Università degli studi di Pavia

Joint Ph.D. program in Mathematics Milano Bicocca - Pavia - INdAM

Doctoral Thesis

Preconditioners for Isogeometric Analysis

Author:
Monica Montardini

Supervisor:
Prof. Giancarlo Sangalli

Co-supervisor:
Dr. Mattia Tani

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor prof. Giancarlo Sangalli for having encouraged me in following my academic dreams. I learned a lot from his vast knowledge, not limited to numerical analysis. I really appreciated his patience and motivation and, in particular, the time he spent in giving me valuable advice in mathematical and non-mathematical topics.

I would also thank my co-supervisor and friend Dr. Mattia Tani. I am very grateful to him for his excellent scientific opinions and for having always encouraged me to do my best. In particular, I would like to thank him for having stimulated me to look at the world from a different side and with a changed soul.

A special thanks goes to prof. Matteo Negri. The discussions on functional analysis problems with him were very fruitful and he was always ready to explain me the issues I could not understand.

I would like also to thank prof. Ulrich Langer, that allowed me to spend three months at the Johannes Kepler University in Linz, where I had the chance to meet high-qualified researchers. I really appreciated the collaborations that started there.

My gratitude is also due to Dr. Michał Bosy. He helped me in understanding domain decomposition methods and in facing everyday university problems.

I would like to thank my fellow Ph.D. students for their feedback, cooperation and of course friendship: Maria Gioia, Nicolò, Barbara, Alberto, Christian, Anderson, Irene, Gabriele, Silvia, Federico and Mai. I really appreciated the time we spent together.

My thanks also goes to my cousins Marco, Silvia and Franco for their continuous encouragement, especially when I was abroad.

Finally, I would like to express my gratitude to my parents Antonio and Paola, my grandmother Rosella and my uncles Tino and Renata, for their undeserved and endless support and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Partial differential equations (PDEs) represent the main tools to model many physical processes, for example fluid flow, heat transfer and solid problems. Typically, the expression of the exact solution can not be found in an analytic way. In these situations numerical methods are the fundamental ingredients to recover an approximated solution.

The most popular technique used to make numerical simulations is the finite element method (FEM). However, one of the main issue of FEM is that the computational domain, usually described by a computer aided design (CAD) file, needs to be approximated, e.g. by triangulation. As a consequence, there is always a systematic geometrical error source and the operation of mesh-refinement results very expensive. The discrepancy between CAD and FEM relies on their mathematical foundation: the former employs splines while the latter uses linear or quadratic interpolation polynomials.

To overcome the gap between FEM and CAD and to improve the interoperability between CAD and PDE solvers, Hughes et al. introduced in the seminal paper [66] an extension of FEM: Isogeometric Analysis (IgA). IgA is based on the adoption of the same functions that describe the CAD geometry (usually B-splines and Non-Uniform Rational B-splines (NURBS)) to construct and represent the approximated solution of the PDE. In this way, thanks to an exact representation of the computational domain the error due to the approximation of the geometry is eliminated. IgA is not limited to B-splines and NURBs as it allows to use a wide variety of spline technologies, e.g. T-splines [5, 98, 7], hierarchical splines [26, 112], Truncated Hierarchical B-splines (THB-splines) [56, 55], Locally Refineable splines (LR-splines) [68, 19].

The refinement of the spline space can be achieved not only by the classical p-refinement (order elevation) and h-refinement (knot insertion) procedures, already present in FEM, but also by the new k-refinement, that is order elevation followed by knot insertion, see [66]. We remark that all kinds of refinement do not alter the geometry and do not need the communication with CAD systems. Despite it does not construct nested spline spaces, the k-refinement leads to possibilities previously unavailable in FEM, as the direct discretization of high order PDEs, the use of continuous stresses and the development of collocation methods. In addition, it paves the way to the so called k-method, that is an isogeometric method where the basis functions have high regularity: unlike the standard FEM functions, that are typically only C^0 continuous, in the k-method, in general, isogeometric basis functions have regularity C^{p-1}, where p represents the polynomial degree. The k-method brings several advantages: higher accuracy per degree-of-freedom [44], robust approximation of non-smooth functions [29], better approximation of the spectrum [38]. However, the high continuity of the k-method introduces challenging problems at the computational level. The two main computational costs (in terms of time and memory) in a solver for PDEs are:

- the formation of the system matrix $\mathbf{A}$;
- the solution of the linear system associated to the PDE

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{1.1.1}$$

Even if the focus of our thesis deals with the second problem, we give a brief overview of the techniques that the isogeometric community has introduced to address the first problem. In the early applications of IgA, the implementation of the k-method was done by extending the existing FEM codes and exploiting their architecture. The system matrix was built performing standard Gaussian quadrature rules within each element and then assembling these local matrices into the global one, as in standard FEM codes. However, there is an important difference with respect to FEM: the higher continuity and, consequently, the larger support of IgA basis functions. Indeed, univariate B-splines of degree p and regularity C^{p-1} can have a support that, usually, consists of $p+1$ elements. Thus, interactions between functions become more costly, as the overlapping support between two functions is bigger and the memory required is higher (see [34]). In addition, Gaussian quadrature rules within each element are far from being optimal and they are the most time-consuming part of the FEM-like assembling codes. The extension to IgA of standard FEM codes was not enough to have a competitive isogeometric assembling method. The development of new and ad-hoc algorithms became an essential goal to pursue. Recently, different alternatives to the standard Gaussian quadrature element-wise assembling have been proposed. Among them, some of the most relevant are sum-factorization techniques [2, 21], low-rank assembling [78, 79] and the weighted quadrature approach [30, 94].

The focus of this thesis regards the study of efficient solution of the isogeometric linear systems, that represents the second main computational cost in the construction of a solver for PDEs. An important issue is that the high continuity of the functions in the k-method affects the sparsity-pattern of the system matrix and increases the computational cost required by the linear solver to find an approximated solution. The studies of [34] highlight that direct solvers do not provide good performances when combined with the k-method. For example, suppose to solve a d-dimensional Poisson problem with a multi-frontal solver, that is a standard direct solver (see [34, Section 2.3]). Let N_{dof} be the number of degrees-of-freedom. Assuming $N_{dof} \gg p^d$, the number of floating point operations (FLOPs) required to solve the system discretized with splines of degree p and regularity C^{p-1} is $O(N_{dof}^2 p^d)$. On the other hand, when we use a standard C^0 FEM discretization with polynomials of degree p, we need $O(N_{dof}^2)$ FLOPs. The degradation of the performances of direct solvers applied within C^{p-1} IgA has switched the attention of the isogeometric community on iterative solvers. The most used iterative methods are the so-called Krylov subspace methods, see e.g. [91]. The rate of convergence of Krylov subspace methods can be bounded by using the distribution of the eigenvalues of the system matrix [91]. For example, when $\mathbf{A}$ is symmetric positive-definite, we can use the conjugate gradient (CG) method [60] to solve the linear system and it is known that the behavior of the solver depends, on the worst case, on the spectral condition number $\kappa(\mathbf{A}) := \frac{\lambda_{max}(\mathbf{A})}{\lambda_{min}(\mathbf{A})}$ as

$$\|\mathbf{u} - \mathbf{u}_k\|_{\mathbf{A}} \leq 2 \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right)^k \|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{A}}, \tag{1.1.2}$$

where $\mathbf{u}_k$ is the k-th iterate and where $\|\cdot\|_{\mathbf{A}}$ is the norm induced by the vector norm $\|\mathbf{v}\|_{\mathbf{A}} := (\mathbf{v}^T \mathbf{A} \mathbf{v})^{\frac{1}{2}}$. The dependence of the rate of convergence on the eigenvalues still holds when the matrix $\mathbf{A}$ is just symmetric and the system is solved by MINRES [87]. If we consider non-symmetric matrices the situation is more complicated and we can not describe the convergence of Krylov subspace methods, like GMRES [92], through the distribution of the eigenvalues. Nevertheless, in many applications it is observed that a clustered spectrum (away from 0 and infinity) makes the convergence faster [57]. Thus, in general, the more clustered are the eigenvalues, the faster is the convergence.

The introduction of a preconditioning operation implies that instead of solving (1.1.1), we solve

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{u} = \mathbf{P}^{-1}\mathbf{b}$$

where $\mathbf{P}$ is a square, invertible matrix called preconditioner. This means that at each iteration of a preconditioned Krylov-subspace method we have to solve a linear system with matrix $\mathbf{P}$ as

$$\mathbf{P}\mathbf{v} = \mathbf{r}$$

where $\mathbf{r}$ represents the current residual. The rates of convergence of the iterative methods can gain benefits, provided that the preconditioned system has better spectral properties than the unpreconditioned one. For example, in the CG case, when we introduce a symmetric positive definite preconditioner $\mathbf{P}$, the convergence estimate (1.1.2) becomes

$$\|\mathbf{u} - \mathbf{u}_k\|_{\mathbf{A}} \leq 2\left(\frac{\sqrt{\kappa(\mathbf{P}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{P}^{-1}\mathbf{A})} + 1}\right)^k \|\mathbf{u} - \mathbf{u}_0\|_{\mathbf{A}}.$$

As isogeometric matrices suffer of large condition numbers when the degree is high or when the mesh-size is small, iterative solvers could perform well only if combined with efficient preconditioning strategies. In this context, we say that the preconditioner $\mathbf{P}$ is *ideal* if it satisfies both of the following properties:

- $\mathbf{P}$ must be *computationally efficient*: the setup and the application of $\mathbf{P}$ should be non-expensive in terms of FLOPs and memory and proportional to the number of DOFs of the problem;
- $\mathbf{P}$ must be *robust*: the eigenvalues of the preconditioned system $\mathbf{P}^{-1}\mathbf{A}$ should be clustered and they should be bounded away from 0 and infinity, independently on the parameters of the problem.

Ideal preconditioners are not easy to obtain with high order methods. The first studies on isogeometric preconditioners are provided by [33], where the problem of interest is the Poisson problem and the iterative method is the CG method. The authors show that the most efficient preconditioners used in FEM applications, as diagonal Jacobi, incomplete LU and SOR, work well with low-order splines, but their computational performances deteriorate when high order functions are employed. That is why, up to some years ago, in practical applications, quadratic or cubic splines were preferred, see [6, 82].

It was clear that the extension of standard FEM preconditioners was not an effective choice and, consequently, in IGA community the development of efficient preconditioning strategies has become a primary task. A lot of efforts have been directed to the Poisson system. One of the first kind of preconditioners for elliptic problems that has been studied is the group of multigrid based preconditioners [41, 52, 65]. An additive multilevel approach has been used in [27]. The authors of [93] exploit the tensor product structure of the isogeometric spaces to design a preconditioner that is the sum of matrices with a special Kronecker structure. Their preconditioner can be efficiently applied by using the fast diagonalization method (FD), that is a direct solver introduced in [77]. Such designed preconditioning strategy reveals to be robust both with respect to spline degree and mesh-size and it can be easily coupled with an overlapping-Schwarz domain decomposition method. Moreover, in practice, it has a computational setup and application cost that is proportional to the number of DOFs of the problem. Overlapping Schwarz methods for elliptic problems in IgA have been also studied in [10, 9], while BDDC methods and preconditioners in [8, 12, 89]. The extension of FETI-DP, a variant of FETI (see [50]), to IgA has been introduced in [70] and further studied in [61, 62].

The recent paper [94] combines the FD preconditioner with a matrix-free weighted-quadrature approach in order to design an efficient Poisson solver. The overall procedure is very efficient in practical applications. The authors are able to provide a computationally-efficient k-method that has superior performances with respect to low-order discretizations. The extension of the global solver to other PDEs is possible, but the efficiency of the method is strictly related to the presence of a good preconditioner. As this paper confirms, the preconditioning step is the most critical part for the development of an efficient isogeometric solver.

Several efforts have been done in the design of efficient preconditioners for PDEs different than the Poisson equation, as it will be highlighted in the introduction of Chapter 3, 4 and 5, and this thesis represents a contribution in this direction.

1.2 Main achievements and structure of the work

The aim of this thesis is the development of efficient solvers for linear systems arising in the isogeometric discretization of different PDEs. Each problem studied yields to linear system matrices with different properties and structure, that thus require ad-hoc preconditioning strategy.

The starting point and the basis of our preconditioners is the FD method and, in particular, the work [93], in which the authors fully exploit the tensor product structure of the isogeometric functions to build a preconditioner obtained by discretizing the Poisson system in the parametric domain and considering constant coefficients.

Having in mind the good results and performances gained in [93] by using the FD method for its application, we tried to use the same or similar ideas in the design of preconditioners for the isogeometric discretization of other PDEs and thus discretizing the problem of interest (or a simplification of it) in the parametric domain. In particular, we focus on two PDEs: the Stokes problem and the parabolic problem. We are able to recover in all cases the tensor-product structure that can be exploited by the FD method for an efficient application. As there is a strong connection between the eigenvalues and the rate of convergence of Krylov subspace methods, we provide proofs of meaningful spectral estimates for the matrices involved. All the preconditioning strategies that we propose are robust both with respect to the spline degree p and the mesh-size h. One important issue we have to face is the fact that the influence of the geometry on the performance of the preconditioning strategy can create a loss of efficiency, especially when the parametrization highly departs from the identity. Ideally, we would like that the efficiency of the preconditioner does not depend on the coefficients and on the geometry of the problem and that is why we investigate strategies that allow to incorporate in the basic version of the preconditioner some of these pieces of information without increasing the original computational cost. The designed variants of the preconditioners that include some information on the parametrization and on the coefficients also show robustness with respect to the geometry.

The first model problem that we examine is the Stokes stationary system. We consider both the isogeometric Taylor-Hood [4] and the Raviart-Thomas discretizations [25, 45] in a single-patch domain. We develop preconditioners suited for the resulting saddle-point linear systems that have the classical block diagonal, block triangular or constrained structure [43]. The FD method is then used to invert the diagonal blocks. We provide spectral estimates that ensure the good convergence properties of MINRES solver and we show both numerically and theoretically that our preconditioner is robust with respect to h and p. Moreover we compare our preconditioner with the more classical one based on a zero fill-in Incomplete Cholesky (IC) factorization and we observe the superiority of the performances of our approach, especially when including some information on the parametrization in the preconditioner.

Then we consider parabolic problems and in particular the heat equation in computational domains that are fixed in time. We want to fully-exploit the high continuity of the k-method by considering a space-time discretization, where the PDE is discretized simultaneously in space and in time. The first formulation of the heat problem we study has its foundation on the least-squares principle. Indeed, we propose a space-time least-squares isogeometric method and we provide a-priori error estimates that guarantee the good convergence properties of the method. The regularity required by the basis functions is higher than for a standard Galerkin formulation, but this does not represent a problem in the framework of the isogeometric k-method. The resulting system is symmetric and positive definite and the same FD-based preconditioning strategy of [93] can be trivially applied. The total cost of the setup and the application of the preconditioner is, in practical applications, proportional to the number of DOFs. The second weak formulation of the heat problem that we consider is the low-order formulation of [103]. We extend to IgA the theory of [103], developed for standard finite elements, and we provide quasi-optimal error estimates. The isogeometric discretization of this formulation does not lead to a straightforward use of FD method. We circumvent this difficulty by introducing an ad-hoc factorization of the matrices that allows to design a solver conceptually similar to the FD method, with similar computational cost and robust with respect to degree and mesh-size. We provide numerical experiments on non-trivial computational domains and we show a comparison with the previous least-squares solver. The preconditioner is robust with respect to degree and mesh size and the version that incorporates information on the parametrization gives very good results also with highly-distorted geometries. The cost is still proportional to the number of DOFs of the problem.

The structure of this work is the following one:

- Chapter 2 introduces the basic concepts of IgA. We define univariate and multivariate B-spline functions, highlighting their main features. Then we present the Kronecker product and the related properties we need. Finally, we report the preconditioning strategy for the Poisson problem based on the FD method.

- Chapter 3 focuses on the stationary Stokes problem. The chapter starts by presenting the isogeometric discretizations that we use: the isogeometric Taylor-Hood and the isogeometric Raviart-Thomas elements. For the resulting saddle-point linear systems we design preconditioners with a block diagonal, a block triangular or a constrained structure and we prove theoretical estimates that assures the robustness of the preconditioners with respect to the spline degree and mesh-size. We also study the issue of incorporating information on the geometry in the basic preconditioner. Our numerical benchmarks confirm the efficiency of the proposed preconditioning strategies.

- Chapter 4 deals with the least-squares space-time variational formulation of the heat equation. After the introduction of the method, we prove the well-posedness of the least-squares formulation and we provide a-priori error estimates. Then, we present the basic version of the preconditioner, we analyze the spectral properties and the computational cost. We also show how to improve the performance of the basic version of the preconditioner by including some information on the geometry parametrization. Numerical results that confirm the error estimates and show the performances of the preconditioning strategies are finally presented.

- Chapter 5 presents a preconditioning strategy suited for a Galerkin space-time discretization of the heat equation. To overcome the problem of the non-stable eigendecomposition of the time-matrices, we build an ad-hoc decomposition of the pencils and we provide an extension of the FD method. The time required for the application of such designed preconditioner still grows as the number of DOFs. In addition, an improved version of the preconditioning technique that includes some information on the parametrization and on the coefficients of the problem is studied. Lastly, some numerical benchmarks that confirms the error estimates and that assess the performance of the proposed preconditioners are provided.

- Chapter 6 contains brief conclusions of our work and possible future directions of research.

Parts of the results presented in this thesis have been published by the author and coauthors in peer reviewed journals or they are available online:

- Chapter 3:

  [81] M. Montardini, G. Sangalli, and M. Tani. "Robust isogeometric preconditioners for the Stokes system based on the Fast Diagonalization method". In: *Computer Methods in Applied Mechanics and Engineering* 338 (2018), pp. 162 - 185.

- Chapter 4:

  [80] M. Montardini, M. Negri, G. Sangalli, and M. Tani. "Space-time least-squares isogeometric method and efficient solver for parabolic problems". In: *Mathematics of Computation* (accepted for publication).

- Chapter 5:

  [76] G. Loli, M. Montardini, G. Sangalli, and M. Tani. "Space-time Galerkin isogeometric method and efficient solver for parabolic problem". In: *arXiv e-prints*, arXiv:1909.07309 (2019).

Chapter 2

Preliminaries

The aim of this chapter is to introduce the preliminary ingredients and the notations that will be used in the rest of the work.

2.1 B-splines

In this section we present the definition and the basic properties of univariate and multivariate B-splines. For a detailed explanation on this topic we refer to [37, 96]. Useful algorithms for the evaluation of B-splines can be found e.g. in [90].

Remark 2.1. *We remark that in our thesis we investigate isogeometric solvers for different PDEs and each problem is discretized by a particular kind of isogeometric space. For this reason and in order to have self-contained chapters, in this section we give just a general overview of the construction of spline spaces in the parametric domain, while at the beginning of each of the following chapters we resume the splines notations and we define the isogeometric spaces needed.*

2.1.1 Univariate B-splines

Given two positive integers m and p, a *knot vector* in $[0, 1]$ is a set of non-decreasing points

$$\Xi := \{0 = \xi_1 \leq \cdots \leq \xi_{m+p+1} = 1\}.$$

Note that the knots can be repeated. Thus, we introduce the *breakpoint vector*

$$\mathcal{Z} := \{\zeta_1, ..., \zeta_s\},$$

that is a vector that contains the knots without repetitions, and the corresponding vector $\{r_1, \ldots, r_m\}$ of knot multiplicities, i.e. r_i is the multiplicity of the knot ζ_i in Ξ:

$$\Xi = \{\underbrace{\zeta_1, \ldots, \zeta_1}_{r_1 \text{ times}}, \underbrace{\zeta_2, \ldots, \zeta_2}_{r_2 \text{ times}}, \ldots, \underbrace{\zeta_m, \ldots, \zeta_m}_{r_m \text{ times}}\},$$

with $\sum_{i=1}^{m} r_i = m + p + 1$. We assume that the multiplicity of each internal knot does not exceed $p + 1$, that is $r_i \leq p + 1$. The *i-th knot span* is defined as the interval $[\xi_i, \xi_{i+1})$ and, if the knots are repeated, it can have null length. A knot span with non-zero length is called *element* and the maximum of length of each element $h := \max\{\xi_{i+1} - \xi_i | i = 1, \ldots, m + p\}$ is called *mesh-size*.

Then, from the knot vector Ξ we define univariate B-splines basis functions $\widehat{b}_{i,p}$ recursively through Cox-De Boor formulas [39] as

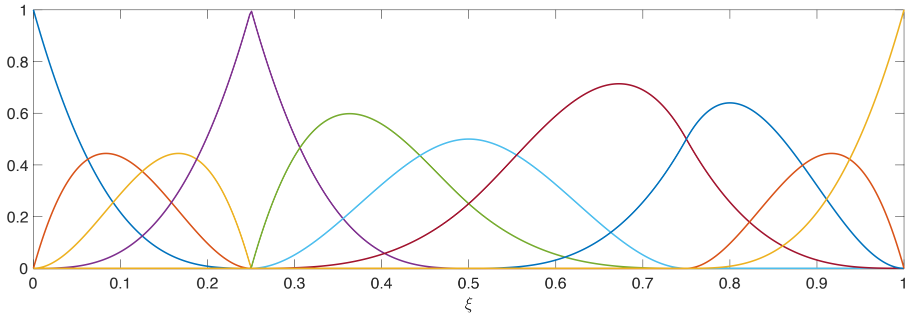


FIGURE 2.1: Example of cubic B-spline basis built from the open knot vector $\{0, 0, 0, 0, 0.25, 0.25, 0.25, 0.5, 0.75, 0.75, 1, 1, 1, 1\}$

for $p = 0$:

$$\widehat{b}_{i,0}(\eta) = \begin{cases} 1 & \text{if } \xi_i \le \eta < \xi_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

for $p \ge 1$:

$$\widehat{b}_{i,p}(\eta) = \begin{cases} \dfrac{\eta - \xi_i}{\xi_{i+p} - \xi_i}\widehat{b}_{i,p-1}(\eta) + \dfrac{\xi_{i+p+1} - \eta}{\xi_{i+p+1} - \xi_{i+1}}\widehat{b}_{i+1,p-1}(\eta) & \text{if } \xi_i \le \eta < \xi_{i+p+1}, \\ 0 & \text{otherwise,} \end{cases}$$

where we adopt the convention $0/0 = 0$. The set of the m B-splines that we have just defined forms a basis of the space of splines of degree p with $\alpha_i := p - r_i$ continuous derivatives at the breakpoint ζ_i. From the definition we have that $-1 \le \alpha_i \le p - 1$ and if $\alpha_i = -1$, then we have a discontinuity at ζ_i. We collect the information on the regularity in the vector $\boldsymbol{\alpha} := \{\alpha_1, \dots, \alpha_s\}$.

We assume that every knot vector is *open*, that is, we set $\xi_1 = \cdots = \xi_{p+1} = 0$ and $\xi_{m+1} = \cdots = \xi_{m+p+1} = 1$. The assumption that the knot vector is open corresponds to the choice $\alpha_1 = \alpha_s = -1$. An example of B-spline basis built from an open knot vector is represented in Figure 2.1. We note that the repetition of the knot causes a loss of regularity and, in particular, we have that at the point 0.25 the basis is just C^0. B-splines built from open knot vectors have useful properties, e.g. they are interpolatory at the extremes of the parametric interval $[0, 1]$ and they are beneficial to impose Dirichlet boundary conditions.

We denote the *univariate spline space* as

$$\widehat{\mathcal{S}}_h^p := \operatorname{span}\{\widehat{b}_{i,p} \ \mid \ i = 1, \dots, m\}. \tag{2.1.1}$$

Some useful properties of B-splines are:

- for all η it holds $\sum_{i=1}^{m} \widehat{b}_{i,p}(\eta) = 1$;
- the support of each $\widehat{b}_{i,p}$ is contained in the interval $[\xi_i, \xi_{i+p+1}]$;
- for all η it holds $\widehat{b}_{i,p}(\eta) \ge 0$.

The proofs of the properties above are elementary and they can be found e.g. in [39]. We remark that for $p = 0$ and $p = 1$, the B-splines are the same piecewise constant and piecewise linear functions, respectively, present in classical FEM, while for higher degree they are different. In particular, for $p \ge 2$, B-splines are not nodal and their support consist not only of one element, in general.

The space $\widehat{\mathcal{S}}_h^p$ can be refined through knot insertion and degree elevation. These two operations generate two kinds of refinement: h-refinement, that is a refinement of the mesh obtained by insertion of new knots, and p-refinement, that corresponds to degree elevation without increasing the regularity and it is obtained by increasing the multiplicity of the knots. In IgA there is also a third kind of refinement, not present in FEM literature, that is the k-refinement that is obtained by degree elevation followed by knot insertion. Differently from h-refinement and p-refinement, the k-refinement does not lead to a sequence of nested spaces: the spaces obtained through this procedure have increased regularity (see [37] for more details).

2.1.2 Tensor product B-splines

The simplest and straightforward way to extend univariate B-splines to the d-dimensional case with $d > 1$ is using a tensor product construction.

The d-dimensional parametric domain is defined as the d-dimensional cube $\widehat{\Omega} := (0,1)^d$.

Let $\Xi_l := \{\xi_{l,1} \leq \dots \leq \xi_{l,m_l+p_l+1}\}$ for $l = 1,\dots,d$ be d knot vectors and $\boldsymbol{p}$ be the vector that contains the degree indexes, i.e. $\boldsymbol{p} := (p_1,\dots,p_d)$. Let also $\mathcal{Z}_l = \{\zeta_{l,1},\dots,\zeta_{l,s_l}\}$ be the corresponding breakpoint vectors and $\boldsymbol{\alpha}_l := \{-1, \alpha_{l,2},\dots,\alpha_{l,s_l},-1\}$ the associated regularity vectors for $l = 1,\dots,d$. We introduce the mesh-sizes $h_l := \max\{\xi_{l,i+1}-\xi_{l,i} \mid i = 1,\dots,m_l+p_l\}$ and the global mesh-size as $h := \max\{h_l \mid l = 1,\dots,d\}$.

Then the multivariate B-splines are defined as

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\eta}) := \widehat{b}_{i_1,p_1}(\eta_1)\dots\widehat{b}_{i_d,p_d}(\eta_d) \tag{2.1.2}$$

where $\boldsymbol{i} := (i_1,\dots,i_d)$ and $\boldsymbol{\eta} = (\eta_1,\dots,\eta_d)$.

The corresponding *multivariate spline space* is defined as

$$\widehat{\mathcal{S}}_h^{\boldsymbol{p}} := \operatorname{span}\left\{\widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \;\middle|\; i_k = 1,\dots,m_k \text{ for } k = 1,\dots,d\right\} = \widehat{\mathcal{S}}_{h_1}^{p_1} \otimes \dots \otimes \widehat{\mathcal{S}}_{h_d}^{p_d}.$$

Multivariate B-splines inherit the same properties as univariate B-splines: they are non-negative, they form a partition of unity and they have local support.

We remark that tensor product B-splines do not allow local refinement. However, there are several generalizations of univariate B-splines to the multivariate case, not considered in this thesis, that make possible the local refinement of the mesh, as T-Splines [5, 98, 7], LR-Splines [68, 19], THB-splines [56, 55] and HB-splines [26, 112].

2.2 Kronecker product

The *Kronecker product* between two matrices $\mathbf{C} \in \mathbb{C}^{n_1 \times n_2}$ and $\mathbf{D} \in \mathbb{C}^{n_3 \times n_4}$ is defined as

$$\mathbf{C} \otimes \mathbf{D} := \begin{bmatrix} [\mathbf{C}]_{1,1}\mathbf{D} & \dots & [\mathbf{C}]_{1,n_2}\mathbf{D} \\ \vdots & \ddots & \vdots \\ [\mathbf{C}]_{n_1,1}\mathbf{D} & \dots & [\mathbf{C}]_{n_1,n_2}\mathbf{D} \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_4},$$

where the ij-th entry of the matrix $\mathbf{C}$ is denoted by $[\mathbf{C}]_{i,j}$.

The properties of the Kronecker product that we will exploit are the following ones:

- it holds
$$(\mathbf{C} \otimes \mathbf{D})^* = \mathbf{C}^* \otimes \mathbf{D}^*, \tag{2.2.1}$$
where * is the complex conjugate transpose operation;

- if $\mathbf{C}$, $\mathbf{D}$, $\mathbf{E}$ and $\mathbf{L}$ are matrices of conforming order, then it holds

$$(\mathbf{C}\otimes\mathbf{D})(\mathbf{E}\otimes\mathbf{L}) = (\mathbf{CE})\otimes(\mathbf{DL}); \tag{2.2.2}$$

- if $\mathbf{C}$ and $\mathbf{D}$ are non-singular, then

$$(\mathbf{C}\otimes\mathbf{D})^{-1} = \mathbf{C}^{-1}\otimes\mathbf{D}^{-1}; \tag{2.2.3}$$

- if $\mathbf{X}\in\mathbb{C}^{n_4\times n_2}$ then

$$(\mathbf{C}\otimes\mathbf{D})\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{DXC}^T) \tag{2.2.4}$$

  where we introduced the vectorization operator "vec", that, when applied to a matrix, simply stacks the columns of the matrix in a vector as

$$[\mathrm{vec}(\mathbf{X})]_{i_1+(i_2-1)n_1} = [\mathbf{X}]_{i_1,i_2} \qquad \text{for } i_j = 1,\dots,n_j \text{ and } j = 1,2.$$

We need to recall the extension of (2.2.4) to the d-dimensional case. We introduce, for $m = 1,\dots,d$, the m-mode product $\times_m$ of a tensor $\mathfrak{X}\in\mathbb{C}^{n_1\times\cdots\times n_d}$ with a matrix $\mathbf{J}\in\mathbb{C}^{w\times n_m}$, that is a tensor of size $n_1\times\cdots\times n_{m-1}\times w\times n_{m+1}\times\dots n_d$ whose elements are defined as

$$[\mathfrak{X}\times_m\mathbf{J}]_{i_1,\dots,i_d} = \sum_{j=1}^{n_m}[\mathfrak{X}]_{i_1,\dots,i_{m-1},j,i_{m+1}\dots,i_d}[\mathbf{J}]_{i_m,j}.$$

Then, given $\mathbf{J}_i\in\mathbb{C}^{w_i\times n_i}$ for $i = 1,\dots,d$, the generalization to the d-dimensional case of (2.2.4), reads as

$$(\mathbf{J}_d\otimes\cdots\otimes\mathbf{J}_1)\,\mathrm{vec}\,(\mathfrak{X}) = \mathrm{vec}\,(\mathfrak{X}\times_1\mathbf{J}_1\times_2\cdots\times_d\mathbf{J}_d)\,, \tag{2.2.5}$$

where the vectorization operator "vec" applied to a tensor stacks its entries into a column vector as

$$[\mathrm{vec}(\mathfrak{X})]_j = [\mathfrak{X}]_{i_1,\dots,i_d} \qquad \text{for } i_l = 1,\dots,n_l \text{ and } l = 1,\dots,d,$$

where $j = i_1 + \sum_{k=2}^d\left[(i_k-1)\Pi_{l=1}^{k-1}n_l\right]$.

