \text{ such that:} \\ \qquad (\nabla^s \mathbf{w}, 2\nu \nabla^s \mathbf{u})_\Omega - (\nabla \cdot \mathbf{w}, p_e)_\Omega + (q_e, \nabla \cdot \mathbf{u})_\Omega = (\mathbf{w}, \mathbf{f})_\Omega \\ \text{for all } \mathbf{w} \in V_0^{\boldsymbol{H}^1}, \text{ and } q_e \in Q_0, \end{cases} \tag{59}$$

where $V_g^{\boldsymbol{H}^1} = \{\mathbf{v} \in \boldsymbol{H}^1(\Omega) : \mathbf{v} = \mathbf{g} \text{ on } \partial\Omega\}$ and $V_0^{\boldsymbol{H}^1} = \{\mathbf{v} \in \boldsymbol{H}^1(\Omega) : \mathbf{v} = 0 \text{ on } \partial\Omega\}$ are the trial and test spaces for the velocity field, respectively. The trial and test spaces for the pressure field correspond to $Q_0 = L_0^2(\Omega) \subset L^2(\Omega)$ that is

Polynomial degree	Method	$N_{\text{elem}} = 512^2$		$N_{\text{elem}} = 1024^2$	
		FLOPs	Time [s]	FLOPs	Time [s]
3	IGA	6.362e+11	37.221	5.924e+12	314.729
	rIGA	2.063e+11	14.920	1.624e+12	95.222
	IGA/rIGA	3.08	2.5	3.64	3.30
4	IGA	1.697e+12	91.853	1.486e+13	947.220
	rIGA	2.626e+11	18.447	1.922e+12	115.483
	IGA/rIGA	6.46	4.98	7.73	8.20
5	IGA	3.576e+12	211.684	2.967e+13	1670.075
	rIGA	3.601e+11	25.638	2.562e+12	151.575
	IGA/rIGA	9.93	8.25	11.58	11.02
6	IGA	6.445e+12	316.894	5.334e+13	***
	rIGA	4.468e+11	30.596	2.944e+12	196.624
	IGA/rIGA	14.42	10.36	18.11	—

Table 4: Number of FLOPs and computational times (in seconds) required to solve the 2D Maxwell problem with the multifrontal direct solver using two mesh sizes and four polynomial degrees (p ranging from 3 to 6).

Polynomial degree	Method	$N_{\text{elem}} = 512^2$		$N_{\text{elem}} = 1024^2$	
		FLOPs	Time [s]	FLOPs	Time [s]
3	IGA	1.362e+11	8.801	1.187e+12	66.899
	rIGA	1.311e+10	1.713	9.121e+10	8.841
	IGA/rIGA	10.38	5.14	13.01	7.56
4	IGA	3.161e+11	19.172	2.703e+12	145.458
	rIGA	1.861e+10	2.275	1.223e+11	11.739
	IGA/rIGA	16.98	8.42	22.10	12.39
5	IGA	6.259e+11	47.657	5.181e+12	317.254
	rIGA	2.877e+10	3.202	1.657e+11	15.647
	IGA/rIGA	21.67	14.88	31.27	20.27
6	IGA	1.035e+12	54.598	9.049e+12	486.781
	rIGA	4.054e+10	4.307	2.102e+11	19.924
	IGA/rIGA	25.53	12.68	43.09	24.43

Table 5: Number of FLOPs and computational times (in seconds) required to solve the 2D Helmholtz problem with the multifrontal direct solver using two mesh sizes and four polynomial degrees (p ranging from 3 to 6).

a space with zero average on Ω. In here, $(\cdot,\cdot)_\Omega$ denotes the L^2 inner product on Ω. The weak problem has a unique solution $(\mathbf{u}, p_e) \in V_g \times Q_0$.

Since the discretization spaces used in this case are designed for $\boldsymbol{H}(\text{div})$ problems, we can strongly set the normal component of the velocity (g_n) at the boundary. The parametric spaces, in this case are

$$\hat{\mathcal{V}}^{\text{div}}_{g_n,h} := \left\{\hat{\mathbf{v}} \in \hat{\mathcal{V}}^{\text{div}}_h(\hat{\Omega}) : \hat{\mathbf{v}} \cdot \hat{\mathbf{n}} = g_n \text{ on } \partial\hat{\Omega}\right\},$$

$$\hat{Q}^{\text{int}}_{0,h} := \left\{\hat{q}_e \in \hat{Q}^{\text{int}}_h(\hat{\Omega}) : \int_\Omega \hat{q}_e = 0\right\}.$$

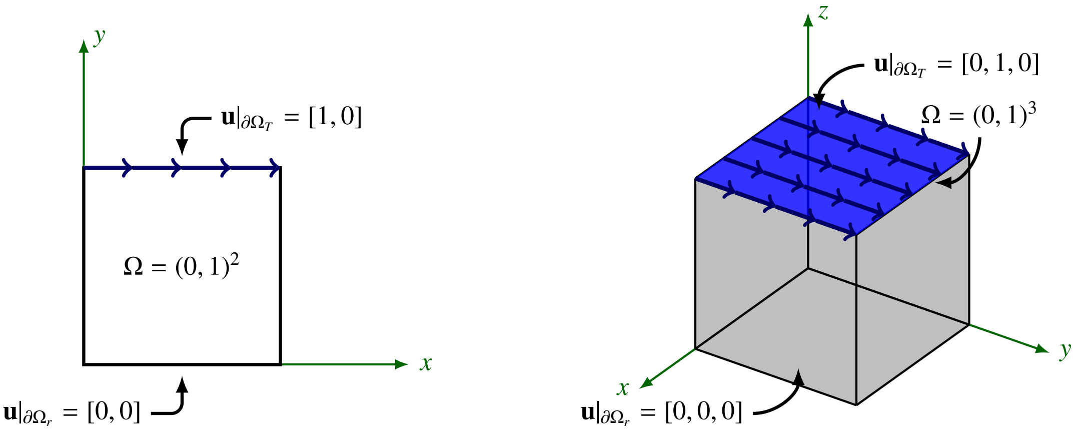
To impose the tangential component of the velocity (g_t), we use Nitsche's method. This approach weakly imposes the tangential component of the vectorial field at the boundary [30]. Now, we introduce the Galerkin formulation that uses the weak imposition of the tangential component of the velocity.

