

The Value of Continuity: Refined Isogeometric Analysis and Fast Direct Solvers

Daniel Garcia^{a,*}, David Pardo^{b,a,c}, Lisandro Dalcin^{d,e,f}, Maciej Paszyński^g, Nathan Collier^h, Victor M. Calo^{i,j}

^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*Center for Numerical Porous Media, King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.*

^e*Consejo Nacional de Investigaciones Científicas y Técnicas, Santa Fe, Argentina.*

^f*Universidad Nacional del Litoral, Santa Fe, Argentina.*

^g*Department of Computer Science, AGH University of Science and Technology, Krakow, Poland.*

^h*Climate Change Science Institute, Oak Ridge National Laboratory, Oak Ridge, Unites States.*

ⁱ*Applied Geology, Western Australian School of Mines, Curtin University, Bentley, WA, Australia.*

^j*Mineral Resources, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Kensington, WA, Australia.*

Abstract

We propose the use of highly continuous finite element spaces interconnected with low continuity hyperplanes to maximize the performance of direct solvers. Starting from a highly continuous Isogeometric Analysis (IGA) discretization, we introduce C^0 -separators to reduce the interconnection between degrees of freedom in the mesh. By doing so, both the solution time and best approximation errors are simultaneously improved. We call the resulting method “refined Isogeometric Analysis (rIGA)”. To illustrate the impact of the continuity reduction, we analyze the number of Floating Point Operations (FLOPs), computational times, and memory required to solve the linear system obtained by discretizing the Laplace problem with structured meshes and uniform polynomial orders. Theoretical estimates demonstrate that an optimal continuity reduction may decrease the total computational time by a factor between p^2 and p^3 , with p being the polynomial order of the discretization. Numerical results indicate that our proposed rIGA method delivers a speed-up factor proportional to p^2 . In a 2D mesh with four million elements and $p = 5$, the 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.

Keywords: Isogeometric Analysis (IGA), Finite Element Analysis (FEA), refined Isogeometric Analysis (rIGA), Direct solvers, Multi-frontal solvers, k-refinement.

1. Introduction

In order to solve numerically a problem governed by partial differential equations (PDEs) and specific boundary conditions (BCs), we often resolve a system of algebraic equations which conforms the discrete representation of the problem. Discretization approaches such as finite element analysis (FEA) or isogeometric analysis (IGA) are frequently employed to generate this algebraic system. These methods discretize the governing PDEs by using a variational formulation and trial and test functions defined by their respective basis functions [1, 2].

In traditional FEA, the basis functions are defined on a reference element, and a mapping to the physical element is employed [1]. Isogeometric Analysis (IGA) seeks to eliminate the need for an interface lying between the geometrical representation built with Computed-Aided Design (CAD) and the analysis performed with FEA, since basis functions themselves are defined using conventional CAD functions. Although many obstacles still remain in the present-day CAD software, this feature avoids the need to define a secondary set of functions (and the corresponding transfer operators) for the numerical analysis [2].

*Corresponding author

Email address: dgarcia@bcamath.org (Daniel Garcia)

URL: www.bcamath.org/dgarcia (Daniel Garcia)

Several solvers can be used to compute the solution of the algebraic systems resulting from the aforementioned methods. We focus here on direct solvers, which are often used to solve stiff linear problems where iterative solvers do not converge or are unreliable. Other scenarios in which direct solvers are convenient include problems with multiple right-hand sides (e.g., when solving inverse problems). Moreover, direct solvers are the main building blocks of many iterative solvers [3]. In this work, we use a multifrontal solver, which is the state-of-the-art direct solver and was originally proposed in [4].

Direct solvers based on graph partitioning, as the multifrontal direct solver, recursively split the system's connectivity graph into pairs of subdomains interconnected by small subsets of degrees of freedom called separators. The order of elimination of degrees of freedom is set by the recursive structure of the partitioned graph, eliminating first the degrees of freedom associated with the subdomains, and then those associated to the separators that connect the subdomains. A detailed explanation is provided in section 3.

The cost to solve an algebraic system of equations, specifically the cost to perform the LU factorization of the matrix, is determined by the discretization method. Previous works presented in [5, 6] show that in IGA, the continuity plays a significant part in the degradation of the direct solvers performance on a per degree of freedom basis. Table 1 illustrates the increment in computational cost resulting from using highly continuous basis functions in the discretization. A maximal continuity IGA discretization is $\mathcal{O}(p^3)$ times more expensive than traditional FEA per unknown, with p being the polynomial order. This analysis assumes uniform continuity and a fixed total number of degrees of freedom (dof) in the system.

| System continuity | FLOPS | |
|--------------------------------------|--|------------------------|
| | Skeleton | Static condensation |
| Traditional FEA (C^0) | $\mathcal{O}\left(\left(N^{(d-1)/d}\right)^3\right)$ | $\mathcal{O}(Np^{2d})$ |
| Maximal continuity IGA (C^{p-1}) | $\mathcal{O}\left(\left(N^{(d-1)/d} p\right)^3\right)$ | $\mathcal{O}(1)$ |

$N = \text{dof}$, $p = \text{polynomial order}$, $d = \text{dimension (2 or 3)}$

Table 1: Summary of FLOPs estimates derived in [5, 6].

According to Table 1, the performance of direct solvers per unknown improves when reducing the inter-element continuity for a given C^{p-1} problem (see Table 1). However, if the number of element is kept fixed and the global continuity is turned to be C^0 , then the total solution cost of the system may become larger than that of the original C^{p-1} system. This larger cost is due to the increased number of degrees of freedom (N) that the reduction of continuity carries (see Figure 1c).

In this work, we analyze the impact of various continuity patterns in the computational cost of the direct solver for a fixed mesh topology and polynomial order p . The strategy we propose reduces the continuity along some separators used in graph partitioning. As the continuity is reduced, the interconnection among the subdomains weakens, and the number of degrees of freedom in the system grows. This is equivalent to discretizing the system using a variation of traditional FEA that employs C^{p-1} subsystems (subdomains) as elements (macro-elements). In the limiting case of reducing continuity along all the separators, the resulting system becomes a traditional C^0 FEA. Alternatively, the method can also be interpreted as a high continuity IGA with certain refinements over specific hyperplanes (separators) that locally reduce the continuity. Figure 1 illustrates the cases with no reduction of continuity (C^{p-1} system) and full reduction of continuity (C^0 system), in addition to a third case with partial reduction of continuity that consists of a C^0 skeleton and four macro-elements (C^{p-1} subsystems). The resulting class of discretizations which have minimal solution cost are the ones defined by the rIGA method we propose. The corresponding spaces are finer than standard maximal continuity IGA spaces and are faster to solve for than both traditional FEA and IGA for meshes with fixed number of elements.

The optimal discretization provided by rIGA solves the problems approximately p^2 times faster than those discretized using highly continuous IGA (C^{p-1}). To understand this boost in performance, we describe the impact of continuity reduction in the solution cost. To simplify the analysis, we assume that the inter-element continuity is

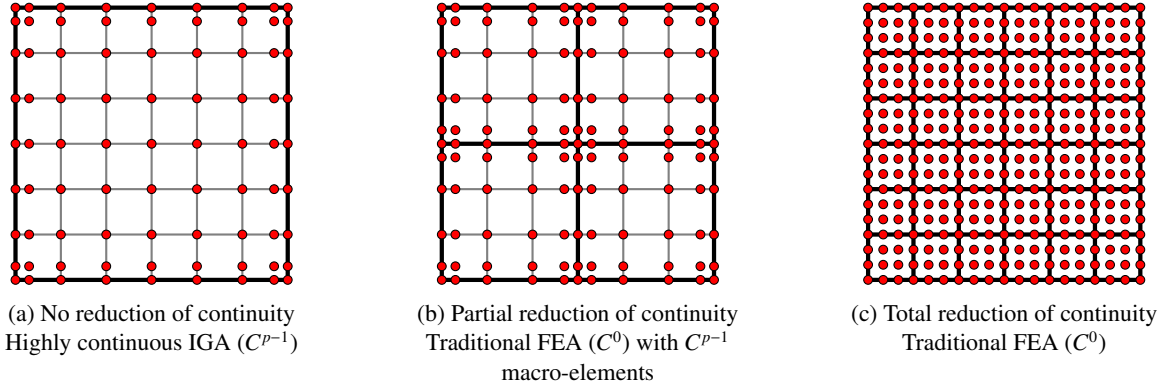


Figure 1: Illustration of FEA discretizations of a 2D system composed of 6×6 elements with polynomial basis functions of order $p = 3$. Red circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton. Bold lines represent lower continuity.

either C^0 or C^{p-1} . The continuity reduction increases the accuracy of the best approximation error (with respect to the C^{p-1} discretization) due to the resulting enrichment of the discrete space while at the same time improves the performance of the solver. The increment in the accuracy of the best approximation error does not always imply an improvement of the result. For example, stability problems in the solution may lead to a worse result even if the best approximation error is reduced.

This paper is organized as follows: Isogeometric analysis is briefly described in Section 2. Section 3 details a state-of-the-art direct solver. The proposed refined isogeometric analysis (rIGA) is presented in Section 4. Section 5 derives computational cost estimates. Sections 6 and 7 describe the model problem and detail the implementation, respectively. Numerical experiments are presented in Section 8. The work concludes in Section 9 summarizing the main features of the proposed refined isogeometric method.

2. Isogeometric Analysis

Isogeometric analysis is a well-established computational approach to solve problems governed by PDEs [2, 7–20]. Since the main ideas of IGA were established in 2005 [2], this approach has been employed in many engineering fields, including phase transition phenomena [7–9, 21, 22], fluid structure interaction (FSI) [10–12], solid mechanics [23–27], fluid dynamics [13–18, 28–33], medical applications [33–36] and electromagnetics [37].

This computational method tries to unify CAD with FEA, seeking to create a single discretization that represents the problems geometry and it is at the same time usable for numerical analysis. Thus, IGA in the future may eliminate the requirement of an interface between the geometrical representation built with CAD and the analysis performed with FEA.

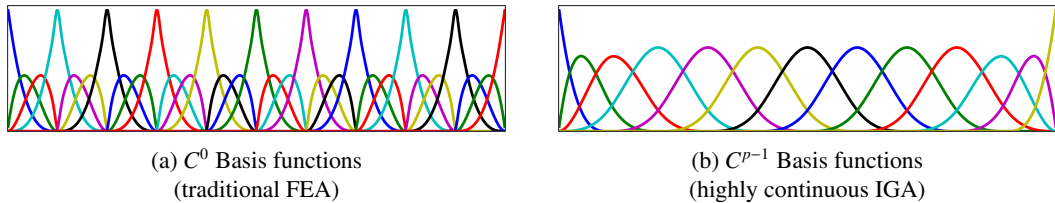


Figure 2: Sets of B-spline basis functions associated to a polynomial order $p = 3$ on a fixed mesh of ten elements.