For more details on the Kronecker product and on its properties we refer to [71].

2.3 Fast diagonalization preconditioner for Poisson problem

In this section we resume the results obtained in [93], where the authors study preconditioning strategies for Poisson problem and they propose the use of the fast diagonalization method to apply their designed preconditioner. For the sake of clarity, we focus on the two-dimensional case.

Let us consider the following Poisson problem endowed, for simplicity, with homogeneous Dirichlet boundary conditions

$$\left\{\begin{array}{rcll} -\nabla\cdot(K(\boldsymbol{x})\nabla u(\boldsymbol{x})) & = & f(\boldsymbol{x}) & \text{in} \quad \Omega, \\ u & = & 0 & \text{on} \quad \partial\Omega, \end{array}\right. \tag{2.3.1}$$

where $\Omega\subset\mathbb{R}^2$ and $K(\boldsymbol{x})$ is a symmetric positive definite matrix for every $\boldsymbol{x}\in\Omega$. For the notations of univariate and multivariate B-splines in the parametric domain, we refer to Section 2.1 We suppose that Ω is given by a regular single-patch spline parametrization $\boldsymbol{G}\in[\widehat{\mathcal{S}}_h^{\boldsymbol{p}}]^2$.

We need to introduce the spline space with boundary conditions in the parametric domain $\widehat{\Omega}$:

$$\widehat{\mathcal{S}}^{\boldsymbol{p}}_{h,0} := \{v \in \widehat{\mathcal{S}}^{\boldsymbol{p}}_{h} \;\mid\; v = 0 \text{ on } \partial\widehat{\Omega}\} = \operatorname{span}\{\widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \;\mid\; 2 \leq i_l \leq m_l - 1 \text{ and } l = 1, 2\}.$$

Note that

$$\widehat{\mathcal{S}}^{\boldsymbol{p}}_{h,0} := \widehat{\mathcal{S}}^{p_2}_{h_2,0} \otimes \widehat{\mathcal{S}}^{p_1}_{h_1,0},$$

where $\widehat{\mathcal{S}}^{p_l}_{h_l,0} := \operatorname{span}\{\widehat{b}_{i,p_l} \;\mid\; i = 2, \dots, m_l - 1\}$. We associate to each multi-index $\boldsymbol{i}$ the number $i = i_1 - 1 + (m_1 - 2)(i_2 - 2)$ and, with abuse of notations, we write

$$\widehat{\mathcal{S}}^{\boldsymbol{p}}_{h,0} = \operatorname{span}\{\widehat{B}_{i,\boldsymbol{p}} \;\mid\; 1 \leq i \leq N_{dof}\}$$

where $N_{dof} := (m_1 - 2)(m_2 - 2)$ is the dimension of the space that incorporates the boundary conditions. We now introduce the corresponding isogeometric space

$$\mathcal{S}^{\boldsymbol{p}}_{h,0} := \left\{\widehat{v} \circ \boldsymbol{G} \;\mid\; v \in \widehat{\mathcal{S}}^{\boldsymbol{p}}_{h,0}\right\} = \operatorname{span}\{B_{\boldsymbol{i},\boldsymbol{p}} := \widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \circ \boldsymbol{G} \;\mid\; i = 1, \dots, N_{dof}\},$$

Then, the Galerkin stiffness matrix associated to the system (2.3.1) takes this form:

$$[\mathbf{A}]_{i,j} := \int_{\Omega} (\nabla B_{i,\boldsymbol{p}})^T K \nabla B_{j,\boldsymbol{p}} \,\mathrm{d}\Omega = \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}}\right)^T Q \nabla \widehat{B}_{j,\boldsymbol{p}} \,\mathrm{d}\widehat{\Omega} \quad i, j = 1, \dots, N_{dof} \tag{2.3.2}$$

where

$$Q = J_{\boldsymbol{G}}^{-1} K J_{\boldsymbol{G}}^{-T} |\det(J_{\boldsymbol{G}})| \tag{2.3.3}$$

and $J_{\boldsymbol{G}}$ is the Jacobian matrix.

2.3.1 The preconditioner

The authors of [93] propose to use as a preconditioner for the CG method applied to the system (2.3.2) the matrix

$$[\mathbf{P}]_{i,j} := \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}}\right)^T \nabla \widehat{B}_{j,\boldsymbol{p}} \,\mathrm{d}\widehat{\Omega} \quad i, j = 1, \dots, N_{dof} \tag{2.3.4}$$

that is obtained by considering in (2.3.2) $\boldsymbol{G}$ equal to the identity map and K equal to the identity matrix. In particular, exploiting the tensor-product structure of the basis functions, we can write

$$\mathbf{P} = K_2 \otimes M_1 + M_2 \otimes K_1, \tag{2.3.5}$$

where K_1 and K_2 are the univariate stiffness matrices while M_1 and M_2 are the univariate mass matrices, that is, for $k = 1, 2$ they are defined for $i, j = 2, \dots, m_k - 1$ as

$$[K_k]_{i-1,j-1} := \int_0^1 \widehat{b}'_{i,p}(\eta_k) \widehat{b}'_{j,p}(\eta_k) \mathrm{d}\eta_k, \quad [M_k]_{i-1,j-1} := \int_0^1 \widehat{b}_{i,p}(\eta_k) \widehat{b}_{j,p}(\eta_k) \mathrm{d}\eta_k.$$

At each CG iteration the application of the preconditioner $\mathbf{P}$ requires to find the solution of the system

$$\mathbf{P}\mathbf{s} = \mathbf{r},$$

that is equivalent to solve

$$(K_2 \otimes M_1 + M_2 \otimes K_1)\mathbf{s} = \mathbf{r}, \tag{2.3.6}$$

where $\mathbf{r}$ is the current residual. The influence of the preconditioner on the rate of convergence of CG is well known (see (1.1))and it depends on the condition number of the preconditioned system. For this reason we report below [93, Proposition 1] that investigates the spectral properties of $\mathbf{P}$.

Proposition 2.1. *It holds*

$$\kappa(\mathbf{P}^{-1}\mathbf{A}) \leq \frac{\sup\limits_{\widehat{\Omega}} \lambda_{\max}(Q)}{\inf\limits_{\widehat{\Omega}} \lambda_{\min}(Q)}, \tag{2.3.7}$$

where Q is given by (2.3.3)*, while $\lambda_{\max}(Q)$ and $\lambda_{min}(Q)$ denote the maximum and the minimum eigenvalue of Q, respectively.*

Note that the bound depends only on the geometry $\boldsymbol{G}$ and on K while it is independent on the mesh-size and the spline degree. Thus, the performance of the preconditioner could deteriorate if the geometry highly departs from the identity or if the coefficient matrix K varies widely.

2.3.2 The application of the preconditioner: the fast diagonalization method

We now consider the problem of the efficient application of $\mathbf{P}$, that is, to find the solution of (2.3.6). Using (2.2.4), we can rewrite (2.3.6) as

$$M_1 S K_2^T + K_1 S M_2^T = R, \tag{2.3.8}$$

where $\text{vec}(S) = \mathbf{s}$ and $\text{vec}(R) = \mathbf{r}$. Equation (2.3.8) is called in literature (generalized) Sylvester equation and arises in many applications. For a recent survey on the solving methods for the Sylvester equation we refer to [101].

In [93] the authors propose to use the fast diagonalization method (FD), that is a direct method for solving equations with the Sylvester-like structure of (2.3.8). The FD method was first proposed in [77] as a method to solve elliptic PDEs discretized using finite differences. Let us suppose for simplicity that the univariate stiffness and mass matrices have dimensions $n \times n$ and that the total number of degrees-of-freedom is $N_{dof} := n^2$. The first step in the two-dimensional FD method is to compute the generalized eigendecompostion of the matrix pencils (K_i, M_i) for $i = 1, 2$, that is

$$K_1 U_1 = M_1 U_1 \Lambda_1, \quad K_2 U_2 = M_2 U_2 \Lambda_2 \tag{2.3.9}$$

where U_1 and U_2 are matrices containing the generalized eigenvectors and Λ_1 and Λ_2 are the diagonal matrices of the corresponding eigenvalues. Note that, as the matrices M_i are symmetric and positive definite, we also have that the eigenvectors are M_i-orthogonal, that is

$$U_1^T M_1 U_1 = \mathbb{I}_n \quad \text{and} \quad U_2^T M_2 U_2 = \mathbb{I}_n, \tag{2.3.10}$$

where $\mathbb{I}_n$ denotes the identity matrix of dimension n. Thus, as a consequence, we have the factorizations

$$U_1^{-T} U_1^{-1} = M_1 \quad \text{and} \quad U_2^{-T} U_2^{-1} = M_2$$

and, by inserting them in (2.3.9), we get

$$U_1^{-T} \Lambda_1 U_1^{-1} = K_1 \quad \text{and} \quad U_2^{-T} \Lambda_2 U_2^{-1} = K_2.$$

Thus, we can rewrite (2.3.6) as

$$(U_2 \otimes U_1)^{-T} (\mathbb{I}_n \otimes \Lambda_1 + \Lambda_2 \otimes \mathbb{I}_n)(U_2 \otimes U_1)^{-1} \mathbf{s} = \mathbf{r},$$

and then the solution of the system is given by

$$\mathbf{s} = (U_2 \otimes U_1)(\mathbb{I}_n \otimes \Lambda_1 + \Lambda_2 \otimes \mathbb{I}_n)^{-1}(U_2 \otimes U_1)^T \mathbf{r}.$$

After a preliminary step in which we have to setup the preconditioner by performing the generalized eigendecomposition, the application of $\mathbf{P}$ just involves two matrix vector multiplications and an inversion of a diagonal matrix. We summarize this procedure in Algorithm 1.

Algorithm 1 2D FD method

```
1: Setup: Compute the generalized eigendecompositions (2.3.9)
2: Application: Compute $\mathbf{t} = (U_2 \otimes U_1)^T \mathbf{r}$
3:              Compute $\mathbf{q} = (\mathbb{I}_n \otimes \Lambda_1 + \Lambda_2 \otimes \mathbb{I}_n)^{-1}\,\mathbf{t}$
4:              Compute $\mathbf{s} = (U_2 \otimes U_1)\ \mathbf{q}$.
```

2.3.3 Computational cost

We briefly analyze the cost of Algorithm 1. For more details see [93].

The setup of the preconditioner, that is Step 1 in Algorithm 1, needs $O(N_{dof}^{3/2})$. We remark that the setup step has to be performed only once, as the matrices do not change during the CG iterative process.

The application of the preconditioner involves Steps 2-4. Step 2 and Step 4 are computed using property (2.2.4) and involve two matrix-matrix product each. The resulting computational cost for both steps is $O(N_{dof}^{3/2})$. Finally, Step 3 is just a diagonal scaling and its cost of $O(N_{dof})$ FLOPs is negligible. The total computational cost of Algorithm 1 is thus of $O(N_{dof}^{3/2})$. The memory required to store the preconditioner is of $O(N_{dof})$.

The other dominant cost in a CG iteration is the product between the matrix $\mathbf{A}$ and a vector, that is the residual computation. This matrix-vector product is twice the number of non-zeros of $\mathbf{A}$, that is approximately equal to $2(2p+1)^2 N_{dof}$.

Chapter 3

The Stokes problem

In this chapter the problem of interest is the Stokes system. We consider two well-known isogeometric discretizations for which stability and convergence is known. One is the extension of the Taylor-Hood element, which is inf-sup stable, see [4, 28, 25, 18, 20]. The other is the extension of the Raviart-Thomas element, which is stable and structure-preserving, in the sense that the discrete solution is pointwise divergence-free; see [25, 45] (and [46, 47] for its extension to Navier-Stokes). Both allow for arbitrary degree and regularity, in the spirit of the k-method.

Isogeometric preconditioners for the Stokes system have also been studied in recent papers: [35, 36] consider block-diagonal and block-triangular preconditioners combined to black-box solvers (either algebraic-multigrid or incomplete factorization); [88] studies the domain-decomposition FETI-DP strategy; [32] focuses on a multigrid strategy; another multigrid approach, which extends the results of [64], can be found in [106].

In this chapter, for both Taylor-Hood and Raviart-Thomas isogeometric discretizations of the Stokes system, we consider preconditioners having the classical block structure (see [43]) and using direct solvers to invert the diagonal blocks.

In the simplest approach, our pressure Schur complement preconditioner is the pressure mass matrix in parametric coordinates, which is solved by exploiting its Kronecker structure. Moreover, our preconditioner for the velocity blocks is a component-wise divergence of the symmetric gradient in parametric coordinates, and its solution is the solution of a Sylvester-like equation. Among many methods, following [93] we adopt the fast diagonalization (FD) method (see also Section 2.3).

An important problem we have to face is the treatment of the geometry parametrization. The simplest approach outlined above does not incorporate any geometry information in the preconditioner, causing a significant loss of efficiency on complex geometry parametrizations. To overcome this limitation, we propose a modification of the preconditioner for a partial inclusion of the geometry information, without increasing its computational cost: in our numerical benchmarking we show the clear benefits of this approach. Indeed, we show theoretically and numerically that our preconditioner is robust with respect to the mesh size h and spline degree p, both for the isogeometric Taylor-Hood and Raviart-Thomas methods. While previous papers considered low-degree splines only (typically quadratics and cubics), we are motivated to consider higher degrees in our tests (up to degree 6 for the velocity and 5 for the pressure, for memory constraints) by the fact that the computational cost of our preconditioner is almost independent of the degree. The iterative solver total computational time is $O(n_{dof}p^3)$, but it is heavily dominated by the matrix-vector multiplication which takes more than the 99% of the overall cost when the pressure degree is 5 and the velocity degree is 6, on a 16^3 elements mesh. In this case our preconditioner is much faster than the alternatives known in literature: for example, about 3 orders of magnitude when comparing to a standard preconditioner based on the incomplete Cholesky factorization, which is known to be an effective choice (see e.g. [35]).

In conclusion our numerical benchmarks confirm that the proposed preconditioner is very efficient and well suited for the k-method. Further advances in the solver performance can be achieved with a matrix-free approach, that accelerates the matrix-vector multiplication operation, for moderate or large degree. A first step in this research direction is [94].

The outline of the chapter is as follows. In Section 3.1 we recall the basic notations on the univariate and multivariate B-splines, while in Section 3.2 we give a short review of the Taylor-Hood and Raviart-Thomas isogeometric discretizations for the Stokes system. The derivation of the discrete Stokes system is given in Section 3.3 and in Section 3.4 we introduce some standard block-structured preconditioners that we will consider in the numerical tests. The core of the chapter is Section 3.5, where we focus on the construction of the preconditioning matrices for the velocity and pressure blocks, discuss their properties and solution strategies. We also propose a modification aimed at improving the preconditioner efficiency by incorporating some information on the geometry parametrization. Numerical results on three different single-patch domains are reported in Section 3.6. Finally, in Section 3.7 we draw the conclusions.

3.1 Notations and main assumptions for the spline spaces

In this section we summarize the notations for the spline spaces we will use thorough the chapter, referring to Section 2.1 for the basic definitions. For the discretization of the Stokes system, the regularity of the basis functions in each parametric direction is a fundamental piece of information, that we want to highlight in the notation.

Let $\Xi := \{\xi_1, ..., \xi_{m+p+1}\}$ be an open knot vector. We assume that Ξ is uniform, i.e. with equally spaced breakpoints and we denote the mesh size with h. We also assume that Ξ has uniform regularity α, that is, we set $\alpha_2 = \cdots = \alpha_{s-1} = \alpha$. Then, we slightly modify the notation presented in 2.1 and, we denote the associated univariate B-splines of degree p and the corresponding univariate spline space as

$$\widehat{b}_{i,p}^{\alpha} \quad \text{and} \quad \widehat{\mathcal{S}}_{\alpha}^{p} = \operatorname{span}\{\widehat{b}_{i,p}^{\alpha} \ \mid \ i = 1, \dots m\},$$

and we set $m_{\alpha}^{p} := m = \dim(\widehat{\mathcal{S}}_{\alpha}^{p})$. The extension of this framework to non-uniform knot vectors and arbitrary regularity is trivial (see, in this context, [20, Remark 4.4] and [28]) and it is considered in our numerical tests.

For 3D problems, the case we address, the univariate open knot vectors $\Xi_l := \{\xi_{l,1}, ..., \xi_{l,m_{\alpha_l}^{p_l}+p_l+1}\}$ for $l = 1, 2, 3$ and degree indices $\boldsymbol{p} = (p_1, p_2, p_3)$ are given and we collect the regularity of each direction in the vector $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \alpha_3)$. Then, for a multi-index $\boldsymbol{i} = (i_1, i_2, i_3)$, the multivariate B-spline is denoted as

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}^{\boldsymbol{\alpha}}(\boldsymbol{\eta}) := \widehat{b}_{i_1,p_1}^{\alpha_1}(\eta_1)\widehat{b}_{i_2,p_2}^{\alpha_2}(\eta_2)\widehat{b}_{i_3,p_3}^{\alpha_3}(\eta_3)$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3)$, and the multivariate spline space as

$$\widehat{\mathcal{S}}_{\boldsymbol{\alpha}}^{\boldsymbol{p}} := \widehat{\mathcal{S}}_{\alpha_1}^{p_1} \otimes \widehat{\mathcal{S}}_{\alpha_2}^{p_2} \otimes \widehat{\mathcal{S}}_{\alpha_3}^{p_3} = \operatorname{span}\{\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}^{\boldsymbol{\alpha}} \mid i_k = 1, ..., m_{\alpha_k}^{p_k}; k = 1, 2, 3\}.$$

We refer to spline spaces as spaces of splines defined on the parametric domain $\widehat{\Omega} := [0, 1]^3$.

3.2 Isogeometric spaces

Let the computational domain $\Omega \subset \mathbb{R}^3$ be given by a single-patch spline parametrization $\boldsymbol{G} \in \widehat{\mathcal{S}}^{\boldsymbol{p}}_{\boldsymbol{\alpha}} \times \widehat{\mathcal{S}}^{\boldsymbol{p}}_{\boldsymbol{\alpha}} \times \widehat{\mathcal{S}}^{\boldsymbol{p}}_{\boldsymbol{\alpha}}$ of degree p in each parametric direction. We assume that $\boldsymbol{G}$ is a *regular* parametrization, in sense that its Jacobian is everywhere invertible.

Isogeometric spaces over Ω are suitable push-forwards, through $\boldsymbol{G}$, of spline spaces. In particular, in the context of the Stokes system, we focus on two discretizations of isogeometric spaces that have been proposed in [20] and [28] respectively. Their definition and properties are summarized in this section, see [4, 18, 20, 28, 45] for further details.

3.2.1 Taylor-Hood isogeometric spaces

The Taylor-Hood (TH) spline spaces are defined as

$$\begin{aligned}\mathcal{V}_h^{TH} &:= \widehat{\mathcal{S}}^{\boldsymbol{p}+1}_{\boldsymbol{\alpha}} \times \widehat{\mathcal{S}}^{\boldsymbol{p}+1}_{\boldsymbol{\alpha}} \times \widehat{\mathcal{S}}^{\boldsymbol{p}+1}_{\boldsymbol{\alpha}},\\ \mathcal{Q}_h^{TH} &:= \widehat{\mathcal{S}}^{\boldsymbol{p}}_{\boldsymbol{\alpha}}.\end{aligned}$$

where $\boldsymbol{p}+1 := (p+1, p+1, p+1)$. For the velocity space we will also need

$$\mathcal{V}_{h,0}^{TH} := \left\{ \widehat{\boldsymbol{v}}_h \in \mathcal{V}_h^{TH} \;\middle|\; \widehat{\boldsymbol{v}}_h = 0 \text{ on } \partial\widehat{\Omega} \right\}.$$

A basis for $\mathcal{V}_h^{TH}$ is

$$\left\{ \mathbf{e}_k \widehat{B}^{\boldsymbol{\alpha}}_{\boldsymbol{i},\boldsymbol{p}+1} \;\middle|\; i_l = 1, ..., m^{p+1}_{\alpha_l};\; k, l = 1, 2, 3 \right\}.$$

where $\mathbf{e}_k$ is the k-th canonical basis vector of $\mathbb{R}^3$.

A basis for $\mathcal{V}_{h,0}^{TH}$ is then

$$\left\{ \mathbf{e}_k \widehat{B}^{\boldsymbol{\alpha}}_{\boldsymbol{i},\boldsymbol{p}+1} \;\middle|\; i_l = 2, ..., m^{p+1}_{\alpha_l} - 1;\; k, l = 1, 2, 3 \right\}. \tag{3.2.1}$$

To each multi-index $\boldsymbol{i}$ present in (3.2.1) we associate a scalar index i, corresponding to the colexicographical ordering of the internal degrees of freedom, such that

$$i = i_1 - 1 + (i_2 - 2)(m^{p+1}_{\alpha_1} - 2) + (i_3 - 2)(m^{p+1}_{\alpha_1} - 2)(m^{p+1}_{\alpha_2} - 2)$$

and, with abuse of notation, we rewrite the basis of $\mathcal{V}_{h,0}^{TH}$ as

$$\left\{ \mathbf{e}_k \widehat{B}^{\boldsymbol{\alpha}}_{i,\boldsymbol{p}+1} \;\middle|\; i = 1, ..., n^{TH}_{V,k};\; k = 1, 2, 3 \right\},$$

where $n^{TH}_{V,1} = n^{TH}_{V,2} = n^{TH}_{V,3} := (m^{p+1}_{\alpha_1} - 2)(m^{p+1}_{\alpha_2} - 2)(m^{p+1}_{\alpha_3} - 2)$. We also define

$$n^{TH}_V := \dim(\mathcal{V}_{h,0}^{TH}) = n^{TH}_{V,1} + n^{TH}_{V,2} + n^{TH}_{V,3}.$$

A basis for $\mathcal{Q}_h^{TH}$ is

$$\left\{ \widehat{B}^{\boldsymbol{\alpha}}_{\boldsymbol{i},\boldsymbol{p}} \;\middle|\; i_l = 1, ..., m^{p}_{\alpha_l};\; l = 1, 2, 3 \right\}. \tag{3.2.2}$$

To each multi-index $\boldsymbol{i}$ present in (3.2.2) we associate a scalar index i, corresponding to the colexicographical ordering of the internal degrees of freedom, such that

$$i = i_1 + (i_2 - 1)m^p_{\alpha_1} + (i_3 - 1)m^p_{\alpha_1} m^p_{\alpha_2} \tag{3.2.3}$$

and, with abuse of notation, we rewrite the basis of $\mathcal{Q}_h^{TH}$ as

$$\left\{ \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}} \;\middle|\; i = 1, ..., n_Q^{TH} \right\}, \tag{3.2.4}$$

where

$$n_Q^{TH} := \dim(\mathcal{Q}_h^{TH}) = m_{\alpha_1}^p m_{\alpha_2}^p m_{\alpha_3}^p. \tag{3.2.5}$$

The TH isogeometric spaces are the isoparametric push-forwards (see [20, 28]):

$$\mathcal{V}_{h,0}^{TH} := \operatorname{span}\left\{ \boldsymbol{\phi}_i^{k,TH} := \mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+1}^{\boldsymbol{\alpha}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, ..., n_{V,k}^{TH};\ k = 1, 2, 3 \right\} \tag{3.2.6a}$$

$$\mathcal{Q}_h^{TH} := \operatorname{span}\left\{ \rho_i^{TH} := \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, ..., n_Q^{TH} \right\}. \tag{3.2.6b}$$

For the discrete variational formulation of the Stokes system we will also need the space

$$\mathcal{Q}_{h,0}^{TH} := \left\{ w \in \mathcal{Q}_h^{TH} \;\middle|\; \int_\Omega w \, \mathrm{d}\Omega = 0 \right\}. \tag{3.2.7}$$

3.2.2 Raviart-Thomas isogeometric spaces

The Raviart-Thomas (RT) spline spaces are defined as

$$\begin{aligned} \mathcal{V}_h^{RT} &:= \widehat{\mathcal{S}}_{\boldsymbol{\alpha}+\mathbf{e}_1}^{\boldsymbol{p}+\mathbf{e}_1} \times \widehat{\mathcal{S}}_{\boldsymbol{\alpha}+\mathbf{e}_2}^{\boldsymbol{p}+\mathbf{e}_2} \times \widehat{\mathcal{S}}_{\boldsymbol{\alpha}+\mathbf{e}_3}^{\boldsymbol{p}+\mathbf{e}_3}, \\ \mathcal{Q}_h^{RT} &:= \widehat{\mathcal{S}}_{\boldsymbol{\alpha}}^{\boldsymbol{p}}. \end{aligned}$$

where $\boldsymbol{p} + \mathbf{e}_1 = (p+1,\ p,\ p)$, $\boldsymbol{p} + \mathbf{e}_2 = (p,\ p+1,\ p)$, $\boldsymbol{p} + \mathbf{e}_3 = (p,\ p,\ p+1)$ and $\boldsymbol{\alpha} + \mathbf{e}_1 = (\alpha_1 + 1,\ \alpha_2,\ \alpha_3)$, $\boldsymbol{\alpha} + \mathbf{e}_2 = (\alpha_1,\ \alpha_2 + 1,\ \alpha_3)$, $\boldsymbol{\alpha} + \mathbf{e}_3 = (\alpha_1,\ \alpha_2,\ \alpha_3 + 1)$.

For the velocity space we will also need

$$\mathcal{V}_{h,0}^{RT} := \left\{ \widehat{\boldsymbol{v}}_h \in \mathcal{V}_h^{RT} \;\middle|\; \widehat{\boldsymbol{v}}_h \cdot \widehat{\boldsymbol{n}} = 0 \text{ on } \partial\widehat{\Omega} \right\}.$$

A basis for $\mathcal{V}_h^{RT}$ is

$$\left\{ \mathbf{e}_k \widehat{B}_{\boldsymbol{i},\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \;\middle|\; i_k = 1, ..., m_{p+1}^{\alpha_k+1};\ i_l = 1, ..., m_p^{\alpha_l};\ l \neq k;\ l, k = 1, 2, 3 \right\}.$$

A basis for $\mathcal{V}_{h,0}^{RT}$ is then

$$\left\{ \mathbf{e}_k \widehat{B}_{\boldsymbol{i},\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \;\middle|\; i_k = 2, ..., m_{p+1}^{\alpha_k+1} - 1;\ i_l = 1, ..., m_p^{\alpha_l};\ l \neq k;\ l, k = 1, 2, 3 \right\}. \tag{3.2.8}$$

To each multi-index $\boldsymbol{i}$ present in (3.2.8) we associate a scalar index i, corresponding to the lexicographical ordering of the internal degrees of freedom, such that

$$\begin{gathered} \text{for } k = 1 \quad i = i_1 - 1 + (i_2 - 1)(m_{p+1}^{\alpha_1+1} - 2) + (i_3 - 1)(m_{p+1}^{\alpha_1+1} - 2) m_p^{\alpha_2}, \\ \text{for } k = 2 \quad i = i_1 + (i_2 - 2) m_p^{\alpha_1} + (i_3 - 1) m_p^{\alpha_1} (m_{p+1}^{\alpha_2+1} - 2), \\ \text{for } k = 3 \quad i = i_1 + (i_2 - 1) m_p^{\alpha_1} + (i_3 - 2) m_p^{\alpha_1} m_p^{\alpha_2} \end{gathered}$$

and, with abuse of notation, we rewrite the basis of $\mathcal{V}_{h,0}^{RT}$ as

$$\left\{ \mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \;\middle|\; i = 1, ..., n_{V,k}^{RT};\ k = 1, 2, 3 \right\}, \tag{3.2.9}$$

where

$$n_{V,1}^{RT} = (m_{p+1}^{\alpha_1+1} - 2)m_p^{\alpha_2}m_p^{\alpha_3}, \quad n_{V,2}^{RT} = m_p^{\alpha_1}(m_{p+1}^{\alpha_2+1} - 2)m_p^{\alpha_3}, \quad n_{V,3}^{RT} = m_p^{\alpha_1}m_p^{\alpha_2}(m_{p+1}^{\alpha_3+1} - 2).$$

We also define

$$n_V^{RT} := \dim(\mathcal{V}_{h,0}^{RT}) = n_{V,1}^{RT} + n_{V,2}^{RT} + n_{V,3}^{RT}.$$

As $\mathcal{Q}_h^{RT} = \mathcal{Q}_h^{TH}$, a basis for $\mathcal{Q}_h^{RT}$ is (3.2.4) and its dimension is denoted by $n_Q^{RT} = n_Q^{TH} = n_Q$ (cfr. (3.2.5)).

The RT isogeometric spaces are defined by suitable push-forwards:

$$\mathcal{V}_{h,0}^{RT} := \operatorname{span}\left\{\boldsymbol{\phi}_i^{k,RT} := \left((\det(J_{\boldsymbol{G}}))^{-1} J_{\boldsymbol{G}}\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right) \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, ..., n_{V,k}^{RT};\; k = 1, 2, 3\right\} \tag{3.2.10a}$$

$$Q_h^{RT} := \operatorname{span}\left\{\rho_i^{RT} := \left((\det(J_{\boldsymbol{G}}))^{-1} \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}}\right) \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, ..., n_Q^{RT}\right\}. \tag{3.2.10b}$$

The push-forward employed for $\mathcal{V}_{h,0}^{RT}$ is the Piola transform and its use is important to assure inf-sup stability, see [28] and Section 3.3.

We remark that although in the parametric domain $\mathcal{Q}_h^{RT} = \mathcal{Q}_h^{TH}$, in general $\mathcal{Q}_h^{RT} \neq \mathcal{Q}_h^{TH}$.