$$\begin{cases} \text{Find } \{\mathbf{u}, p_e\}\,, \text{ with } \mathbf{u} \in \mathcal{V}^{\text{div}}_{g_n,h}, \text{ and } p_e \in \mathcal{Q}^{\text{int}}_{0,h}, \text{ such that:} \\ \quad (\nabla^s \mathbf{w}, 2\nu\nabla^s \mathbf{u})_\Omega - (\nabla\cdot\mathbf{w}, p_e)_\Omega + (q_e, \nabla\cdot\mathbf{u})_\Omega \qquad\qquad (\mathbf{w}, \mathbf{f})_\Omega \\ \qquad\qquad -(\mathbf{w}, 2\nu\nabla^s\mathbf{u}\cdot\mathbf{n})_\Gamma + \left(\mathbf{w}, 2\nu\mathbf{u}\,\alpha_p\right)_\Gamma \quad = \quad +\left(\mathbf{w}, 2\nu\mathbf{g}\,\alpha_p\right)_\Gamma \\ \qquad\qquad\qquad\qquad -(\mathbf{u}, 2\nu\nabla^s\mathbf{w}\cdot\mathbf{n})_\Gamma \qquad\quad -(\mathbf{g}, 2\nu\nabla^s\mathbf{w}\cdot\mathbf{n})_\Gamma \\ \text{for all } \mathbf{w} \in \mathcal{V}^{\text{div}}_{0,h}, \text{ and } q_e \in \mathcal{Q}^{\text{int}}_{0,h}, \end{cases} \tag{60}$$

where $\mathcal{V}^{\text{div}}_{0,h}$ and $\mathcal{Q}^{\text{int}}_{0,h}$ are computed using (15) and (18), respectively. Γ refers to the boundary in which the tangential condition is impossed, $\mathbf{g}$ is the boundary condition data that consists in a tangential value g_t and a normal value g_n, and $\alpha_p = C_{pen}/h_f$. $C_{pen} = 5(p+1)$ is the penalty parameter that depends upon the polynomial degree of the discretization and h_f, which is the wall normal mesh size [31]. In the Galerkin formulation, the term in blue comes from the natural boundary condition (consistency), the penalization terms are in red and the terms in green conform the adjoint consistency.

5.2.1. Lid-driven cavity problem (Stokes flow)

We solve a lid-driven cavity problem over a unitary domain, assuming a Stokes flow. We impose a tangential velocity at the top boundary of the domain ($\partial\Omega_T$) equal to one, and on the remaining boundaries ($\partial\Omega_r$) we consider a no-slip condition. Moreover, no external forces are considered ($\mathbf{f} = \mathbf{0}$). Figure 20 illustrates the structure of the lid-driven cavity test problem.


y
$\mathbf{u}|_{\partial\Omega_T} = [1,0]$
$\Omega = (0,1)^2$
x
$\mathbf{u}|_{\partial\Omega_r} = [0,0]$
z
$\mathbf{u}|_{\partial\Omega_T} = [0,1,0]$
$\Omega = (0,1)^3$
y
x
$\mathbf{u}|_{\partial\Omega_r} = [0,0,0]$

(a) 2D domain (b) 3D domain

Figure 20: Lid-driven cavity fluid flow test problem.

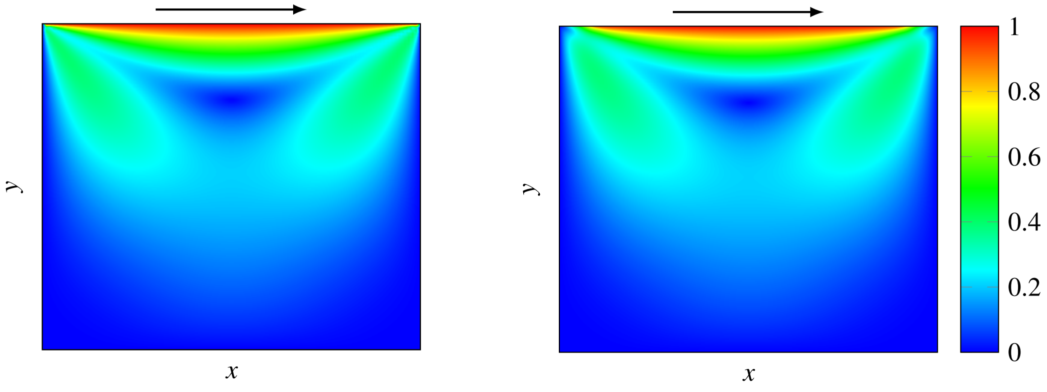
The pressure and the stress fields, in this problem, involve corner singularities. Due to this, the exact solution of the velocity lies in a Sobolev space $\mathcal{W}^{1,p}(\Omega)$, being $1 < p \leq 2$, instead of lying in $\boldsymbol{H}(\text{div})$ or $\boldsymbol{H}^1$. In [14], Evans et al. showed that the $\boldsymbol{H}(\text{div}) \times L^2$ IGA discretization approximates well the smooth portions of the flow and properly resolve the flow close to the corner singularities showing that the vorticity nearby those corners slowly converges to the highly accurate pseudospectral results presented in [32].