IGA creates discrete representations with high inter-element continuity. It employs smooth basis functions with larger supports than those used in traditional FEA. Figure 2 illustrates a set of cubic basis functions for both traditional C^0 FEA and maximal continuity C^{p-1} IGA for a fixed mesh size. The use of high continuity basis functions provides additional robustness in linear elasticity, structural vibration and wave propagation calculations [2, 24–26], as well as

in fluid mechanics problems [18–20, 28, 30]. Moreover, IGA often provides discretizations with a lower number of degrees of freedom than those delivered by traditional FEA in order to achieve the same error level [2, 23, 38].

3. Direct solvers for sparse systems

There exist several types of direct solvers, such as those based on LU and QR factorizations. The idea behind these solvers is to perform a suitable decomposition of the original matrix in terms of the multiplication of two auxiliary matrices leading to two linear systems of equations that can be easily solved. The fastest direct solvers are based on LU factorization (Cholesky factorization in the case of symmetric positive definite systems). These solvers decompose the matrix into a lower-triangular matrix (L) and an upper-triangular one (U), and solve the respective triangular systems sequentially in order to obtain the solution of the original problem.

For sparse systems (which arise in FEA due to the local support of the basis functions), a reordering of rows and columns of the matrix is performed before proceeding with the factorization in order to minimize the subsequent fill-in in the L and U factors, thus improving the performance of the solver (Figure 3). The matrix is commonly reordered according to a nested-dissection technique [39], since this ordering algorithm is optimal for minimizing the fill-in of the L and U factors for the case of structured grids with an equal number of elements in each spatial direction, and it is quasi-optimal for many other cases.

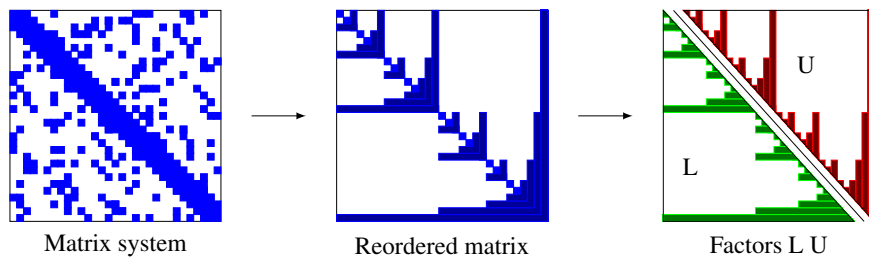


Figure 3: LU decomposition of a sparse matrix.

The state-of-the-art implementation is based on the multifrontal solver, a method of solution of sparse linear systems proposed in [4]. This solver is a generalization of the frontal direct solver presented in [40]. The multifrontal solver performs a recursive partitioning of the mesh into pairs of disconnected pieces (subdomains) that are interconnected by small subsets of degrees of freedom called separators (Figure 4). Figure 5 illustrates an example where a 2D mesh is recursively partitioned four times.

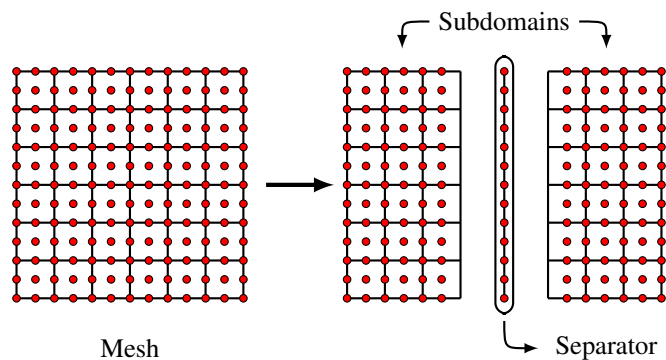


Figure 4: Subdomains resulting from the first recursive partition of a 2D problem. For simplicity, we sketch a finite element discretization using a polynomial order $p = 2$ and C^0 continuity. Red circles represent the nodal degrees of freedom in the system, while black lines denote the mesh skeleton (lower continuity lines).

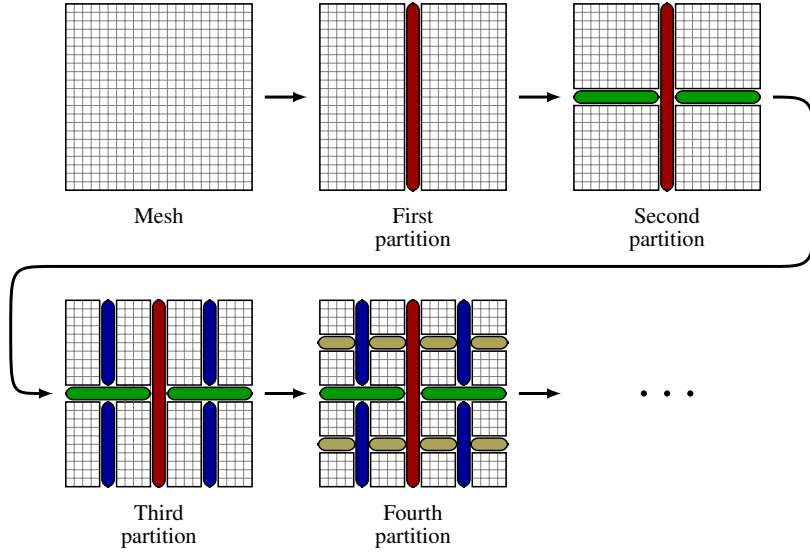


Figure 5: First four recursive partitions of a 2D mesh.

In the factorization process, the elimination of the degrees of freedom follows the recursive structure of the mesh partition. The unknowns associated to the subdomains are eliminated first. Then, the remaining unknowns are eliminated at every separator once they can be fully expressed in terms of the separator degrees of freedom. Once all separators are processed, a single subset of fully assembled degrees of freedom (first separator) is solved, and a backward substitution is executed following the structure of the system partition in order to recover the eliminated degrees of freedom and solve the problem. An illustration of the factorization procedure and solution of a 2D problem recursively partitioned into two subdomains is presented in Figure 6. For additional details on the factorization process, we refer to [4, 6, 40, 41].

The selected discretization heavily influences the performance of the recursive elimination of the system. For traditional FEA, the interconnection between subdomains is weak due to the minimal inter-element continuity. Subdomains are connected by narrow separators. In C^{p-1} IGA, the high inter-element continuity strengthens the interconnection between subdomains, since the growth of the basis function support increments the number of degrees of freedom shared between elements. Therefore, wider separators are required to interconnect the subdomains (Figure 7).

Highly continuous discretizations degrade the performance of the direct solver per unknown, increasing time and memory requirements. Indeed, highly continuous IGA is p^3 times more expensive than traditional FEA per unknown, as indicated in Table 1 and Equations 2 and 3. This cost increment occurs because the size of the separators increases and results in more expensive matrix factorizations [3].

4. Refined Isogeometric Analysis

We propose a discretization strategy that we call refined isogeometric analysis (rIGA). This approach seeks the discretization that delivers the fastest solution time for a given mesh with a fixed polynomial order. rIGA decreases the overall cost to solve the system of equations when using direct solvers by reducing the inter-element continuity while controlling the total number of degrees of freedom added to the system. More precisely, we reduce the continuity along the inter-subdomains boundaries of the system that result from the recursive partitioning of the mesh. The continuity reduction is performed in such a way that it becomes zero in between the subdomains, so the interconnection is weakened (Figure 8). The reduction of continuity narrows the separators that interconnect the subdomains in order to make the partial matrix factorizations inexpensive. Unfortunately, the reduction of continuity increases the number of degrees of freedom (Figure 9) that needs to be controlled.

In order to control the overhead paid due to the higher number of degrees of freedom, we perform localized reductions of continuity. The optimal continuity reduction decreases the total cost of performing LU factorization

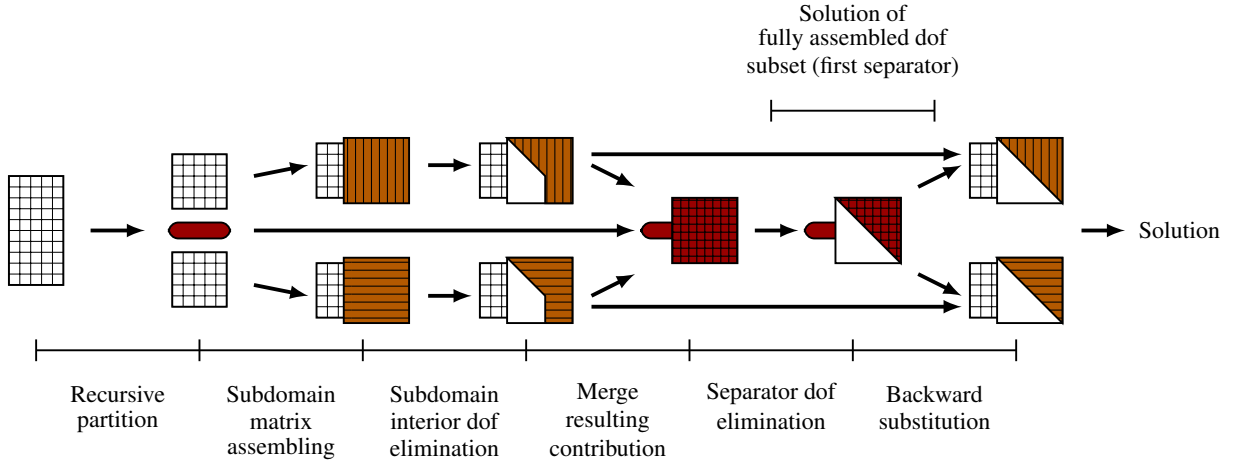


Figure 6: Factorization procedure for a 2D system recursively partitioned into two subdomains. dof stands for degrees of freedom.

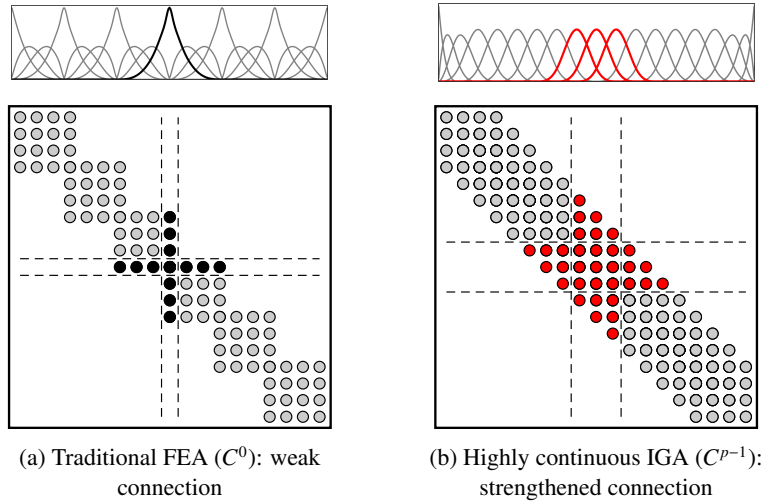


Figure 7: Illustration of a separator that interconnects two 1D subdomains (using $p = 3$ basis functions). Higher continuous discretizations (C^{p-1} system) involve wider separators and more dof to connect the subdomains. Traditional FEA (C^0 system) benefits from narrow separators.

while keeping the total number of degrees of freedom under control, since a larger increment in the number of degrees of freedom would involve a higher factorization cost. To make the problem tractable, we reduce the search of optimal discretizations (in terms of minimizing the number of FLOPs needed to perform the factorization) to C^{p-1} discretizations enriched by an arbitrary number of C^0 -separators.