For the discrete variational formulation of the Stokes system we will also need the space

$$\mathcal{Q}_{h,0}^{RT} := \left\{w \in \mathcal{Q}_h^{RT} \;\middle|\; \int_\Omega w \;\mathrm{d}\Omega = 0\right\}. \tag{3.2.11}$$

3.3 Isogeometric analysis of the Stokes system

As usual, let $L^2(\Omega)$ be the space of square integrable functions, $L^\infty(\Omega)$ the space of essentially bounded measurable functions and $H^1(\Omega)$ the space of functions in $L^2(\Omega)$ whose first-order derivatives belong to $L^2(\Omega)$. We define the vectorial spaces $\mathbf{L}^2(\Omega) := \left[L^2(\Omega)\right]^3$ and $\mathbf{H}_0^1(\Omega) := \left[H_0^1(\Omega)\right]^3$ endowed with the standard norms that we denote with $\|\cdot\|_{\mathbf{L}^2(\Omega)}$ and $\|\cdot\|_{\mathbf{H}^1(\Omega)}$, respectively. The standard $\mathbf{L}^2$-scalar product is denoted with $(\cdot,\cdot)_{\mathbf{L}^2(\Omega)}$.

Then, the Stokes system reads as

$$\begin{aligned} -\nabla\cdot(2\nu\,\nabla^s\boldsymbol{u}) + \nabla q = \boldsymbol{f} \qquad & \text{in } \Omega \\ \nabla\cdot\boldsymbol{u} = 0 \qquad & \text{in } \Omega \end{aligned}$$

where $\nabla^s = \frac{1}{2}\left(\nabla + \nabla^T\right)$, $\boldsymbol{u}$ is the velocity, q is the scalar pressure and $\nu > 0$ is the kinematic viscosity. We assume $\nu \in L^\infty(\Omega)$ and $\boldsymbol{f} \in \mathbf{L}^2(\Omega)$. We consider no-slip boundary conditions, that is we impose $\boldsymbol{u} = 0$ on $\partial\Omega$. The pressure is determined up to a constant.

The standard (mixed) variational formulation of the problem reads: find $\boldsymbol{u} \in \mathbf{H}_0^1(\Omega)$ and $q \in L_0^2(\Omega) := \{w \in L^2(\Omega) \mid \int_\Omega w \;\mathrm{d}\Omega = 0\}$ such that

$$\mathcal{A}(\boldsymbol{u},\boldsymbol{v}) + \mathcal{B}(\boldsymbol{u},q) = (\boldsymbol{f},\boldsymbol{v})_{\mathbf{L}^2(\Omega)} \qquad \forall\,\boldsymbol{v} \in \mathbf{H}_0^1(\Omega) \tag{3.3.1a}$$

$$\mathcal{B}(\boldsymbol{u},r) = 0 \qquad \forall\,r \in L_0^2(\Omega), \tag{3.3.1b}$$

where the bilinear forms $\mathcal{A}(\cdot,\cdot)$ and $\mathcal{B}(\cdot,\cdot)$ are defined as

$$\begin{aligned} \mathcal{A}(\boldsymbol{w},\boldsymbol{v}) &= \int_\Omega 2\nu\,\nabla^s\boldsymbol{w} : \nabla^s\boldsymbol{v}\;\mathrm{d}\Omega \\ \mathcal{B}(\boldsymbol{v},r) &= -\int_\Omega r\,\nabla\cdot\boldsymbol{v}\;\mathrm{d}\Omega. \end{aligned} \tag{3.3.2}$$

The isogeometric Taylor-Hood (TH) discretization of Stokes system is a standard Galerkin method for (3.3.1) and reads: find $\boldsymbol{u}_h^{TH} \in \mathcal{V}_{h,0}^{TH}$ and $q_h^{TH} \in \mathcal{Q}_{h,0}^{TH}$ such that

$$\begin{aligned} &\mathcal{A}(\boldsymbol{u}_h^{TH}, \boldsymbol{v}_h) + \mathcal{B}(\boldsymbol{v}_h, q_h^{TH}) = (\boldsymbol{f}, \boldsymbol{v}_h)_{\mathbf{L}^2(\Omega)} && \forall\, \boldsymbol{v}_h \in \mathcal{V}_{h,0}^{TH}, && (3.3.3a)\\ &\mathcal{B}(\boldsymbol{u}_h^{TH}, r_h) = 0 && \forall\, r_h \in \mathcal{Q}_{h,0}^{TH}, && (3.3.3b)\end{aligned}$$

where $\mathcal{V}_{h,0}^{TH}$ and $\mathcal{Q}_{h,0}^{TH}$ are defined as (3.2.6a) and (3.2.7). A detailed analysis on the well posedness of this problem can be found in [4, 18, 20].

The isogeometric Raviart-Thomas (RT) discretization we adopt is based on a Nitsche formulation for the weak imposition of the tangential Dirichlet boundary condition to ensure stability (see [45]).

The method reads: find $\boldsymbol{u}_h^{RT} \in \mathcal{V}_{h,0}^{RT}$ and $q_h^{RT} \in \mathcal{Q}_{h,0}^{RT}$ such that

$$\begin{aligned} &\mathcal{A}(\boldsymbol{u}_h^{RT}, \boldsymbol{v}_h) + \sigma(\boldsymbol{u}_h^{RT}, \boldsymbol{v}_h) + \mathcal{B}(\boldsymbol{v}_h, q_h^{RT}) = (\boldsymbol{f}, \boldsymbol{v}_h)_{\mathbf{L}^2(\Omega)} && \forall\, \boldsymbol{v}_h \in \mathcal{V}_{h,0}^{RT}, && (3.3.4a)\\ &\mathcal{B}(\boldsymbol{u}_h^{RT}, r_h) = 0 && \forall\, r_h \in \mathcal{Q}_{h,0}^{RT}, && (3.3.4b)\end{aligned}$$

where $\mathcal{V}_{h,0}^{RT}$ and $\mathcal{Q}_{h,0}^{RT}$ are defined as (3.2.10a) and (3.2.11) and the bilinear form $\sigma(\cdot,\cdot)$ is defined as

$$\sigma(\boldsymbol{w}_h, \boldsymbol{v}_h) := \int_{\partial\Omega} 2\nu \left[\frac{C_{pen}}{h} \boldsymbol{w}_h \cdot \boldsymbol{v}_h - ((\nabla^s \boldsymbol{w}_h)\, \boldsymbol{n}) \cdot \boldsymbol{v}_h - ((\nabla^s \boldsymbol{v}_h)\, \boldsymbol{n}) \cdot \boldsymbol{w}_h \right] \mathrm{d}\Gamma, \qquad (3.3.5)$$

with $C_{pen} > 0$ a penalty parameter. The well-posedness of RT discretization for Stokes problem and the choice of C_{pen} are analysed in [45].

In practice, we build the linear system by replacing $\mathcal{Q}_{h,0}^{TH}$ and $\mathcal{Q}_{h,0}^{RT}$ by $\mathcal{Q}_h^{TH}$ and $\mathcal{Q}_h^{RT}$, respectively. This means that we do not incorporate the zero-mean-value constraint into the pressure space, since this will be taken care of by the Krylov iterative solver later.

Then, the discrete Stokes system matrix is

$$\mathbf{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \qquad (3.3.6)$$

where

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \quad B = [B_1\ B_2\ B_3],$$

and for TH discretization, $i = 1, ..., n_{V,r}^{TH}$, $j = 1, ..., n_{V,s}^{TH}$, $r, s = 1, 2, 3$ and $l = 1, ..., n_Q$

$$\begin{aligned} [A_{rs}^{TH}]_{i,j} &:= \mathcal{A}(\boldsymbol{\phi}_i^{r,TH}, \boldsymbol{\phi}_j^{s,TH}),\\ [B_r^{TH}]_{l,j} &:= \mathcal{B}(\boldsymbol{\phi}_j^{r,TH}, \rho_l^{TH}),\end{aligned}$$

while for RT discretization, $i = 1, ..., n_{V,r}^{RT}$, $j = 1, ..., n_{V,s}^{RT}$, $r, s = 1, 2, 3$ and $l = 1, ..., n_Q$

$$\begin{aligned} [A_{rs}^{RT}]_{i,j} &:= \mathcal{A}(\boldsymbol{\phi}_i^{r,RT}, \boldsymbol{\phi}_j^{s,RT}) + \sigma(\boldsymbol{\phi}_i^{r,RT},\ \boldsymbol{\phi}_j^{s,RT}),\\ [B_r^{RT}]_{l,j} &:= \mathcal{B}(\boldsymbol{\phi}_j^{r,RT}, \rho_l^{RT}),\end{aligned}$$

referring to Section 3.2 for the notations of the basis.

In particular the following expressions will be useful for the construction of the preconditioner: for $k = 1, 2, 3$ and $i, j = 1, ..., n_{V,k}^{TH}$

$$[A_{kk}^{TH}]_{i,j} {=} \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+1}^{\boldsymbol{\alpha}}\right)^T \; \boldsymbol{C}_k^{TH} \; \nabla \widehat{B}_{j,\boldsymbol{p}+1}^{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{\eta}, \tag{3.3.7}$$

where

$$\boldsymbol{C}_k^{TH} = \nu (J_{\boldsymbol{G}}^{-1} J_{\boldsymbol{G}}^{-T} + D_k D_k^T) \; |\det(J_{\boldsymbol{G}})| \tag{3.3.8}$$

and $D_k := J_{\boldsymbol{G}}^{-1} \mathbf{e}_k$, while for $k = 1, 2, 3$ and $i, j = 1, ..., n_{V,k}^{RT}$

$$\begin{aligned}
[A_{kk}^{RT}]_{i,j} {:=} & \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right)^T \; \boldsymbol{C}_k^{RT} \; \nabla \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \, \mathrm{d}\boldsymbol{\eta} \\
& {+}\sigma \left(\left(\widetilde{J}_{\boldsymbol{G}} \mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \circ \boldsymbol{G}^{-1}, \left(\widetilde{J}_{\boldsymbol{G}} \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \circ \boldsymbol{G}^{-1} \right) \\
& + \int_{[0,1]^3} 2\nu \left\{ \left[R_k \left(\nabla \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} J_{\boldsymbol{G}}^{-1} \right) \right]^s : \left[(\mathbb{H}_{\boldsymbol{G}} \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}) J_{\boldsymbol{G}}^{-1} \right]^s \right. \\
& + \left[(\mathbb{H}_{\boldsymbol{G}} \mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}) J_{\boldsymbol{G}}^{-1} \right]^s : \left[R_k \left(\nabla \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} J_{\boldsymbol{G}}^{-1} \right) \right]^s \\
& \left. + \left\| \left[(\mathbb{H}_{\boldsymbol{G}} \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}) J_{\boldsymbol{G}}^{-1} \right]^s \right\|_F^2 \right\} |\det(J_{\boldsymbol{G}})| \, \mathrm{d}\boldsymbol{\eta},
\end{aligned} \tag{3.3.9}$$

where

$$\boldsymbol{C}_k^{RT} = \nu |\det(J_{\boldsymbol{G}})| J_{\boldsymbol{G}}^{-1} \left(\|R_k\|_2^2 I + R_k R_k^T \right) J_{\boldsymbol{G}}^{-T}, \tag{3.3.10}$$

$\widetilde{J}_{\boldsymbol{G}} := \left(\det(J_{\boldsymbol{G}})\right)^{-1} J_{\boldsymbol{G}}$, $R_k := \widetilde{J}_{\boldsymbol{G}} \, \mathbf{e}_k$ and $\mathbb{H}_{\boldsymbol{G}}$ is the (trivariate) Hessian tensor $\mathbb{H}_{\boldsymbol{G}} := \nabla \widetilde{J}_{\boldsymbol{G}}$, with the convention that, for a given vector $\mathbf{w} \in \mathbb{R}^3$,

$$\mathbb{H}_{\boldsymbol{G}} \mathbf{w} := \left[\left(\partial_{\eta_1} \widetilde{J}_{\boldsymbol{G}} \right) \mathbf{w}, \;\; \left(\partial_{\eta_2} \widetilde{J}_{\boldsymbol{G}} \right) \mathbf{w}, \;\; \left(\partial_{\eta_3} \widetilde{J}_{\boldsymbol{G}} \right) \mathbf{w} \right].$$

Here and throughout, $\| \cdot \|_2$ denotes the Euclidean vector norm and the induced matrix norm, $\| \cdot \|_F$ refers to the Frobenius matrix norm and $[\, \cdot \,]^s$ denotes the symmetric part. Note that the last integral in (3.3.9) is zero when $\boldsymbol{G}$ is the identity map.

3.4 Preconditioners for the whole system

In this section we introduce the preconditioning strategies that we consider in our numerical tests. In what follows P_V represents a preconditioning matrix for the block A and P_Q a preconditioning matrix for S, where

$$S = BA^{-1}B^T \tag{3.4.1}$$

is the (negative) Schur complement.

Once P_V and P_Q are constructed (this will be discussed in the next section), one can set up suitable preconditioners to be used in the context of Krylov iterative methods [15, 43, 114, 100]. We select three approaches.

In the first one, we consider the block diagonal preconditioner [83]

$$\mathbf{P}_D = \begin{bmatrix} P_V & 0 \\ 0 & P_Q \end{bmatrix}, \tag{3.4.2}$$

which, being symmetric and positive definite, preserves the symmetry of the problem. Therefore it can be coupled with a method for symmetric systems such as MINRES [87]. In the other two approaches, we respectively consider the block triangular [83] and constrained [69] preconditioners

$$\mathbf{P}_T = \begin{bmatrix} P_V & B^T \\ 0 & -P_Q \end{bmatrix} \tag{3.4.3}$$

and

$$\mathbf{P}_C = \begin{bmatrix} P_V & B^T \\ B & BP_V^{-1}B^T - P_Q \end{bmatrix}, \tag{3.4.4}$$

both coupled with the GMRES method [92]. We remark that $\mathbf{P}_C^{-1}$ can be applied efficiently thanks to the factorization

$$\mathbf{P}_C^{-1} = \begin{bmatrix} \mathbb{I}_{n_V} & -P_V^{-1}B^T \\ 0 & \mathbb{I}_{n_Q} \end{bmatrix} \begin{bmatrix} \mathbb{I}_{n_V} & 0 \\ 0 & -P_Q^{-1} \end{bmatrix} \begin{bmatrix} \mathbb{I}_{n_V} & 0 \\ -B & \mathbb{I}_{n_Q} \end{bmatrix} \begin{bmatrix} P_V^{-1} & 0 \\ 0 & \mathbb{I}_{n_Q} \end{bmatrix},$$

where, as in the rest of the thesis, $\mathbb{I}_k$ denotes the identity matrix of dimension $k \times k$.

3.5 Preconditioners for P_V and P_Q

Our choice for the preconditioning block P_V has a block diagonal structure:

$$P_V := \begin{bmatrix} P_{V,1} & 0 & 0 \\ 0 & P_{V,2} & 0 \\ 0 & 0 & P_{V,3} \end{bmatrix}; \tag{3.5.1}$$

the blocks $P_{V,k}$ are a simplified version of the blocks A_{kk} where the geometry map and the kinematic viscosity are replaced by the identity map and identity function, respectively. In other words, analogously to (3.3.2) and (3.3.5), we define in the parametric domain

$$\widehat{\mathcal{A}}(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{v}}) := \int_{\widehat{\Omega}} 2\, \nabla^s \widehat{\boldsymbol{w}} : \nabla^s \widehat{\boldsymbol{v}} \, \mathrm{d}\widehat{\Omega},$$

$$\widehat{\sigma}(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{v}}) := \int_{\partial\widehat{\Omega}} 2 \left[\frac{C_{pen}}{h} \widehat{\boldsymbol{w}} \cdot \widehat{\boldsymbol{v}} - ((\nabla^s \widehat{\boldsymbol{w}})\widehat{\boldsymbol{n}}) \cdot \widehat{\boldsymbol{v}} - ((\nabla^s \widehat{\boldsymbol{v}})\widehat{\boldsymbol{n}}) \cdot \widehat{\boldsymbol{w}} \right] \mathrm{d}\widehat{\Gamma},$$

where $\widehat{\boldsymbol{n}}$ is the exterior normal to the boundary $\partial\widehat{\Omega}$. Therefore for TH discretization, according to (3.3.7), for $i, j = 1, ..., n_{V,k}^{TH}$ and $k = 1, 2, 3$ we define

$$\left[P_{V,k}^{TH}\right]_{i,j} := \widehat{\mathcal{A}}(\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+1}^{\boldsymbol{\alpha}}, \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+1}^{\boldsymbol{\alpha}}) = \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+1}^{\boldsymbol{\alpha}}\right)^T \mathbf{T}_k \, \nabla \widehat{B}_{j,\boldsymbol{p}+1}^{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{\eta}, \tag{3.5.2}$$

where $\mathbf{T}_k = \mathbb{I} + \mathbf{e}_k \mathbf{e}_k^T$, while for RT discretization, according to (3.3.9), for $i, j = 1, ..., n_{V,k}^{RT}$ and $k = 1, 2, 3$ we define

$$\begin{aligned} \left[P_{V,k}^{RT}\right]_{i,j} &:= \widehat{\mathcal{A}}(\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}, \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}) + \widehat{\sigma}\left(\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}, \ \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right) \\ &= \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right)^T \mathbf{T}_k \nabla \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \, \mathrm{d}\boldsymbol{\eta} + \widehat{\sigma}\left(\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}, \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right). \end{aligned} \tag{3.5.3}$$

Exploiting the tensor product structure of the basis functions, we can write

$$
\begin{aligned}
P_{V,1}^{TH} &= K_3^{TH} \otimes M_2^{TH} \otimes M_1^{TH} + M_3^{TH} \otimes K_2^{TH} \otimes M_1^{TH} + 2 M_3^{TH} \otimes M_2^{TH} \otimes K_1^{TH}, && (3.5.4a)\\
P_{V,2}^{TH} &= K_3^{TH} \otimes M_2^{TH} \otimes M_1^{TH} + 2 M_3^{TH} \otimes K_2^{TH} \otimes M_1^{TH} + M_3^{TH} \otimes M_2^{TH} \otimes K_1^{TH}, && (3.5.4b)\\
P_{V,3}^{TH} &= 2 K_3^{TH} \otimes M_2^{TH} \otimes M_1^{TH} + M_3^{TH} \otimes K_2^{TH} \otimes M_1^{TH} + M_3^{TH} \otimes M_2^{TH} \otimes K_1^{TH}, && (3.5.4c)
\end{aligned}
$$

and

$$
\begin{aligned}
P_{V,1}^{RT} &= \widetilde{K}_3^{RT} \otimes \widetilde{M}_2^{RT} \otimes M_1^{RT} + \widetilde{M}_3^{RT} \otimes \widetilde{K}_2^{RT} \otimes M_1^{RT} + 2 \widetilde{M}_3^{RT} \otimes \widetilde{M}_2^{RT} \otimes K_1^{RT}, && (3.5.5a)\\
P_{V,2}^{RT} &= \widetilde{K}_3^{RT} \otimes M_2^{RT} \otimes \widetilde{M}_1^{RT} + 2 \widetilde{M}_3^{RT} \otimes K_2^{RT} \otimes \widetilde{M}_1^{RT} + \widetilde{M}_3^{RT} \otimes M_2^{RT} \otimes \widetilde{K}_1^{RT}, && (3.5.5b)\\
P_{V,3}^{RT} &= 2 K_3^{RT} \otimes \widetilde{M}_2^{RT} \otimes \widetilde{M}_1^{RT} + M_3^{RT} \otimes \widetilde{K}_2^{RT} \otimes \widetilde{M}_1^{RT} + M_3^{RT} \otimes \widetilde{M}_2^{RT} \otimes \widetilde{K}_1^{RT}, && (3.5.5c)
\end{aligned}
$$

where for $k = 1, 2, 3$ the univariate matrix factors are

$$
\begin{aligned}
\left[K_k^{TH}\right]_{l-1,s-1} &= \int_{[0,1]} (\widehat{b}_{l,p+1}^{\alpha_k})'(\eta_k)(\widehat{b}_{s,p+1}^{\alpha_k})'(\eta_k)\, \mathrm{d}\eta_k, && l, s = 2, ..., m_{\alpha_k}^{p+1} - 1,\\
\left[M_k^{TH}\right]_{l-1,s-1} &= \int_{[0,1]} \widehat{b}_{l,p+1}^{\alpha_k}(\eta_k)\ \widehat{b}_{s,p+1}^{\alpha_k}(\eta_k)\, \mathrm{d}\eta_k, && l, s = 2, ..., m_{\alpha_k}^{p+1} - 1,
\end{aligned}
$$

and

$$
\begin{aligned}
\left[K_k^{RT}\right]_{l-1,s-1} &= \int_{[0,1]} (\widehat{b}_{l,p+1}^{\alpha_k+1})'(\eta_k)(\widehat{b}_{s,p+1}^{\alpha_k+1})'(\eta_k)\, \mathrm{d}\eta_k, && l, s = 2, ..., m_{p+1}^{\alpha_k+1} - 1,\\
\left[M_k^{RT}\right]_{l-1,s-1} &= \int_{[0,1]} \widehat{b}_{l,p+1}^{\alpha_k+1}(\eta_k)\ \widehat{b}_{s,p+1}^{\alpha_k+1}(\eta_k)\, \mathrm{d}\eta_k, && l, s = 2, ..., m_{p+1}^{\alpha_k+1} - 1,\\
\left[\widetilde{K}_k^{RT}\right]_{l,s} &= \int_{[0,1]} (\widehat{b}_{l,p}^{\alpha_k})'(\eta_k)(\widehat{b}_{s,p}^{\alpha_k})'(\eta_k)\, \mathrm{d}\eta_k - \Big[(\widehat{b}_{l,p}^{\alpha_k})'(1)\widehat{b}_{s,p}^{\alpha_k}(1) - (\widehat{b}_{l,p}^{\alpha_k})'(0)\widehat{b}_{s,p}^{\alpha_k}(0) \\
&\quad + (\widehat{b}_{s,p}^{\alpha_k})'(1)\widehat{b}_{l,p}^{\alpha_k}(1) - (\widehat{b}_{s,p}^{\alpha_k})'(0)\widehat{b}_{l,p}^{\alpha_k}(0) - 2\frac{C_{pen}}{h} \quad \big(\widehat{b}_{l,p}^{\alpha_k}(1)\widehat{b}_{s,p}^{\alpha_k}(1) \\
&\quad + \widehat{b}_{l,p}^{\alpha_k}(0)\widehat{b}_{s,p}^{\alpha_k}(0)\big)\Big], && l, s = 1, ..., m_p^{\alpha_k},\\
\left[\widetilde{M}_k^{RT}\right]_{l,s} &= \int_{[0,1]} \widehat{b}_{l,p}^{\alpha_k}(\eta_k)\ \widehat{b}_{s,p}^{\alpha_k}(\eta_k)\, \mathrm{d}\eta_k, && l, s = 1, ..., m_p^{\alpha_k}.
\end{aligned}
$$

Now we consider the construction of P_Q. The Schur complement S is spectrally equivalent to the (weighted) pressure mass matrix

$$
\begin{aligned}
\left[Q^{TH}\right]_{i,j} &:= \int_{\Omega} \nu^{-1} \rho_i^{TH} \rho_j^{TH}\, \mathrm{d}\Omega = \int_{\widehat{\Omega}} \nu^{-1} \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}} \widehat{B}_{j,\boldsymbol{p}}^{\boldsymbol{\alpha}}\ g^{TH}\, \mathrm{d}\boldsymbol{\eta},\\
\left[Q^{RT}\right]_{i,j} &:= \int_{\Omega} \nu^{-1} \rho_i^{RT} \rho_j^{RT}\, \mathrm{d}\Omega = \int_{\widehat{\Omega}} \nu^{-1} \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}} \widehat{B}_{j,\boldsymbol{p}}^{\boldsymbol{\alpha}}\ g^{RT}\, \mathrm{d}\boldsymbol{\eta},
\end{aligned}
\qquad (3.5.6)
$$

for $i, j = 1, ..., n_Q$, where $g^{TH}(\boldsymbol{\eta}) := |\det(J_{\boldsymbol{G}})|$ and $g^{RT}(\boldsymbol{\eta}) := |\det(J_{\boldsymbol{G}})|^{-1}$. The equivalence holds uniformly with respect to a variable kinematic viscosity ν, see [58]. However, as for P_V, in our simple approach we drop the dependence on ν and the geometry mapping, by selecting:

$$
\left[P_Q^{TH}\right]_{i,j} := \left[P_Q^{RT}\right]_{i,j} := \int_{\widehat{\Omega}} \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}} \widehat{B}_{j,\boldsymbol{p}}^{\boldsymbol{\alpha}}\, \mathrm{d}\,\boldsymbol{\eta} \quad i, j = 1, ..., n_Q;
$$

as for (3.5.2) and (3.5.3). Exploiting again the tensor product structure of the basis we can write P_Q as

$$
P_Q = M_3 \otimes M_2 \otimes M_1, \qquad (3.5.7)
$$

where for $k = 1, 2, 3$

$$[M_k]_{l,s} = \int_{[0,1]} \widehat{b}^{\alpha_k}_{l,p}(\eta_k)\, \widehat{b}^{\alpha_k}_{s,p}(\eta_k)\, \mathrm{d}\eta_k, \quad l, s = 1, ..., m^p_{\alpha_k}.$$

3.5.1 Spectral properties

A desirable requirement for all the strategies proposed in Section 3.4 is that P_V and P_Q are spectrally equivalent to A and Q, respectively. We analyse here the spectral properties of $P_V^{-1}A$ and $P_Q^{-1}Q$. We refer to [43, Section 4.2], where such properties are used to derive explicit bounds for the eigenvalues of the preconditioned system $\mathbf{P}^{-1}\mathbf{A}$, in the special case of the block diagonal preconditioner. In particular, if the eigenvalues of $P_V^{-1}A$ and $P_Q^{-1}Q$ are bounded away from 0 and infinity uniformly with respect to h and p, then so are the eigenvalues of the full system.

The bilinear forms $\mathcal{A}(\cdot,\cdot)$ and $\widehat{\mathcal{A}}(\cdot,\cdot)$ satisfy

$$2C_{\text{Korn}}\nu_{\min} \left|\boldsymbol{v}\right|^2_{\mathbf{H}^1(\Omega)} \leq \mathcal{A}(\boldsymbol{v},\boldsymbol{v}) \leq 2\nu_{\max} \left|\boldsymbol{v}\right|^2_{\mathbf{H}^1(\Omega)} \qquad \forall\, \boldsymbol{v} \in \mathbf{H}^1_0(\Omega), \tag{3.5.8}$$

$$2\widehat{C}_{\text{Korn}} \left|\widehat{\boldsymbol{v}}\right|^2_{\mathbf{H}^1(\widehat{\Omega})} \leq \widehat{\mathcal{A}}(\widehat{\boldsymbol{v}},\widehat{\boldsymbol{v}}) \leq 2 \left|\widehat{\boldsymbol{v}}\right|^2_{\mathbf{H}^1(\widehat{\Omega})} \qquad \forall\, \widehat{\boldsymbol{v}} \in \mathbf{H}^1_0(\widehat{\Omega}), \tag{3.5.9}$$

where C_{Korn} and $\widehat{C}_{\text{Korn}}$ are the Korn constants (for homogeneous Dirichlet boundary conditions on the whole boundary we have $C_{\text{Korn}} = \widehat{C}_{\text{Korn}} = 1/2$, see [31, Section 6.3]) and

$$\nu_{\min} := \inf_{\Omega} \nu, \qquad \nu_{\max} := \sup_{\Omega} \nu.$$

We also have that the bilinear forms $\mathcal{A}(\cdot,\cdot) + \sigma(\cdot,\cdot)$ and $\widehat{\mathcal{A}}(\cdot,\cdot) + \widehat{\sigma}(\cdot,\cdot)$ in the discrete spaces satisfy

$$C_1 \|\boldsymbol{v}_h\|^2_{\mathbf{H}^1_{pen}(\Omega)} \leq \mathcal{A}(\boldsymbol{v}_h,\boldsymbol{v}_h) + \sigma(\boldsymbol{v}_h,\boldsymbol{v}_h) \leq C_2 \|\boldsymbol{v}_h\|^2_{\mathbf{H}^1_{pen}(\Omega)} \qquad \forall\, \boldsymbol{v}_h \in \mathcal{V}^{RT}_{h,0}, \tag{3.5.10}$$

$$\widehat{C}_1 \|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{H}^1_{pen}(\widehat{\Omega})} \leq \widehat{\mathcal{A}}(\widehat{\boldsymbol{v}}_h,\widehat{\boldsymbol{v}}_h) + \widehat{\sigma}(\widehat{\boldsymbol{v}}_h,\widehat{\boldsymbol{v}}_h) \leq \widehat{C}_2 \|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{H}^1_{pen}(\widehat{\Omega})} \qquad \forall\, \widehat{\boldsymbol{v}}_h \in \mathcal{V}^{RT}_{h,0}, \tag{3.5.11}$$

where the norm $\|\cdot\|_{\mathbf{H}^1_{pen}(\widehat{\Omega})}$ is defined as $\|\cdot\|^2_{\mathbf{H}^1_{pen}(\Omega)} := \|\cdot\|^2_{\mathbf{H}^1(\Omega)} + \frac{C_{pen}}{h}\|\cdot\|^2_{\mathbf{L}^2(\partial\Omega)}$ and C_1, C_2, $\widehat{C}_1$ and $\widehat{C}_2$ are constants depending on C_{pen} and on the inverse estimate constants of the discrete spaces $\mathcal{V}^{RT}_{h,0}$ and $\mathcal{V}^{RT}_{h,0}$ respectively: these inequalities follow from [45, Lemma 6.2], [45, Lemma 6.3],[45, Eq. (6.9)] and the equivalence between $\|\cdot\|_{\mathbf{H}^1_{pen}(\Omega)}$ and $|\cdot|^2_{\mathbf{H}^1_{pen}(\Omega)} := |\cdot|^2_{\mathbf{H}^1(\Omega)} + \frac{C_{pen}}{h}\|\cdot\|^2_{\mathbf{L}^2(\partial\Omega)}$.

We start by proving bounds on the eigenvalues of $P_V^{-1}A$.

Theorem 3.1. *It holds*

$$\delta \leq \lambda_{\min}\left(P_V^{-1}A\right), \qquad \lambda_{\max}\left(P_V^{-1}A\right) \leq \Delta, \tag{3.5.12}$$

where δ and Δ are positive constants that do not depend on h or on p.