In addition to this classical Stokes benchmark problem, we consider a 2D case in which the boundary condition at the top boundary of the domain consists of a continuous function

$$\mathbf{u}|_{\partial\Omega_T} = \begin{cases} [10x, 0], & \forall\, x < 0.1 \\ [1, 0]. & \forall\, 0.1 \leq x \leq 0.9 \\ [10(1-x), 0]. & \forall\, 0.9 \leq x \leq 1 \end{cases} \tag{61}$$

In this case, the problem does not experience corner singularities, and the exact solution of the fluid flow problem is in the proper spaces.

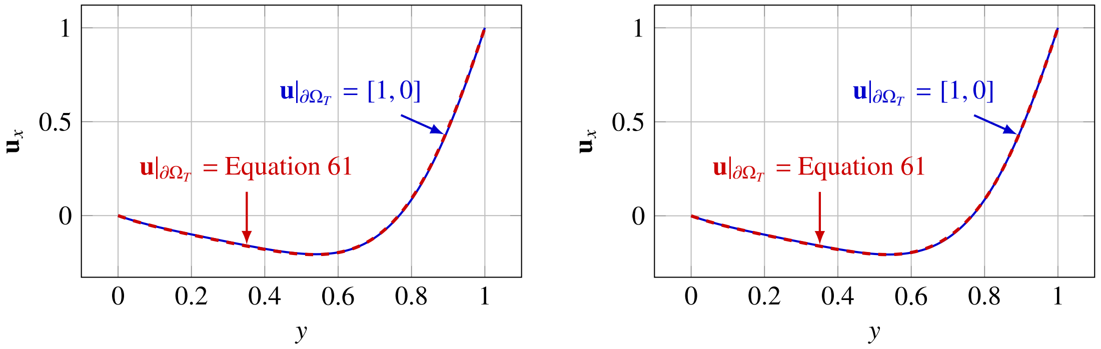
Figure 21 illustrates the magnitude of the velocity for the Stokes problem solved using a mesh size of 1024^2 elements and polynomial degree $p = 5$ in 2D. In Figure 22, we plot the horizontal velocity along the vertical centerline when using the top boundary condition $\mathbf{u}|_{\partial\Omega_T} = [1, 0]$ and the one defined with (61) and Figure 23 compares the horizontal velocity along the vertical centerline for IGA and the optimal case of rIGA in both 2D and 3D with $\mathbf{u}|_{\partial\Omega_T} = [1, 0]$ (that corresponds to a classical Stokes benchmark problem). The smooth portion of the flow inside the domain is accurately approximated, and the continuity reduction shows no notorious impact on the solution, as expected.



(a) Case with the top boundary condition of $\mathbf{u}|_{\partial\Omega_T} = [1, 0]$

(b) Case with the top boundary condition defined by (61)

Figure 21: Magnitude of the velocity ($\mathbf{u}$) of the 2D Stokes problem. Problem solution approximated over a mesh size of 1024^2 elements and a polynomial degree $p = 5$.



(a) IGA

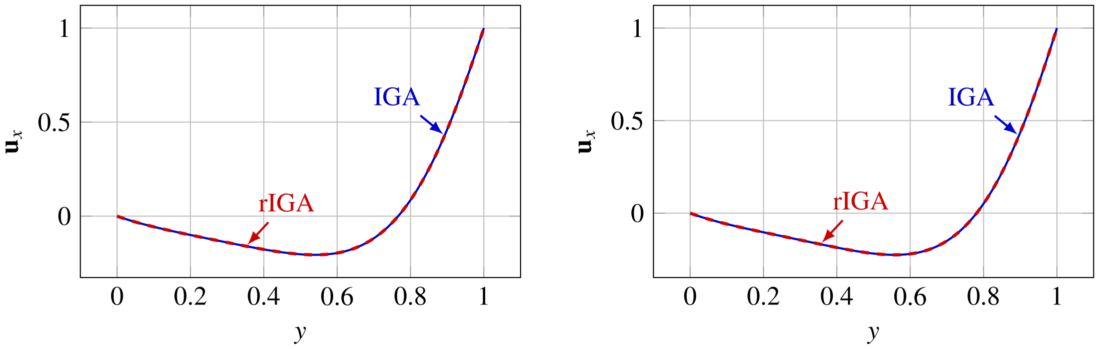
(b) rIGA

Figure 22: Horizontal velocity $\mathbf{u}_x$ along the vertical centerline. Comparison between using a top boundary condition of $\mathbf{u}|_{\partial\Omega_T} = [1, 0]$ and $\mathbf{u}|_{\partial\Omega_T}$ defined by (61).

Figures 24 and 25 display the number of FLOPs required to solve the Stokes problem in 2D and 3D, respectively. The number of FLOPs is plotted with respect to the macro-elements size $s = n_{\text{elem}}/2^{\ell}$, being ℓ the number of partition levels over which we perform continuity reduction.

For 2D, we use a constant factor close to 16 to fit the theoretical estimates provided in (28) and (44) with the computed number of FLOPs. For 3D, the constant factor used to fit the theoretical estimates with the numerical results is close to 7. These constants include the contribution of forming the Schur complements and the total number of FLOPs that LAPACK performs to factorize the system, and are slightly dependent on the polynomial degree.