Since the tensor product structure used in NURBS-based IGA limits the continuity reduction over local mesh zones, we adopt a simple implementation that consists of reducing the continuity over hyperplanes that cross the entire mesh in a single direction. The hyperplanes correspond to certain subdomains boundaries (separators). Figure 10 illustrates the structure of this implementation for a 2D problem.

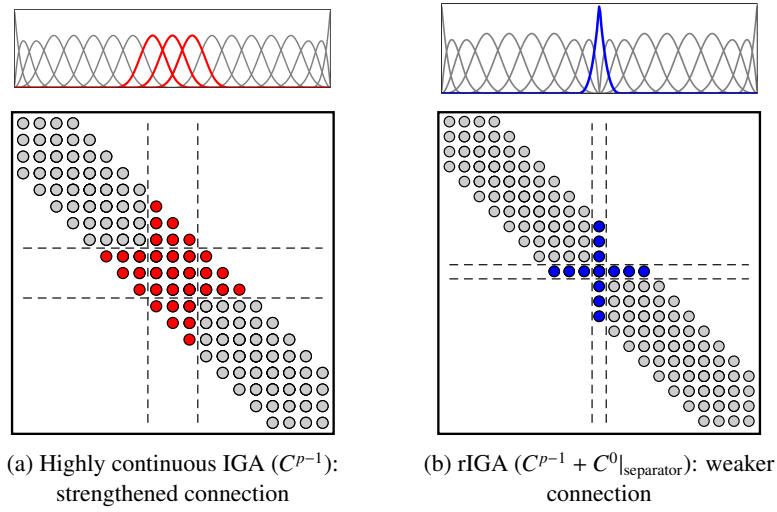


Figure 8: Illustration of a separator that interconnects two 1D subdomains (using $p = 3$ basis functions). rIGA discretization involves narrower separators than IGA but it increases the number of the total system dof.

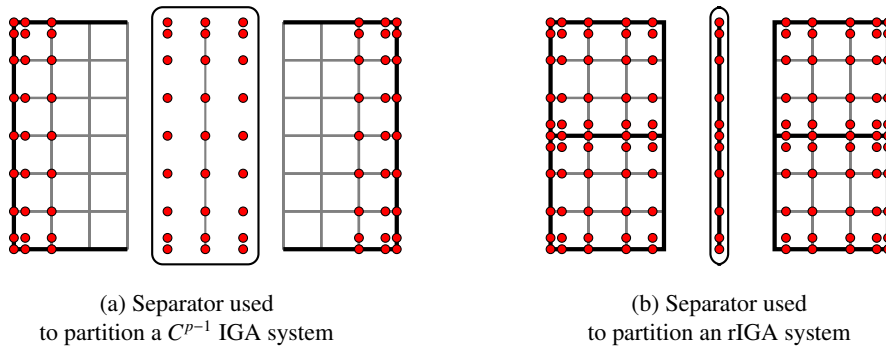


Figure 9: Illustration of the width reduction and the increment in the number of unknowns of a separator used to interconnect two subdomains that result from partitioning a 2D system with 6×6 elements and polynomial order $p = 3$. In this example, rIGA introduces two C^0 -separators.

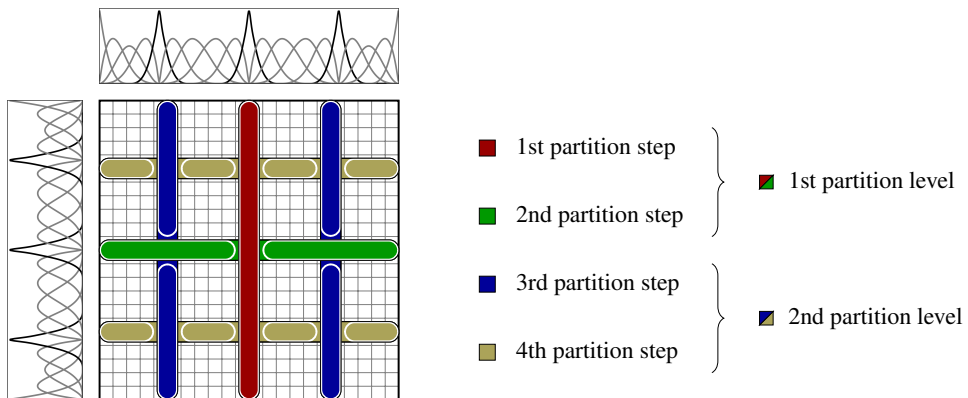


Figure 10: Tensor product based rIGA in a 2D system.

5. Theoretical estimation of the direct solver computational cost

This section provides theoretical estimates of the number of Floating Points Operations (FLOPs) required to solve an rIGA system using a direct solver. In particular, we explain why rIGA can be several times faster than both IGA and FEA when applied to a fixed mesh size with $p > 1$. For simplicity, we assume that the discretization has the same number of elements in each spatial dimension. Moreover, for deriving the theoretical estimates, we assume that the system is solved with a multi-frontal direct solver using a nested-dissection ordering.

5.1. Cost estimates for finite element and isogeometric analyses

The cost to solve a system of linear equations using the multifrontal solver was analyzed in [3, 5, 6]. This cost consists of three parts. The first part considers the matrix reordering cost. The remaining parts are the costs to perform the matrix factorization and backward substitution. We derive the theoretical estimates assuming that the matrix reordering and backward substitution costs are negligible, which is always the case for moderate to large size systems of equations.

In the multifrontal direct solver, the matrix decomposition procedure consists of the partial elimination of the subsets of degrees of freedom, either subdomains or separators. The cost to perform a partial LU (Cholesky) factorization of a dense matrix is $\mathcal{O}(q^3)$, with q being the size of the subsets of degrees of freedom. The total cost of the matrix decomposition is obtained by adding the contribution of all partial factorizations (Figure 11).

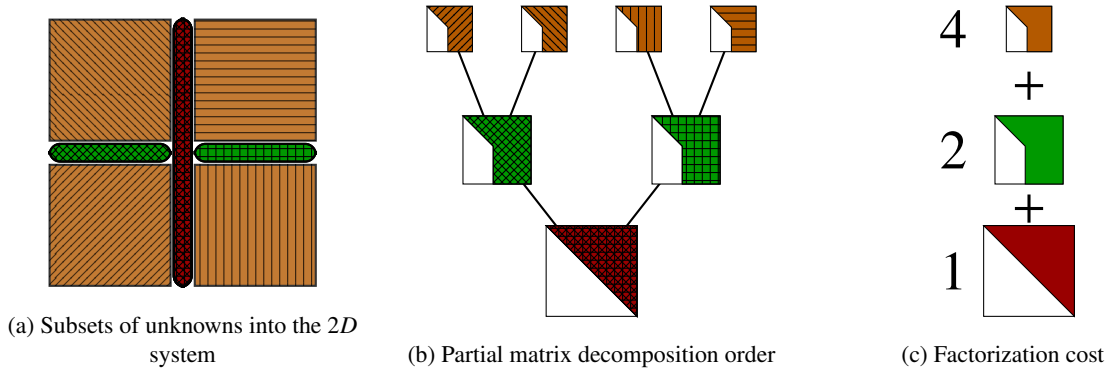


Figure 11: Illustration of the elimination of unknowns of a 2D system with one level of partition (two steps of partition). The cost of factorization is the sum of all partial decompositions.

The size of the separators (number of unknowns) is

$$q_{sep} = \mathcal{O}\left(N^{(d-1)/d}(k+1)\right),$$

where N is the total number of degrees of freedom, d is the dimension, and k is the inter-elemental continuity. We denote the separator continuity k . Thus, $(k+1)$ stands for the separator width, which is 1 in traditional FEA and p for IGA. The size of the minimal subdomains (leaves of the tree, see [41]) is equal to the number of bubble basis functions that every subdomain contains. For traditional FEA, the subsystem size is $q_{sub} = (p-1)^d$, since each subdomain corresponds to one element. In highly continuous IGA, the minimal subdomain size is $q_{sub} \approx 1$ since for collections of $p+1$ elements in each dimension, one basis function can be eliminated. See [3] for a detailed description of the process.

The cost of matrix factorization is then given by

$$\begin{aligned} \sum_{\eta_{sep}} \mathcal{O}\left((q_{sep})^3\right) + \eta_{sub} \cdot \mathcal{O}\left((q_{sub})^3\right) &= \sum_{\eta_{sep}} \mathcal{O}\left(\left(N^{(d-1)/d}(k+1)\right)^3\right) + \eta_{sub} \cdot \mathcal{O}\left((q_{sub})^3\right) \\ &= \underbrace{\mathcal{O}\left(\left(N^{(d-1)/d}(k+1)\right)^3\right)}_{\text{Skeleton}} + \underbrace{\eta_{sub} \cdot \mathcal{O}\left((q_{sub})^3\right)}_{\text{Interior dof}}, \quad [\text{FLOPs}] \end{aligned} \quad (1)$$

where η_{sep} is the number of separators used to partition the system, and η_{sub} is the number of subdomains. The first term corresponds to the skeleton cost, that is, the cost to eliminate the degrees of freedom of the separators while the last term is the cost of static condensation, i.e., the cost of eliminating the interior subdomain degrees of freedom.

The cost estimates for traditional FEA and highly continuous IGA become:

$$\text{FEA: } \mathcal{O}\left(\left(N^{(d-1)/d}\right)^3\right) + N_e(p-1)^{3d} \approx \mathcal{O}\left(N^{3(d-1)/d}\right) + \mathcal{O}\left(Np^{2d}\right), \quad [\text{FLOPs}] \quad (2)$$

$$\text{IGA: } \mathcal{O}\left(\left(N^{(d-1)/d}p\right)^3\right) + \eta_{sub}\mathcal{O}(1) \approx \mathcal{O}\left(N^{3(d-1)/d}p^3\right), \quad [\text{FLOPs}] \quad (3)$$

where N_e is the number of elements, and the number of degrees of freedom in FEA is $N = \mathcal{O}(N_e p^d)$. These cost estimates match with those derived in [5] and presented here in Table 1. Defining $n_e = N_e^{1/d}$ as the number of elements in each spatial direction, the number of degrees of freedom is given by

$$\text{FEA: } N = (n_e p + 1)^d$$

$$\text{IGA: } N = (n_e + p)^d$$

and the cost estimates to solve a problem with a mesh with a fixed number of elements and a given polynomial order are

$$\text{FEA: } \approx \mathcal{O}\left((n_e p + 1)^{3(d-1)}\right) + \mathcal{O}\left(n_e^d p^{3d}\right), \quad [\text{FLOPs}]$$

$$\text{IGA: } \approx \mathcal{O}\left((n_e + p)^{3(d-1)} p^3\right). \quad [\text{FLOPs}]$$

In $2D$, the solution cost of both C^0 FEA and C^{p-1} IGA is similar to each other (up to lower order terms), while in $3D$, C^0 FEA is $\mathcal{O}(p^3)$ more expensive than C^{p-1} IGA. Indeed, the elimination of the skeleton degrees of freedom in $3D$ requires a large number of FLOPs, specifically, $\mathcal{O}(p^3)$ times the number of FLOPs required to solve the corresponding C^{p-1} IGA system.