Proof. We begin with TH discretization case, proving (3.5.12) for $\delta = \delta^{TH}$ and $\Delta = \Delta^{TH}$. Let $\widehat{\boldsymbol{v}}_h \in \mathcal{V}^{TH}_{h,0}$ and let $\boldsymbol{v}_h := \widehat{\boldsymbol{v}}_h \circ \boldsymbol{G}^{-1} \in \mathcal{V}^{TH}_{h,0}$. Moreover, let $\mathbf{v}$ be the coordinate vector of $\widehat{\boldsymbol{v}}_h$ with respect to the basis (3.2.8). By the Courant-Fischer theorem, (3.5.12) is equivalent to find δ^{TH} and Δ^{TH} such that

$$\delta^{TH} \leq \frac{\mathbf{v}^T A^{TH} \mathbf{v}}{\mathbf{v}^T P_V^{TH} \mathbf{v}} \leq \Delta^{TH}.$$

Using (3.5.8), we have

$$2C_{\text{Korn}}\nu_{\min}\,|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)} \le \mathbf{v}^T A^{TH}\mathbf{v} \le 2\nu_{\max}\,|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)}\,.$$

Using (3.5.9) and decomposing $\widehat{\boldsymbol{v}}_h = \widehat{\boldsymbol{v}}_{h,1}+\widehat{\boldsymbol{v}}_{h,2}+\widehat{\boldsymbol{v}}_{h,3}$, where $\widehat{\boldsymbol{v}}_{h,k}$ are the Cartesian components of $\widehat{\boldsymbol{v}}_h$, we have for $k = 1, 2, 3$,

$$2\widehat{C}_{\text{Korn}}\,|\widehat{\boldsymbol{v}}_{h,k}|^2_{\mathbf{H}^1(\widehat{\Omega})} \le \widehat{\mathcal{A}}(\widehat{\boldsymbol{v}}_{h,k},\widehat{\boldsymbol{v}}_{h,k}) \le 2\,|\widehat{\boldsymbol{v}}_{h,k}|^2_{\mathbf{H}^1(\widehat{\Omega})}\,;$$

summing the three bounds above and using $\widehat{\mathcal{A}}(\widehat{\boldsymbol{v}}_{h,1},\widehat{\boldsymbol{v}}_{h,1}) + \widehat{\mathcal{A}}(\widehat{\boldsymbol{v}}_{h,2},\widehat{\boldsymbol{v}}_{h,2}) + \widehat{\mathcal{A}}(\widehat{\boldsymbol{v}}_{h,3},\widehat{\boldsymbol{v}}_{h,3}) = \mathbf{v}^T P_V^{TH}\mathbf{v}$ yields

$$2\widehat{C}_{\text{Korn}}\,|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})} \le \mathbf{v}^T P_V^{TH}\mathbf{v} \le 2\,|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})}\,;$$

in conclusion it suffices to prove

$$\frac{\delta^{TH}}{C_{\text{Korn}}\nu_{\min}} \le \frac{|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)}}{|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})}} \le \frac{\widehat{C}_{\text{Korn}}\Delta^{TH}}{\nu_{\max}}, \tag{3.5.13}$$

for suitable δ^{TH} and Δ^{TH} and for all all $\widehat{\boldsymbol{v}}_h \in \mathcal{V}^{TH}_{h,0}$ with $\boldsymbol{v}_h =: \widehat{\boldsymbol{v}}_h \circ \boldsymbol{G}^{-1} \in \mathcal{V}^{TH}_{h,0}$. In other words, we just need to prove the equivalence between $|\boldsymbol{v}_h|_{\mathbf{H}^1(\Omega)}$ and $|\widehat{\boldsymbol{v}}_h|_{\mathbf{H}^1(\widehat{\Omega})}$. One of the two bounds is

$$\begin{aligned}|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)} &= \int_\Omega \|\nabla \boldsymbol{v}_h\|_F^2\ \mathrm{d}\Omega = \int_{\widehat{\Omega}} |\det(J_{\boldsymbol{G}})|\,\left\|\nabla\widehat{\boldsymbol{v}}_h J_{\boldsymbol{G}}^{-1}\right\|_F^2\ \mathrm{d}\boldsymbol{\eta}\\ &\le \sup_{\widehat{\Omega}}\left\{|\det(J_{\boldsymbol{G}})|\,\left\|J_{\boldsymbol{G}}^{-1}\right\|_2^2\right\}\int_{\widehat{\Omega}}\|\nabla\widehat{\boldsymbol{v}}_h\|_F^2\ \mathrm{d}\boldsymbol{\eta} = \sup_{\widehat{\Omega}}\left\{|\det(J_{\boldsymbol{G}})|\,\left\|J_{\boldsymbol{G}}^{-1}\right\|_2^2\right\}|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})}\,,\end{aligned} \tag{3.5.14}$$

where we used the fact that, given any two matrices X, Y with conforming dimensions, it holds $\|XY\|_F^2 \le \|X\|_F^2\,\|Y\|_2^2$. For the other bound, just observe that $\widehat{\boldsymbol{v}}_h := \boldsymbol{v}_h \circ \boldsymbol{G}$, and then

$$|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})} \le \sup_{\Omega}\left\{|\det(J_{\boldsymbol{G}^{-1}})|\,\left\|J^{-1}_{\boldsymbol{G}^{-1}}\right\|_2^2\right\}|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)} = \sup_{\widehat{\Omega}}\left\{\frac{\|J_{\boldsymbol{G}}\|_2^2}{|\det(J_{\boldsymbol{G}})|}\right\}|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)}\,. \tag{3.5.15}$$

This concludes the proof for the TH case.

The RT case is similar, we just highlight the differences. As above, from (3.5.10) and (3.5.11), we get

$$C_1\|\boldsymbol{v}_h\|^2_{\mathbf{H}^1_{pen}(\Omega)} \le \mathbf{v}^T A^{RT}\mathbf{v} \le C_2\|\boldsymbol{v}_h\|^2_{\mathbf{H}^1_{pen}(\Omega)}, \tag{3.5.16}$$

$$\widehat{C}_1\|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{H}^1_{pen}(\widehat{\Omega})} \le \mathbf{v}^T P_V^{RT}\mathbf{v} \le \widehat{C}_2\|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{H}^1_{pen}(\widehat{\Omega})}, \tag{3.5.17}$$

where $\widehat{\boldsymbol{v}}_h \in \mathcal{V}^{RT}_{h,0}$, $\boldsymbol{v}_h = ((\det(J_{\boldsymbol{G}}))^{-1}J_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h)\circ\boldsymbol{G}^{-1} = (\widetilde{J}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h)\circ\boldsymbol{G}^{-1} \in \mathcal{V}^{RT}_{h,0}$ and $\mathbf{v}$ is the common coordinate vector. Then, we look for δ^{RT} and Δ^{RT} such that

$$\delta^{RT}\frac{\widehat{C}_2}{C_1} \le \frac{\|\boldsymbol{v}_h\|^2_{\mathbf{H}^1_{pen}(\Omega)}}{\|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{H}^1_{pen}(\widehat{\Omega})}} \le \frac{\widehat{C}_1}{C_2}\Delta^{RT}.$$

Direct computation shows that $\nabla\left(\widetilde{J}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h\right)=\widetilde{J}_{\boldsymbol{G}}\nabla\widehat{\boldsymbol{v}}_h+\mathbb{H}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h$, where $\widetilde{J}_{\boldsymbol{G}}$ and $\mathbb{H}_{\boldsymbol{G}}$ as in Section 3.3. It holds

$$\begin{aligned}
|\boldsymbol{v}_h|^2_{\mathbf{H}^1(\Omega)}=&\int_\Omega \|\nabla\boldsymbol{v}_h\|_F^2\ \mathrm{d}\Omega=\int_{\widehat{\Omega}}|\det(J_{\boldsymbol{G}})|\left\|\nabla\left(\widetilde{J}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h\right)J_{\boldsymbol{G}}^{-1}\right\|_F^2\mathrm{d}\widehat{\Omega}\\
&\le 2\int_{\widehat{\Omega}}|\det(J_{\boldsymbol{G}})|\left(\left\|\widetilde{J}_{\boldsymbol{G}}\nabla\widehat{\boldsymbol{v}}_h J_{\boldsymbol{G}}^{-1}\right\|_F^2+\left\|\left(\mathbb{H}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h\right)J_{\boldsymbol{G}}^{-1}\right\|_F^2\right)\mathrm{d}\widehat{\Omega}\\
&\le 2\sup_{\widehat{\Omega}}\left\{|\det(J_{\boldsymbol{G}})|\left\|J_{\boldsymbol{G}}^{-1}\right\|_2^2\left\|\widetilde{J}_{\boldsymbol{G}}\right\|_2^2,|\det(J_{\boldsymbol{G}})|\left\|J_{\boldsymbol{G}}^{-1}\right\|_2^2\left\|\mathbb{H}_{\boldsymbol{G}}\right\|_F^2\right\}\left\|\widehat{\boldsymbol{v}}_h\right\|^2_{\mathbf{H}^1(\widehat{\Omega})},
\end{aligned}$$

where $\|\mathbb{H}_{\boldsymbol{G}}\|_F^2$ is the Frobenius tensor norm of $\mathbb{H}_{\boldsymbol{G}}$. Moreover, it holds

$$\begin{aligned}
\|\boldsymbol{v}_h\|^2_{\mathbf{L}^2(\Omega)}&=\int_\Omega|\boldsymbol{v}_h|^2\mathrm{d}\Omega=\int_{\widehat{\Omega}}|\det(J_{\boldsymbol{G}})|\left\|\widetilde{J}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h\right\|_2^2\mathrm{d}\widehat{\Omega}\\
&\le\sup_{\widehat{\Omega}}\left\{|\det(J_{\boldsymbol{G}})|\left\|\widetilde{J}_{\boldsymbol{G}}\right\|_2^2\right\}\left\|\widehat{\boldsymbol{v}}_h\right\|^2_{\mathbf{L}^2(\widehat{\Omega})},\\
\|\boldsymbol{v}_h\|^2_{\mathbf{L}^2(\partial\Omega)}&=\int_{\partial\Omega}|\boldsymbol{v}_h|^2\mathrm{d}\Gamma\le\|\mathrm{cof}(\nabla\boldsymbol{G})\|_{L^\infty(\widehat{\Omega}),l}\int_{\partial\widehat{\Omega}}\left\|\widetilde{J}_{\boldsymbol{G}}\widehat{\boldsymbol{v}}_h\right\|_2^2\mathrm{d}\widehat{\Gamma}\\
&\le\|\mathrm{cof}(\nabla\boldsymbol{G})\|_{L^\infty(\widehat{\Omega}),l}\sup_{\partial\widehat{\Omega}}\left\{\left\|\widetilde{J}_{\boldsymbol{G}}\right\|_2^2\right\}\left\|\widehat{\boldsymbol{v}}_h\right\|^2_{\mathbf{L}^2(\partial\widehat{\Omega})}
\end{aligned}$$

where $\mathrm{cof}(\cdot)$ refers to the matrix of the cofactors and $\|\cdot\|_{L^\infty(\widehat{\Omega}),l}$ is defined as in [48].

By observing that $\widehat{\boldsymbol{v}}_h=(\widetilde{J}_{\boldsymbol{G}^{-1}}\boldsymbol{v}_h)\circ\boldsymbol{G}$, we can use a similar argument to show that

$$\begin{aligned}
|\widehat{\boldsymbol{v}}_h|^2_{\mathbf{H}^1(\widehat{\Omega})}&\le 2\sup_{\Omega}\left\{|\det(J_{\boldsymbol{G}^{-1}})|\left\|J_{\boldsymbol{G}^{-1}}^{-1}\right\|_2^2\left\|\widetilde{J}_{\boldsymbol{G}^{-1}}\right\|_2^2,|\det(J_{\boldsymbol{G}^{-1}})|\left\|J_{\boldsymbol{G}^{-1}}^{-1}\right\|_2^2\left\|\mathbb{H}_{\boldsymbol{G}^{-1}}\right\|_F^2\right\}\|\boldsymbol{v}_h\|^2_{\mathbf{H}^1(\Omega)},\\
\|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{L}^2(\widehat{\Omega})}&\le\sup_{\Omega}\left\{|\det(J_{\boldsymbol{G}^{-1}})|\left\|\widetilde{J}_{\boldsymbol{G}^{-1}}\right\|_2^2\right\}\|\boldsymbol{v}_h\|^2_{\mathbf{L}^2(\Omega)},\\
\|\widehat{\boldsymbol{v}}_h\|^2_{\mathbf{L}^2(\partial\widehat{\Omega})}&\le\left\|\mathrm{cof}(\nabla\boldsymbol{G}^{-1})\right\|_{L^\infty(\Omega),l}\sup_{\partial\Omega}\left\{\left\|\widetilde{J}_{\boldsymbol{G}^{-1}}\right\|_2^2\right\}\|\boldsymbol{v}_h\|^2_{\mathbf{L}^2(\partial\Omega)}.
\end{aligned}$$

This concludes the analysis of the RT case. □

We next analyse $P_Q^{-1}Q$.

Theorem 3.2. *It holds*

$$\theta\le\lambda_{\min}\left(P_Q^{-1}Q\right),\qquad\lambda_{\max}\left(P_Q^{-1}Q\right)\le\Theta,\tag{3.5.18}$$

where θ and Θ are positive constants that do not depend on h or on p.

Proof. We report the proof for TH discretization. The proof for the RT discretization can be derived in an analogous way.

By Courant-Fischer theorem, we need to prove

$$\theta\le\frac{\langle Q\mathbf{g},\mathbf{g}\rangle}{\langle P_Q\mathbf{g},\mathbf{g}\rangle}\le\Theta\quad\forall\,\mathbf{g}\in\mathbb{R}^{n_Q}.$$

Let $\mathbf{g} \in \mathbb{R}^{n_Q}$ and $g_h = \sum_{i=1}^{n_Q} [\mathbf{g}]_i \widehat{B}_{i,\boldsymbol{p}}^{\boldsymbol{\alpha}}$. It holds

$$\begin{aligned}\mathbf{g}^T Q^{TH} \mathbf{g} &= \int_{\widehat{\Omega}} g_h^2 \nu^{-1} |\det(J_{\mathbf{G}})| \, \mathrm{d}\widehat{\Omega} \leq \sup_{\widehat{\Omega}} \left(|\det(J_{\mathbf{G}})| \; \nu^{-1} \right) \int_{\widehat{\Omega}} g_h^2 \, \mathrm{d}\widehat{\Omega} \\ &\leq \sup_{\widehat{\Omega}} \left(|\det(J_{\mathbf{G}})| \; \nu^{-1} \right) \mathbf{g}^T P_Q^{TH} \mathbf{g}, \end{aligned} \tag{3.5.19}$$

and, in an analogous way, one can prove the other side of the inequality. □

Remark 3.1. *The constants δ, Δ, θ and Θ depend on the parametrization $\boldsymbol{G}$ and on the kinematic viscosity ν. This dependence can be inferred from the proof of Theorems 3.1–3.2. Considering for example the TH case, from* (3.5.13)*–*(3.5.15) *and using*

$$\left[\sup_{\widehat{\Omega}} \left\{ \frac{\|J_{\boldsymbol{G}}\|_2^2}{|\det(J_{\boldsymbol{G}})|} \right\} \right]^{-1} = \inf_{\widehat{\Omega}} \left\{ \frac{|\det(J_{\boldsymbol{G}})|}{\|J_{\boldsymbol{G}}\|_2^2} \right\},$$

we get to the following admissible choices

$$\delta^{TH} = C_{\text{Korn}} \nu_{\min} \inf_{\widehat{\Omega}} \left\{ \frac{|\det(J_{\boldsymbol{G}})|}{\|J_{\boldsymbol{G}}\|_2^2} \right\} \;\; \textit{and} \;\; \Delta^{TH} = \widehat{C}_{\text{Korn}}^{-1} \nu_{\max} \sup_{\widehat{\Omega}} \left\{ |\det(J_{\boldsymbol{G}})| \, \|J_{\boldsymbol{G}}\|_2^2 \right\}.$$

In a similar way, from (3.5.19)*, we have following admissible choices*

$$\theta^{TH} = \inf_{\widehat{\Omega}} (|\det(J_{\mathbf{G}})| \nu^{-1}), \quad \textit{and} \; \Theta^{TH} = \sup_{\widehat{\Omega}} (|\det(J_{\mathbf{G}})| \nu^{-1}).$$

3.5.2 Preconditioners application by the fast diagonalization method

At each iteration of our iterative solver we have to solve

$$\mathbf{P}\mathbf{s} = \mathbf{r}, \tag{3.5.20}$$

where $\mathbf{r}$ is the current residual and $\mathbf{P}$ is a preconditioner, that can be either matrix from (3.4.2), (3.4.3) or (3.4.4). Besides multiplications by B or B^T, to accomplish this task we need to solve the linear systems with matrices P_V and P_Q. Thanks to (2.2.3), (2.2.5) and the band structure of the univariate factors in (3.5.7), the solution of a linear system with matrix P_Q is obtained in a direct way with only $O(pn_Q)$ FLOPs.

On the other hand, the solution of a linear system with matrix P_V requires to solve three Sylvester-like equations, one for each diagonal block $P_{V,k}$. Following [93], to accomplish this aim we use the fast diagonalization (FD) direct method. We refer to Section 2.3 for the two-dimensional case. Here, we briefly explain its main features in three dimensions.

Consider the general Sylvester-like system:

$$R\mathbf{q} := \left(K_3 \otimes M_2 \otimes M_1 + M_3 \otimes K_2 \otimes M_1 + M_3 \otimes M_2 \otimes K_1 \right) \mathbf{q} = \mathbf{t}, \tag{3.5.21}$$

with both M_i and K_i symmetric and positive definite matrices for $i = 1, 2, 3$. We assume for simplicity that the matrices K_i and M_i all have the same order n for $i = 1, 2, 3$. Let

$$K_i U_i = M_i U_i D_i, \quad i = 1, 2, 3, \tag{3.5.22}$$

be the eigendecomposition of the pencils (K_i, M_i), where D_i are diagonal matrices containing the eigenvalues of $M_i^{-1} K_i$ and $U_i^T M_i U_i = \mathbb{I}_n$. We have $M_i = U_i^{-T} U_i^{-1}$ and $K_i = U_i^{-T} D_i U_i^{-1}$.

Then, we can factorize R as

$$R = (U_3 \otimes U_2 \otimes U_1)^{-T} (\mathbb{I}_n \otimes \mathbb{I}_n \otimes D_1 + \mathbb{I}_n \otimes D_2 \otimes \mathbb{I}_n + D_3 \otimes \mathbb{I}_n \otimes \mathbb{I}_n)(U_3 \otimes U_2 \otimes U_1)^{-1}.$$

Exploiting (2.2.1), (2.2.5) and the factorization above, the solution of (3.5.21) can be computed by the following algorithm.

Algorithm 2 3D FD method

```
1: Setup: Compute the generalized eigendecompositions (3.5.22)
2: Application: Compute $\widetilde{\mathbf{t}} = (U_3 \otimes U_2 \otimes U_1)^T \mathbf{t}$
3:              Compute $\widetilde{\mathbf{q}} = (\mathbb{I}_n \otimes \mathbb{I}_n \otimes D_1 + \mathbb{I}_n \otimes D_2 \otimes \mathbb{I}_n + D_3 \otimes \mathbb{I}_n \otimes \mathbb{I}_n)^{-1} \widetilde{\mathbf{t}}$
4:              Compute $\mathbf{q} = (U_3 \otimes U_2 \otimes U_1)\, \widetilde{\mathbf{q}}$
```

Algorithm 2 requires $12n^4 + O(n^3) = 12N_{dof}^{4/3} + O(N_{dof})$ FLOPs, where $N_{dof} = n^3$ denotes the order of R. Step 1, i.e the setup, and Step 3 are optimal as they require only $O(N_{dof})$ FLOPs. The asymptotic dominant cost, i.e. $12N_{dof}^{4/3}$ FLOPs, is related to the matrix-matrix products of step 2 and step 4. However Step 2 and Step 4, being BLAS level 3 operations, are typically implemented in a highly efficient way on modern computers. As a consequence, despite their superlinear computational cost, in practice they do not dominate the computational time of the overall iterative strategy (see the numerical experiments of [93] and the ones in the present chapter for more details on this important point).

3.5.3 Inclusion of the geometry and coefficients information in P_V and P_Q

The proposed preconditioners P_V and P_Q are robust with respect to the mesh size and spline degree. However they do not incorporate any information from the coefficients (either the geometry map $\boldsymbol{G}$ and or the kinematic viscosity ν) and in fact the preconditioner's quality is affected from the coefficients. This is reflected in the analysis of Section 3.5.1 (see Remark 3.1 for the TH case). Numerical tests of Section 3.6 confirm this expectation. We therefore present two strategies that partially incorporate ν and $\boldsymbol{G}$ in P_V and P_Q, without increasing the preconditioners computational cost.

First, we consider a diagonal scaling. In particular, we replace P_Q by $P_Q^{\boldsymbol{G}} := D_Q^{1/2} P_Q D_Q^{1/2}$, where D_Q is a diagonal matrix having diagonal entries

$$[D_Q]_{i,i} = [Q]_{i,i} \,/\, [P_Q]_{i,i} \,.$$

Even though we postpone a mathematical analysis of it to a further work, the numerical tests in Section 3.6 show that this cheap modification of the preconditioner is sufficient to give $P_Q^{\boldsymbol{G}}$ robustness with respect to the coefficients (not only $\boldsymbol{G}$, as indicated, but also ν).

The same idea, applied to P_V, while able to incorporate efficiently the contribution of the scalar coefficient ν, is less effective when the geometry parametrization is far from a scaled identity. In this case we propose to incorporate some components of the geometry parametrization into the univariate matrix factors appearing in (3.5.4) and (3.5.5) in order to build a preconditioner $\widehat{P}_V$ such that Algorithm 2 can still be used. We focus on the diagonal blocks A_{kk} and we incorporate in P_V some information on the parametrization present in the A_{kk} by making approximations of the full matrix $\boldsymbol{C}_k$ (see equations (3.3.8), (3.3.10)), whose entries are functions of three variables that we denote with $[\boldsymbol{C}_k]_{i,j}(\boldsymbol{\eta})$:

$$\boldsymbol{C}_k(\boldsymbol{\eta}) = \begin{bmatrix} [\boldsymbol{C}_k]_{1,1}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{1,2}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{1,3}(\boldsymbol{\eta}) \\ [\boldsymbol{C}_k]_{2,1}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{2,2}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{2,3}(\boldsymbol{\eta}) \\ [\boldsymbol{C}_k]_{3,1}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{3,2}(\boldsymbol{\eta}) & [\boldsymbol{C}_k]_{3,3}(\boldsymbol{\eta}) \end{bmatrix}. \tag{3.5.23}$$

We discard the off-diagonal terms and approximate the diagonal entries $[\boldsymbol{C}_k]_{1,1}(\boldsymbol{\eta})$, $[\boldsymbol{C}_k]_{2,2}(\boldsymbol{\eta})$ and $[\boldsymbol{C}_k]_{3,3}(\boldsymbol{\eta})$ by the algorithm reported in Appendix A as follows

$$\boldsymbol{C}_k(\boldsymbol{\eta}) \approx \widehat{\boldsymbol{C}}_k(\boldsymbol{\eta}) := \begin{bmatrix} \omega_1^k(\eta_1)\mu_2^k(\eta_2)\mu_3^k(\eta_3) & 0 & 0 \\ 0 & \mu_1^k(\eta_1)\omega_2^k(\eta_2)\mu_3^k(\eta_3) & 0 \\ 0 & 0 & \mu_1^k(\eta_1)\mu_2^k(\eta_2)\omega_3^k(\eta_3) \end{bmatrix}.$$

The approximation above is computed directly at the quadrature points, hence no function space has to be selected a-priori. The cost of this algorithm is proportional to the number of quadrature points, hence in our setting it requires $O(n_{el}p^d)$ FLOPs. This cost could be easily reduced by computing the approximation on a coarser grid of points, and then extending by interpolation, as we do in the other chapter. However this is not necessary, since such cost is already negligible in the context of the iterative procedures considered in this chapter, as can be seen e.g. by comparing Tables 3.3 and 3.7.

Keeping the block-diagonal structure of P_V (cfr. (3.5.1)), we define for the TH discretization, $k = 1, 2, 3$ and $i, j = 1, ..., n_{V,k}^{TH}$

$$\left[\widehat{P}_{V,k}^{TH}\right]_{i,j} := \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+1}^{\boldsymbol{\alpha}}\right)^T \widehat{\boldsymbol{C}}_k^{TH} \nabla \widehat{B}_{j,\boldsymbol{p}+1}^{\boldsymbol{\alpha}} \, \mathrm{d}\boldsymbol{\eta},$$

while for the RT discretization, $k = 1, 2, 3$ and $i, j = 1, ..., n_{V,k}^{RT}$

$$\begin{aligned} \left[\widehat{P}_{V,k}^{RT}\right]_{i,j} := & \int_{\widehat{\Omega}} \left(\nabla \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k}\right)^T \widehat{\boldsymbol{C}}_k^{RT} \nabla \ \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \, \mathrm{d}\boldsymbol{\eta} + 2 \int_{\partial\widehat{\Omega}} \left[\frac{C_{pen}}{h} \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \mathbf{e}_k \cdot \left(\widehat{\boldsymbol{C}}_k^{RT} \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \right. \\ & - \left(\left(\nabla^s \left(\mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \widehat{\boldsymbol{n}} \right) \right) \cdot \left(\widehat{\boldsymbol{C}}_k^{RT} \mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \\ & \left. - \left(\left(\nabla^s \left(\mathbf{e}_k \widehat{B}_{j,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \widehat{\boldsymbol{n}} \right) \right) \cdot \left(\widehat{\boldsymbol{C}}_k^{RT} \mathbf{e}_k \widehat{B}_{i,\boldsymbol{p}+\mathbf{e}_k}^{\boldsymbol{\alpha}+\mathbf{e}_k} \right) \right] \mathrm{d}\widehat{\Gamma}. \end{aligned}$$

The preconditioners $\widehat{P}_{V,k}$ maintain the tensor structure of (3.5.4) and (3.5.5):

$$\begin{aligned} \widehat{P}_{V,1}^{TH} &= K_3^{1,TH} \otimes M_2^{1,TH} \otimes M_1^{1,TH} + M_3^{1,TH} \otimes K_2^{1,TH} \otimes M_1^{1,TH} + M_3^{1,TH} \otimes M_2^{1,TH} \otimes K_1^{1,TH}, \\ \widehat{P}_{V,2}^{TH} &= K_3^{2,TH} \otimes M_2^{2,TH} \otimes M_1^{2,TH} + M_3^{2,TH} \otimes K_2^{2,TH} \otimes M_1^{2,TH} + M_3^{2,TH} \otimes M_2^{2,TH} \otimes K_1^{2,TH}, \\ \widehat{P}_{V,3}^{TH} &= K_3^{3,TH} \otimes M_2^{3,TH} \otimes M_1^{3,TH} + M_3^{3,TH} \otimes K_2^{3,TH} \otimes M_1^{3,TH} + M_3^{3,TH} \otimes M_2^{3,TH} \otimes K_1^{3,TH}, \end{aligned}$$

$$\begin{aligned} \widehat{P}_{V,1}^{RT} &= \widetilde{K}_3^{1,RT} \otimes \widetilde{M}_2^{1,RT} \otimes M_1^{1,RT} + \widetilde{M}_3^{1,RT} \otimes \widetilde{K}_2^{1,RT} \otimes M_1^{1,RT} + \widetilde{M}_3^{1,RT} \otimes \widetilde{M}_2^{1,RT} \otimes K_1^{1,RT}, \\ \widehat{P}_{V,2}^{RT} &= \widetilde{K}_3^{2,RT} \otimes M_2^{2,RT} \otimes \widetilde{M}_1^{2,RT} + \widetilde{M}_3^{2,RT} \otimes K_2^{2,RT} \otimes \widetilde{M}_1^{2,RT} + \widetilde{M}_3^{2,RT} \otimes M_2^{2,RT} \otimes \widetilde{K}_1^{2,RT}, \\ \widehat{P}_{V,3}^{RT} &= K_3^{3,RT} \otimes \widetilde{M}_2^{3,RT} \otimes \widetilde{M}_1^{3,RT} + M_3^{3,RT} \otimes \widetilde{K}_2^{3,RT} \otimes \widetilde{M}_1^{3,RT} + M_3^{3,RT} \otimes \widetilde{M}_2^{3,RT} \otimes \widetilde{K}_1^{3,RT}, \end{aligned}$$

where, for $d, k = 1, 2, 3$ and $l, s = 2, ..., m_{\alpha_k}^{p+1} - 1$ the new pairs (K_k^d, M_k^d) are defined as

$$\begin{aligned} \left[K_k^{d,TH}\right]_{l-1,s-1} &= \int_{[0,1]} \omega_k^{d,TH}(\eta_k) (\widehat{b}_{l,p+1}^{\alpha_k})'(\eta_k) (\widehat{b}_{s,p+1}^{\alpha_k})'(\eta_k) \, \mathrm{d}\eta_k, \\ \left[M_k^{d,TH}\right]_{l-1,s-1} &= \int_{[0,1]} \mu_k^{d,TH}(\eta_k) \widehat{b}_{l,p+1}^{\alpha_k}(\eta_k) \ \widehat{b}_{s,p+1}^{\alpha_k}(\eta_k) \, \mathrm{d}\eta_k, \\ \left[K_k^{d,RT}\right]_{l-1,s-1} &= \int_{[0,1]} \omega_k^{d,RT}(\eta_k) (\widehat{b}_{l,p+1}^{\alpha_k+1})'(\eta_k) (\widehat{b}_{s,p+1}^{\alpha_k+1})'(\eta_k) \, \mathrm{d}\eta_k, \\ \left[M_k^{d,RT}\right]_{l-1,s-1} &= \int_{[0,1]} \mu_k^{d,RT}(\eta_k) \widehat{b}_{l,p+1}^{\alpha_k+1}(\eta_k) \ \widehat{b}_{s,p+1}^{\alpha_k+1}(\eta_k) \, \mathrm{d}\eta_k, \end{aligned}$$

while for $d,k=1,2,3$ and $l,s=1,...,m_p^{\alpha_k}$ the pairs $(\widetilde{K}_k^d,\widetilde{M}_k^d)$ are defined as

$$\begin{aligned}
\left[\widetilde{K}_k^{d,RT}\right]_{l,s} =& \int_{[0,1]} \omega_k^{d,RT}(\eta_k)(\widehat{b}_{l,p}^{\alpha_k})'(\eta_k)(\widehat{b}_{s,p}^{\alpha_k})'(\eta_k)\,\mathrm{d}\eta_k \\
&-\Bigg[\omega_k^{d,RT}(1)(\widehat{b}_{l,p}^{\alpha_k})'(1)\widehat{b}_{s,p}^{\alpha_k}(1)-\omega_k^{d,RT}(0)(\widehat{b}_{l,p}^{\alpha_k})'(0)\widehat{b}_{s,p}^{\alpha_k}(0) \\
&+\omega_k^{d,RT}(1)(\widehat{b}_{s,p}^{\alpha_k})'(1)\widehat{b}_{l,p}^{\alpha_k}(1)-\omega_k^{d,RT}(0)(\widehat{b}_{s,p}^{\alpha_k})'(0)\widehat{b}_{l,p}^{\alpha_k}(0) \\
&-2\frac{C_{pen}}{h}\left(\omega_k^{d,RT}(1)\widehat{b}_{l,p}^{\alpha_k}(1)\widehat{b}_{s,p}^{\alpha_k}(1)+\omega_k^{d,RT}(0)\widehat{b}_{l,p}^{\alpha_k}(0)\widehat{b}_{s,p}^{\alpha_k}(0)\right)\Bigg], \\
&, \\
\left[\widetilde{M}_k^{d,RT}\right]_{l,s} =& \int_{[0,1]} \mu_k^{d,RT}(\eta_k)\widehat{b}_{l,p}^{\alpha_k}(\eta_k)\ \widehat{b}_{s,p}^{\alpha_k}(\eta_k)\,\mathrm{d}\eta_k.
\end{aligned}$$

Then, we apply a diagonal scaling. This leads to an effective preconditioner having the form $P_V^{\boldsymbol{G}} := D_V^{1/2}\widehat{P}_V D_V^{1/2}$, where D_V has diagonal entries $[D_V]_{i,i} = [A]_{i,i} \,/\, \left[\widehat{P}_V\right]_{i,i}$.