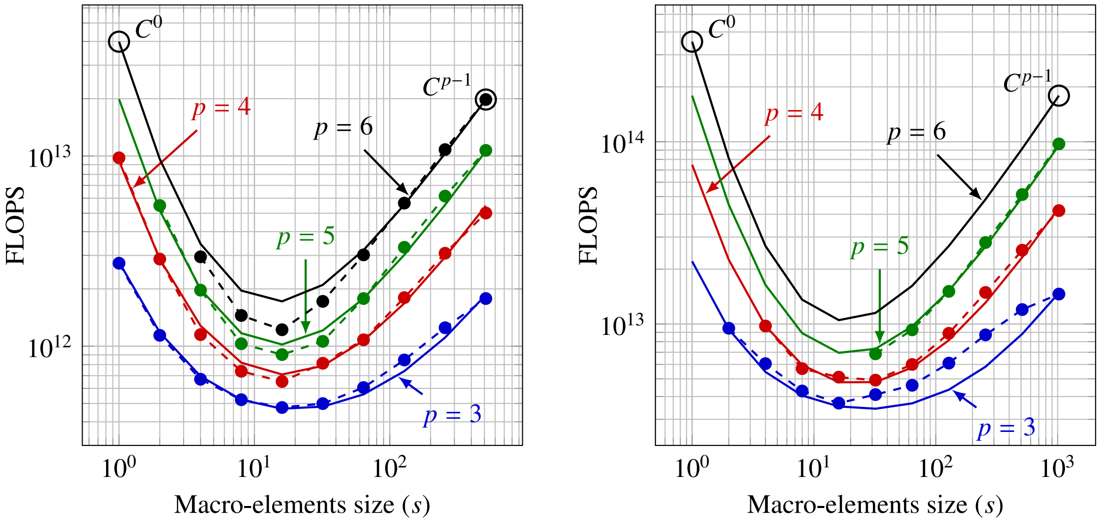
Tables 6 and 7 provide the number of FLOPs and computational times (in seconds) for the corresponding IGA and optimal rIGA discretizations in 2D and 3D, respectively.



(a) 2D: Mesh size of 1024^2 elements and $p = 4$

(b) 3D: Mesh size of 32^2 elements and $p = 4$

Figure 23: Comparison of the horizontal velocity $\mathbf{u}_x$ along the vertical centerline for IGA and the optimal case of rIGA.



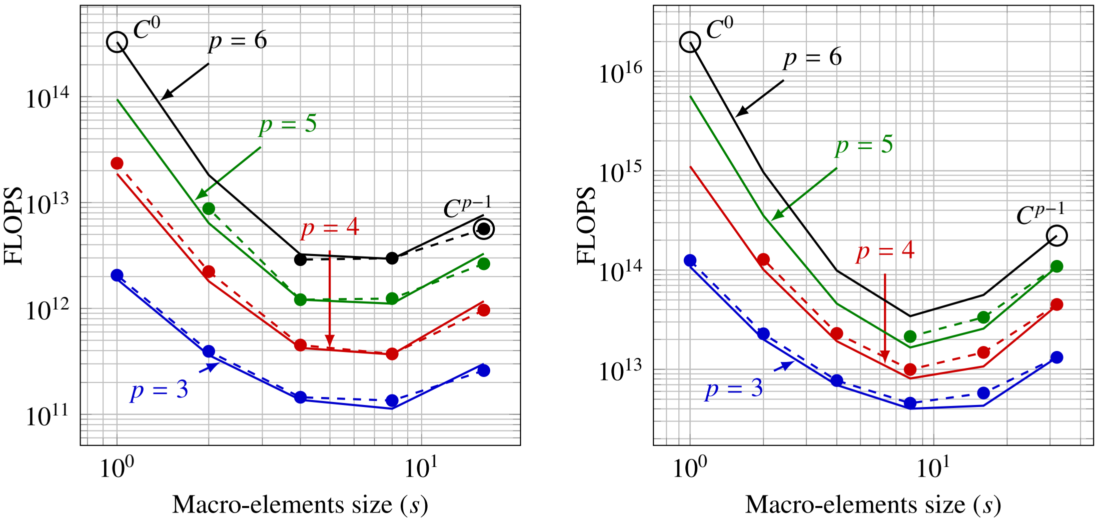
(a) Mesh size of 512^2 elements

(b) Mesh size of 1024^2 elements

Figure 24: Number of FLOPs required to solve the 2D Stokes problem with the multifrontal direct solver. The dashed lines with rounded markers correspond to the numerical results and the solid lines represent the theoretical estimates.

The theoretical estimates describe well the behaviour we observe in the numerical experiments. In particular, these estimates predict the macro-element size that delivers the maximum cost reduction. The numerical results show that rIGA decreases the number of FLOPs required for solving the Stokes problem by a factor up to $O(p^2)$ in 2D. For instance, in the case that we solve the model problem using a mesh size of 1024^2 elements and polynomial degree $p = 5$, the optimal rIGA case reports a reduction with respect to C^{p-1} IGA of approximately 14 times in terms of the number of FLOPs. In 3D, the reduction factor of the number of FLOPs is approximately p. In 3D, the examples we can solve are in the pre-asymptotic regime, and due to that, the maximum reduction factor of the number of FLOPs we can observe is $O(p)$. As we reach the asymptotic regime (for a sufficiently large grid), the reduction factor of the number of FLOPs becomes $O(p^2)$. For example, when using the FLOPs estimates of (44) with $p = 6$ and $N_{\text{elem}} = 256^3$ we obtain a factor of 31 in the computational time savings.

In terms of computational times, rIGA reduces the computational time with respect to the maximum continuity IGA by an order of magnitude when solving the Stokes problem in 2D, with a mesh size of 1024^2 elements and polynomial degree $p = 5$. That is, rIGA takes seven minutes to solve a problem where IGA for the same mesh takes over 80 minutes. In 3D, the case solved with a mesh size of 32^3 elements and polynomial degree $p = 4$ reports the



(a) Mesh size of 16^3 elements

(b) Mesh size of 32^3 elements

Figure 25: Number of FLOPs required to solve the 3D Stokes problem with the multifrontal direct solver. The dashed lines with rounded markers correspond to the numerical results and the solid lines represent the theoretical estimates.