5.2. Cost estimate for rIGA

To compute the cost to solve a refined isogeometric system we assume that the factorization is performed in two steps (Figure 12). First, we eliminate the degrees of freedom contained in the C^{p-1} subsystems (macro-elements). Then, we eliminate the remaining degrees of freedom associated to the C^0 -separators.

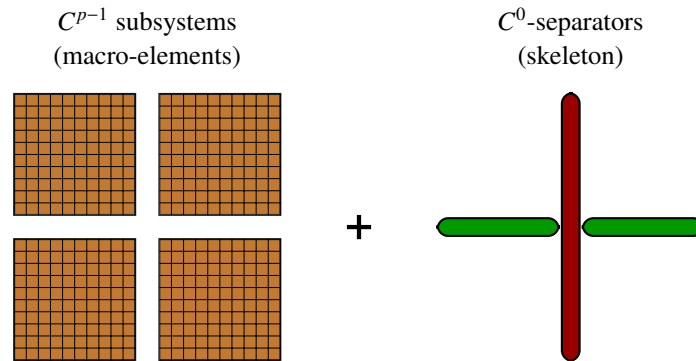


Figure 12: Illustration of C^{p-1} subsystems (macro-elements) and C^0 -separators (skeleton) that result from partitioning a $2D$ system.

The total cost to factorize an rIGA system is given by (1) and can be expressed as

$$\text{rIGA: } \underbrace{\eta_{m-e} \cdot \mathcal{O}\left(\left(q_{m-e}\right)^3\right)}_{C^{p-1} \text{ macro-elements}} + \underbrace{\mathcal{O}\left(\left(N^{(d-1)/d}(k+1)\right)^3\right)}_{C^0 \text{ skeleton}},$$

where the coefficient k in the skeleton term is zero due to the reduction of continuity. At the ℓ -th partition level, rIGA splits the original system into $\eta_{m-e} = 2^{d\ell}$ macro-elements of size $N/2^{d\ell}$. The cost of the macro-elements is the same as for the C^{p-1} IGA systems we derived in (3). Then, $q_{m-e} = (N/2^{d\ell})^{(d-1)/d} p$, and the total cost becomes

$$\underbrace{2^{d\ell} \cdot \mathcal{O}\left(\left(\frac{N}{2^{d\ell}}\right)^{3(d-1)/d} p^3\right)}_{C^{p-1} \text{ macro-elements}} + \underbrace{\mathcal{O}\left(N^{3(d-1)/d}\right)}_{C^0 \text{ skeleton}}.$$

The continuity reduction enriches the resulting rIGA system. Every cut that splits the original system adds $(p-1)$ degrees of freedom in the spatial direction perpendicular to the cut. For instance, in Figure 12, the vertical cut (that corresponds to the C^0 -separator) adds $(p-1)$ degrees of freedom in the horizontal direction, increasing the system size to $(n_e + p + (p-1))(n_e + p)$. The horizontal cut adds $(p-1)$ new unknowns in the vertical direction, which increases the total number of degrees of freedom in the system to $(n_e + p + (p-1))^2$. Thus, the number of degrees of freedom for a given mesh ($n_e = 2^s : s \in \mathbb{N}^+$) partitioned into $2^{d\ell}$ macro-elements is

$$\begin{aligned} N = n^d &= (n_e + p + \underbrace{(2^\ell - 1)(p-1)}_{\text{Enrichment}})^d \\ &= (2^s + p + (2^\ell - 1)(p-1))^d, \end{aligned}$$

where $(2^\ell - 1)$ is the number of cuts performed in each spatial dimension, and the total cost to factorize an rIGA system is given by

$$\begin{aligned} \theta &= \theta_{\text{macro-element}} + \theta_{C^0\text{-separator}} \\ &= 2^{d\ell} \cdot \mathcal{O}\left(\left(\frac{n}{2^\ell}\right)^{3(d-1)} p^3\right) + \mathcal{O}\left(n^{3(d-1)}\right) \\ &= \left(\underbrace{2^{(3-2d)\ell} \cdot \mathcal{O}\left(n^{3(d-1)} p^3\right)}_{C^{p-1} \text{ macro-elements contribution}} + \underbrace{\mathcal{O}\left(n^{3(d-1)}\right)}_{C^0\text{-separators contribution}} \right). \quad [\text{FLOPs}] \end{aligned}$$

6. Model Problem

We use the Laplace equation in (4) as a model problem to exemplify the performance of rIGA.

$$\left\{ \begin{array}{l} \text{Find } u \text{ such that :} \\ \nabla \cdot (\nabla u) = 0 \quad \text{in } \Omega \\ u = 1 \quad \text{on } \partial\Omega_1 \\ u = 0 \quad \text{on } \partial\Omega_0 \\ \nabla u \cdot n = 0 \quad \text{on } \partial\Omega_w, \end{array} \right. \quad (4)$$

where $\Omega = [0, 1]^d$, with d being the dimension, $\partial\Omega_w \cup \partial\Omega_0 \cup \partial\Omega_1 = \partial\Omega$, $\partial\Omega_w \cap \partial\Omega_0 = \emptyset$ and $\partial\Omega_w \cap \partial\Omega_1 = \emptyset$. The problem domains for 2D and 3D are illustrated in Figure 13.

7. Implementation details

We use unmapped B-splines to build the model problem assuming a unitary domain. The tensor product of the unmapped B-splines defines the domain of the problem, obtaining a regular and structured mesh. The meshes

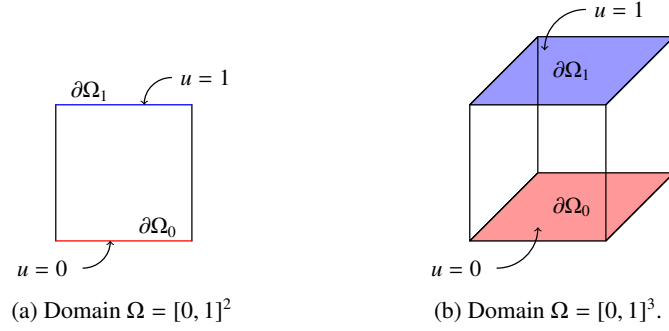


Figure 13: Model problem domain.

| n_e | Dimension | |
|-------|-------------------|-----------------|
| | $d = 2$ | $d = 3$ |
| | $512^2 (s = 9)$ | $32^3 (s = 5)$ |
| 2^s | $1024^2 (s = 10)$ | $64^3 (s = 6)$ |
| | $2048^2 (s = 11)$ | $128^3 (s = 7)$ |

Table 2: Mesh sizes used to solve the model problem.

incorporate the same number of elements in each spatial dimension, $n_e = 2^s$, in order to obtain a uniform structure when partitioning the system. Table 2 presents the implemented mesh sizes.

Four polynomial orders p ranging from 2 to 5 are used to perform the discretization. These polynomial orders are kept constant in each problem.

For every mesh size, we consider a range of cases with particular number of partition levels using C^0 -separators. The first case assumes no reduction of continuity (no level of partition uses C^0 -separators, which corresponds to the conventional tensor-product IGA), while the last one involves a reduction of the continuity along *all* the inter-elements boundaries (all levels of partition employ C^0 -separators, which correspond to the conventional FEA). This allows us to analyze the impact of the local reduction of continuity in the computational cost, besides finding the optimal continuity reduction for rIGA.

Our implementation is based on library PetIGA, which is a high-performance software platform for IGA [42] based on PETSc [43, 44]. PetIGA has been used to model many engineering applications since its inception [3, 5, 8, 9, 42, 45–49].

The sequential version of the multifrontal solver MUMPS [50, 51] is used to solve the algebraic system resulting from the discretization. The automatic choice of partitioning technique made by MUMPS is used, resulting in a METIS implementation for all the cases.

All computational tests are solved sequentially on TACC Stampede system. Each node is outfitted with turbo boost 2.7 GHz cores (up to peak 3.5 GHz in turbo mode) and 1TB of memory (URL: <http://www.tacc.utexas.edu>).

8. Numerical Results

In this section, we present the numerical results. We report the FLOPs, computational times (in seconds) and memory requirements (that correspond to the non-zero entries in factors L and U expressed in Mbytes) for all the analyzed variations in mesh size and polynomial order.

8.1. Fit of estimates

In order to fit the theoretical estimates with the computed number of FLOPS required to factor the systems, we introduce two constants, namely, \mathbf{A} and \mathbf{B} , as follows:

$$\theta = \mathbf{A} 2^{(3-2d)\ell} \left(n^{3(d-1)} p^3 \right) + \mathbf{B} \left(n^{3(d-1)} \right)$$

$$2D: \quad \theta = \left(\mathbf{A} 2^{-\ell} p^3 + \mathbf{B} \right) \left(2^s + p + (2^\ell - 1)(p - 1) \right)^3 + L.O.T., \quad [\text{FLOPs}]$$

$$3D: \quad \theta = \left(\mathbf{A} 2^{-3\ell} p^3 + \mathbf{B} \right) \left(2^s + p + (2^\ell - 1)(p - 1) \right)^6 + L.O.T., \quad [\text{FLOPs}]$$

where *L.O.T.* stands for lower order terms. For each p , we estimate \mathbf{A} and \mathbf{B} by solving a least square fitting problem for a large mesh size. Specific values of \mathbf{A} and \mathbf{B} are presented in Table 3. Constants \mathbf{A} and \mathbf{B} are almost independent of

| Constants | | Polynomial order p | | | |
|-----------|--------------|----------------------|------|------|------|
| | | 2 | 3 | 4 | 5 |
| 2D | \mathbf{A} | 24.5 | 21.5 | 21.5 | 20.0 |
| | \mathbf{B} | 26.5 | 24.5 | 22.5 | 20.0 |
| 3D | \mathbf{A} | 6.6 | 6.3 | 6.0 | 5.7 |
| | \mathbf{B} | 7.8 | 7.2 | 7.2 | 7.2 |

Table 3: Fitting constants computed for every polynomial order in 2D and 3D.

the polynomial order, but depend on the problem dimension. Both the contribution of forming the Schur complements and the total number of FLOPs performed by LAPACK to factorize the system [52] are included in these constants.