We use the following notation: $\mathbf{P}_D^{\boldsymbol{G}}$, $\mathbf{P}_T^{\boldsymbol{G}}$ and $\mathbf{P}_C^{\boldsymbol{G}}$ are the preconditioner matrices for the Stokes system obtained by replacing P_V and P_Q with $P_V^{\boldsymbol{G}}$ and $P_Q^{\boldsymbol{G}}$ in (3.4.2), (3.4.3) and (3.4.4), respectively. The corresponding preconditioned strategies are then referred to as $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES, $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES.

3.6 Numerical results

We present here numerical experiments to show the performance of our preconditioning strategies. All the tests are performed by Matlab (version 8.5.0.197613 R2015a) and using the GeoPDEs toolbox [111], on a Intel Xeon i7-5820K processor, running at 3.30 GHz, and with 64 GB of RAM. We restrict our tests to a single computational thread. Indeed, even though our strategy would likely benefit from parallelization on a multicore hardware, as its main computational efforts are matrix products, a careful analysis of the parallel implementation would require an in-depth study, which is beyond the scope of this work.

In the construction and application of our preconditioner the two dominant steps are the eigendecomposition of the univariate matrices (step 1 in Algorithm 2) and the multiplication of Kronecker matrices (steps 2 and 4 in Algorithm 2). These two key operations are performed by the `eig` Matlab function and by the Tensorlab toolbox [102], respectively. The tolerance of both MINRES and GMRES is set to 10^{-8} and the initial guess is the null vector in all tests.

As a comparison, we consider a block-diagonal preconditioner based on an incomplete Cholesky factorization. In our case, the zero-fill incomplete Cholesky factorization, denoted IC(0), is computed by the MATLAB `ichol` routine for the matrix

$$\begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ 0 & 0 & A_{33} & 0 \\ 0 & 0 & 0 & Q \end{bmatrix}$$

and then used in a Conjugate Gradient (CG) inner iteration in order to approximate the application of the ideal preconditioner

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 \\ A_{21} & A_{22} & A_{23} & 0 \\ A_{31} & A_{32} & A_{33} & 0 \\ 0 & 0 & 0 & Q \end{bmatrix}. \tag{3.6.1}$$

This strategy is denoted IC(0)-MINRES. The tolerance of this inner CG loop is set to 10^{-2} as this maximizes the efficiency of the overall strategy in the numerical tests we consider below. The inner loop is needed to achieve robustness with respect to h, while robustness with respect to p is common for incomplete factorizations. For this reason, incomplete factorizations are often adopted in IGA as preconditioners: in the context of the Stokes system, see [35] where a similar approach is considered and benchmarked.

We remark that the geometry parametrization, without simplifications, is directly incorporated in the preconditioner (3.6.1). Therefore, as it is seen in the tests below, IC(0)-MINRES behaves quite robustly with respect to the geometry parametrizations (since $\lambda_{\max}\left(Q^{-1}BA^{-1}B^T\right)$ and $\lambda_{\min}\left(Q^{-1}BA^{-1}B^T\right)$ depend on Ω, some dependence on the shape of the domain is unavoidable), while the geometry parametrization has a critical role in our strategies. Also for this reason, IC(0)-MINRES is an important term of comparison.

We consider three different geometries, with increasing complexity (from the point of view of the geometry parametrization): the cube, the eighth of annulus, and a hollow torus with an eccentric annular cross-section (see Figure 5.1).

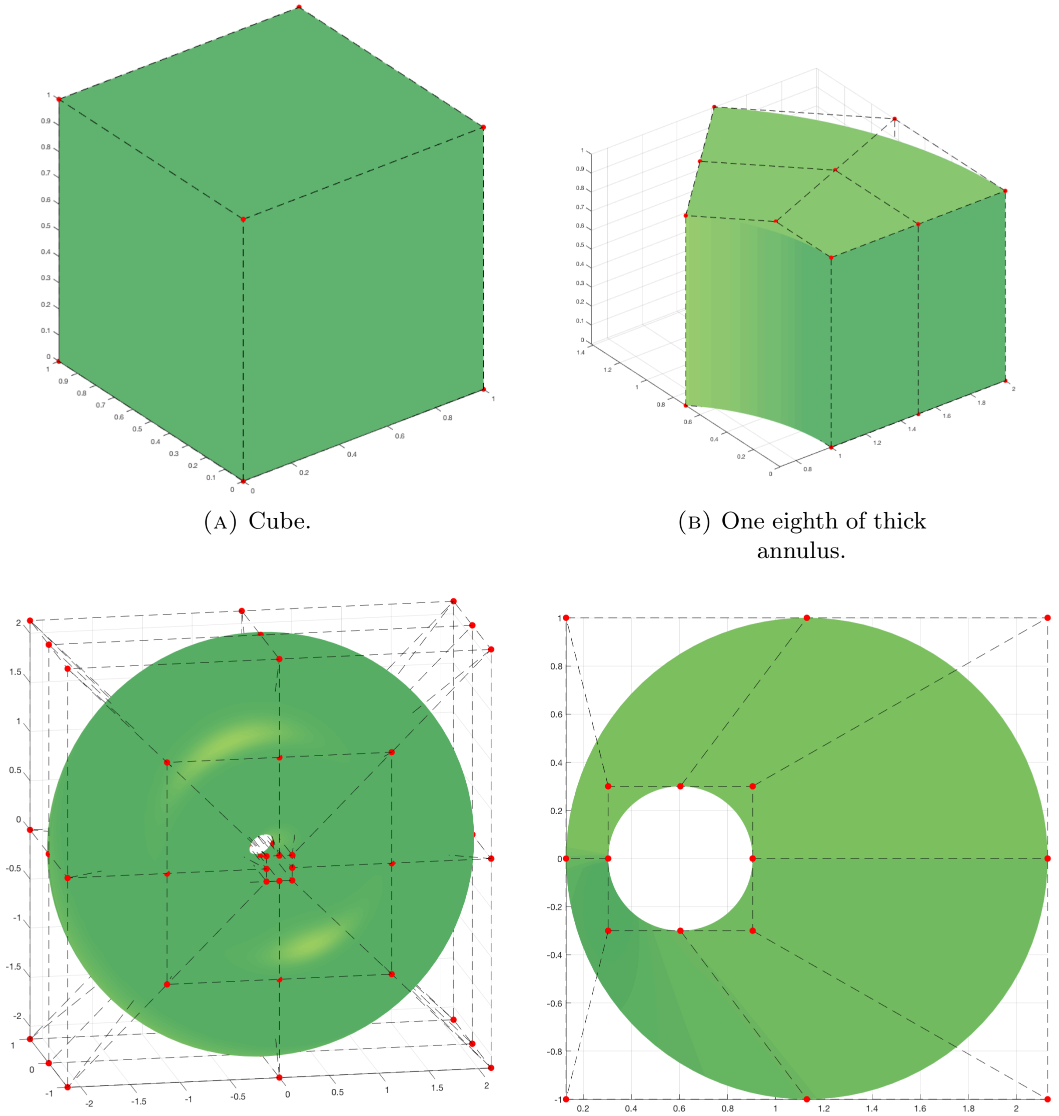
As discussed in Section 3.3, the Stokes problem is discretized using the spaces $\mathcal{V}_{h,0}^{TH}$, $\mathcal{Q}_{h,0}^{TH}$, $\mathcal{V}_{h,0}^{RT}$ and $\mathcal{Q}_{h,0}^{RT}$ defined respectively in (3.2.6a), (3.2.7), (3.2.10a) and (3.2.11). In all our tests we choose a uniform regularity $\boldsymbol{\alpha} = (\alpha, \alpha, \alpha)$ with $\alpha = p - 1$, except for the hollow torus domain where the spaces are C^0 at the boundary of the initial mesh elements, and C^α, $\alpha = p - 1$, once the mesh is refined. Note that p always refers to the spline degree of the pressure space. For Raviart-Thomas discretizations we choose $C_{pen} = 5(\alpha + 1)$ in (3.3.5), as it numerically leads to stable schemes (see [45]).

Tables 3.1–3.10 report the total solving time, which includes the preconditioner setup and the MINRES/ GMRES iterations. However, we exclude the time for the formation of the pressure mass matrix Q, which is needed in IC(0) and $\mathbf{P}_D^{\boldsymbol{G}}$, $\mathbf{P}_T^{\boldsymbol{G}}$, $\mathbf{P}_C^{\boldsymbol{G}}$ setup (though only the main diagonal of Q is needed in our approaches, and, in all cases, only a low-order approximation of Q is needed for preconditioning). Indeed, it is well known that the formation of isogeometric matrices is expensive unless ad-hoc routines are adopted (e.g. the weighted-quadrature approach [30] or the low-rank approach [78]). In this work, we only focus on the solver and do not address the efficient formation of the matrix. We denote by n_{el} the number of elements in each parametric direction. The symbol "$*$" denotes the impossibility of formation of the matrix $\mathbf{A}$, due to memory requirements.

In Table 3.7 we report, only for the eighth of annulus testcase, the preconditioner setup time and the preconditioner application time, separately, and in Table 3.8 we report the percentage of computing time spent in the preconditioner application. Finally, Table 3.11 contains number of iterations and solving times obtained with three different choices of variable kinematic viscosity ν in the hollow torus domain.

Cube. We first consider the symmetric driven cavity problem in $\Omega = \widehat{\Omega} = [0,1]^3$ (Figure 4.1b). In this case, $\boldsymbol{G}$ is the identity map and therefore $A_{kk} = P_{V,k}$. Homogeneous boundary conditions for the velocity on the lateral sides of the cube and a velocity equal to $[1,0,0]^T$ at the top and to $[-1,0,0]^T$ at the bottom are imposed, while f is the null function and $\nu = 1$.

In Table 3.1 we report, for the TH discretization, $\mathbf{P}_D$-MINRES and IC(0)-MINRES performances. The former is much faster, especially for high degree. $\mathbf{P}_D$-MINRES results with RT discretization are reported in Table 3.2. The computational time is lower compared to TH discretization since, for equal mesh sizes, the TH velocity space is about 2^3 times bigger than the one for RT. In all cases the number of iterations is uniformly bounded with respect to p and n_{el}.



(A) Cube.

(B) One eighth of thick annulus.

(C) Hollow torus.

(D) Hollow torus (cross section).

FIGURE 3.1: Stokes system. Computational domains.

	(TH) $\mathbf{P}_D$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	48 / 0.16	51 / 0.21	52 / 0.43	52 / 0.81
8	53 / 0.74	53 / 1.49	53 / 3.01	53 / 5.70
16	56 / 5.61	56 / 12.76	56 / 26.54	56 / 51.00
32	56 / 52.23	56 / 114.07	$*$	$*$

	(TH) IC(0)-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	35 / 0.22	37 / 0.69	37 / 1.71	37 / 3.77
8	34 / 2.82	37 / 7.22	35 / 16.10	36 / 33.76
16	35 / 35.09	35 / 74.34	35 / 151.87	35 / 305.90
32	36 / 482.25	36 / 902.51	$*$	$*$

TABLE 3.1: Stokes system. Cube domain (TH). Performance of $\mathbf{P}_D$-MINRES (upper table) and IC(0)-MINRES (lower table).

	(RT) $\mathbf{P}_D$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	43 / 0.13	46 / 0.18	48 / 0.23	48 / 0.39
8	54 / 0.23	52 / 0.44	52 / 0.85	52 / 1.59
16	55 / 0.95	53 / 2.56	52 / 4.77	52 / 9.02
32	55 / 6.39	54 / 16.67	52 / 34.58	$*$

TABLE 3.2: Stokes system. Cube domain (RT). Performance of $\mathbf{P}_D$-MINRES.

One eighth of thick annulus. Now we consider the eighth of a thick annulus domain (Figure 3.1b). The internal radius and the height are equal to 1, while the external radius is equal to 2. The boundary data represent a generalization of the symmetric driven cavity boundary conditions, i.e. the velocity is constrained to be $[-1,0,0]^T$ on the set $\{y=0\}$ and $[\sqrt{2}/2, \sqrt{2}/2, 0]^T$ on the opposite side, while homogeneous boundary conditions are imposed anywhere else. Note that in this case $A_{kk} \neq P_{V,k}$. The kinematic viscosity ν is constant and equal to 1.

Table 3.3 shows the results of $\mathbf{P}_D$-MINRES, $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES and IC(0)-MINRES for TH discretization. Again, IC(0)-MINRES is not competitive with $\mathbf{P}_D$-MINRES and $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES in terms of computing time. The use of $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES halves the number of iterations and the solving time w.r.t. $\mathbf{P}_D$-MINRES, indicating that the inclusion of some geometry information improves the performance of the preconditioner. In Table 3.4 we report results for $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES with RT discretization. The performances of $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES with TH and RT discretizations are reported in Table 3.5 and Table 3.6 respectively. We do not report results for $\mathbf{P}_T$-GMRES and $\mathbf{P}_C$-GMRES, as the effect of not including any geometry in the preconditioners is similar to the case of the block diagonal preconditioner. We see that, though the number of iterations of both $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES is lower than $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES, they are comparable to it in terms of CPU time. This is due to the higher application cost of the block triangular and constraint preconditioners (which is mainly related to the matrix-vector products with B and B^T). We emphasize that, again, in all the FD-based strategies the number of iterations is uniformly bounded with respect to p and n_{el}.

In order to better understand the behaviour of the preconditioners, and identify directions of further improvements, we analyse in Table 3.7 the computational costs for the setup and the application of the preconditioners. We recall that for IC(0)-MINRES, the application corresponds to the execution of the inner CG iterative solver with residual tolerance 10^{-2}. In all cases, we assume the pressure mass matrix Q is given. Table 3.7 reports the total time spent in the preconditioner setup and application. We clearly see that the FD-based preconditioners are much faster than the incomplete factorization. Note that the setup time for $\mathbf{P}_D^{\boldsymbol{G}}$ is higher than for $\mathbf{P}_D$ due to the cost of computing the separable approximation of the geometry (see the Appendix): further studies and tune up of this procedure will be considered in our following works.

In Table 3.8, preconditioner application time is compared with the overall computation time of the iterative solver. With $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES strategy, the percentage of time spent for the preconditioner is negligible, e.g. when $p=5$ and $n_{el}=16$ it is less than 1%. The computation time is indeed mainly spent in the matrix-vector multiplication. This situation suggests that further improvements could be obtained shifting towards a matrix-free implementation [94].

The results of Table 3.7 and 3.8 clearly show that the suboptimal asymptotic cost $O(N_{dof}^{4/3})$ of the preconditioner is not seen in practice, up to the largest problem tested. Note in particular from Table 3.7 that the application times of the FD-based preconditioners scale with respect to h much better than the asymptotic cost would suggest. This is due to the high efficiency of the routines that computes the dense matrix-matrix products that are the core of the FD method.

Hollow torus. The last domain examined is a torus with a hole (Figure 3.1c), obtained by revolving an eccentric annulus (Figure 3.1d) around the y axis. We take $\nu = 1$, $f = [\cos(\arctan(x/z)),\ \sin(4\pi x),\ \sin(\arctan(x/z))]^T$ and we impose homogeneous Dirichlet boundary conditions anywhere on the external boundary. We consider here the periodic setting, imposing C^0 periodic continuity in the function space. For this problem, we present only TH discretization results and focus on the effects of the geometry parametrization on the performances of the preconditioning strategies. Computing time and number of iterations of

	(TH) $\mathbf{P}_D$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	116 / 0.39	128 / 0.56	137 / 1.12	146 / 2.14
8	146 / 1.66	153 / 4.02	158 / 8.79	160 / 16.83
16	163 / 16.53	164 / 38.54	165 / 75.95	162 / 138.17
32	169 / 181.68	166 / 337.37	$*$	$*$

	(TH) $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	65 / 0.21	68 / 0.33	69 / 0.57	72 / 1.09
8	72 / 0.91	74 / 2.06	74 / 4.24	75 / 8.01
16	77 / 8.11	77 / 18.82	77 / 36.70	77 / 67.74
32	79 / 90.56	79 / 168.60	$*$	$*$

	(TH) IC(0)-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	39 / 0.28	39 / 0.79	41 / 1.64	41 / 32.69
8	39 / 3.13	39 / 7.44	39 / 16.47	39 / 32.69
16	40 / 39.44	39 / 80.53	37 / 157.37	37 / 281.24
32	38 / 611.55	38 / 1085.21	$*$	$*$

Table 3.3: Stokes system. One eighth of thick annulus domain (TH). Performance of $\mathbf{P}_D$-MINRES (upper table), $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES (middle table) and IC(0)-MINRES (lower table).

	(RT) $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	59 / 0.22	58 / 0.17	62 / 0.30	63 / 0.54
8	63 / 0.29	63 / 0.58	61 / 1.09	64 / 2.10
16	67 / 1.36	65 / 3.23	65 / 6.37	66 / 12.07
32	65 / 8.71	66 / 23.73	66 / 48.38	$*$

Table 3.4: Stokes system. One eighth of thick annulus domain (RT). Performance of $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES.

	(TH) $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	38 / 0.20	42 / 0.28	42 / 0.56	47 / 1.17
8	41 / 0.78	42 / 1.78	43 / 4.50	45 / 8.50
16	43 / 7.57	44 / 17.52	45 / 35.43	46 / 66.21
32	45 / 76.69	46 / 165.72	$*$	$*$

	(TH) $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	35 / 0.21	37 / 0.30	39 / 0.59	41 / 1.15
8	37 / 0.80	38 / 1.77	39 / 4.33	41 / 8.25
16	38 / 7.19	39 / 16.51	40 / 33.47	41 / 62.98
32	39 / 61.29	40 / 152.44	$*$	$*$

Table 3.5: Stokes system. One eighth of thick annulus domain (TH). Performance of $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES (upper table) and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES (lower table).

	(RT) $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	41 / 0.19	44 / 0.20	46 / 0.35	48 / 0.69
8	46 / 0.34	47 / 0.71	49 / 1.48	50 / 5.55
16	47 / 1.72	49 / 7.77	50 / 16.57	52 / 32.86
32	48 / 21.15	50 / 56.50	52 / 120.06	$*$

	(RT) $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	37 / 0.19	38 / 0.22	39 / 0.36	40 / 0.68
8	38 / 0.34	40 / 0.71	41 / 1.41	42 / 4.98
16	39 / 1.63	40 / 6.81	41 / 14.42	42 / 28.18
32	39 / 18.30	40 / 48.05	41 / 100.99	$*$

Table 3.6: Stokes system. One eighth of thick annulus domain (RT). Performance of $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES (upper table) and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES (lower table).

	(TH) $\mathbf{P}_D$ Setup times / Total application times			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	0.02 / 0.19	0.02 / 0.20	0.02 / 0.20	0.03 / 0.21
8	0.04 / 0.27	0.04 / 0.29	0.04 / 0.33	0.04 / 0.37
16	0.05 / 0.87	0.06 / 0.95	0.06 / 1.09	0.06 / 1.18
32	0.09 / 7.21	0.12 / 9.94	$*$	$*$

	(TH) $\mathbf{P}_D^{\boldsymbol{G}}$ Setup times / Total application times			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	0.05 / 0.88	0.06 / 0.10	0.06 / 0.10	0.07 / 0.11
8	0.09 / 0.13	0.12 / 1.49	0.16 / 0.16	0.21 / 0.18
16	0.28 / 0.46	0.49 / 0.51	0.76 / 0.56	1.14 / 0.62
32	1.57 / 3.86	3.20 / 3.93	$*$	$*$

	(TH) IC(0) Setup times / Total application times			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	0.01 / 0.21	0.03 / 0.59	0.12 / 1.43	0.38 / 3.04
8	0.09 / 2.55	0.45 / 6.02	1.46 / 13.05	4.23 / 23.98
16	0.94 / 34.49	4.36 / 66.68	13.90 / 125.35	40.91 / 207.12
32	9.09 / 558.27	46.65 / 889.03	$*$	$*$

Table 3.7: Stokes system. One eight of thick annulus domain (TH). Setup times and total application times of $\mathbf{P}_D$ (top table), $\mathbf{P}_D^{\boldsymbol{G}}$ (middle table) and IC(0) (bottom table).

	(TH) $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
8	14.28%	6.79%	3.77%	2.24%
16	5.67%	2.70%	1.52%	0.91%
32	4.26 %	2.33%	$*$	$*$

	(TH) IC(0)-MINRES			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
8	81.46%	80.91%	79.23%	73.35%
16	87.44%	82.80%	79.65%	73.64%
32	91.28%	81.92%	$*$	$*$

Table 3.8: Stokes system. One eight of thick annulus domain (TH). Percentage of computing time of the preconditioner application in each MINRES iteration: $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES (top table) and IC(0)-MINRES (bottom table).

	(TH) $\mathbf{P}_D$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	2004 / 6.42	4125 / 39.16	6411 / 153.95	8305 / 478.69
8	5524 / 80.73	7875 / 360.15	9914 / 1117.12	11032 / 3286.67
16	9931 / 1081.01	11780 / 3763.90	12964 / 8776.73	13553 / 18626.03
32	12864 / 10244.45	13426 / 29344.81	$*$	$*$

	(TH) $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	77 / 0.31	87 / 0.89	97 / 2.59	104 / 6.24
8	96 / 1.52	104 / 4.99	110 / 12.82	115 / 34.70
16	119 / 13.87	124 / 40.89	133 / 91.82	139 / 197.30
32	142 / 116.95	147 / 344.34	$*$	$*$

	(TH) IC(0)-MINRES Iterations / Time			
n_{el}	$p = 2$	$p = 3$	$p = 4$	$p = 5$
4	49 / 1.05	46 / 3.74	50 / 11.79	50 / 31.42
8	45 / 5.42	45 / 18.52	45 / 51.18	45 / 126.83
16	45 / 45.11	43 / 125.60	45 / 307.79	45 / 660.63
32	45 / 493.12	44 / 1352.81	$*$	$*$

TABLE 3.9: Stokes system. Hollow torus domain (TH). Performance of $\mathbf{P}_D$-MINRES (upper table), $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES (middle table) and IC(0)-MINRES (lower table).

$\mathbf{P}_D$-MINRES, $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES and IC(0)-MINRES are reported in Table 3.9. As expected, the geometry parametrization of the hollow torus has a non-negligible influence on the performance of our preconditioners.

This is especially true for the $\mathbf{P}_D$-MINRES strategy, that requires thousands of iterations to converge. On the other hand, this influence is greatly reduced with partial inclusion of the geometry ($\mathbf{P}_D^{\boldsymbol{G}}$-MINRES). Here the number of iterations and the CPU times are two orders of magnitude lower than for $\mathbf{P}_D$-MINRES. CPU times for $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES are also significantly better than for IC(0)-MINRES, despite the fact the number of iterations is higher. We also remark that the number of iterations for $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES is only three times higher than $\mathbf{P}_D$-MINRES on the cube.

Finally, in Table 3.10 we present the computing times and the number of iterations of $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES. Also in this case, we do not report the performance of $\mathbf{P}_T$-GMRES and $\mathbf{P}_C$-GMRES because the effects of the non-inclusion of the geometry information are the same as for the block diagonal preconditioner. As for the cube domain, we see that the computing times of both GMRES based strategies are comparable with the computing times of $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES, even if the number of iterations is lower.

Hollow torus: variable ν. In this paragraph we investigate the effect of a variable kinematic viscosity ν on our preconditioning strategies. We consider the hollow torus domain with

	(TH) $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES	Iterations / Time		
n_{el}	$p=2$	$p=3$	$p=4$	$p=5$
4	44 / 0.30	50 / 0.80	57 / 2.39	61 / 6.89
8	49 / 1.25	54 / 4.54	58 / 11.98	62 / 31.52
16	58 / 10.78	60 / 32.86	63 / 73.52	67 / 159.46
32	68 / 105.31	71 / 275.54	$*$	$*$

	(TH) $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES	Iterations / Time		
n_{el}	$p=2$	$p=3$	$p=4$	$p=5$
4	37 / 0.28	41 / 0.74	45 / 2.09	50 / 6.09
8	41 / 1.16	45 / 4.07	49 / 10.82	53 / 28.59
16	51 / 10.27	55 / 31.73	59 / 72.91	63 / 158.12
32	69 / 113.81	72 / 299.62	$*$	$*$

Table 3.10: Stokes system. Hollow torus domain (TH). Performance of $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES (upper table) and $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES (lower table)

	$\mathbf{P}_D$-MINRES	$\mathbf{P}_D^{\boldsymbol{G}}$-MINRES	$\mathbf{P}_T^{\boldsymbol{G}}$-GMRES
$k=1$	13426 / 29344.81	147 / 344.34	71 / 275.54
$k=100$	17254 / 37667.04	180 / 400.46	84 / 325.02
$k=10000$	–	180 / 407.68	84 / 326.78

Table 3.11: Stokes system. Hollow torus domain (TH). Performance of $\mathbf{P}_D$-MINRES, $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES and $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES for $p=3$ and $n_{el}=32$. The symbol "–" denotes the fact the the solver does not converge because of stagnation.

$\nu = 1+(k-1)(1+\cos(\arctan(x/z)))/2)$ depending on a parameter k, $p=3$ and $n_{el}=32$ and we compare in Table 3.11 the performances of $\mathbf{P}_D$-MINRES, $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES and $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES.

$\mathbf{P}_D$-MINRES is the worse strategy both in terms of number of iterations and in computing times for all values of k and in the case $k=10000$ it does not even converge. The geometry inclusion strategy, on the other hand, succeeds in capturing the effect of the variable ν; the number of iterations of $\mathbf{P}_D^{\boldsymbol{G}}$-MINRES and $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES remains stable when k varies.

We remark that $\mathbf{P}_C^{\boldsymbol{G}}$-GMRES has a behaviour similar to $\mathbf{P}_T^{\boldsymbol{G}}$-GMRES, as it is also highlighted in the previous testcases, and for this reason we do not consider it in the table.

3.7 Conclusions

In this chapter we have addressed the problem of finding good preconditioners for isogeometric discretizations of the Stokes system. Our approach exploits the tensor-product structure of the multivariate B-spline basis. The application of our preconditioners $\mathbf{P}_D$, $\mathbf{P}_T$ and $\mathbf{P}_C$ (and their coefficients-including variants $\mathbf{P}_D^{\boldsymbol{G}}$, $\mathbf{P}_T^{\boldsymbol{G}}$ and $\mathbf{P}_C^{\boldsymbol{G}}$) requires the solution of linear systems that have a Kronecker structure, or a Sylvester-like equation structure. This can be performed by direct solvers with the highest efficiency. This also guarantees robustness with respect to both the spline degree p and mesh resolution. Numerical tests show that $\mathbf{P}_D^{\boldsymbol{G}}$, $\mathbf{P}_T^{\boldsymbol{G}}$ and $\mathbf{P}_C^{\boldsymbol{G}}$ allow to maintain the performance also in case of non-trivial geometries and highly oscillating coefficients.

We have performed a comparative numerical benchmarking with respect to a more common approach which uses a similar block structure for the preconditioner but applies it by an incomplete Cholesky factorization and an inner conjugate gradient. The solution time is always in favour of our preconditioners, despite that they are influenced by the geometry parametrization. Even more important is that our preconditioners are well suited for a matrix-free approach, which should lead to solvers that are orders of magnitude faster.

Chapter 4

The heat equation: least-squares method

In this chapter, we design and analyze an isogeometric method for parabolic equations, focusing on the heat equation as model problem. The most common numerical methods for time-dependent PDEs are obtained by discretizing separately in time (e.g, by difference schemes) and in space (e.g., by a Galerkin method). We consider instead the alternative approach of discretizing the PDE simultaneously in space and time, that is, the so-called space-time (variational) approach. A first idea of space-time finite element method has been introduced in [51, 85, 86] and developed for the heat conduction problem in [24]. Further pioneering studies on space-time methods have been [99, 67], where the authors consider a Galerkin formulation and add a least-squares operator to enhance stability and mitigate spurious oscillations.

More recently, the mathematical analysis of Galerkin space-time methods for parabolic equations has been developed in [97] for a wavelet discretization, and in [103] for a Galerkin finite element discretization. In the IGA framework, the idea of using smooth splines in time has been first proposed in [107]. In [17] the authors consider C^0 coupling between the space-time slabs with a suitable stabilized formulation that also yields to a sequential scheme. Space-time Isogeometric Analysis involving fluid-structure interaction, again based on discontinuous approximation in time, are proposed in [108, 109, 110]. A stabilized space-time isogeometric method for the heat equation has been proposed in [74, 75] and its time-parallel multigrid solver has been developed in [63].