Polynomial degree	Method	$N_{\text{elem}} = 512^2$		$N_{\text{elem}} = 1024^2$	
		FLOPs	Time [s]	FLOPs	Time [s]
3	IGA rIGA IGA/rIGA	1.78e+12 4.76e+11 3.73	106.27 34.51 3.07	1.45e+13 3.68e+12 3.94	794.28 226.29 3.51
4	IGA rIGA IGA/rIGA	5.01e+12 6.52e+11 7.68	277.83 46.24 6.01	4.19e+13 4.92e+12 8.37	2191.69 309.11 7.09
5	IGA rIGA IGA/rIGA	1.07e+13 9.03e+11 11.85	583.97 62.67 9.32	9.73e+13 6.85e+12 14.20	4975.04 429.88 11.57
6	IGA rIGA IGA/rIGA	1.98e+13 1.24e+12 15.97	1038.72 83.32 12.47	1.93e+14 8.04e+12 24.00	12996.36 645.24 20.14

Note: Results in blue were computed with Pardiso and PETSc using 64bit indices.

Table 6: Number of FLOPs and computational times (in seconds) required to solve the 2D Stokes problem with the multifrontal direct solver using two mesh sizes and four polynomial degrees (p ranging from 3 to 6).

largest solvable problem using sequential MUMPS. In this case, the improvement in scaling is just above four times, as the meshes are rather small (i.e., the 90 minutes of IGA reduce to 20 with rIGA).

6. Conclusions

We extend the refined isogeometric analysis (rIGA) to solve multi-field problems. This extension delivers a reduction in the computational cost and provides better approximability than C^{p-1} IGA. rIGA computes the solution of the 2D electromagnetic problem with the spline-generalization of the Ndelec finite element spaces approximately

Polynomial degree	Method	$N_{\text{elem}} = 16^3$		$N_{\text{elem}} = 32^3$	
		FLOPs	Time [s]	FLOPs	Time [s]
3	IGA	2.59e+11	15.36	1.32e+13	667.42
	rIGA	1.34e+11	8.79	4.56e+12	242.99
	IGA/rIGA	1.93	1.74	2.89	2.75
4	IGA	9.63e+11	53.42	4.49e+13	2222.29
	rIGA	3.72e+11	22.78	9.96e+12	530.63
	IGA/rIGA	2.59	2.34	4.51	4.19
5	IGA	2.63e+12	138.27	1.09e+14	5394.60
	rIGA	1.24e+12	69.58	2.14e+13	1179.58
	IGA/rIGA	2.12	1.98	5.09	4.57
6	IGA	5.64e+12	285.07	2.26e+14	14674.84
	rIGA	2.89e+12	165.49	4.50e+13	3019.07
	IGA/rIGA	1.95	1.72	5.02	4.86

Note: Results in blue were computed with Pardiso and PETSc using 64bit indices.

Table 7: Number of FLOPs and computational times (in seconds) required to solve the 3D Stokes problem with the multifrontal direct solver using two mesh sizes and four polynomial degrees (p ranging from 3 to 6).

p^2 faster than C^{p-1} IGA. Similarly, the numerical results of the incompressible fluid flow problem solved with the spline-based generalization of the Raviart-Thomas finite elements spaces show a reduction factor in the computational cost up to p^2 for 2D. In 3D, the maximum size of a problems that we can solve with sequential MUMPS is in the pre-asymptotic regime. Thus, the maximum gain factors are of $O(p)$. In multi-field problems, we require additional time (larger grids than in scalar problems) to arrive at the asymptotic limit and reap the full benefits. For sufficiently large grids (asymptotic regime), the theoretical estimates show that the gain factor scales like $O(p^2)$.

The optimal discretizations obtained with rIGA consists of enriched/nested spaces with respect to C^{p-1} IGA. Therefore, the best approximation error is improved by definition. Similarly, the best approximation error of the corresponding C^0 FEA discretization is smaller than that of rIGA. Nevertheless, since rIGA is orders of magnitude faster than FEM in 2D and especially in 3D, the slight improvement in the errors do not justify the use of FEM from the computational cost point of view. The total numerical error for stable elliptic problems improves when going from IGA to rIGA discretizations.

As future work, we plan to apply rIGA to solve hyperbolic and parabolic PDEs systems in order to perform a deeper analysis on the effect of the continuity reduction on the total numerical error. In particular, we shall evaluate how significant is the impact of the continuity reduction on the problems' stability. As an observation, the first work in this topic has been just published in 2018. In this work, the authors study the spectral approximation properties of rIGA and show how the local reduction of continuity impact on the error in the eigenvalues and eigenfunctions [33].

Acknowledgement

David Pardo has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 777778, the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2016-76329-R (AEI/FEDER, EU), and MTM2016-81697-ERC/AEI, the BCAM Severo Ochoa accreditation of excellence SEV-2018-0718, and the Basque Government through the BERC 2014-2017 program, and the Consolidated Research Group Grant IT649-13 on Mathematical Modeling, Simulation, and Industrial Applications (M2SI). This publication was also made possible in part by the Deep Earth Imaging Enterprise Future Science Platforms of the Commonwealth Scientific Industrial Research Organisation, CSIRO, of Australia, the Mega-grant of the Russian Federation Government (N14.Y26.31.0013) and the Curtin Institute for Computation. The J. Tinsley Oden Faculty Fellowship Research Program at the Institute for Computational Engineering and Sciences (ICES) of the University of Texas at Austin has partially supported the visits of VMC to ICES.

The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper.

[1] J. A. Cottrell, TJR. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, Ltd, Singapore, Ma, 2009.

[2] J. A. Cottrell, TJR. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Computer Methods in Applied Mechanics and Engineering 196 (4144) (2007) 4160 – 4183.

[3] TJR. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (3941) (2005) 4135 – 4195.

[4] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, V.M. Calo, The cost of continuity: A study of the performance of isogeometric finite elements using direct solvers, Computer Methods in Applied Mechanics and Engineering 213-216 (0) (2012) 353 – 361.