8.2. Number of FLOPs

Figures 14 and 15 show the number of FLOPs required to factor the algebraic system for 2D and 3D, respectively. The number of FLOPs is plotted with respect to the macro-element size (n_{m-e}^d). The macro-element size diminishes as we add C^0 -separators. Thus, the C^0 FEA case corresponds to a macro-element size equal to 1, while the C^{p-1} IGA case corresponds to the largest macro-element size. These numerical results confirm the following:

- The theoretical estimates approximate well the numerical results.
- The optimal discretization (in terms of minimizing the number of FLOPs of the direct solver) is in an intermediate stage between uniformly global continuity C^0 and C^{p-1} . Neither of the two extreme cases provides optimal discretizations.
- The reduction factor in the number of FLOPs by using an optimal discretization is approx. $(p + 1)^2$ for 2D and p^2 for 3D.

In Figure 14, rIGA shows a maximum reduction factor of the number of FLOPs of 40 with respect to C^{p-1} IGA. This reduction factor is obtained when the model problem is solved with $N_e = 2048^2$ and a polynomial order $p = 5$. Further numerical results showed that when we solve the problem with $N_e = 2048^2$ and a polynomial order $p = 9$, rIGA reduces the number of FLOPs by a factor of 70 with respect to C^{p-1} IGA, and 90 with respect to C^0 FEA.

In 3D, the maximum reduction factor of the number of FLOPs observed when using rIGA is about 26 with respect to C^{p-1} IGA, and even larger with respect to C^0 FEA. This reduction factor was analytically obtained because the C^{p-1} IGA case cannot be computed due to memory limitations.

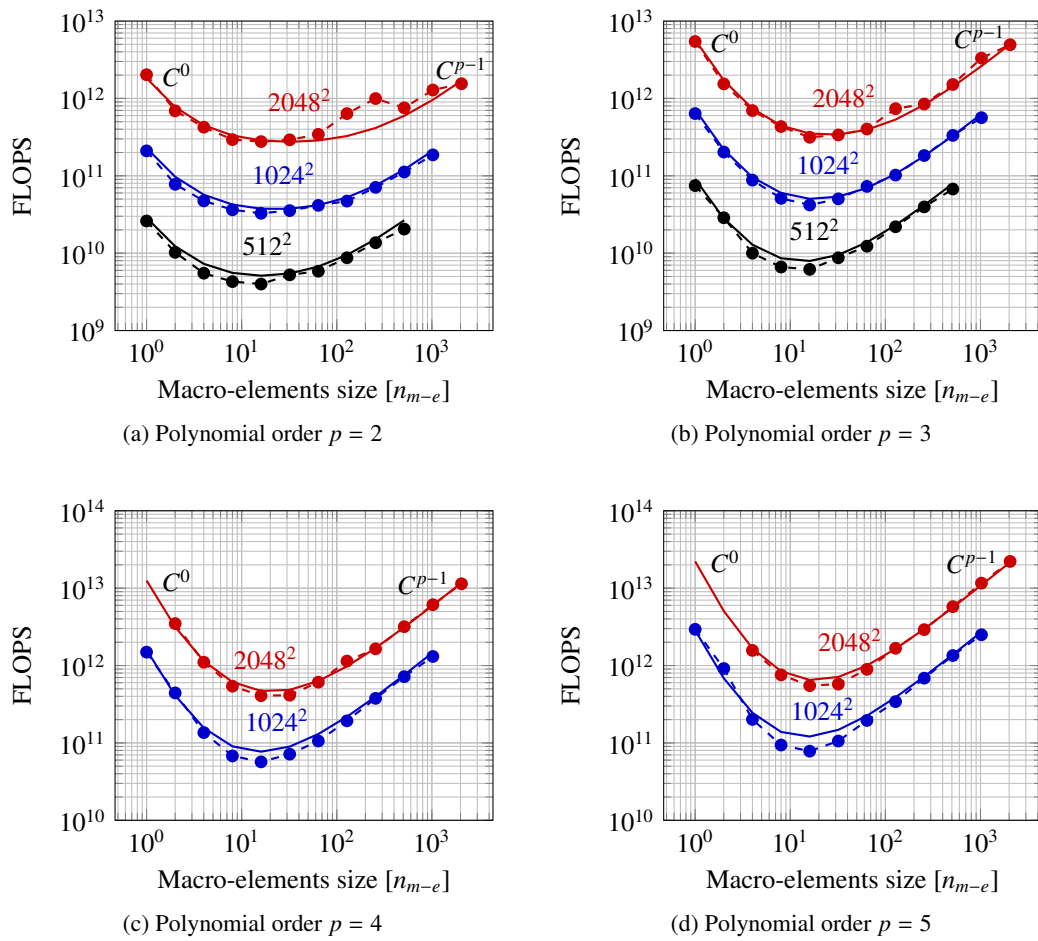


Figure 14: Number of FLOPs required to eliminate the degrees of freedom in the 2D model problem. The solid lines (—) correspond to the theoretical estimates and the dashed lines with rounded markers (-o-) represent the numerical results.

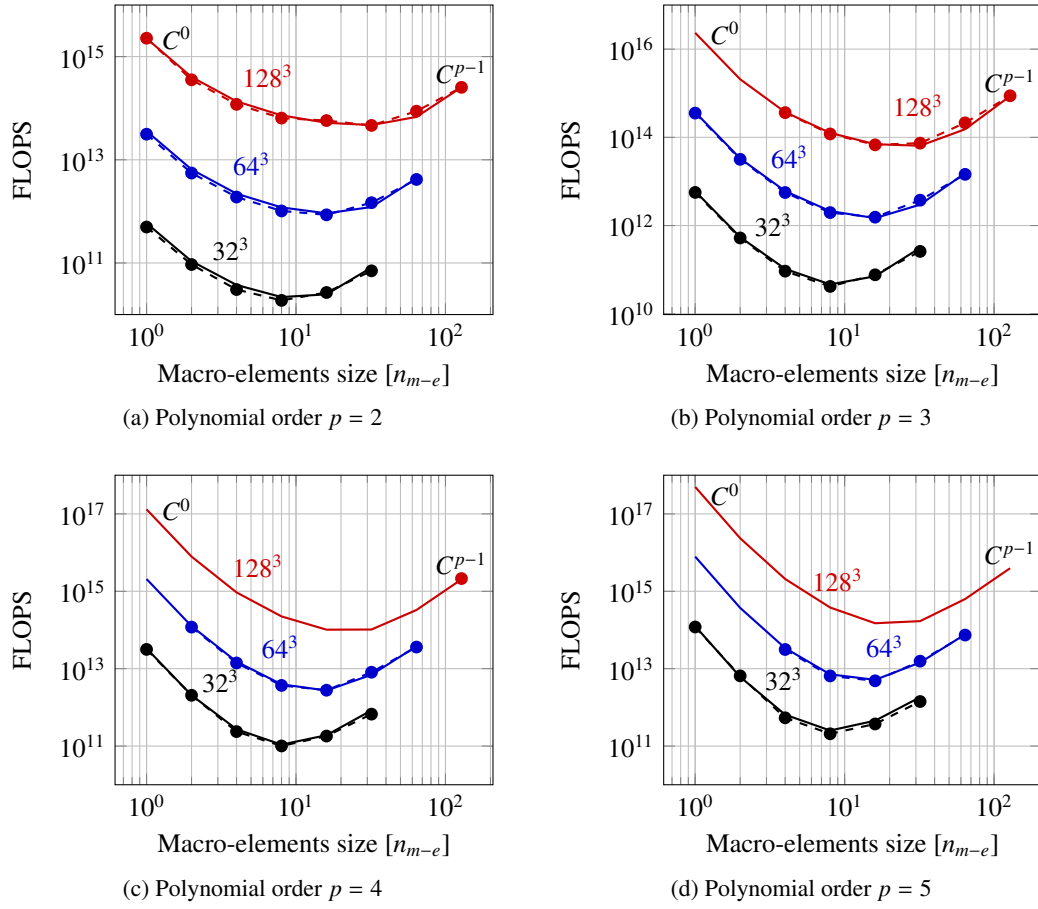


Figure 15: Number of FLOPs required to eliminate the degrees of freedom in the 3D model problem. The solid lines (—) correspond to the theoretical estimates and the dashed lines with rounded markers (—o—) represent the numerical results.

8.3. Computational times

Figures 16 and 17 provide computational times for 2D and 3D. The plots present the solution time with respect to the macro-element size. Table 4 quantifies the computational times for C^0 FEA, C^{p-1} IGA and the optimal case (rIGA) for every polynomial order in a fine mesh.

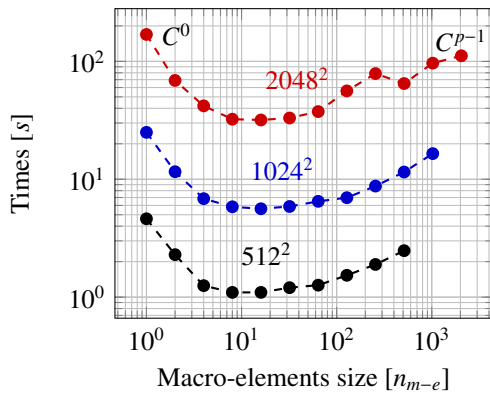
| Mesh size | $N_e = 2048^2$ | | | | $N_e = 128^3$ | | | |
|-----------|----------------|---------|--------------|-------|---------------|-------|--------------|-------|
| | C^{p-1} | C^0 | Optimal case | Gain | C^{p-1} | C^0 | Optimal case | Gain |
| 2 | 1.1e+02 | 1.7e+02 | 3.2e+01 | 3.44 | 1.4e+04 | *** | 3.4e+03 | 4.28 |
| 3 | 3.2e+02 | 4.2e+02 | 3.7e+01 | 8.65 | 5.4e+04 | *** | 3.9e+03 | 13.69 |
| 4 | 6.9e+02 | *** | 4.6e+01 | 15 | 2.1e+03 | *** | 1.9e+02 | 11.32 |
| 5 | 1.3e+03 | *** | 5.9e+01 | 22.03 | 4.2e+03 | *** | 3.2e+02 | 13.08 |
| 9 | 6.7e+03 | *** | 1.8e+02 | 37.2 | - | - | - | - |

Note: * * * The computation exceeds the maximum available physical memory, thus the solution failed (out of memory).

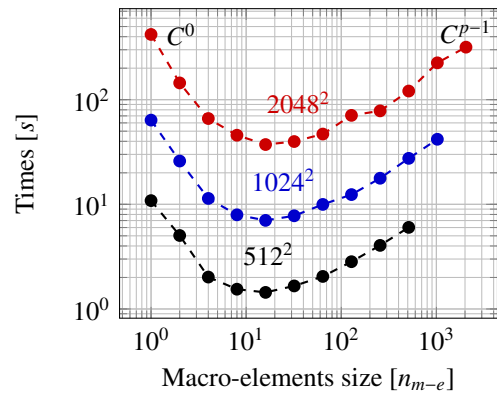
Blue stands for a mesh size $N_e = 64^3$.

Table 4: Computational time (in seconds) for the model problem discretized with the asymptotic mesh and maximum gain of the optimal case with respect to the C^{p-1} IGA discretization.