In contrast to the existing space-time IGA works, in this work we adopt an L^2 least-squares approximation. The first appearance of a least-squares space-time formulation was in [84]. However, as discussed in [13, 14], the discretized formulation of [84] departs from the least-squares minimization principle. In [13, 14] the authors consider a least-squares finite element method for unsteady fluid dynamics problems. For second-order differential equations, the L^2 minimization of the equation residual would require C^1-continuous functions in the spatial variables, however [13, 14] recast the second-order equation into a set of first-order equations, whose least-squares formulation allows C^0 functions. Furthermore, [13, 14] introduce a time-marching approach to lower the memory requirement and the computational time. Henceforth, the most relevant contributions on space-time least-squares methods have retained these two features: 1) the minimization of first-order residuals and 2) the time-marching technique (similar to the use of time-slabs or discontinuous-in-time approximation). We refer to the book [16] for a review of the literature.

Our work departs from the setting described above: we consider high degree and smoothness splines in time and space with the following implications: 1) exploiting the C^1-continuity of our approximating function, we directly minimize the second-order residual and 2) we need to solve a global-in-time linear system. Point 1) represents an advantage while point 2) is addressed by exploiting the tensor product structure of the spline basis functions: we do not need to form the global space-time matrix, which is given as sum of Kronecker products of matrices, and we set up a preconditioner that relies on the solution of a Sylvester-like equation. Indeed, the least-squares formulation allows us to use the same preconditioning technique introduced in [93] for the Poisson problem, based the fast diagonalization method (see also Section 2.3). For the space-time least-squares formulation, the computational cost of the preconditioner setup is at most $O(N_{dof})$ floating-point operations (FLOPs) while its application is $O(N_{dof}^{1+1/d})$ FLOPs, where d is the number of space dimensions and N_{dof} denotes the total number of degrees-of-freedom (for simplicity, here we consider the same number of degrees-of-freedom in time and in each space direction). In our numerical benchmarks the measured computational time of the preconditioner, for serial single-core execution, is close to optimality, that is proportional to N_{dof}, with no dependence on p. Therefore, the preconditioner is robust with respect to the polynomial degree. Moreover, under the assumption that the coefficients of the equation do not depend on time, our approach requires a significantly small amount of memory compared to other space-time approaches: denoting by N_s the total number of degrees-of-freedom in space (and assuming the number of degrees-of-freedom in time is not too large, as in typical applications) the storage cost is $O(p^d N_s + N_{dof})$. This is exactly what one would get for low-order time-marching schemes.

Space-time methods facilitate the full parallelization of the solver, see [42, 54]. The preconditioner we propose fits in the framework, e.g., of [72]. We do not address this important issue in our work, that will be the focus of our further research.

The outline of the chapter as follows. In Section 4.1 we recall the notations for the univariate and multivariate B-Splines basis functions while the isogeometric spaces that we need for the discrete analysis are introduced in Section 4.2. The parabolic model problem is presented in Section 4.3, where we also discuss the well-posedness of the least-squares approximation and the a-priori error estimates. Section 4.4 focuses on preconditioning strategy and its spectral analysis. We show numerical results to assess the performance of the proposed preconditioner and to confirm the a-priori error estimates in Section 4.5. In Section 4.6 we draw conclusions and highlight future research directions. Section 4.7 contains some technical results while the last section resumes useful classical theorems used in this chapter.

4.1 Notations and main assumptions for the spline spaces

In this section we summarize the notations and the assumptions for the univariate and multivariate spline space that we employ in the rest of the chapter.

We consider functions of space and time, where the space domain is d-dimensional. Even if the analysis works for a general d, in the numerical tests we will focus on $d = 2, 3$, which are the most interesting cases in practical applications. Therefore we introduce $d + 1$ univariate knot vectors $\Xi_l := \{\xi_{l,1} \leq \cdots \leq \xi_{l,m_l+p_l+1}\}$ for $l = 1, \ldots, d$ and $\Xi_t := \{\xi_{t,1} \leq \cdots \leq \xi_{t,m_t+p_t+1}\}$. For the definition of univariate B-splines in each parametric direction we refer to Section 2.1.1. We collect the degree indexes in a vector $\boldsymbol{p} := (\boldsymbol{p}_s, p_t)$, where $\boldsymbol{p}_s := (p_1, \ldots, p_d) \in \mathbb{N}^d$. For the sake of simplicity, we consider $p_1 = \cdots = p_d =: p_s$ but the general case is similar.

In the following, h_s will denote the maximum mesh size in all spatial directions and h_t the mesh size in the time direction. We assume that the following quasi-uniformity condition on the knot vectors holds.

Assumption 4.1. *We assume that the knot vectors are quasi-uniform, that is, there exists α such that $0 < \alpha \leq 1$, independent of h_s and h_t, such that each non-empty knot span $(\xi_{l,i}, \xi_{l,i+1})$ fulfills $\alpha h_s \leq \xi_{l,i+1} - \xi_{l,i} \leq h_s$, for $1 \leq l \leq d$, and each non-empty knot span $(\xi_{t,i}, \xi_{t,i+1})$ fulfills $\alpha h_t \leq \xi_{t,i+1} - \xi_{t,i} \leq h_t$.*

We introduce the univariate spline spaces $\widehat{\mathcal{S}}_{h_s}^{p_s}$ and $\widehat{\mathcal{S}}_{h_t}^{p_t}$ and we denote by $\widehat{\Omega} := (0,1)^d$ the spatial parameter domain. Following (2.1.2), we denote the multivariate B-spline on $\widehat{\Omega} \times [0,1]$ as

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\eta},\tau) := \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta})\widehat{b}_{i_t,p_t}(\tau),$$

where $\widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta}) := \widehat{b}_{i_1,p_s}(\eta_1)\ldots\widehat{b}_{i_d,p_s}(\eta_d)$, $\boldsymbol{i_s} := (i_1,\ldots,i_d)$, $\boldsymbol{i} := (\boldsymbol{i_s}, i_t)$ and $\boldsymbol{\eta} = (\eta_1,\ldots,\eta_d)$. The corresponding spline space is denoted as

$$\widehat{\mathcal{S}}_h^{\boldsymbol{p}} := \operatorname{span}\left\{ \widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \;\middle|\; i_k = 1,...,m_k \text{ for } k = 1,\ldots,d; i_t = 1,\ldots,m_t \right\},$$

where $h := \max\{h_s, h_t\}$. We have $\widehat{\mathcal{S}}_h^{\boldsymbol{p}} = \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}} \otimes \widehat{\mathcal{S}}_{h_t}^{p_t} = \widehat{\mathcal{S}}_{h_s}^{p_s} \otimes \cdots \otimes \widehat{\mathcal{S}}_{h_s}^{p_s} \otimes \widehat{\mathcal{S}}_{h_t}^{p_t}$, where $\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}} := \operatorname{span}\left\{ \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta}) \;\middle|\; i_k = 1,...,m_k \text{ for } k = 1,\ldots,d \right\}$.

The minimum regularity of the spline spaces that we assume is the following.

Assumption 4.2. *We assume that $p_s \geq 2$, $\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}} \subset C^1(\widehat{\Omega})$, $p_t \geq 1$ and $\widehat{\mathcal{S}}_{h_t}^{\boldsymbol{p_t}} \subset C^0(\widehat{\Omega})$.*

4.2 Isogeometric spaces

The space domain $\Omega \subset \mathbb{R}^d$ is given as a spline non-singular single-patch, that is, the following conditions are fulfilled.

Assumption 4.3. *We assume that $\boldsymbol{F} : \widehat{\Omega} \to \Omega$, with $\boldsymbol{F} \in \left[\widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}}\right]^d$ on the closure of $\widehat{\Omega}$.*

Assumption 4.4. *We assume that $\boldsymbol{F}^{-1}$ has piecewise bounded derivatives of any order.*

Let $\boldsymbol{x} = (x_1,\ldots,x_d) := \boldsymbol{F}(\boldsymbol{\eta})$. Given $T > 0$, the space-time computational domain $\Omega \times [0,T]$ is given by the parametrization $\boldsymbol{G} \in \left[\widehat{\mathcal{S}}_h^{\boldsymbol{p}}\right]^{d+1}$ such that $\boldsymbol{G} : \widehat{\Omega} \times [0,1] \to \Omega \times [0,T]$ with $\boldsymbol{G}(\boldsymbol{\eta},\tau) := (\boldsymbol{F}(\boldsymbol{\eta}), T\tau) = (\boldsymbol{x}, t)$, and where $t := T\tau$. We introduce, in the parametric domain, the space with boundary conditions

$$\widehat{\mathcal{V}}_{h,0} := \left\{ \widehat{v}_h \in \widehat{\mathcal{S}}_h^{\boldsymbol{p}} \;\middle|\; \widehat{v}_h = 0 \text{ on } \partial\widehat{\Omega} \times (0,1) \text{ and } \widehat{v}_h = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\}.$$

Note that $\widehat{\mathcal{V}}_{h,0} = \widehat{\mathcal{V}}_{s,h_s,0} \otimes \widehat{\mathcal{V}}_{t,h_t,0}$, where

$$\begin{aligned}\widehat{\mathcal{V}}_{s,h_s,0} &:= \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}} \;\middle|\; \widehat{w}_h = 0 \text{ on } \partial\widehat{\Omega} \right\} \\ &= \operatorname{span}\left\{ \widehat{b}_{i_1,p_s}\ldots\widehat{b}_{i_d,p_s} \;\middle|\; i_k = 2,\ldots,m_k - 1;\; k = 1,\ldots,d \right\}, \qquad (4.2.1a)\\ \widehat{\mathcal{V}}_{t,h_t,0} &:= \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_t}^{p_t} \;\middle|\; \widehat{w}_h(0) = 0 \right\} = \operatorname{span}\left\{ \widehat{b}_{i_t,p_t} \;\middle|\; i_t = 2,\ldots,m_t \right\}. \qquad (4.2.1b)\end{aligned}$$

Reordering the basis and then introducing the colexicographical ordering of the degrees-of-freedom, we have

$$\widehat{\mathcal{V}}_{s,h_s,0} = \operatorname{span}\left\{ \widehat{b}_{i_1,p_s}\ldots\widehat{b}_{i_d,p_s} \middle| i_k = 1,\ldots,n_{s,k};\; k = 1,\ldots,d \right\} = \operatorname{span}\left\{ \widehat{B}_{i,\boldsymbol{p_s}} \middle| i = 1,\ldots,N_s \right\},$$

$$\widehat{\mathcal{V}}_{t,h_t,0} = \operatorname{span}\left\{ \widehat{b}_{i,p_t} \;\middle|\; i = 1,\ldots,n_t \right\}$$

and

$$\widehat{\mathcal{V}}_{h,0} = \operatorname{span}\left\{ \widehat{B}_{i,\boldsymbol{p}} \;\middle|\; i = 1,\ldots,N_{dof} \right\}, \qquad (4.2.2)$$

where we have defined

$$n_t := m_t - 1, \qquad n_{s,k} := m_k - 2, \qquad N_s := \prod_{k=1}^{d} n_{s,k}, \qquad N_{dof} := N_s n_t.$$

The isogeometric space we consider is the isoparametric push-forward of $\widehat{\mathcal{V}}_{h,0}$, i.e.

$$\mathcal{V}_{h,0} := \operatorname{span}\left\{ B_{i,\boldsymbol{p}} := \widehat{B}_{i,\boldsymbol{p}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, \dots, N_{dof} \right\}. \tag{4.2.3}$$

Note that $\mathcal{V}_{h,0}$ can be written as

$$\mathcal{V}_{h,0} = \mathcal{V}_{s,h_s,0} \otimes \mathcal{V}_{t,h_t,0},$$

where

$$\mathcal{V}_{s,h_s,0} := \operatorname{span}\left\{ B_{i,\boldsymbol{p}_s} := \widehat{B}_{i,\boldsymbol{p}_s} \circ \boldsymbol{F}^{-1} \;\middle|\; i = 1, \dots, N_s \right\},$$

$$\mathcal{V}_{t,h_t,0} := \operatorname{span}\left\{ b_{i,p_t} := \widehat{b}_{i,p_t}(\cdot/T) \;\middle|\; i = 1, \dots, n_t \right\}.$$

4.3 Parabolic model problem and its discretization

4.3.1 The heat equation and the regularity of its solution

In this section, after the definition of the model problem, we prove some results on the regularity of its solution. In order to help the reading of the proofs, we report in Section 4.8 the classical results of functional analysis that we employ, rewritten in our simplified setting.

We denote by ∂_t the partial time derivative and by Δ the laplacian w.r.t. spatial variables. If A and B are Hilbert spaces, $A \otimes B$ denotes the closure of their tensor product (see [3, Definition 12.3.2]). We also identify the spaces $H^m((0,T); H^n(\Omega))$, $H^n(\Omega) \otimes H^m(0,T)$ and $H^{n,m}(\Omega \times (0,T))$, (see [3, Section 12.7]). We denote by $H_\Delta(\Omega)$ the space $\left\{ z \in L^2(\Omega) \;\middle|\; \Delta z \in L^2(\Omega) \right\}$, we have the following result.

Proposition 4.1. *Under Assumptions 4.2–4.4, there exists a constant $C_\Delta > 0$, depending only on the space parametrization $\boldsymbol{F}$, such that*

$$\|z\|^2_{H^2(\Omega)} \le C_\Delta \|\Delta z\|^2_{L^2(\Omega)} \qquad \forall z \in H^1_0(\Omega) \cap H^2(\Omega). \tag{4.3.1}$$

Proof. From Assumptions 4.2–4.4, Ω has a piecewise smooth boundary with bounded curvature and, in particular, it has non-null interior angles (see the definition in [73, Chapitre III, pag. 161], reported in Definition 4.1) of Section 4.8. Then, we can use [73, Chapitre III, Lemme 11.1], reported in Lemma 4.8 of Section 4.8. □

We define the space

$$\mathcal{V}_0 := \left\{ v \in \left[\left(H^1_0(\Omega) \cap H^2(\Omega) \right) \otimes L^2(0,T) \right] \cap \left[L^2(\Omega) \otimes H^1(0,T) \right] \;\middle|\; v = 0 \text{ on } \Omega \times \{0\} \right\},$$

endowed with the norm

$$\|v\|^2_{\mathcal{V}_0} := \int_0^T \|\Delta v(\cdot,t)\|^2_{L^2(\Omega)}\, \mathrm{dt} + \int_0^T \|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)}\, \mathrm{dt}. \tag{4.3.2}$$

Thanks to Proposition 4.1, $\mathcal{V}_0$ is a Hilbert space and the $\|\cdot\|_{\mathcal{V}_0}$-norm is equivalent to

$$|||v|||^2 := \|v\|^2_{H^2(\Omega)\otimes L^2(0,T)} + \|v\|^2_{L^2(\Omega)\otimes H^1(0,T)}. \tag{4.3.3}$$

Our model problem is the heat equation, with initial and homogeneous boundary conditions: we seek for a solution u such that

$$\begin{cases} \partial_t u - \Delta u \;=\; f & \text{in} \quad \Omega \times (0,T), \\ u \;=\; 0 & \text{on} \quad \partial\Omega \times (0,T), \\ u \;=\; 0 & \text{in} \quad \Omega \times \{0\}. \end{cases} \tag{4.3.4}$$

with $f \in L^2(\Omega \times (0,T))$. Before proving the theorem assessing the regularity of the solution u of (4.3.4), we need the following lemma.

Lemma 4.1. *Let Assumptions 4.3–4.4 hold and let $r \in L^2(\Omega)$. Then, there exists a unique weak solution $z \in H^2(\Omega)$ to the Poisson problem*

$$\begin{cases} -\Delta z \;=\; r & \text{in} \quad \Omega, \\ \quad z \;=\; 0 & \text{on} \quad \partial\Omega. \end{cases} \tag{4.3.5}$$

Moreover, there exists a constant C depending only on $\boldsymbol{F}$ such that

$$\|z\|_{H^2(\Omega)} \leq C\|r\|_{L^2(\Omega)}. \tag{4.3.6}$$

Proof. We recall that z is a weak solution of (4.3.5) if $z \in H^1_0(\Omega)$ and if $\int_\Omega \nabla z \cdot \nabla q \,\mathrm{d}\Omega = \int_\Omega rq \,\mathrm{d}\Omega \;\forall q \in H^1_0(\Omega)$. Then, we have that $z \in H^1_0(\Omega)$ is a weak solution of (4.3.5) if and only if $w := z \circ \boldsymbol{F} \in H^1_0(\widehat{\Omega})$ is a weak solution of

$$\begin{cases} -\nabla \cdot (\boldsymbol{R}\,\nabla w) \;=\; g & \text{in} \quad \widehat{\Omega}, \\ \qquad w \;=\; 0 & \text{on} \quad \partial\widehat{\Omega}, \end{cases} \tag{4.3.7}$$

where $g := |\det(J_{\boldsymbol{F}})| r \circ \boldsymbol{F}$ and $\boldsymbol{R} := J_{\boldsymbol{F}}^{-1} J_{\boldsymbol{F}}^{-T} |\det(J_{\boldsymbol{F}})|$. Thanks to Assumptions 4.3–4.4, we have that $\boldsymbol{F} : \widehat{\Omega} \to \Omega$ fulfils $\boldsymbol{F} \in C^{1,1}$ on the closure of $\widehat{\Omega}$ and $\boldsymbol{F}^{-1} \in C^{1,1}(\overline{\Omega})$. Therefore, we have that the entries of the matrix $\boldsymbol{R}$ are Lipschitz continuous and we can apply [59, Theorem 3.2.1.2], reported in Theorem 4.5 in Section 4.8, to conclude that there exists a unique solution $w \in H^2(\widehat{\Omega})$ of problem (4.3.7). Thanks to [73, Chapitre III, Lemme 11.1], reported in Lemma 4.8 in Section 4.8, we also have

$$\|w\|^2_{H^2(\widehat{\Omega})} \leq c_1 \left(\|\nabla \cdot (\boldsymbol{R}\,\nabla w)\,\|^2_{L^2(\widehat{\Omega})} + \|w\|^2_{L^2(\widehat{\Omega})} \right) \leq c_2 \|\nabla \cdot (\boldsymbol{R}\,\nabla w)\|^2_{L^2(\widehat{\Omega})} = c_2 \|g\|^2_{L^2(\widehat{\Omega})},$$

where c_1 and c_2 are constants depending only on $\boldsymbol{R}$, that is, on $\boldsymbol{F}$ and its inverse. Finally, we conclude

$$\|z\|_{H^2(\Omega)} \leq C_1 \|w\|_{H^2(\widehat{\Omega})} \leq C_2 \|g\|_{L^2(\widehat{\Omega})} \leq C\|r\|_{L^2(\Omega)},$$

where the constants C_1, C_2 and C depend only on $\boldsymbol{F}$. □

Theorem 4.1. *Let $f \in L^2(\Omega \times (0,T))$ and let Assumptions 4.1-4.4 hold. Then there exists a unique weak solution (as defined in [49, Chapter 7], see also Definition 4.2 in Section 4.8) $u \in \left(H^2(\Omega) \otimes L^2(0,T)\right) \cap \left(L^2(\Omega) \otimes H^1(0,T)\right) \cap \left(H^1_0(\Omega) \otimes L^\infty(0,T)\right)$ of (4.3.4). We also have*

$$\|u\|_{H^2(\Omega)\otimes L^2(0,T)} + \|u\|_{L^2(\Omega)\otimes H^1(0,T)} + \|u\|_{H^1_0(\Omega)\otimes L^\infty(0,T)} \leq C\|f\|_{L^2(\Omega\times(0,T))},$$

where C is a constant depending only on $\boldsymbol{F}$.

Proof. Following the same arguments of step 1 and step 2 of the proof of [49, Chapter 7, Theorem 5] (see Theorem 4.6 in Section 4.8), we conclude that $u \in \left(H^1_0(\Omega) \otimes L^\infty(0,T)\right) \cap$

$\big(L^2(\Omega) \otimes H^1(0,T)\big)$ and that

$$\|u\|_{L^2(\Omega)\otimes H^1(0,T)} + \|u\|_{H^1_0(\Omega)\otimes L^\infty(0,T)} \leq D_1 \|f\|_{L^2(\Omega\times(0,T))}, \tag{4.3.8}$$

where D_1 is a constant depending only on $\boldsymbol{F}$.

We write for a.e. $t \in [0,T]$

$$\int_\Omega \nabla u(\boldsymbol{x},t)\cdot\nabla v(\boldsymbol{x})\,\mathrm{d}\Omega = \int_\Omega r(\boldsymbol{x},t)\; v(\boldsymbol{x})\,\mathrm{d}\Omega \quad \forall v \in H^1_0(\Omega),$$

where $r := f - \partial_t u \in L^2(\Omega\times(0,T))$ and in particular $r(\cdot,t) \in L^2(\Omega)$ for a.e. $t \in [0,T]$. Therefore, thanks to Lemma 4.1, we conclude that $u(\cdot,t) \in H^2(\Omega)$ for a.e. $t \in [0,T]$ and thus $u \in H^2(\Omega)\otimes L^2(0,T)$: indeed, integrating in time, (4.3.6) and (4.3.8) yield to the following estimate

$$\|u\|^2_{H^2(\Omega)\otimes L^2(0,T)} \leq C^2\|r\|^2_{L^2(\Omega\times(0,T))} \leq C^2(\|f\|^2_{L^2(\Omega\times(0,T))}+\|u\|^2_{L^2(\Omega)\otimes H^1(0,T)}) \leq D_2^2\|f\|^2_{L^2(\Omega\times(0,T))},$$

where $D_2^2 := C^2 + D_1^2$. This concludes the proof. □

More generally, non-homogeneous initial and boundary conditions are allowed. For example, if $u = u_0$ in $\Omega\times\{0\}$, with $u_0 \in H^1_0(\Omega)$, we lift[1] u_0 to $\widetilde{u}_0 \in (H^1_0(\Omega)\cap H^2(\Omega))\otimes L^2(0,T) \cap L^2(\Omega)\otimes H^1(0,T)$. Then $\widetilde{u} = u - \widetilde{u}_0 \in \mathcal{V}_0$ is the solution of

$$\begin{cases} \partial_t\widetilde{u} - \Delta\widetilde{u} \;=\; \widetilde{f} & \text{in} \quad \Omega\times(0,T), \\ \widetilde{u} \;=\; 0 & \text{on} \;\; \partial\Omega\times(0,T), \\ \widetilde{u} \;=\; 0 & \text{in} \quad \Omega\times\{0\}, \end{cases} \tag{4.3.9}$$

where $\widetilde{f} := f - \partial_t\widetilde{u}_0 + \Delta\widetilde{u}_0$. For a detailed description of the variational formulation of problems (4.3.4)–(4.3.9) and their well-posedness see, for example, [49, 97].

4.3.2 Space-time least-squares variational formulation

We consider the following variational formulation for the system (4.3.4): find $u \in \mathcal{V}_0$ such that

$$u = \arg\min_{v\in\mathcal{V}_0} \tfrac{1}{2}\,\|\partial_t v - \Delta v - f\|^2_{L^2(\Omega\times(0,T))}\,. \tag{4.3.10}$$

Its Euler-Lagrange equation is

$$\mathcal{A}(u,v) = \mathcal{F}(v) \quad \forall v \in \mathcal{V}_0, \tag{4.3.11}$$

where the bilinear form $\mathcal{A}(\cdot,\cdot)$ and the linear form $\mathcal{F}(\cdot)$ are defined as

$$\mathcal{A}(v,w) := \int_0^T\int_\Omega (\partial_t v\,\partial_t w + \Delta v\,\Delta w - \partial_t v\,\Delta w - \Delta v\,\partial_t w)\;\mathrm{d}\Omega\,\mathrm{dt}, \tag{4.3.12}$$

$$\mathcal{F}(w) := \int_0^T\int_\Omega f\,(\partial_t w - \Delta w)\,\mathrm{d}\Omega\,\mathrm{dt}.$$

For an equivalent way of writing the minimization problem (4.3.10), we refer to Section 4.7.2. The variational formulation (4.3.11) is well-posed, thanks to the following Lemmas 4.2–4.4 and Proposition 4.2.

[1] We can use the same argument as in Theorem 4.1 that is, the proof of [49, Chapter 7, Theorem 5] (see also Theorem 4.6 in Section 4.8), where step 3 therein uses the elliptic regularity property which is given, in our case, by Lemma 4.1.

Lemma 4.2. *The bilinear form $\mathcal{A}(\cdot,\cdot)$ is continuous in $\mathcal{V}_0$. Particularly, it holds*

$$|\mathcal{A}(v,w)| \leq 2\|v\|_{\mathcal{V}_0}\|w\|_{\mathcal{V}_0} \quad \forall v,w \in \mathcal{V}_0.$$

Proof. Given $v,w \in \mathcal{V}_0$, by Cauchy-Schwarz inequality

$$\begin{aligned}
|\mathcal{A}(v,w)| &\leq \|v\|_{\mathcal{V}_0}\|w\|_{\mathcal{V}_0} + \int_0^T\int_\Omega |\partial_t v\,\Delta w|\,\mathrm{d}\Omega\,\mathrm{dt} + \int_0^T\int_\Omega |\Delta v\,\partial_t w|\,\mathrm{d}\Omega\,\mathrm{dt} \\
&\leq \|v\|_{\mathcal{V}_0}\|w\|_{\mathcal{V}_0} + \left[\int_0^T \left(\|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)} + \|\Delta v(\cdot,t)\|^2_{L^2(\Omega)}\right)\mathrm{dt}\right]^{1/2} \\
&\quad * \left[\int_0^T \left(\|\partial_t w(\cdot,t)\|^2_{L^2(\Omega)} + \|\Delta w(\cdot,t)\|^2_{L^2(\Omega)}\right)\mathrm{dt}\right]^{1/2} \\
&\leq 2\|v\|_{\mathcal{V}_0}\|w\|_{\mathcal{V}_0},
\end{aligned}$$

which concludes the proof. □

Lemma 4.3. *The bilinear form $\mathcal{A}(\cdot,\cdot)$ is $\mathcal{V}_0$-elliptic. In particular, it holds*

$$\mathcal{A}(v,v) \geq \|v\|^2_{\mathcal{V}_0} \quad \forall v \in \mathcal{V}_0.$$

Proof. Let $v \in \mathcal{V}_0$. Thanks to [23, Lemme 3.3] (see also Lemma 4.9 of Section 4.8), we can write

$$-2\int_0^T\int_\Omega \partial_t v\,\Delta v\,\mathrm{d}\Omega\,\mathrm{dt} = \int_\Omega |\nabla v(\boldsymbol{x},T)|^2\,\mathrm{d}\Omega - \int_\Omega |\nabla v(\boldsymbol{x},0)|^2\,\mathrm{d}\Omega,$$

where $\nabla := [\partial_{x_1},\ldots,\partial_{x_d}]^T$ denotes the gradient w.r.t. spatial variables $x_1,\ldots,x_d$. In particular, as $\nabla v(\boldsymbol{x},0) = 0$, we have that

$$\mathcal{A}(v,v) = \int_0^T \|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt} + \int_0^T \|\Delta v(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt} + \int_\Omega |\nabla v(\boldsymbol{x},T)|^2\,\mathrm{d}\Omega \geq \|v\|^2_{\mathcal{V}_0} \quad \forall v \in \mathcal{V}_0,$$

which concludes the proof. □

Lemma 4.4. *The linear form $\mathcal{F}(\cdot)$ is continuous in $\mathcal{V}_0$. In particular it holds*

$$\mathcal{F}(v) \leq \sqrt{2}\|f\|_{L^2(\Omega\times(0,T))}\|v\|_{\mathcal{V}_0} \quad \forall v \in \mathcal{V}_0.$$

Proof. Given $v \in \mathcal{V}_0$, by Cauchy-Schwarz inequality we get

$$\begin{aligned}
|\mathcal{F}(v)| &\leq \|f\|_{L^2(\Omega\times(0,T))}\left(\int_0^T \|\partial_t v(\cdot,t) - \Delta v(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt}\right)^{1/2} \\
&\leq \sqrt{2}\|f\|_{L^2(\Omega\times(0,T))}\left(\int_0^T \|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt} + \int_0^T \|\Delta v(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt}\right)^{1/2} \\
&= \sqrt{2}\|f\|_{L^2(\Omega\times(0,T))}\|v\|_{\mathcal{V}_0},
\end{aligned}$$

which concludes the proof. □

Proposition 4.2. *Under Assumptions 4.2–4.4, the minimization problem* (4.3.10) *and the variational problem* (4.3.11) *are equivalent and they admit a unique solution $u \in \mathcal{V}_0$.*

Proof. The proof follows using Lemmas 4.2–4.4 and the Lax-Milgram theorem. □

4.3.3 Space-time least-squares approximation

Thanks to Assumption 4.2, we have

$$\mathcal{V}_{h,0} \subset (H^1_0(\Omega) \cap H^2(\Omega)) \otimes H^1(0,T) \subset \mathcal{V}_0. \tag{4.3.13}$$

Therefore, we consider a Galerkin method for (4.3.11), that is, the least-squares approximation of the system (4.3.4): find $u_h \in \mathcal{V}_{h,0}$ such that

$$u_h = \arg\min_{v_h \in \mathcal{V}_{h,0}} \tfrac{1}{2} \left\| \partial_t v_h - \Delta v_h - f \right\|^2_{L^2(\Omega \times (0,T))}. \tag{4.3.14}$$

Its Euler-Lagrange equation is

$$\mathcal{A}(u_h, v_h) = \mathcal{F}(v_h) \quad \forall v_h \in \mathcal{V}_{h,0}. \tag{4.3.15}$$

Well-posedness and quasi-optimality follow from standard arguments.

Proposition 4.3. *The minimization problem* (4.3.14) *and the variational problem* (4.3.15) *are equivalent and they admit a unique solution* $u_h \in \mathcal{V}_{h,0}$. *It also holds:*

$$\|u - u_h\|_{\mathcal{V}_0} \leq \sqrt{2} \inf_{v_h \in \mathcal{V}_{h,0}} \|u - v_h\|_{\mathcal{V}_0}. \tag{4.3.16}$$

Proof. The proof of the equivalence and of the existence and uniqueness of a solution follow by using Lemmas 4.2–4.4 and the Lax-Milgram theorem, while the proof of (4.3.16) is a consequence of the Céa Lemma and the symmetry of the bilinear form $\mathcal{A}$. □

The following result states the convergence of our method.