[5] N. Collier, L. Dalcin, V.M. Calo, On the computational efficiency of isogeometric methods for smooth elliptic problems using direct solvers, International Journal for Numerical Methods in Engineering 100 (8) (2014) 620–632.

[6] V.M. Calo, N. O. Collier, D. Pardo, M. R. Paszynski, Computational complexity and memory usage for multi-frontal direct solvers used in p finite element analysis, Procedia Computer Science 4 (0) (2011) 1854 – 1861.

[7] D. Garcia, D. Pardo, L. Dalcin, M. Paszyski, N. Collier, V.M. Calo, The value of continuity: Refined isogeometric analysis and fast direct solvers, Computer Methods in Applied Mechanics and Engineering 316 (2017) 586–605, special Issue on Isogeometric Analysis: Progress and Challenges.

[8] D. Garcia, M. Bartoň, D. Pardo, Optimally refined isogeometric analysis, Procedia Computer Science 108 (2017) 808–817, international Conference on Computational Science, (ICCS) 2017, 12-14 June 2017, Zurich, Switzerland.

[9] D. Garcia, D. Pardo, L. Dalcin, V.M. Calo, Refined isogeometric analysis for a preconditioned conjugate gradient solver, Computer Methods in Applied Mechanics and Engineering 335 (2018) 490 – 509.

[10] A. Buffa, C. de Falco, G. Sangalli, Isogeometric analysis: Stable elements for the 2D Stokes equation, International Journal for Numerical Methods in Fluids 65 (11-12) (2011) 1407–1422.

[11] M. G. Cox, The numerical evaluation of B-splines, IMA Journal of Applied Mathematics 10 (2) (1972) 134–149.

[12] C. de Boor, On calculating with B-splines, Journal of Approximation Theory 6 (1) (1972) 50 – 62.

[13] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Computer Methods in Applied Mechanics and Engineering 199 (17-20) (2010) 1143 – 1152.

[14] J. A. Evans, TJR. Hughes, Isogeometric divergence-conforming B-splines for the Darcy-Stokes-Brinkman equations, Mathematical Models and Methods in Applied Sciences 23 (04) (2013) 671–741.

[15] J. A. Evans, TJR. Hughes, Isogeometric divergence-conforming B-splines for the steady Navier-Stokes equations, Mathematical Models and Methods in Applied Sciences 23 (08) (2013) 1421–1478.

[16] J. A. Evans, TJR. Hughes, Isogeometric divergence-conforming B-splines for the unsteady Navier-Stokes equations, Journal of Computational Physics 241 (2013) 141–167.

[17] A. Sarmiento, D. Garcia, L. Dalcin, N. Collier, V. Calo, Micropolar fluids using B-spline divergence conforming spaces, Procedia Computer Science 29 (2014) 991 – 1001.

[18] P. Vignal, A. Sarmiento, A. M. Côrtes, L. Dalcin, V.M. Calo, Coupling Navier-Stokes and Cahn-Hilliard equations in a two-dimensional annular flow configuration, Procedia Computer Science 51 (2015) 934 – 943.

[19] A. Sarmiento, A. Côrtes, D. Garcia, L. Dalcin, N. Collier, V.M. Calo, PetIGA-MF: A multi-field high-performance toolbox for structure-preserving B-splines spaces, Journal of Computational Science 18 (2017) 117–131.

[20] L. Dalcin, N. Collier, P. Vignal, A. Côrtes, V.M. Calo, PetIGA: A framework for high-performance isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 308 (2016) 151–181.

[21] L. F. R. Espath, A. F. Sarmiento, P. Vignal, B. O. N. Varga, A. M. A. Cortes, L. Dalcin, V. M. Calo, Energy exchange analysis in droplet dynamics via the Navier-Stokes-Cahn-Hilliard model, Journal of Fluid Mechanics 797 (2016) 389430.

[22] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc Web page (2016).
URL `http://www.mcs.anl.gov/petsc`

[23] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.7, Argonne National Laboratory (2016).
URL `http://www.mcs.anl.gov/petsc`

[24] P. R. Amestoy, I. S. Duff, J.-Y. LÉxcellent, J. Koster, A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling, SIAM Journal on Matrix Analysis and Applications 23 (1) (2001) 15–41.

[25] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Computing 32 (2) (2006) 136 – 156.

[26] Karypis Laboratory, METIS, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview (2016).
URL `http://glaros.dtc.umn.edu/gkhome/metis/metis/overview`

[27] C. G. Petra, O. Schenk, M. Lubin, K. Gärtner, An augmented incomplete factorization approach for computing the schur complement in stochastic optimization, SIAM Journal on Scientific Computing 36 (2) (2014) C139–C162.

[28] C. G. Petra, O. Schenk, M. Anitescu, Real-time stochastic optimization of complex energy systems on high-performance computers, IEEE Computing in Science & Engineering 16 (5) (2014) 32–42.

[29] O. Schenk, K. Gärtner, W. Fichtner, Efficient sparse lu factorization with left-right looking strategy on shared memory multiprocessors, BIT Numerical Mathematics 40 (1) (2000) 158–176.

[30] J. Nitsche, Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, in: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, Vol. 36, Springer, 1971, pp. 9–15.

[31] Y. Bazilevs, C. Michler, V.M. Calo, TJR. Hughes, Weak Dirichlet boundary conditions for wall-bounded turbulent flows, Computer Methods in Applied Mechanics and Engineering 196 (49-52) (2007) 4853–4862.

[32] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Computers & Fluids 27 (4) (1998) 421–433.

[33] V. Puzyrev, Q. Deng, V. Calo, Spectral approximation properties of isogeometric analysis with variable continuity, Computer Methods in Applied Mechanics and Engineering 334 (2018) 22–39.