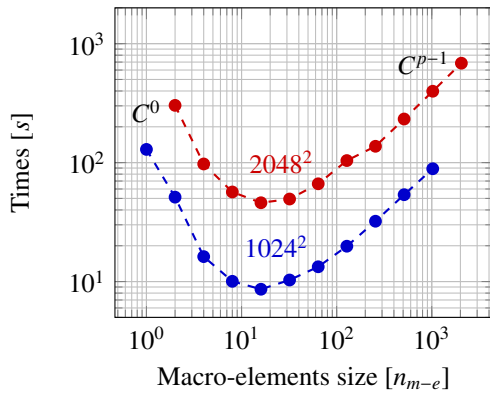
The system that results from discretizing the model problem with $N_e = 2048^2$ and a polynomial order $p = 9$ reports the maximum gain in solution time for 2D (corresponding to a time factor of 37.2). The rIGA optimal system requires almost 3 minutes to be solved, while the system obtained from C^{p-1} IGA is solved in approximately 2 hours. In 3D, the system discretized with $N_e = 128^3$ and polynomial order $p = 3$ reports a time reduction factor of 13.69. Thus, the problem is solved in 1 hour with rIGA instead of the 15 hours required when using C^{p-1} IGA. This is the maximum reproducible gain we obtained in 3D, since the C^{p-1} cases solved with a mesh size $N_e = 128^3$ and polynomial orders $p = 4$ and $p = 5$ could not be resolved (due to memory limitations). Nonetheless, theoretical estimates clearly indicate that gains associated to rIGA vastly increase as we increment p .



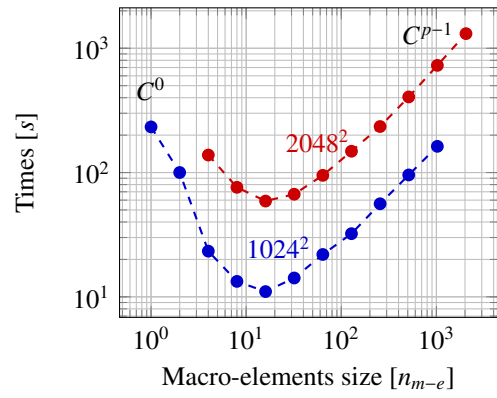
(a) Polynomial order $p = 2$



(b) Polynomial order $p = 3$

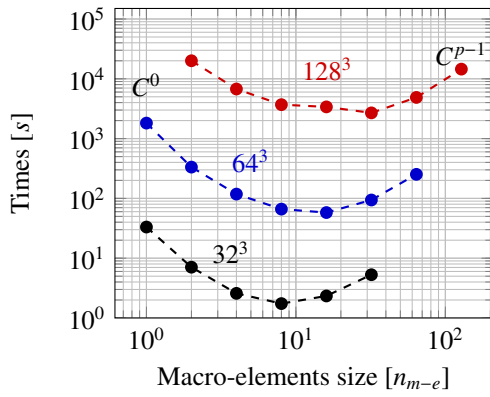


(c) Polynomial order $p = 4$

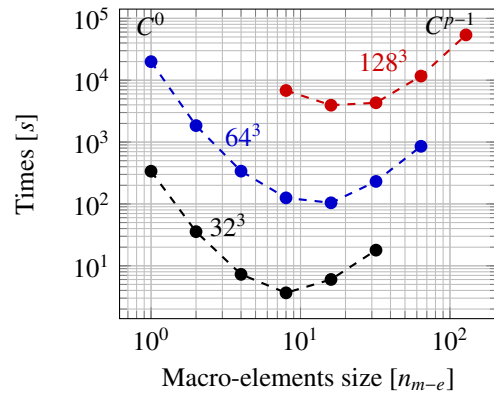


(d) Polynomial order $p = 5$

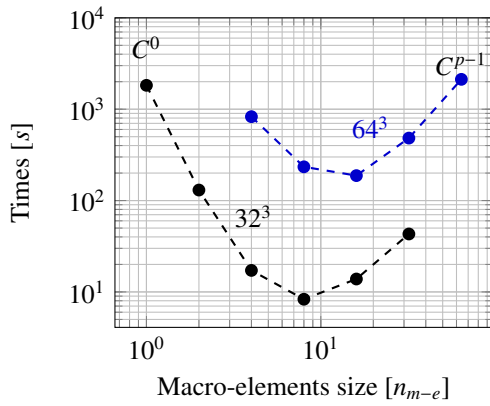
Figure 16: Computational time (in seconds) to factorize the 2D model problem.



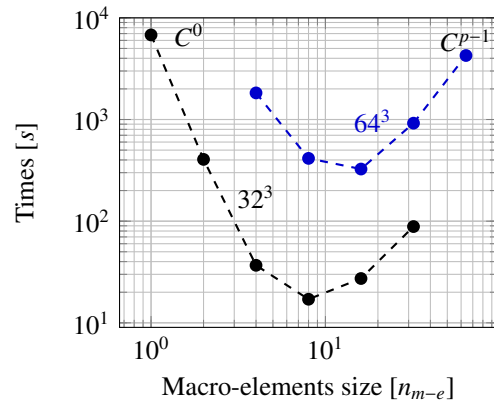
(a) Polynomial order $p = 2$



(b) Polynomial order $p = 3$



(c) Polynomial order $p = 4$



(d) Polynomial order $p = 5$

Figure 17: Computational time (in seconds) to factorize the 3D model problem.

8.4. Memory requirements

Figures 18 and 19 show the memory requirements (in Mbytes) for 2D and 3D, respectively. We display the memory usage reported by the multifrontal solver MUMPS along with the theoretical estimation computed with equations 5 and 6, which are derived in Appendix A.

$$2D: \quad \theta = O\left(\left(\frac{3}{2} \log_2(2^s)^2 p^2 + \frac{3}{2} \log_2(2^{-i})\right) n^2\right) + L.O.T., \quad [Mbytes] \quad (5)$$

$$3D: \quad \theta = O\left(\left(\frac{7}{2} (2^{-i}) p^2 + \frac{7}{2} (1 - 2^{-i})\right) n^4\right) + L.O.T. \quad [Mbytes] \quad (6)$$

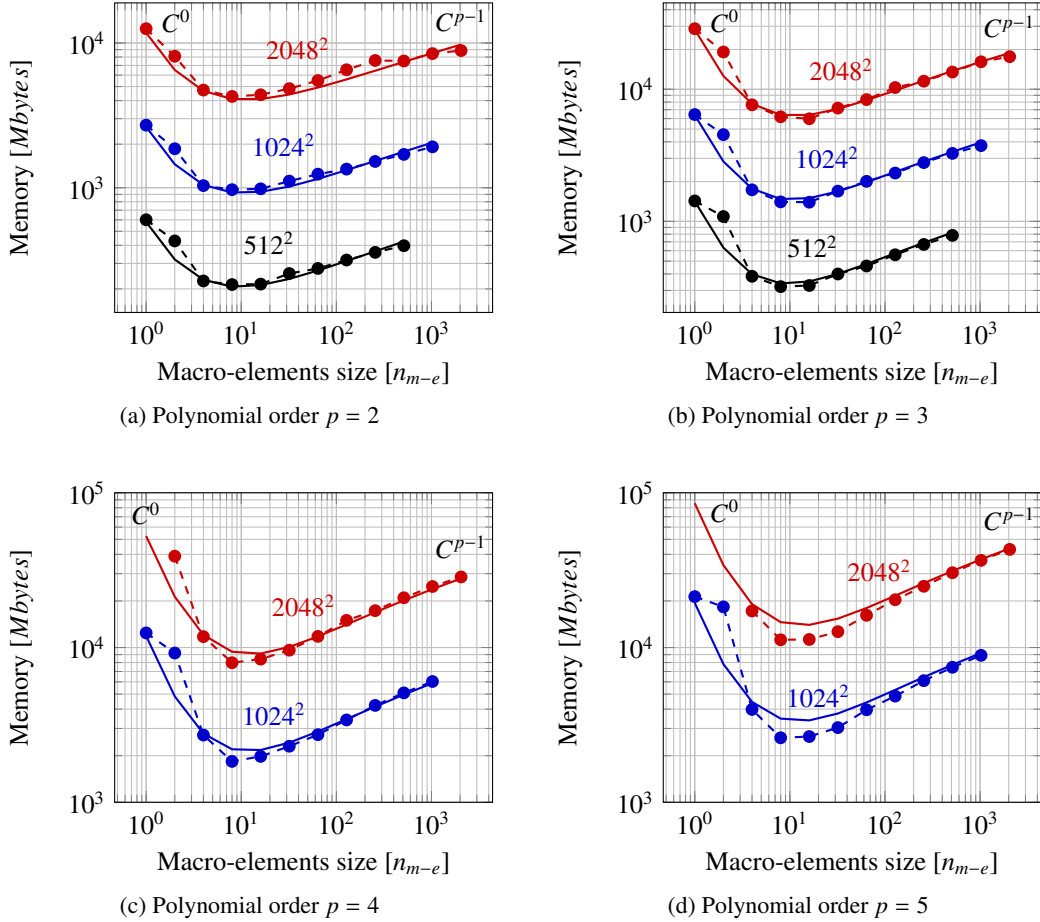


Figure 18: Memory requirements for the factorization of the 2D model problem. The solid lines (—) correspond to the theoretical estimates and the dashed lines with rounded markers (-o-) represent the numerical results.

In 2D, the maximum reduction in memory usage among those considered in Figure 18 corresponds to the case with $N_e = 2048^2$ and polynomial order $p = 5$. The factor of memory reduction is 4. In 3D, the system discretized using $N_e = 128^3$ and polynomial orders $p = 5$ reports the maximum reduction in memory requirements. In this case, the theoretical factor is 2.9. Systems discretized with higher polynomial orders involve larger reduction of memory usage.