Theorem 4.2. *Under Assumptions 4.2–4.4, we have* $\lim_{h\to 0} \|u - u_h\|_{\mathcal{V}_0} = 0$.

Proof. To prove the theorem, we show that

$$\lim_{h \to 0} \inf_{v_h \in \mathcal{V}_{h,0}} \|u - v_h\|_{\mathcal{V}_0} = 0, \tag{4.3.17}$$

and then use (4.3.16).

Given $u \in \mathcal{V}_0$, let $\widehat{u} = u \circ \boldsymbol{G}^{-1}$ be its pullback. Since $\boldsymbol{G}$ and $\boldsymbol{G}^{-1}$ are both of class $W^{2,\infty}$ and since the $\mathcal{V}_0$-norm (4.3.2) is equivalent to the $|||\cdot|||$-norm (4.3.3), the pullback is an isomorphism between $\mathcal{V}_0$ and

$$\widehat{\mathcal{V}}_0 = \left\{ v \in \left[\left(H^2(\widehat{\Omega}) \cap H^1_0(\widehat{\Omega}) \right) \otimes L^2(0,1) \right] \cap \left[L^2(\widehat{\Omega}) \otimes H^1(0,1) \right] \;\middle|\; v = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\},$$

endowed with the norm

$$\|v\|^2_{\widehat{\mathcal{V}}_0} := \int_0^1 \|\Delta v(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})} \,\mathrm{d}\tau + \int_0^1 \|\partial_\tau v(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})} \,\mathrm{d}\tau.$$

Then, by using Lemma 4.7 reported in Section 4.7.1, we can approximate, as close as we want, $\widehat{u} \in \widehat{\mathcal{V}}_0$ by a smooth function fulfilling the same boundary conditions of $\widehat{u}$, and then by a spline in $\widehat{\mathcal{V}}_{h,0}$ (see (4.2.2)), on a fine enough mesh. This implies (4.3.17). □

4.3.4 A priori error analysis

We investigate in this section the approximation properties of the isogeometric space $\mathcal{V}_{h,0}$ under h-refinement.

Proposition 4.4. *Let q_s and q_t be two integers such that $2 \leq q_s \leq p_s + 1$ and $1 \leq q_t \leq p_t + 1$. Under Assumption 4.1, there exists a projection $\Pi_h : \mathcal{V}_0 \cap \big(H^{q_s}(\Omega) \otimes H^1(0,T)\big) \cap \big(H^2(\Omega) \otimes H^{q_t}(0,T)\big) \to \mathcal{V}_{h,0}$ such that*

$$\|v - \Pi_h v\|_{\mathcal{V}_0} \leq C \left(h_s^{q_s-2} \|v\|_{H^{q_s}(\Omega)\otimes H^1(0,T)} + h_t^{q_t-1} \|v\|_{H^2(\Omega)\otimes H^{q_t}(0,T)} \right) \tag{4.3.18}$$

where the constant C depends on p_s, p_t, α and the parametrization $\boldsymbol{G}$.

Proof. The result follows from the anisotropic approximation estimates that are developed in [11]. We remark that [11] states its error analysis for 2 dimensions, but the results therein straightforwardly generalize to higher dimension. We give an overview of the proof, for the sake of completeness.

As space and time coordinates in $\Omega \times [0,T]$ are orthogonal, the parametric coordinate (tangent) vectors are

$$\boldsymbol{g}_i(\boldsymbol{x}) := \partial_{\eta_i} \boldsymbol{G} \circ \boldsymbol{G}^{-1}(\boldsymbol{x},t) = \begin{bmatrix} \partial_{\eta_i} \boldsymbol{F} \circ \boldsymbol{F}^{-1}(\boldsymbol{x}) \\ 0 \end{bmatrix} \in \mathbb{R}^d \times \{0\} \subset \mathbb{R}^{d+1} \quad \text{for } i = 1, \dots, d,$$

$$\boldsymbol{g}_t(t) := \partial_\tau \boldsymbol{G} \circ \boldsymbol{G}^{-1}(\boldsymbol{x},t) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ T \end{bmatrix} \in \mathbb{R}^{d+1}.$$

Then, given $v \in \mathcal{V}_0$, the directional derivatives w.r.t. $\boldsymbol{g}_i$ and $\boldsymbol{g}_t$ that are used in [11, Section 5], become

$$\begin{bmatrix} \frac{\partial v(\boldsymbol{x},t)}{\partial \boldsymbol{g}_1} \\ \vdots \\ \frac{\partial v(\boldsymbol{x},t)}{\partial \boldsymbol{g}_d} \end{bmatrix} = \left(J_{\boldsymbol{F}} \circ \boldsymbol{F}^{-1}(\boldsymbol{x}) \right)^T \nabla_{\boldsymbol{x}} v(\boldsymbol{x},t), \qquad \frac{\partial v}{\partial \boldsymbol{g}_t}(\boldsymbol{x},t) = T\, \partial_t v(\boldsymbol{x},t).$$

Higher-order directional derivatives can be defined similarly, as in [11, Section 5]. We also have that

$$\left\| \frac{\partial}{\partial \boldsymbol{g}_{i_1}} \left(\cdots \frac{\partial v}{\partial \boldsymbol{g}_{i_k}} \right) \right\|_{L^2(\Omega\times(0,T))} \leq C \|v\|_{H^k(\Omega)\otimes L^2(0,T)}, \tag{4.3.19a}$$

$$\left\| \frac{\partial^k v}{\partial \boldsymbol{g}_t^k} \right\|_{L^2(\Omega\times(0,T))} \leq C \|v\|_{L^2(\Omega)\otimes H^k(0,T)}, \tag{4.3.19b}$$

for a suitable constant C, $k \geq 0$ and $i_j \in \{1, \dots, d\}$, $j = 1, \dots, k$. Therefore, [11, Theorem 5.1] generalized to $d+1$ dimensions gives the existence of a projection Π_h on the space $\mathcal{V}_{h,0}$ such that

$$\|v - \Pi_h v\|_{H^2(\Omega)\otimes L^2(0,T)} \leq C \left(h_s^{q_s-2} \|v\|_{H^{q_s}(\Omega)\otimes L^2(0,T)} + h_t^{q_t-1} \|v\|_{H^2(\Omega)\otimes H^{q_t-1}(0,T)} \right),$$

$$\|v - \Pi_h v\|_{L^2(\Omega)\otimes H^1(0,T)} \leq C \left(h_s^{q_s-2} \|v\|_{H^{q_s-2}(\Omega)\otimes H^1(0,T)} + h_t^{q_t-1} \|v\|_{L^2(\Omega)\otimes H^{q_t}(0,T)} \right),$$

with C depending only on p_s, p_t, α and the space parametrization $\boldsymbol{G}$. Squaring and summing the two inequalities above, using (4.3.19) and that $\int_0^T \|\Delta(v - \Pi_h v)(\cdot,t)\|^2_{L^2(\Omega)}\, \mathrm{dt} \leq$

$\|v-\Pi_h v\|^2_{H^2(\Omega)\otimes L^2(0,T)}$, leads to

$$\begin{aligned}\|v-\Pi_h v\|_{\mathcal{V}_0} \leq & C h_s^{q_s-2}\left(\|v\|_{H^{q_s}(\Omega)\otimes L^2(0,T)}+\|v\|_{H^{q_s-2}(\Omega)\otimes H^1(0,T)}\right)\\ &+C h_t^{q_t-1}\left(\|v\|_{H^2(\Omega)\otimes H^{q_t-1}(0,T)}+\|v\|_{L^2(\Omega)\otimes H^{q_t}(0,T)}\right),\end{aligned}$$

and finally (4.3.18) by the obvious upperbound of the right-hand-side norms. □

As a direct corollary of Proposition 4.3 and 4.4, we can now state the a-priori error estimate for the least-squares method.

Theorem 4.3. *Let q_s and q_t be two integers such that $q_s \geq 2$ and $q_t \geq 1$. If $u \in \mathcal{V}_0 \cap \left(H^{q_s}(\Omega)\otimes H^1(0,T)\right) \cap \left(H^2(\Omega)\otimes H^{q_t}(0,T)\right)$ is the solution of* (4.3.4) *and $u_h \in \mathcal{V}_{h,0}$ is the solution of* (4.3.15)*, then*

$$\|u-u_h\|_{\mathcal{V}_0} \leq C\left(h_s^{k_s-2}\|u\|_{H^{k_s}(\Omega)\otimes H^1(0,T)}+h_t^{k_t-1}\|u\|_{H^2(\Omega)\otimes H^{k_t}(0,T)}\right) \tag{4.3.20}$$

where $k_s := \min\{q_s, p_s+1\}$, $k_t := \min\{q_t, p_t+1\}$, C is a constant that depends only on p_s, p_t, α and the parametrization $\boldsymbol{G}$.

4.3.5 Discrete system

Before introducing the discrete system, we rewrite the bilinear form $\mathcal{A}(\cdot,\cdot)$ in an equivalent way, through the following Lemma.

Lemma 4.5. *The bilinear form $\mathcal{A}(\cdot,\cdot)$ can be written as*

$$\mathcal{A}(v_h,w_h)=\int_0^T\int_\Omega \partial_t v_h\,\partial_t w_h\,\mathrm{d}\Omega\,\mathrm{d}t+\int_0^T\int_\Omega \Delta v_h\,\Delta w_h\,\mathrm{d}\Omega\,\mathrm{d}t+\int_\Omega \nabla v_h(\boldsymbol{x},T)\cdot\nabla w_h(\boldsymbol{x},T)\,\mathrm{d}\Omega \tag{4.3.21}$$

for all $v_h, w_h \in \mathcal{V}_{h,0}$.

Proof. Let $v_h, w_h \in \mathcal{V}_{h,0}$. First note that $\partial_t v_h, \partial_t w_h \in \left(H^1_0(\Omega)\cap H^2(\Omega)\right)\otimes L^2(0,T)$, from (4.3.13), and $\partial_t v_h = \partial_t w_h = 0$ on $\partial\Omega\times[0,T]$. Using Green formula and integrating by parts yields to

$$\begin{aligned}-\int_0^T\int_\Omega (\partial_t v_h\,\Delta w_h+\partial_t w_h\Delta v_h)\,\mathrm{d}\Omega\,\mathrm{d}t \;=\; & -\int_0^T\int_{\partial\Omega}(\partial_t v_h\nabla w_h\cdot\boldsymbol{\nu}+\partial_t w_h\nabla v_h\cdot\boldsymbol{\nu})\,\mathrm{d}\Omega\,\mathrm{d}t\\ &+\int_0^T\int_\Omega[\nabla(\partial_t v_h)\cdot\nabla w_h+\nabla(\partial_t w_h)\cdot\nabla v_h]\,\mathrm{d}\Omega\,\mathrm{d}t\\ =\; & \int_0^T\left[\partial_t\left(\int_\Omega \nabla v_h\cdot\nabla w_h\,\mathrm{d}\Omega\right)\right]\mathrm{d}t\\ =\; & \int_\Omega[\nabla v_h(\boldsymbol{x},T)\cdot\nabla w_h(\boldsymbol{x},T)-\nabla v_h(\boldsymbol{x},0)\cdot\nabla w_h(\boldsymbol{x},0)]\,\mathrm{d}\Omega\\ =\; & \int_\Omega \nabla v_h(\boldsymbol{x},T)\cdot\nabla w_h(\boldsymbol{x},T)\,\mathrm{d}\Omega,\end{aligned}$$

where $\boldsymbol{\nu}\in\mathbb{R}^d$ is the external normal unit vector to $\partial\Omega$. Then (4.3.21) follows. □

Remark 4.1. *Note that the identity* (4.3.21) *holds also in the continuous setting (see Section 4.7.2).*

After the introduction of the basis (4.2.3) for $\mathcal{V}_{h,0}$, the linear system associated to (4.3.15) is

$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

where $[\mathbf{A}]_{i,j} := \mathcal{A}(B_{i,\boldsymbol{p}}, B_{j,\boldsymbol{p}})$ and $[\mathbf{b}]_i := \mathcal{F}(B_{i,\boldsymbol{p}})$. The discrete system matrix $\mathbf{A}$ can be written as the sum of Kronecker product matrices (see (4.3.21))

$$\mathbf{A} = K_t \otimes M_s + M_t \otimes L_s + S_t \otimes K_s, \tag{4.3.22}$$

where the time matrices are for $i, j = 1, \dots, n_t$

$$[K_t]_{i,j} := \int_0^T b'_{i,p_t}(t)\, b'_{j,p_t}(t)\, \mathrm{dt}, \quad [M_t]_{i,j} := \int_0^T b_{i,p_t}(t)\, b_{j,p_t}(t)\, \mathrm{dt}, \quad [S_t]_{i,j} := b_{i,p_t}(T)\, b_{j,p_t}(T),$$

and the spatial matrices are for $i, j = 1, \dots, N_s$

$$[L_s]_{i,j} := \int_\Omega \Delta B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\, \Delta B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\, \mathrm{d}\Omega, \qquad [M_s]_{i,j} := \int_\Omega B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\, B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\, \mathrm{d}\Omega,$$

$$[K_s]_{i,j} := \int_\Omega \nabla B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\, \nabla B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\, \mathrm{d}\Omega.$$

4.4 Preconditioner definition and application

In this section we analyze solving strategies for the least-squares method (4.3.15) and we present a suitable preconditioner. Thanks to the least-squares formulation of the heat equation, the matrix $\mathbf{A}$ in (4.3.22) is symmetric and positive definite. Thus, we can design and analyze a suitable symmetric positive definite preconditioner to be used for a preconditioned Conjugate Gradient method.

The simpler version of our preconditioner is associated with the bilinear form $\widehat{\mathcal{P}} : \widehat{\mathcal{V}}_{h,0} \times \widehat{\mathcal{V}}_{h,0} \to \mathbb{R}$ defined as

$$\widehat{\mathcal{P}}(w_h, v_h) := \int_0^1 \int_{\widehat{\Omega}} \partial_\tau w_h\, \partial_\tau v_h \, \mathrm{d}\widehat{\Omega}\, \mathrm{d}\tau + \sum_{k=1}^d \int_0^1 \int_{\widehat{\Omega}} \frac{\partial^2 w_h}{\partial \eta_k^2} \frac{\partial^2 v_h}{\partial \eta_k^2} \, \mathrm{d}\widehat{\Omega}\, \mathrm{d}\tau \tag{4.4.1}$$

and with the corresponding norm

$$\|v_h\|^2_{\widehat{\mathcal{P}}} := \widehat{\mathcal{P}}(v_h, v_h). \tag{4.4.2}$$

The preconditioner matrix is given by

$$[\mathbf{P}]_{i,j} = \widehat{\mathcal{P}}(\widehat{B}_{i,\boldsymbol{p}}(\boldsymbol{\eta}, \tau), \widehat{B}_{j,\boldsymbol{p}}(\boldsymbol{\eta}, \tau)) \qquad i, j = 1, \dots, N_{dof}$$

and has the following structure:

$$\mathbf{P} = \widehat{K}_t \otimes \widehat{M}_s + \widehat{M}_t \otimes \widetilde{L}_s, \tag{4.4.3}$$

where, referring to (4.2.1) for the notation of the basis functions, we have defined for $i, j = 1, \dots, n_t$

$$[\widehat{K}_t]_{i,j} := \int_0^1 \widehat{b}'_{i,p_t}(\tau)\, \widehat{b}'_{j,p_t}(\tau)\, \mathrm{d}\tau, \qquad [\widehat{M}_t]_{i,j} := \int_0^1 \widehat{b}_{i,p_t}(\tau)\, \widehat{b}_{j,p_t}(\tau)\, \mathrm{d}\tau,$$

and for $i, j = 1, \dots, N_s$

$$[\widetilde{L}_s]_{i,j} := \sum_{k=1}^{d} \int_{\widehat{\Omega}} \frac{\partial^2 \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{\eta})}{\partial \eta_k^2} \frac{\partial^2 \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{\eta})}{\partial \eta_k^2} \, \mathrm{d}\widehat{\Omega}, \qquad [\widehat{M}_s]_{i,j} := \int_{\widehat{\Omega}} \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{\eta}) \; \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{\eta}) \, \mathrm{d}\widehat{\Omega}.$$

Note that $\widehat{K}_t$, $\widehat{M}_t$ and $\widehat{M}_s$ correspond to K_t, M_t and M_s, respectively, where the integration is performed on the parametric domain $\widehat{\Omega}$. The matrices $\widetilde{L}_s$ and $\widehat{M}_s$ can be further factorized as sum of Kronecker products as

$$\widetilde{L}_s = \sum_{k=1}^{d} \widehat{M}_d \otimes \cdots \otimes \widehat{M}_{k+1} \otimes \widehat{L}_k \otimes \widehat{M}_{k-1} \otimes \cdots \otimes \widehat{M}_1, \qquad \widehat{M}_s = \widehat{M}_d \otimes \cdots \otimes \widehat{M}_1,$$

where for $k = 1, \dots, d$ and for $i, j = 1, \dots, n_{s,k}$

$$[\widehat{L}_k]_{i,j} := \int_0^1 \widehat{b}''_{i,p_s}(\eta_k) \; \widehat{b}''_{j,p_s}(\eta_k) \, \mathrm{d}\eta_k, \quad [\widehat{M}_k]_{i,j} := \int_0^1 \widehat{b}_{i,p_s}(\eta_k) \; \widehat{b}_{j,p_s}(\eta_k) \, \mathrm{d}\eta_k.$$

If $d = 3$, that is the case addressed in the numerical tests, we have that (4.4.3) becomes

$$\mathbf{P} = \widehat{K}_t \otimes \widehat{M}_3 \otimes \widehat{M}_2 \otimes \widehat{M}_1 + \widehat{M}_t \otimes \widehat{L}_3 \otimes \widehat{M}_2 \otimes \widehat{M}_1 + \widehat{M}_t \otimes \widehat{M}_3 \otimes \widehat{L}_2 \otimes \widehat{M}_1 + \widehat{M}_t \otimes \widehat{M}_3 \otimes \widehat{M}_2 \otimes \widehat{L}_1.$$

4.4.1 Spectral properties

We now focus on the spectral analysis of $\mathbf{P}^{-1}\mathbf{A}$. We need to define the bilinear form $\mathcal{P} : \mathcal{V}_{h,0} \times \mathcal{V}_{h,0} \to \mathbb{R}$

$$\mathcal{P}(w_h, v_h) := \int_0^T \int_\Omega \partial_t w_h \, \partial_t v_h \, \mathrm{d}\Omega \, \mathrm{dt} + \sum_{k=1}^{d} \int_0^T \int_\Omega \frac{\partial^2 w_h}{\partial x_k^2} \frac{\partial^2 v_h}{\partial x_k^2} \, \mathrm{d}\Omega \, \mathrm{dt}$$

and the associated norm

$$\|v_h\|_{\mathcal{P}}^2 := \mathcal{P}(v_h, v_h).$$

Note that $\mathcal{P}(\cdot,\cdot)$ and $\|\cdot\|_{\mathcal{P}}$ are analogous to $\widehat{\mathcal{P}}(\cdot,\cdot)$ and $\|\cdot\|_{\widehat{\mathcal{P}}}$ but integration is performed on the physical domain (see (4.4.1) and (4.4.2)).

We first prove the equivalence between the norms $\|\cdot\|_{\mathcal{P}}$ and $\|\cdot\|_{\mathcal{V}_0}$ in $\mathcal{V}_{h,0}$.

Proposition 4.5. *Under Assumptions 4.2–4.3, it holds*

$$\frac{1}{C_\Delta} \|v_h\|_{\mathcal{P}}^2 \leq \|v_h\|_{\mathcal{V}_0}^2 \leq d \|v_h\|_{\mathcal{P}}^2 \quad \forall v_h \in \mathcal{V}_{h,0},$$

where C_Δ is the constant defined in (4.3.1).

Proof. Given $v_h \in \mathcal{V}_{h,0}$, recalling (4.3.13) and thanks to (4.3.1), we have that

$$\begin{aligned} \sum_{k=1}^{d} \int_0^T \int_\Omega \left| \frac{\partial^2 v_h}{\partial x_k^2} \right|^2 \mathrm{d}\Omega \, \mathrm{dt} &\leq \int_0^T \int_\Omega \left(\sum_{k,l=1}^{d} \left| \frac{\partial^2 v_h}{\partial x_k \partial x_l} \right|^2 \right) \mathrm{d}\Omega \, \mathrm{dt} = \int_0^T |v_h(\cdot,t)|_{H^2(\Omega)}^2 \, \mathrm{dt} \\ &\leq \int_0^T \|v_h(\cdot,t)\|_{H^2(\Omega)}^2 \, \mathrm{dt} \leq C_\Delta \int_0^T \|\Delta v_h(\cdot,t)\|_{L^2(\Omega)}^2 \, \mathrm{dt}. \end{aligned}$$

Thus, the first inequality holds. We also have

$$\begin{aligned}\int_0^T \|\Delta v_h(\cdot,t)\|^2_{L^2(\Omega)}\,\mathrm{dt} &= \sum_{k,l=1}^{d}\int_0^T\int_\Omega \frac{\partial^2 v_h}{\partial x_k^2}\frac{\partial^2 v_h}{\partial x_l^2}\,\mathrm{d}\Omega\,\mathrm{dt}\\ &\leq \frac{1}{2}\sum_{k,l=1}^{d}\int_0^T\left[\left\|\frac{\partial^2 v_h}{\partial x_k^2}(\cdot,t)\right\|^2_{L^2(\Omega)}+\left\|\frac{\partial^2 v_h}{\partial x_l^2}(\cdot,t)\right\|^2_{L^2(\Omega)}\right]\mathrm{dt}\\ &\leq d\sum_{k=1}^{d}\int_0^T\left\|\frac{\partial^2 v_h}{\partial x_k^2}(\cdot,t)\right\|^2_{L^2(\Omega)}\mathrm{dt} = d\sum_{k=1}^{d}\int_0^T\int_\Omega\left|\frac{\partial^2 v_h}{\partial x_k^2}\right|^2\mathrm{d}\Omega\,\mathrm{dt}\end{aligned}$$

and we can conclude that the second inequality holds. □

Corollary 4.1. *Under Assumptions 4.2–4.3, it holds*

$$\frac{1}{C_\Delta}\|v_h\|^2_{\mathcal{P}} \leq \mathcal{A}(v_h,v_h) \leq 2d\|v_h\|^2_{\mathcal{P}} \qquad \forall v_h \in \mathcal{V}_{h,0}. \tag{4.4.4}$$

Proof. The statement follows from Lemma 4.2, Lemma 4.3 and Proposition 4.5. □

Proposition 4.6. *Under Assumptions 4.2–4.4, there exist constants $Q_1, Q_2 > 0$ independent of h_s, h_t, p_s, p_t, but dependent on $\boldsymbol{G}$ such that*

$$Q_1\|v_h\|^2_{\mathcal{P}} \leq \|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}} \leq Q_2\|v_h\|^2_{\mathcal{P}} \quad \forall \widehat{v}_h \ \in \widehat{\mathcal{V}}_{h,0} \text{ and } v_h := \widehat{v}_h\circ\boldsymbol{G}^{-1}.$$

Proof. Let $\widehat{v}_h \in \widehat{\mathcal{V}}_{h,0}$ and $v_h := \widehat{v}_h \circ \boldsymbol{G}^{-1} \in \mathcal{V}_{h,0}$. First we prove the first inequality. Observing that $\boldsymbol{G}^{-1}(\boldsymbol{x},t) = (\boldsymbol{F}^{-1}(\boldsymbol{x}), t/T)$, we get

$$\begin{aligned}\int_0^T\int_\Omega (\partial_t v_h)^2\ \mathrm{d}\Omega\ \mathrm{dt} &= \frac{1}{T}\int_0^1\int_{\widehat{\Omega}}(\partial_\tau\widehat{v}_h)^2\,|\mathrm{det}\,(J_{\boldsymbol{G}})|\ \mathrm{d}\widehat{\Omega}\ \mathrm{d}\tau\\ &\leq \frac{1}{T}\sup_{\widehat{\Omega}}\{|\mathrm{det}(J_{\boldsymbol{G}})|\}\int_0^1\|\partial_\tau\widehat{v}_h(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})}\ \mathrm{d}\tau\\ &\leq \frac{1}{T}\sup_{\widehat{\Omega}}\{|\mathrm{det}(J_{\boldsymbol{G}})|\}\,\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}.\end{aligned}$$

Let $\mathbb{H}_{\widehat{v}_h}$ be the Hessian of $\widehat{v}_h$ with respect to the spatial parametric variables $\eta_1,\dots,\eta_d$, i.e. $\mathbb{H}_{\widehat{v}_h} \in \mathbb{R}^{d\times d}$ with $[\mathbb{H}_{\widehat{v}_h}]_{i,j} = \frac{\partial^2\widehat{v}_h}{\partial\eta_i\partial\eta_j}$ for $i,j = 1,\dots,d$, and let $[J_{\boldsymbol{G}}^{-1}]_{\cdot,i} \in \mathbb{R}^d$ denote the *i-th* column of $J_{\boldsymbol{G}}^{-1}$. Then, for $i = 1,\dots,d$, it holds

$$\begin{aligned}\int_0^T\int_\Omega\left(\frac{\partial^2 v_h}{\partial x_i^2}\right)^2\ \mathrm{d}\Omega\ \mathrm{dt} &= \int_0^1\int_{\widehat{\Omega}}\left([J_{\boldsymbol{G}}^{-1}]^T_{\cdot,i}\mathbb{H}_{\widehat{v}_h}[J_{\boldsymbol{G}}^{-1}]_{\cdot,i} + \nabla\widehat{v}_h^T\frac{\partial[J_{\boldsymbol{G}}^{-1}]_{\cdot,i}}{\partial\eta_i}\right)^2 T|\,\mathrm{det}(J_{\boldsymbol{G}})|\ \mathrm{d}\widehat{\Omega}\ \mathrm{d}\tau\\ &\leq \int_0^1\int_{\widehat{\Omega}}\left(\widehat{C}_1\|\mathbb{H}_{\widehat{v}_h}\|^2_F + \widehat{C}_2\|\nabla\widehat{v}_h\|^2_2\right)\ \mathrm{d}\widehat{\Omega}\ \mathrm{d}\tau,\end{aligned}$$

where $\|\cdot\|_F$ and $\|\cdot\|_2$ denote the Frobenius norm and the two-norm of matrices (the norm induced by the Euclidean vector norm), respectively, $\widehat{C}_1 := 2T\max_i\sup_{\widehat{\Omega}}\left\{\left(\|[J_{\boldsymbol{G}}^{-1}]_{\cdot,i}\|_2\right)^4|\,\mathrm{det}(J_{\boldsymbol{G}})|\right\}$, $\widehat{C}_2 := 2T\max_i\sup_{\widehat{\Omega}}\left\{\left(\left\|\frac{\partial[J_{\boldsymbol{G}}^{-1}]_{\cdot,i}}{\partial\eta_i}\right\|_2\right)^2|\,\mathrm{det}(J_{\boldsymbol{G}})|\right\}$ and where we used that $\|\mathbb{H}_{\widehat{v}_h}\|_2 \leq \|\mathbb{H}_{\widehat{v}_h}\|_F$.

Following the proof of Proposition 4.5, we can prove that

$$\int_0^1 \|\Delta \widehat{v}_h(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})}\,\mathrm{d}\tau \le d\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}} \quad \forall \widehat{v}_h \in \widehat{\mathcal{V}}_{h,0}.$$

Thus it holds

$$\int_0^1 \int_{\widehat{\Omega}} \|\mathbb{H}_{\widehat{v}_h}\|^2_F \,\mathrm{d}\widehat{\Omega}\,\mathrm{d}\tau \le 2\int_0^1 |\widehat{v}_h(\cdot,\tau)|^2_{H^2(\widehat{\Omega})}\,\mathrm{d}\tau \le 2\widehat{C}_\Delta \int_0^1 \|\Delta\widehat{v}_h(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})}\,\mathrm{d}\tau \ \le 2d\widehat{C}_\Delta\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}$$

$$\int_0^1 \int_{\widehat{\Omega}} \|\nabla\widehat{v}_h\|^2_2 \,\mathrm{d}\widehat{\Omega}\,\mathrm{d}\tau \ = \int_0^1 |\widehat{v}_h(\cdot,\tau)|^2_{H^1(\widehat{\Omega})}\,\mathrm{d}\tau \le \widehat{C}_\Delta \int_0^1 \|\Delta\widehat{v}_h(\cdot,\tau)\|^2_{L^2(\widehat{\Omega})}\,\mathrm{d}\tau \le d\widehat{C}_\Delta\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}},$$

where $\widehat{C}_\Delta > 0$ is the constant such that $\|z\|^2_{H^2(\widehat{\Omega})} \le \widehat{C}_\Delta\|\Delta z\|^2_{L^2(\widehat{\Omega})}$, for $z \in H^1_0(\widehat{\Omega}) \cap H^2(\widehat{\Omega})$. Therefore, we have

$$\int_0^T \int_\Omega \left(\frac{\partial^2 v_h}{\partial x_i^2}\right)^2 \mathrm{d}\Omega\ \mathrm{dt} \le d\widehat{C}_\Delta \left(2\widehat{C}_1 + \widehat{C}_2\right)\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}$$

and, summing all terms that define $\|\cdot\|_{\mathcal{P}}$, we conclude

$$Q_1\|v_h\|^2_{\mathcal{P}} \le \|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}$$

with $\frac{1}{Q_1} := \frac{1}{T}\sup_{\widehat{\Omega}}\{|\det(J_{\boldsymbol{G}})|\} + d^2\widehat{C}_\Delta\left(2\widehat{C}_1+\widehat{C}_2\right)$.