Appendices

A. Curl-conforming IGA spaces: Separators' size

The size of the separators for the case in which we use curl-conforming IGA spaces in 2D is given by

$$q_{sep} = q_{sep}^{\mathrm{u}_x} + q_{sep}^{\mathrm{u}_y},$$

where $q_{sep}^{\mathrm{u}_x}$ and $q_{sep}^{\mathrm{u}_y}$ refer to the size of the separators in each of the curl-conforming spaces. Table 8 presents the size of the separators in each space where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

Separator	$q_{sep}^{\mathrm{u}_x}$	$q_{sep}^{\mathrm{u}_y}$
$q_{sep}\|_y$ (Vertical)	$O\left(2^{-(i-1)}\,(n_{\mathrm{elem}}+p)(k)\right)$	$O\left(2^{-(i-1)}\,(n_{\mathrm{elem}}+p-1)(k+1)\right)$
$q_{sep}\|_x$ (Horizontal)	$O\left(2^{-i}\,(n_{\mathrm{elem}}+p-1)(k+1)\right)$	$O\left(2^{-i}(n_{\mathrm{elem}}+p)(k)\right)$

Table 8: Size of the separators in each of the fields corresponding to the curl-conforming spaces.

$$\begin{aligned} q_{sep}|_y &= O\Big(2^{-(i-1)}\,((n_{\mathrm{elem}}+p)(2k+1)-(k+1))\Big), && (62)\\ q_{sep}|_x &= O\Big(2^{-i}\,((n_{\mathrm{elem}}+p)(2k+1)-(k+1))\Big). && (63)\end{aligned}$$

B. Divergence-conforming IGA spaces: Separators' size

The size of the separators for the case in which we use divergence-conforming IGA spaces in 2D is given by

$$q_{sep} = q_{sep}^{\mathrm{u}_x} + q_{sep}^{\mathrm{u}_y},$$

where $q_{sep}^{\mathrm{u}_x}$ and $q_{sep}^{\mathrm{u}_y}$ refer to the size of the separators in each of the spaces. Table 9 presents the size of the separators in those spaces

Separator	$q_{sep}^{\mathrm{u}_x}$	$q_{sep}^{\mathrm{u}_y}$
$q_{sep}\|_y$ (Vertical)	$O\left(2^{-(i-1)}\,(n_{\mathrm{elem}}+p-1)(k+1)\right)$	$O\left(2^{-(i-1)}\,(n_{\mathrm{elem}}+p)(k)\right)$
$q_{sep}\|_x$ (Horizontal)	$O\left(2^{-i}\,(n_{\mathrm{elem}}+p)(k)\right)$	$O\left(2^{-i}(n_{\mathrm{elem}}+p-1)(k+1)\right)$

Table 9: Size of the separators in each of the fields corresponding to the divergence-conforming spaces.

where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

$$\begin{aligned} q_{sep}|_y &= O\Big(2^{-(i-1)}\,((n_{\mathrm{elem}}+p)(2k+1)-(k+1))\Big), && (64)\\ q_{sep}|_x &= O\Big(2^{-(i)}\,((n_{\mathrm{elem}}+p)(2k+1)-(k+1))\Big). && (65)\end{aligned}$$

C. Integral-conforming IGA spaces: Separators' size

The size of the separators for the case in which we use an integral-conforming IGA space in 2D is $q_{sep} = q_{sep}^{\phi}$, where q_{sep}^{ϕ} is given in Table 10.

Separator	q_{sep}^{ϕ}
$q_{sep}\|_y$ (Vertical)	$O\left(2^{-(i-1)}\,(n_{\text{elem}}+p-1)(k)\right)$
$q_{sep}\|_x$ (Horizontal)	$O\left(2^{-i}(n_{\text{elem}}+p-1)(k)\right)$

Table 10: Size of the separators for the integral-conforming spaces.

where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

$$q_{sep}|_y \;=\; O\Big(2^{-(i-1)}(n_{\text{elem}}+p-1)(k)\Big), \tag{66}$$

$$q_{sep}|_x \;=\; O\Big(2^{-(i)}(n_{\text{elem}}+p-1)(k)\Big). \tag{67}$$

D. Curl-conforming rIGA spaces: Separators and macro-elements size

D.1. Separators' size

The size of the separators for the case in which we use curl-conforming rIGA spaces in 2D is given by

$$q_{sep} = q_{sep}^{\mathrm{u}_x} + q_{sep}^{\mathrm{u}_y},$$

where $q_{sep}^{\mathrm{u}_x}$ and $q_{sep}^{\mathrm{u}_y}$ refer to the size of the separators in each of the curl-conforming spaces. Table 11 presents the size of the separators in each space where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

Separator	$q_{sep}\|_y$ (Vertical)	$q_{sep}\|_x$ (Horizontal)
$q_{sep}^{\mathrm{u}_x}$	$O\left(2^{-(i-1)}\,(n_{\text{elem}}+p+(2^i-1)(k-1))(1)\right)$	$O\left(2^{-i}\,(n_{\text{elem}}+p-1+(2^i-1)(k-1))(2)\right)$
$q_{sep}^{\mathrm{u}_y}$	$O\left(2^{-(i-1)}\,(n_{\text{elem}}+p-1+(2^i-1)(k-1))(2)\right)$	$O\left(2^{-i}(n_{\text{elem}}+p+(2^i-1)(k-1))(1)\right)$

Table 11: Size of the separators in each of the fields corresponding to the curl-conforming spaces.

$$q_{sep}|_y \;=\; O\Big(2^{-(i-1)}\left((n_{\text{elem}}+p+(2^i-1)(k-1))\right)\Big), \tag{68}$$

$$q_{sep}|_x \;=\; O\Big(2^{-i}\left((n_{\text{elem}}+p+(2^i-1)(k-1))\right)\Big), \tag{69}$$

D.2. Macro-elements' size

The size of the macro-elements in 2D is $n_{\text{m-e}} = n_{\text{m-e}}|_x \cdot n_{\text{m-e}}|_y$, being $n_{\text{m-e}}|_x$ and $n_{\text{m-e}}|_y$ the size of the macro-element in the horizontal and vertical spatial directions, respectively. Table 12 provides the macro-elements sizes.