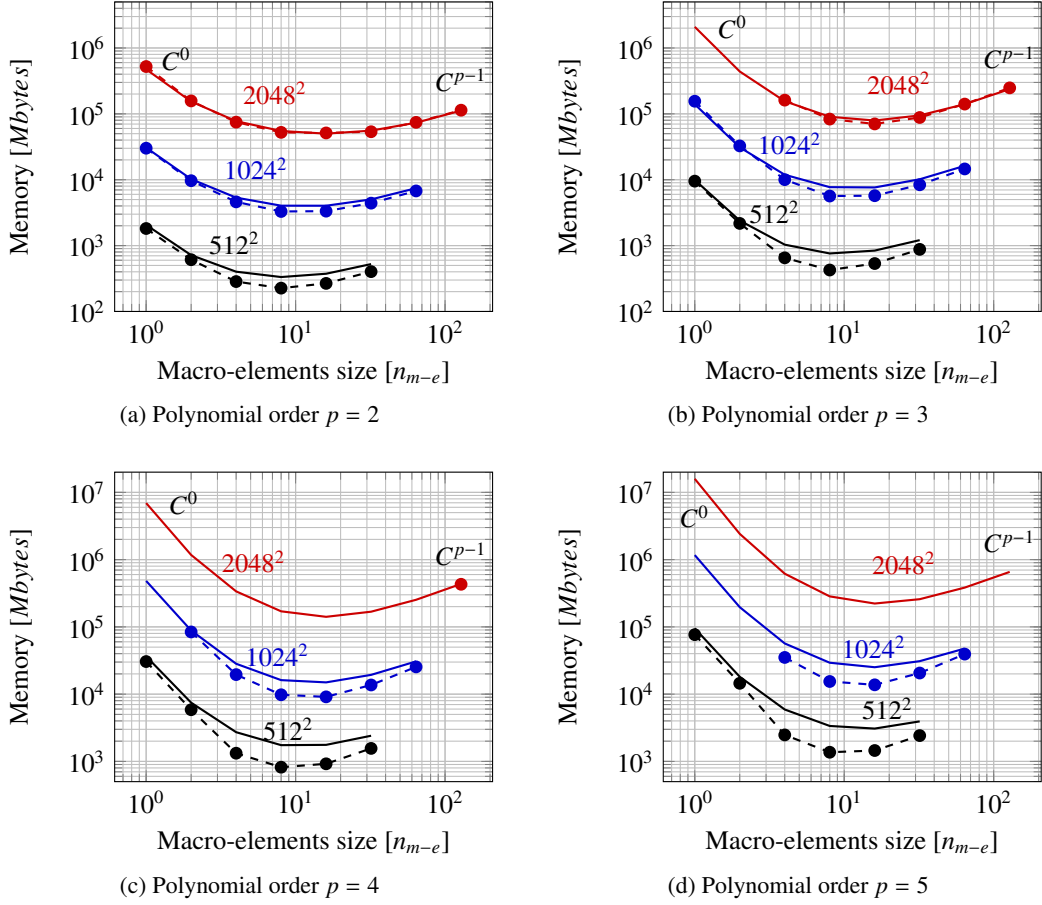


Figure 19: Memory requirements for the factorization of the 3D model problem. The solid lines (–) correspond to the theoretical estimates and the dashed lines with rounded markers (–o–) represent the numerical results.

In 3D, the discrepancies observed for high p and low n between the theoretical estimates and the recorded data for memory usage are due to the exclusion of lower order terms from the theoretical estimates. Such lower order terms (in the 3D memory estimates) are only one power of n smaller than the dominant cost, but they are multiplied by a larger power of p . Thus, they become dominant on the pre-asymptotic regime (low n and large p). This situation does not occur in the FLOPs estimates, because for that case, the dominant cost is n^3 times larger than lower order terms, and thus, estimates rapidly arrive to the asymptotic regime.

9. Conclusions

In this paper, we propose a refined isogeometric analysis (rIGA) method to solve problems governed by linear partial differential equations. Starting from a highly continuous isogeometric analysis discretization, our strategy reduces the continuity over certain hyperplanes that act as separators during the elimination of degrees of freedom. To illustrate the impact of the continuity reduction on a direct solver, we report the floating point operations (FLOPs), computational times, and memory requirements to solve linear systems resulting from the refined isogeometric discretization with structured meshes and uniform polynomial orders. Neither traditional finite elements nor isogeometric analysis provides the optimal number of FLOPs for a fixed mesh size. The optimal is achieved by the rIGA method we introduce herein.

The refined isogeometric discretization (rIGA) solves problems (with large mesh sizes and high polynomial orders) approximately p^2 times faster than problems discretized using C^{p-1} isogeometric analysis (IGA), and the gains with

respect to finite element analysis (FEA) are even larger, especially in 3D. The memory requirements also decrease.

In $2D$, the cases solved using macro-elements (subdomains) with a size of 16^2 elements report the lowest computational times in most of the configurations tested. In $3D$, a wider range of macro-element size options is observed. For small mesh sizes ($N_e = 32^3$), optimal macro-elements consist of 8^3 elements, while as the mesh size increases to $N_e = 128^3$, the optimal macro-element size tends to 16^3 elements.

The maximum polynomial order that we use to solve the model problem is $p = 9$. The optimal results, in this case, report a gain factor with respect to IGA (C^{p-1}) of almost 70 times in terms of FLOPS and more than 35 times in terms of time. The theoretical gain factor of rIGA with respect to FEA (C^0) is almost 90 times in FLOPS. In $2D$, the problem that we solve in 3 minutes with rIGA, requires approximately 2 hours to be solved with IGA.

In $3D$, the maximum reproducible gain with respect to C^{p-1} is 13.69, which corresponds to $N_e = 128^3$ and polynomial order $p = 3$. In this case, the problem that we solve in one hour with rIGA requires approximately 15 hours to be solved with IGA, and more than 100 hours if one employs FEA.

Since the optimal refined isogeometric discretization consists of an enriched/nested space with respect to C^{p-1} IGA, the best approximation error is improved by definition. Similarly, the best approximation error of the corresponding FEA discretization is smaller than that of rIGA. The total numerical error for stable elliptic problems improves when going from IGA to rIGA discretizations. For the case of hyperbolic and parabolic problems, stability may play a crucial role on the total approximation error. A detailed error analysis in hyperbolic and parabolic systems will be performed in a future work.

Possible lines of future research include the study of non-tensor product refined isogeometric analysis discretizations, including those incorporating T-splines. The savings for this strategy may vary depending upon the particular discretization. The main idea is to employ C^0 T-splines on the top level of a given macro-element in order to *separate* the interior of the possibly locally-refined macro-element from the rest of the computational domain.

We are also working on a parallel implementation of the refined isogeometric analysis that will be analyzed in a forthcoming publication. The use of refined isogeometric analysis is expected to be beneficial in terms of distributed memory parallel computations, since separators diminish the amount of information that needs to be shared among neighbouring processors, thus, minimizing the communication cost and increasing its parallel scalability.

A third line of future research consists of expanding the refined isogeometric analysis strategy to the case of iterative solvers by using an static-condensation technique at the macro-element level.

Acknowledgement

David Pardo and Daniel Garcia have received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 644602, the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824-P and MTM2016-76329-R, the BCAM Severo Ochoa accreditation of excellence SEV-2013-0323, and the Basque Government through the BERC 2014-2017 program, the Consolidated Research Group Grant IT649-13 on "Mathematical Modeling, Simulation, and Industrial Applications (M2SI)", and the ICERMAR Project KK-2015/0000097. The work of Maciej Paszyński has been supported by National Science Centre, Poland, grant no. DEC-2015/17/B/ST6/01867. This publication was made possible in part by a National Priorities Research Program grant 7-1482-1-278 from the Qatar National Research Fund (a member of The Qatar Foundation) and the Center for Numerical Porous Media at King Abdullah University of Science and Technology (KAUST). 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 Advance Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported in this paper.

References

- [1] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, New York, Mi, 2000.
- [2] T. J. R. 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 (39-41) (2005) 4135 – 4195. doi:http://dx.doi.org/10.1016/j.cma.2004.10.008.
- [3] 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. doi:http://dx.doi.org/10.1016/j.cma.2011.11.002.
- [4] I. S. Duff, J. K. Reid, *The Multifrontal Solution of Indefinite Sparse Symmetric Linear*, *ACM Trans. Math. Softw.* 9 (3) (1983) 302–325. doi:10.1145/356044.356047.
- [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. doi:10.1002/nme.4769.
- [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. doi:http://dx.doi.org/10.1016/j.procs.2011.04.201.
- [7] H. Gómez, V. M. Calo, Y. Bazilevs, T. J. R. Hughes, *Isogeometric analysis of the Cahn-Hilliard phase-field model*, *Computer Methods in Applied Mechanics and Engineering* 197 (4950) (2008) 4333 – 4352. doi:http://dx.doi.org/10.1016/j.cma.2008.05.003.
- [8] P. A. Vignal, N. Collier, V. M. Calo, *Phase Field Modeling Using PetIGA*, *Procedia Computer Science* 18 (2013) 1614 – 1623. doi:http://dx.doi.org/10.1016/j.procs.2013.05.329.
- [9] P. Vignal, L. Dalcin, D. L. Brown, N. Collier, V. M. Calo, *An energy-stable convex splitting for the phase-field crystal equation*, *Computers & Structures* 158 (2015) 355 – 368. doi:http://dx.doi.org/10.1016/j.compstruc.2015.05.029.
- [10] Y. Bazilevs, V. M. Calo, Y. Zhang, T. J. R. Hughes, *Isogeometric Fluid-structure Interaction Analysis with Applications to Arterial Blood Flow*, *Computational Mechanics* 38 (4-5) (2006) 310–322. doi:10.1007/s00466-006-0084-3.
- [11] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, Y. Zhang, *Isogeometric fluid-structure interaction: theory, algorithms, and computations*, *Computational Mechanics* 43 (1) (2008) 3–37. doi:10.1007/s00466-008-0315-x.
- [12] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, T. J. R. Hughes, *An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves*, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 1005 – 1053. doi:http://dx.doi.org/10.1016/j.cma.2014.10.040.
- [13] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, G. Scovazzi, *Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows*, *Computer Methods in Applied Mechanics and Engineering* 197 (1-4) (2007) 173 – 201. doi:http://dx.doi.org/10.1016/j.cma.2007.07.016.
- [14] Y. Bazilevs, C. Michler, V. M. Calo, T. J. R. Hughes, *Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes*, *Computer Methods in Applied Mechanics and Engineering* 199 (13-16) (2010) 780 – 790. doi:http://dx.doi.org/10.1016/j.cma.2008.11.020.
- [15] K. Chang, T. Hughes, V. Calo, *Isogeometric variational multiscale large-eddy simulation of fully-developed turbulent flow over a wavy wall*, *Computers & Fluids* 68 (2012) 94–104. doi:http://dx.doi.org/10.1016/j.compfluid.2012.06.009.
- [16] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, S. Hulshoff, *The role of continuity in residual-based variational multiscale modeling of turbulence*, *Computational Mechanics* 41 (3) (2008) 371–378. doi:10.1007/s00466-007-0193-7.
- [17] Y. Bazilevs, I. Akkerman, *Large eddy simulation of turbulent TaylorCouette flow using isogeometric analysis and the residual-based variational multiscale method*, *Journal of Computational Physics* 229 (9) (2010) 3402 – 3414. doi:http://dx.doi.org/10.1016/j.jcp.2010.01.008.
- [18] Y. G. Motlagh, H. T. Ahn, T. J. R. Hughes, V. M. Calo, *Simulation of laminar and turbulent concentric pipe flows with the isogeometric variational multiscale method*, *Computers & Fluids* 71 (2013) 146 – 155. doi:http://dx.doi.org/10.1016/j.compfluid.2012.09.006.
- [19] P. N. Nielsen, A. R. Gersborg, J. Gravesen, N. L. Pedersen, *Discretizations in isogeometric analysis of NavierStokes flow*, *Computer Methods in Applied Mechanics and Engineering* 200 (45-46) (2011) 3242 – 3253. doi:http://dx.doi.org/10.1016/j.cma.2011.06.007.
- [20] A. Tagliabue, L. Dedé, A. Quarteroni, *Isogeometric Analysis and error estimates for high order partial differential equations in fluid dynamics*, *Computers & Fluids* 102 (2014) 277 – 303. doi:http://dx.doi.org/10.1016/j.compfluid.2014.07.002.
- [21] P. Vignal, A. Sarmiento, A. M. Crtes, 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. doi:http://dx.doi.org/10.1016/j.procs.2015.05.228.
- [22] P. Vignal, L. Dalcin, N. Collier, V. Calo, *Modeling Phase-transitions Using a High-performance, Isogeometric Analysis Framework*, *Procedia Computer Science* 29 (2014) 980 – 990. doi:http://dx.doi.org/10.1016/j.procs.2014.05.088.
- [23] J. A. Cottrell, T. J. R. Hughes, A. Reali, *Studies of refinement and continuity in isogeometric structural analysis*, *Computer Methods in Applied Mechanics and Engineering* 196 (4144) (2007) 4160 – 4183. doi:http://dx.doi.org/10.1016/j.cma.2007.04.007.
- [24] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. R. Hughes, *Isogeometric analysis of structural vibrations*, *Computer Methods in Applied Mechanics and Engineering* 195 (41-43) (2006) 5257 – 5296. doi:http://dx.doi.org/10.1016/j.cma.2005.09.027.
- [25] F. Auricchio, L. B. da Veiga, A. Buffa, C. Lovadina, A. Reali, G. Sangalli, *A fully locking-free isogeometric approach for plane linear elasticity problems: A stream function formulation*, *Computer Methods in Applied Mechanics and Engineering* 197 (1-4) (2007) 160 – 172. doi:http://dx.doi.org/10.1016/j.cma.2007.07.005.
- [26] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, T. J. R. Hughes, *Robustness of isogeometric structural discretizations under severe mesh distortion*, *Computer Methods in Applied Mechanics and Engineering* 199 (5-8) (2010) 357 – 373. doi:http://dx.doi.org/10.1016/j.cma.2009.01.022.
- [27] L. M. Bernal, V. M. Calo, N. Collier, G. A. Espinosa, F. Fuentes, J. C. Mahecha, *Isogeometric Analysis of Hyperelastic Materials Using PetIGA*, *Procedia Computer Science* 18 (2013) 1604 – 1613. doi:http://dx.doi.org/10.1016/j.procs.2013.05.328.
- [28] H. Gómez, T. J. R. Hughes, X. Nogueira, V. M. Calo, *Isogeometric analysis of the isothermal Navier-Stokes-Korteweg equations*, *Computer Methods in Applied Mechanics and Engineering* 199 (25-28) (2010) 1828 – 1840. doi:http://dx.doi.org/10.1016/j.cma.2010.02.010.
- [29] 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. doi:10.1002/fld.2337.