	$n_{\text{m-e}}\|_x$	$n_{\text{m-e}}\|_y$
u_x	$2^{-\ell}\left(n_{\text{elem}}+p-1+(2^\ell-1)(k-1)\right)$	$2^{-\ell}\left(n_{\text{elem}}+p+(2^\ell-1)(k-1)\right)$
u_y	$2^{-\ell}\left(n_{\text{elem}}+p+(2^\ell-1)(k-1)\right)$	$2^{-\ell}\left(n_{\text{elem}}+p-1+(2^\ell-1)(k-1)\right)$

Table 12: Macro-element sizes in the vectorial field corresponding to the curl-conforming discrete spaces.

E. Divergence-conforming rIGA spaces: Separators and macro-elements size

E.1. Separators' size

The size of the separators for the case in which we use divergence-conforming IGA spaces in 2D is given by

$$q_{sep} = q_{sep}^{\mathrm{u}_x} + q_{sep}^{\mathrm{u}_y},$$

where $q_{sep}^{\mathrm{u}_x}$ and $q_{sep}^{\mathrm{u}_y}$ refer to the size of the separators in each of the spaces. Table 13 presents the size of the separators in those spaces where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

Separator	$q_{sep}\|_y$ (Vertical)	$q_{sep}\|_x$ (Horizontal)
$q_{sep}^{\mathrm{u}_x}$	$O\left(2^{-(i-1)}\,(n_{\text{elem}} + p - 1 + (2^i - 1)(k - 1))(2)\right)$	$O\left(2^{-i}(n_{\text{elem}} + p + (2^i - 1)(k - 1))(1)\right)$
$q_{sep}^{\mathrm{u}_y}$	$O\left(2^{-(i-1)}\,(n_{\text{elem}} + p + (2^i - 1)(k - 1))(1)\right)$	$O\left(2^{-i}\,(n_{\text{elem}} + p - 1 + (2^i - 1)(k - 1))(2)\right)$

Table 13: Size of the separators in each of the fields corresponding to the divergence and integral-conforming spaces.

$$q_{sep}|_y = O\left(2^{-(i-1)}\left((n_{\text{elem}} + p + (2^i - 1)(k - 1))\right)\right), \tag{70}$$

$$q_{sep}|_x = O\left(2^{-i}\left((n_{\text{elem}} + p + (2^i - 1)(k - 1))\right)\right), \tag{71}$$

E.2. Macro-elements' size

In 2D, the size of the macro-elements is $n_{\text{m-e}} = n_{\text{m-e}}|_x \cdot n_{\text{m-e}}|_y$, being $n_{\text{m-e}}|_x$ and $n_{\text{m-e}}|_y$ the size of the macro-element in the horizontal and vertical spatial directions, respectively. Table 16 provides the macro-elements sizes for the fields, assuming that the macro-elements are C^{p-1} systems.

	$n_{\text{m-e}}\|_x$	$n_{\text{m-e}}\|_y$
u_x	$2^{-\ell}\left(n_{\text{elem}} + p + (2^\ell - 1)(k - 1)\right)$	$2^{-\ell}\left(n_{\text{elem}} + p - 1 + (2^\ell - 1)(k - 1)\right)$
u_y	$2^{-\ell}\left(n_{\text{elem}} + p - 1 + (2^\ell - 1)(k - 1)\right)$	$2^{-\ell}\left(n_{\text{elem}} + p + (2^\ell - 1)(k - 1)\right)$

Table 14: Macro-element size in the vectorial field corresponding to the divergence-conforming spaces.

F. Integral-conforming rIGA spaces: Separators and macro-elements size

F.1. Separators' size

The size of the separators for the case in which we use an integral-conforming IGA space in 2D is given by $q_{sep} = q_{sep}^{\phi}$, where q_{sep}^{ϕ} is given in Table 15

Separator	q_{sep}^{ϕ}
$q_{sep}\|_y$ (Vertical)	$O\left(2^{-(i-1)}\,(n_{\text{elem}} + p - 1 + (2^i - 1)(k - 1))(1)\right)$
$q_{sep}\|_x$ (Horizontal)	$O\left(2^{-i}(n_{\text{elem}} + p - 1 + (2^i - 1)(k - 1))(1)\right)$

Table 15: Size of the separators for the integral-conforming spaces.

where the first term inside parenthesis is the length and the second corresponds to the thickness of the separators. Then, the size of the separators at the i-th partition level is

$$q_{sep}|_y = O\left(2^{-(i-1)}\left((n_{\text{elem}} + p + (2^i - 1)(k - 1))\right)\right), \tag{72}$$

$$q_{sep}|_x = O\left(2^{-i}\left((n_{\text{elem}} + p + (2^i - 1)(k - 1))\right)\right), \tag{73}$$

F.2. Macro-elements' size

In 2D, the size of the macro-elements is $n_{\text{m-e}} = n_{\text{m-e}}|_x \cdot n_{\text{m-e}}|_y$, being $n_{\text{m-e}}|_x$ and $n_{\text{m-e}}|_y$ the size of the macro-element in the horizontal and vertical spatial directions, respectively. Table 16 provides the macro-elements sizes for the field, assuming that the macro-elements are C^{p-1} systems.

	$n_{\text{m-e}}\|_x$	$n_{\text{m-e}}\|_y$
ϕ	$2^{-\ell}\left(n_{\text{elem}} + p - 1 + (2^{\ell} - 1)(k - 1)\right)$	$2^{-\ell}\left(n_{\text{elem}} + p - 1 + (2^{\ell} - 1)(k - 1)\right)$

Table 16: Macro-element size in the scalar field corresponding to the integral-conforming space.