- [30] J. A. Evans, T. J. R. Hughes, Isogeometric divergence-conforming B-splines for the unsteady NavierStokes equations, *Journal of Computational Physics* 241 (2013) 141 – 167. doi:<http://dx.doi.org/10.1016/j.jcp.2013.01.006>.
- [31] M.-C. Hsu, Y. Bazilevs, V. Calo, T. Tezduyar, T. Hughes, Improving stability of stabilized and multiscale formulations in flow simulations at small time steps, *Computer Methods in Applied Mechanics and Engineering* 199 (13-16) (2010) 828–840. doi:<http://dx.doi.org/10.1016/j.cma.2009.06.019>.
- [32] Y. Bazilevs, C. Michler, V. Calo, T. Hughes, Weak Dirichlet boundary conditions for wall-bounded turbulent flows, *Computer Methods in Applied Mechanics and Engineering* 196 (49-52) (2007) 4853–4862. doi:<http://dx.doi.org/10.1016/j.cma.2007.06.026>.
- [33] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, T. J. R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, *Computer Methods in Applied Mechanics and Engineering* 196 (29-30) (2007) 2943 – 2959. doi:<http://dx.doi.org/10.1016/j.cma.2007.02.009>.
- [34] Y. Bazilevs, J. Gohean, T. Hughes, R. Moser, Y. Zhang, Patient-specific isogeometric fluidstructure interaction analysis of thoracic aortic blood flow due to implantation of the jarvik 2000 left ventricular assist device, *Computer Methods in Applied Mechanics and Engineering* 198 (45-46) (2009) 3534 – 3550. doi:<http://dx.doi.org/10.1016/j.cma.2009.04.015>.
- [35] S. S. Hossain, S. F. A. Hossain, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, Mathematical modeling of coupled drug and drug-encapsulated nanoparticle transport in patient-specific coronary artery walls, *Computational Mechanics* 49 (2) (2011) 213–242. doi:10.1007/s00466-011-0633-2.
- [36] V. M. Calo, N. F. Brasher, Y. Bazilevs, T. J. R. Hughes, Multiphysics model for blood flow and drug transport with application to patient-specific coronary artery flow, *Computational Mechanics* 43 (1) (2008) 161–177. doi:10.1007/s00466-008-0321-z.
- [37] 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. doi:<http://dx.doi.org/10.1016/j.cma.2009.12.002>.
- [38] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, G. Sangalli, Isogeometric Analysis: Approximation, Stability and Error estimates for h-refined meshes, *Mathematical Models and Methods in Applied Sciences* 16 (07) (2006) 1031–1090. doi:10.1142/S0218202506001455.
- [39] A. George, Nested Dissection of a Regular Finite Element Mesh, *SIAM Journal on Numerical Analysis* 10 (2) (1973) 345–363. doi:10.1137/0710032.
- [40] B. M. Irons, A frontal solution program for finite element analysis, *International Journal for Numerical Methods in Engineering* 2 (1970) 5–32.
- [41] M. Paszyński, Fast solvers for mesh based computations, Taylor & Francis, CRC Press, 2015.
- [42] L. Dalcin, N. O. Collier, P. Vignal, M. A. Cortes, V. M. Calo, PetIGA: A framework for high-performance isogeometric analysis, arxiv (1305.4452), <http://arxiv.org/abs/1305.4452v3>.
- [43] 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, PETSc Web page. URL <http://www.mcs.anl.gov/petsc>
- [44] 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, PETSc Users Manual (ANL-95/11 - Revision 3.6). URL <http://www.mcs.anl.gov/petsc>
- [45] A. M. A. Côrtes, P. Vignal, A. Sarmiento, D. Garcia, N. Collier, L. Dalcin, V. M. Calo, Solving Nonlinear, High-Order Partial Differential Equations Using a High-Performance Isogeometric Analysis Framework, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [46] A. Côrtes, A. Coutinho, L. Dalcin, V. M. Calo, Performance evaluation of block-diagonal preconditioners for the divergence-conforming b-spline discretization of the stokes system, *Journal of Computational Science* 11 (2015) 123 – 136. doi:<http://dx.doi.org/10.1016/j.jocs.2015.01.005>.
- [47] L. F. R. Espath, A. F. Sarmiento, P. Vignal, B. O. N. Varga, A. M. A. Côrtes, L. Dalcin, V. M. Calo, Energy exchange analysis in droplet dynamics via the navierstokescahnhilliard model 797 (2016) 389–430. doi:10.1017/jfm.2016.277.
- [48] A. Sarmiento, D. Garcia, L. Dalcin, N. Collier, V. M. Calo, Micropolar fluids using b-spline divergence conforming spaces, *Procedia Computer Science* 29 (2014) 991 – 1001. doi:<http://dx.doi.org/10.1016/j.procs.2014.05.089>.
- [49] 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* (2016) –doi:<http://dx.doi.org/10.1016/j.jocs.2016.09.010>.
- [50] 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. doi:10.1137/S0895479899358194.
- [51] P. R. Amestoy, A. Guermouche, J.-Y. LÉxcellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, *Parallel Computing* 32 (2) (2006) 136 – 156. doi:<http://dx.doi.org/10.1016/j.parco.2005.07.004>.
- [52] S. Blackford, J. Dongarra, LAPACK Working Note 41 "Installation Guide for LAPACK", Department of Computer Science, University of Tennessee.

Appendix A: Derivation of memory estimation

We compute the number of non-zero entries in factors L and U based on the matrix decomposition procedure performed by the multifrontal direct solver. The matrix decomposition eliminates the dof in sets (either subsystems or separators). Each set consists of q fully assembled dof and its elimination results in $\mathcal{O}(q^2)$ non-zero entries. The total number of non-zero entries on the factors L and U is given by

$$\begin{aligned} & \mathcal{O}\left(\Psi|_{C^0} (q_{sep}|_{C^0})^2\right) + \eta_{m-e} \left(\mathcal{O}\left(\Psi|_{C^{p-1}} (q_{m-e})^2\right)\right) \\ & \mathcal{O}\left(\Psi|_{C^0} (N^{(d-1)/d})^2\right) + 2^{id} \mathcal{O}\left(\Psi|_{C^{p-1}} \left(\left(\frac{N}{2^{id}}\right)^{(d-1)/d} p\right)^2\right) \\ & \mathcal{O}\left(\Psi|_{C^0} n^{2(d-1)}\right) + \frac{2^{id}}{2^{2i(d-1)}} \mathcal{O}\left(\Psi|_{C^{p-1}} n^{2(d-1)} p^2\right) + L.O.T. \end{aligned}$$

where 2^{id} and $N/2^{id}$ are the number and size of the macro-elements. $\Psi|_{C^0}$ and $\Psi|_{C^{p-1}}$ include the contribution of all the separators to the total number of non-zero entries. We compute $\Psi|_{C^0}$ as

$$\Psi|_{C^0} = \sum_{j=0}^{i-1} \sum_{m=0}^{d-1} 2^{dj+m} \left(2^{-(j(d-1)+m)}\right)^2 = \begin{cases} \frac{3}{2} (\log_2(2^i)) & 2D \\ \frac{7}{2} (1 - 2^{-i}) & 3D \end{cases}$$

where j refers to the level of partition, and $j + m$ is the number of steps of partition. Moreover, we assume that $\Psi|_{C^{p-1}}$ behaves as

$$\Psi|_{C^{p-1}} = \begin{cases} \frac{3}{2} (\log_2(2^\iota))^2 & 2D \\ \frac{7}{2} & 3D \end{cases}$$

where ι is the number of levels of the macro-elements partitioning. The total number of non-zero entries for $2D$ and $3D$ are

$$\begin{aligned} 2D: \quad \theta &= \mathcal{O}\left(\underbrace{\left(\frac{3}{2} \log_2(2^{-i})\right)}_{C^0\text{-separators contribution}} + \underbrace{\left(\frac{3}{2} \log_2(2^\iota)^2 p^2\right)}_{C^{p-1}\text{ macro-elements contribution}}\right) n^2 + L.O.T., \\ 3D: \quad \theta &= \mathcal{O}\left(\underbrace{\left(\frac{7}{2} (1 - 2^{-i})\right)}_{C^0\text{-separators contribution}} + \underbrace{\left(\frac{7}{2} 2^{-i} p^2\right)}_{C^{p-1}\text{ macro-elements contribution}}\right) n^4 + L.O.T. \end{aligned}$$

Note that these equations are multiplied by $8 \cdot 10^{-6}$ in order to express the memory estimation in *Mbytes*. We use $8 \cdot 10^{-6}$ since the computations are performed with double precision.