Now we prove the other bound. We observe that $\widehat{v}_h = v_h \circ \boldsymbol{G}$ and $\boldsymbol{G}(\boldsymbol{\eta},\tau) = (\boldsymbol{F}(\boldsymbol{\eta}), T\tau)$. Thus, with similar arguments and using (4.3.1), we have

$$\int_0^1 \int_{\widehat{\Omega}} \partial_\tau \widehat{v}_h^2 \,\mathrm{d}\widehat{\Omega}\ \mathrm{d}\tau \le T\sup_{\Omega}\{|\det(J_{\boldsymbol{G}^{-1}})|\}\,\|v_h\|^2_{\mathcal{P}}$$

and

$$\int_0^1 \int_{\widehat{\Omega}} \left(\frac{\partial^2 \widehat{v}_h}{\partial \eta_i^2}\right)^2 \mathrm{d}\widehat{\Omega}\ \mathrm{d}\tau \le dC_\Delta\left(2C_1 + C_2\right)\|v_h\|^2_{\mathcal{P}},$$

where

$$C_1 := 2\frac{1}{T}\max_i\sup_\Omega\left\{\left(\|[J^{-1}_{\boldsymbol{G}^{-1}}]_{\cdot,i}\|_2\right)^4 |\det(J_{\boldsymbol{G}^{-1}})|\right\}$$

and

$$C_2 := 2\frac{1}{T}\max_i\sup_\Omega\left\{\left(\left\|\frac{\partial[J^{-1}_{\boldsymbol{G}^{-1}}]_{\cdot,i}}{\partial\eta_i}\right\|_2\right)^2 |\det(J_{\boldsymbol{G}^{-1}})|\right\}.$$

We conclude that

$$\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}} \le Q_2\|v_h\|^2_{\mathcal{P}}$$

with $Q_2 := T\sup_\Omega\{|\det(J_{\boldsymbol{G}^{-1}})|\} + d^2C_\Delta\left(2C_1 + C_2\right)$. □

Theorem 4.4. *Under Assumptions 4.2–4.4, it holds*

$$\theta \le \lambda_{\min}(\mathbf{P}^{-1}\mathbf{A}), \qquad \lambda_{\max}(\mathbf{P}^{-1}\mathbf{A}) \le \Theta,$$

where θ and Θ are positive constants that do not depend on h_s, h_t, p_s and p_t.

Proof. Let $\widehat{v}_h \in \widehat{\mathcal{V}}_{h,0}$, $\mathbf{v}$ its coordinate vector with respect to the basis (4.2.2) and $v_h = \widehat{v}_h \circ \boldsymbol{G}^{-1} \in \mathcal{V}_{h,0}$. Thanks to Courant-Fischer theorem, we have to show that there are bounds θ and Θ such that

$$\theta \leq \frac{\mathbf{v}^T\mathbf{A}\mathbf{v}}{\mathbf{v}^T\mathbf{P}\mathbf{v}} \leq \Theta$$

holds for all $\mathbf{v}$. Equivalently, using (4.4.4) and noting that $\mathbf{v}^T\mathbf{A}\mathbf{v} = \mathcal{A}(v_h, v_h)$ and $\mathbf{v}^T\mathbf{P}\mathbf{v} = \widehat{\mathcal{P}}(\widehat{v}_h, \widehat{v}_h) = \|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}$, it is sufficient to show that there are bounds θ and Θ such that

$$\theta C_\Delta \leq \frac{\|v_h\|^2_{\mathcal{P}}}{\|\widehat{v}_h\|^2_{\widehat{\mathcal{P}}}} \leq \frac{\Theta}{2d} \quad \forall \widehat{v}_h \in \widehat{\mathcal{V}}_{h,0},$$

with $v_h = \widehat{v}_h \circ \boldsymbol{G}^{-1}$. Using Proposition 4.6, we can conclude that the previous inequalities hold with $\theta := \frac{1}{C_\Delta Q_2}$ and $\Theta := \frac{2d}{Q_1}$. □

4.4.2 Preconditioner application by fast diagonalization method

The application of the preconditioner is a solution of a Sylvester-like equation: given $\mathbf{r}$ find $\mathbf{s}$ such that

$$\mathbf{P}\mathbf{s} = \mathbf{r}. \tag{4.4.5}$$

Following [93], to solve (4.4.5), we use FD method. It is a direct method that, at the first step, computes the eigendecomposition of the pencils $(\widehat{M}_i, \widehat{L}_i)$ for $i = 1, \dots, d$ and of $(\widehat{M}_t, \widehat{K}_t)$, i.e.

$$\widehat{L}_i U_i = \widehat{M}_i U_i \Lambda_i, \qquad \widehat{K}_t U_t = \widehat{M}_t U_t \Lambda_t \tag{4.4.6}$$

where Λ_i and Λ_t are diagonal eigenvalue matrices while the columns of U_i and U_t contain the corresponding generalized eigenvectors and they are such that

$$\widehat{M}_i = U_i^{-T}U_i^{-1}, \qquad \widehat{L}_i = U_i^{-T}\Lambda_i U_i^{-1}, \qquad \widehat{M}_t = U_t^{-T}U_t^{-1}, \qquad \widehat{K}_t = U_t^{-T}\Lambda_t U_t^{-1}.$$

Then, we can rewrite $\widehat{M}_s$ as

$$\begin{aligned}
\widehat{M}_s &= (U_d^{-T}U_d^{-1}) \otimes \cdots \otimes (U_1^{-T}U_1^{-1}) = (U_d^{-T} \otimes \cdots \otimes U_1^{-T})(U_d^{-1} \otimes \cdots \otimes U_1^{-1}) && \text{using (2.2.2),}\\
&= (U_d \otimes \cdots \otimes U_1)^{-T}(U_d \otimes \cdots \otimes U_1)^{-1} = U_s^{-T}U_s^{-1} && \text{using (2.2.1) and (2.2.3),}
\end{aligned}$$

where $U_s := U_d \otimes \cdots \otimes U_1$. Similarly, denoting with $\mathbb{I}_m \in \mathbb{R}^{m\times m}$ the identity matrix of size m and defining $\Lambda_s := \sum_{i=1}^d \mathbb{I}_{n_s^{i-1}} \otimes \Lambda_i \otimes \mathbb{I}_{n_s^{d-i}}$, we rewrite $\widetilde{L}_s$ as

$$\begin{aligned}
\widetilde{L}_s &= \textstyle\sum_{i=1}^d (U_d^{-T}U_d^{-1}) \otimes \cdots \otimes (U_{i+1}^{-T}U_{i+1}^{-1}) \otimes (U_i^{-T}\Lambda_i U_i^{-1}) \otimes (U_{i-1}^{-T}U_{i-1}^{-1}) \otimes \cdots \otimes (U_1^{-T}U_1^{-1})\\
&= \textstyle\sum_{i=1}^d (U_d^{-T} \otimes \cdots \otimes U_1^{-T})(\mathbb{I}_{n_s^{i-1}} \otimes \Lambda_i \otimes \mathbb{I}_{n_s^{d-i}})(U_d^{-1} \otimes \cdots \otimes U_1^{-1}) && \text{using (2.2.2),}\\
&= U_s^{-T} \otimes \Lambda_s \otimes U_s^{-1} && \text{using (2.2.1), (2.2.2) and (2.2.3).}
\end{aligned}$$

Then, $\mathbf{P}$ can be factorized as

$$\mathbf{P} = (U_t \otimes U_s)^{-T}(\Lambda_t \otimes \mathbb{I}_{n_s^d} + \mathbb{I}_{n_t} \otimes \Lambda_s)(U_t \otimes U_s)^{-1},$$

where we have used (2.2.1), (2.2.2) and (2.2.3). Therefore, after introducing the tensors $\mathfrak{R}, \widetilde{\mathfrak{Q}} \in \mathbb{R}^{n_{s,1}\times \dots n_{s,d} \times n_t}$ s.t. $\mathrm{vec}(\mathfrak{R}) = \mathbf{r}$ and $\mathrm{vec}\left(\widetilde{\mathfrak{Q}}\right) = \widetilde{\mathbf{q}}$, the solution of (4.4.5) can be obtained by the following algorithm.

Algorithm 3 $(d+1)$-dimensional FD

```
Setup: Compute the generalized eigendecompositions (4.4.6)
Application: Compute $\widetilde{\mathbf{r}}=(U_t\otimes U_s)^T\mathbf{r} = (U_t\otimes U_d\otimes\cdots\otimes U_1)^T\boldsymbol{r}=\mathfrak{R}\times_1 U_1^T\cdots\times_{d+1}U_t^T$.
             Compute $\widetilde{\mathbf{q}} = \left(\Lambda_t\otimes\mathbb{I}_{n_s^d}+\mathbb{I}_{n_t}\otimes\Lambda_s\right)^{-1}\widetilde{\mathbf{r}}$.
             Compute $\mathbf{s} = (U_t\otimes U_s)\,\widetilde{\mathbf{q}} = (U_t\otimes U_d\otimes\cdots\otimes U_1)\,\widetilde{\mathbf{q}} = \widetilde{\mathfrak{Q}}\times_1 U_1\cdots\times_{d+1}U_t$.
```

4.4.3 Inclusion of the geometry and coefficient information in the preconditioner

The spectral estimates in Section 4.4.1 show the dependence on $\boldsymbol{G}$ (see the proof of Theorem 4.4): the geometry parametrization affects the performance of our preconditioner (4.4.3), as it is confirmed by the numerical tests in Section 4.5. In this section, we present a strategy to partially incorporate $\boldsymbol{G}$ in the preconditioner, without increasing its computational cost. The same idea has been used in Chapter 3 for the Stokes problem (see also [81]).

We begin by splitting the bilinear form $\mathcal{A}(\cdot,\cdot)$ as

$$\mathcal{A}(v_h,w_h) = \mathcal{K}_t(v_h,w_h)+\mathcal{K}_s(v_h,w_h)-\mathcal{O}(v_h,w_h) \qquad \forall v_h,w_h\in\mathcal{V}_{h,0}$$

where

$$\mathcal{K}_t(v_h,w_h) \;:=\; \int_0^T\int_\Omega \partial_t v_h\,\partial_t w_h \;\mathrm{d}\Omega\;\mathrm{d}t, \qquad \mathcal{K}_s(v_h,w_h) \;:=\; \int_0^T\int_\Omega \Delta v_h\,\Delta w_h\;\mathrm{d}\Omega\;\mathrm{d}t,$$

$$\mathcal{O}(v_h,w_h) := \int_0^T\int_\Omega (\partial_t v_h\,\Delta w_h+\partial_t w_h\,\Delta v_h)\;\mathrm{d}\Omega\;\mathrm{d}t.$$

Using that $v_h := \widehat{v}_h\circ\boldsymbol{G}^{-1}$, $w_h := \widehat{w}_h\circ\boldsymbol{G}^{-1}$ and

$$\frac{\partial^2 v_h}{\partial x_i^2} = \sum_{j,k=1}^d \frac{\partial^2\widehat{v}_h\circ\boldsymbol{G}^{-1}}{\partial\eta_j\partial\eta_k}[J_{\boldsymbol{G}}^{-1}]_{k,i}[J_{\boldsymbol{G}}^{-1}]_{j,i}+\sum_{j=1}^d\frac{\partial\widehat{v}_h\circ\boldsymbol{G}^{-1}}{\partial\eta_j}\frac{\partial[J_{\boldsymbol{G}}^{-1}]_{j,i}}{\partial\eta_i},$$

we can rewrite $\mathcal{K}_t$ and $\mathcal{K}_s$ as

$$\mathcal{K}_t(v_h,w_h) \;=\int_0^1\int_{\widehat{\Omega}} c_{d+1}\partial_\tau\widehat{v}_h\,\partial_\tau\widehat{w}_h\;\mathrm{d}\widehat{\Omega}\;\mathrm{d}\tau, \qquad \mathcal{K}_s(v_h,w_h)\;=\mathcal{K}_{s,1}(\widehat{v}_h,\widehat{w}_h)\;+\mathcal{K}_{s,2}(\widehat{v}_h,\widehat{w}_h)$$

where

$$\mathcal{K}_{s,1}(\widehat{v}_h,\widehat{w}_h) := \sum_{k=1}^d\int_0^1\int_{\widehat{\Omega}} c_k\frac{\partial^2\widehat{v}_h}{\partial\eta_k^2}\frac{\partial^2\widehat{w}_h}{\partial\eta_k^2}\;\mathrm{d}\widehat{\Omega}\;\mathrm{d}\tau,$$

$$\mathcal{K}_{s,2}(\widehat{v}_h,\widehat{w}_h) := \sum_{\substack{r,s=1\\ r\neq s}}^d\sum_{\substack{j,k=1\\ j\neq k}}^d\int_0^1\int_{\widehat{\Omega}} g^1_{rsjk}\frac{\partial^2\widehat{v}_h}{\partial\eta_k\partial\eta_j}\frac{\partial^2\widehat{w}_h}{\partial\eta_r\partial\eta_s}\;\mathrm{d}\widehat{\Omega}\,\mathrm{d}t+\sum_{j,k=1}^d\int_0^1\int_{\widehat{\Omega}} g^2_{jk}\frac{\partial\widehat{v}_h}{\partial\eta_k}\frac{\partial\widehat{w}_h}{\partial\eta_j}\;\mathrm{d}\widehat{\Omega}\,\mathrm{d}t$$
$$+\sum_{r=1}^d\sum_{j,k=1}^d\int_0^1\int_{\widehat{\Omega}} g^3_{rjk}\left(\frac{\partial^2\widehat{v}_h}{\partial\eta_k\partial\eta_j}\frac{\partial\widehat{w}_h}{\partial\eta_r}+\frac{\partial^2\widehat{w}_h}{\partial\eta_k\partial\eta_j}\frac{\partial\widehat{v}_h}{\partial\eta_r}\right)\;\mathrm{d}\widehat{\Omega}\,\mathrm{d}t$$

and where we have defined

$$c_k := \left(\left\|[J_{\boldsymbol{G}}^{-1}]_{\cdot,k}\right\|_2\right)^4|\det(J_{\boldsymbol{F}})|T \quad \text{for } k=1,\ldots,d, \quad c_{d+1} := |\det(J_{\boldsymbol{F}})|T^{-1}, \tag{4.4.7}$$

while $g^1_{rsjk}, g^2_{jk}, g^3_{rjk}$ are functions that depend on the parametrization $\boldsymbol{G}$.

The preconditioner will be based on an approximation of $\mathcal{K}_t + \mathcal{K}_{s,1}$ only. In particular we approximate c_k, for $k = 1, \dots, d+1$ as

$$\begin{aligned} &c_k(\boldsymbol{\eta}, \tau) \approx \mu_1(\eta_1) \dots \mu_{k-1}(\eta_{k-1})\omega_k(\eta_k)\mu_{k+1}(\eta_{k+1}) \dots \mu_d(\eta_d)\mu_{d+1}(\tau) \qquad \text{for } k = 1, \dots, d, \\ &c_{d+1}(\boldsymbol{\eta}, \tau) \approx \mu_1(\eta_1) \dots \mu_d(\eta_d)\omega_{d+1}(\tau). \end{aligned} \tag{4.4.8}$$

The functions c_k in (4.4.8) are first interpolated by constants in each element and then the construction of the univariate factors μ_k, and ω_k is performed by the separation of variable algorithm detailed in the Appendix A. The resulting computational cost is therefore proportional to the number of elements, which for smooth splines is roughly equal to N_{dof}, and independent of the degrees p_s and p_t. As a consequence, the computation of (4.4.8) has a negligible cost in the whole iterative strategy. This first step leads to a matrix of this form

$$\overline{\mathbf{P}^{\boldsymbol{G}}} := \widehat{K}_t^{\boldsymbol{G}} \otimes \widehat{M}_s^{\boldsymbol{G}} + \widehat{M}_t^{\boldsymbol{G}} \otimes \widetilde{L}_s^{\boldsymbol{G}},$$

where, referring to (4.2.1) for the notation of the basis functions, for $i, j = 1, \dots, n_t$,

$$\left[\widehat{K}_t^{\boldsymbol{G}}\right]_{i,j} := \int_0^1 \omega_{d+1}(\tau)\, \widehat{b}'_{i,p_t}(\tau)\, \widehat{b}'_{j,p_t}(\tau)\, \mathrm{d}\tau, \qquad \left[\widehat{M}_t^{\boldsymbol{G}}\right]_{i,j} := \int_0^1 \mu_{d+1}(\tau)\, \widehat{b}_{i,p_t}(\tau)\, \widehat{b}_{j,p_t}(\tau)\, \mathrm{d}\tau \tag{4.4.9}$$

$$\widetilde{L}_s^{\boldsymbol{G}} := \sum_{k=1}^{d} \widehat{M}_d^{\boldsymbol{G}} \otimes \dots \otimes \widehat{M}_{k+1}^{\boldsymbol{G}} \otimes \widehat{L}_k^{\boldsymbol{G}} \otimes \widehat{M}_{k-1}^{\boldsymbol{G}} \otimes \dots \otimes \widehat{M}_1^{\boldsymbol{G}}, \qquad \widehat{M}_s^{\boldsymbol{G}} := \widehat{M}_d^{\boldsymbol{G}} \otimes \dots \otimes \widehat{M}_1^{\boldsymbol{G}},$$

with for $i, j = 1, \dots, n_{s,k}$ and $k = 1, \dots, d$,

$$[\widehat{L}_k^{\boldsymbol{G}}]_{i,j} := \int_0^1 \omega_k(\eta_k)\, \widehat{b}''_{i,p_s}(\eta_k)\, \widehat{b}''_{j,p_s}(\eta_k)\, \mathrm{d}\eta_k, \qquad [\widehat{M}_k^{\boldsymbol{G}}]_{i,j} := \int_0^1 \mu_k(\eta_k)\, \widehat{b}_{i,p_s}(\eta_k)\, \widehat{b}_{j,p_s}(\eta_k)\, \mathrm{d}\eta_k. \tag{4.4.10}$$

The matrix $\overline{\mathbf{P}^{\boldsymbol{G}}}$ maintains the Kronecker structure of (4.4.3) and Algorithm 3 can still be used to compute its application.

Finally, as in [81], we apply a diagonal scaling and we define the preconditioner as $\mathbf{P}^{\boldsymbol{G}} := \mathbf{D}^{1/2}\, \overline{\mathbf{P}^{\boldsymbol{G}}}\, \mathbf{D}^{1/2}$ where $\mathbf{D}$ is the diagonal matrix whose diagonal entries are $[\mathbf{D}]_{i,i} := [\mathbf{A}]_{i,i}/[\overline{\mathbf{P}^{\boldsymbol{G}}}]_{i,i}$.

Remark 4.2. *For the model problem considered in this chapter, the approximation of the geometry parametrization in the time direction is trivial. Notice that the coefficients in* (4.4.7) *do not depend on τ. Indeed, in our case it holds*

$$K_t = \frac{1}{T}\widehat{K}_t, \qquad M_t = T\widehat{M}_t,$$

and hence we could set explicitly $\widehat{K}_t^{\boldsymbol{G}} = K_t$ and $\widehat{M}_t^{\boldsymbol{G}} = M_t$, which is exact. However, we want to present the more general approximating strategy above which could be used also when the spatial geometry or equation's coefficients depend on time.

4.4.4 Computational cost and memory consumption of the linear solver

The cost of our preconditioning strategies consists of two parts: setup cost and application cost.

The setup cost of both $\mathbf{P}$ and $\mathbf{P}^{\boldsymbol{G}}$ includes the eigendecomposition of the pencils $(\widehat{L}_i, \widehat{M}_i)$ and $(\widehat{K}_t, \widehat{M}_t)$ or $(\widehat{L}_i^{\boldsymbol{G}}, \widehat{M}_i^{\boldsymbol{G}})$ and $(\widehat{K}_t^{\boldsymbol{G}}, \widehat{M}_t^{\boldsymbol{G}})$, respectively, that is, Step 1 of Algorithm 1. If we assume for simplicity that $\widehat{L}_i, \widehat{M}_i$, $\widehat{L}_i^{\boldsymbol{G}}, \widehat{M}_i^{\boldsymbol{G}}$ for $i = 1, \ldots, d$ have size $n_s \times n_s$ and that $\widehat{K}_t$, $\widehat{M}_t$, $\widehat{K}_t^{\boldsymbol{G}}$ and $\widehat{M}_t^{\boldsymbol{G}}$ have size $n_t \times n_t$, then the cost of the eigendecomposition is $O(dn_s^3 + n_t^3)$ FLOPs. This cost is optimal for $d = 2$ and negligible for $d = 3$, provided that $n_t \approx n_s$. For $\mathbf{P}^{\boldsymbol{G}}$, we also have to include in the setup cost the creation of the diagonal matrix $\mathbf{D}$, which is negligible, and the construction of the $2(d+1)$ univariate approximations $\mu_1, \ldots, \mu_{d+1}$ and $\omega_1, \ldots, \omega_{d+1}$, that are used to incorporate some geometry information into the preconditioner. As explained in Section 4.4.3, this has a cost which is $O(N_{dof})$ FLOPs.

The application of $\mathbf{P}$ and $\overline{\mathbf{P}^{\boldsymbol{G}}}$, is performed by Algorithm 3, Steps 2–4. Step 2 and Step 4 are efficiently performed exploiting property (2.2.5) and they need a total of $4(dn_s^{d+1}n_t + n_t^2 n_s^d) = 4N_{dof}(dn_s + n_t)$ FLOPs, while Step 3 has an optimal cost, as it requires $O(N_{dof})$ FLOPs. Thus, the total cost of Algorithm 1 is $4N_{dof}(dn_s + n_t) + O(N_{dof})$ FLOPs. The non-optimal dominant cost is given by the dense matrix-matrix products of Step 2 and Step 4, which, however, are usually implemented on modern computers in a high-efficient way, as they are BLAS level 3 operations. In our numerical tests, the overall serial computational time grows almost as $O(N_{dof})$ up to the largest problem considered, as we will show in Section 4.5.

Clearly, the computational cost of each iteration of the CG solver depends on both the preconditioner application and the residual computation. For the sake of completeness, we also discuss the cost of the residual computation, which consists in the multiplication between $\mathbf{A}$ and a vector. Note that this multiplication can be computed by exploiting the special structure (4.3.22) and the formula (2.2.4). In this case, we do not need to compute and store the whole matrix $\mathbf{A}$, but only its factors K_t, S_t, M_t, K_s, L_s and M_s. With this matrix-free approach, noting that the time matrices K_t, S_t, M_t are banded matrices with a band of width $2p_t + 1$ and the spatial matrices K_s, L_s, M_s have a number of non-zeros per row approximately equal to $(2p_s + 1)^d$, the computational cost of a single matrix-vector product is $6\left[(2p_s + 1)^d + 2p_t + 1\right] N_{dof} \approx 6(2p + 1)^d N_{dof}$, if $p = p_s \approx p_t$. Even if this cost is lower than what one would get by using $\mathbf{A}$ explicitly, the comparison with the cost of the preconditioner shows that the residual computation easily turns out to be the dominant cost of the iterative solver (see Table 4.3 in Section 4.5). This issue was already recognized in [93, 81] (see also Chapter 3).

We now analyze the memory consumption. For the preconditioner, we need to store the eigenvector matrices $U_t, U_1, \ldots, U_d$ and the diagonal eigenvalue matrix $\Big(\Lambda_t \otimes I_{n_s^d} + I_{n_t} \otimes \Lambda_s\Big)$. The memory required is

$$n_t^2 + dn_s^2 + N_{dof}.$$

For the system matrix, we need to store the matrices K_t, M_t, M_s, K_s and L_s (the storage of S_t is negligible). The memory required is roughly

$$2\,(2p_t + 1)\,n_t + 3\,(2p_s + 1)^d\,N_s.$$

These numbers show that memory-wise our space-time strategy is very appealing when compared to other approaches, even when spatial and time variables are discretized separately, e.g., with finite differences in time or other time-stepping schemes. To see this, take $d = 3$ and $p_t \approx p_s = p$, and assume $n_t^2 \leq Cp^3 N_s$. In this case, the total memory consumption is then $O\left(p^3 N_s + N_{dof}\right)$ which is the memory required to store the Galerkin matrices associated to spatial variables, plus the memory required to store the solution of the problem.

We emphasize that it is possible, though beyond the scope of this work, to take the matrix-free paradigm one step further by using the approach developed in [94]. Using this approach, where even the factors of $\mathbf{A}$ as in (4.3.22) are not needed, would significantly improve the overall iterative solver in terms of memory and computational cost (both for the setup and for the matrix-vector computations).

4.5 Numerical results

In this section, we show numerical experiments that confirm the convergence behaviour (4.3.20) of the least-squares approximation method defined in Section 4.3.3, and then we present some numerical results regarding the performance of our preconditioner.

The tests are performed with Matlab R2015a and GeoPDEs toolbox [111], on a Intel Core i7-5820K processor, running at 3.30 GHz, with 64 GB of RAM.

In Algorithm 3, the eigendecomposition of Step 1 is done by `eig` Matlab function, while the multiplications of Kronecker matrices, appearing in Step 2 and 4, are performed by Tensorlab toolbox [102]. We fix the tolerance of CG equal to 10^{-8} and the initial guess equal to the null vector in all tests.

We set $h_s = h_t =: h$, and we denote the number of subdivisions in each parametric direction by n_{sub}.

4.5.1 Orders of convergence

We perform accuracy tests in a 2D spatial domain since the calculation of the numerical errors on 3D spatial domains is expensive in terms of computational time, when element-wise Gaussian quadrature is adopted. We set $T = 1$ and we consider a 2D spatial domain: the quarter of annulus with internal radius equal to 1 and external radius equal to 2 (see Figure 4.1a). The initial and Dirichlet boundary conditions and the source term f are fixed such that the exact solution is $u = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2 \sin(\pi t)$. We solved the linear system with Matlab direct solver (backslash "\" operator). Figure 4.2a shows the $\|\cdot\|_{\mathcal{V}_0}$ relative errors with splines of degree $p_s = p_t$ from 2 to 6: the rate of convergence of $O(h^{p_t-1})$ confirms the results of Theorem 4.3. As predicted by the theory, if we increase the degree of spatial B-splines and we set $p_s = p_t + 1$, we can gain an order of convergence. Indeed, Figure 4.2b shows that in this case the $\|\cdot\|_{\mathcal{V}_0}$ relative errors have order p_t.

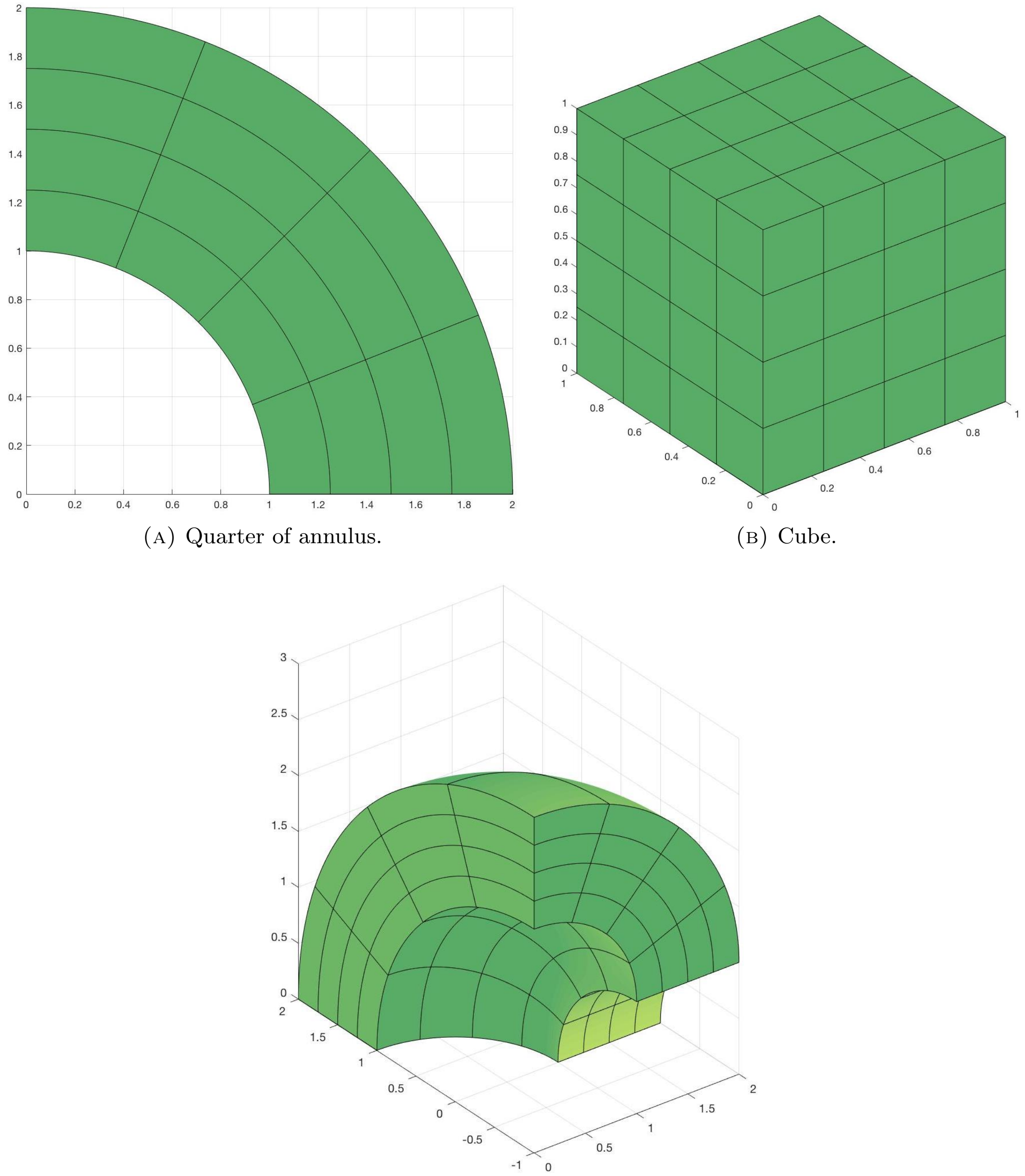
Even if theoretical results do not cover this case, we also analyze in Figures 4.2c and 4.2d the error behaviour for $p_t = p_s$ in $L^2(\Omega \times [0, T])$ and $H^1(\Omega \times [0, T])$ norms, respectively. While the H^1 errors are optimal for every p_t considered, i.e. they are of order p_t for $p_t \geq 2$, the orders of convergence in L^2 norm are optimal and thus equal to $p_t + 1$, only for $p_t \geq 3$. The suboptimal behaviour of the error in L^2 norm for $p_t = p_s = 2$ is in fact consistent with the Aubin-Nitsche type estimate and with the a-priori error estimates for fourth-order PDEs (see in particular the classical result [105, Theorem 3.7]).

4.5.2 Performance of the preconditioner

To assess the performance of our preconditioning strategy, we set $T = 1$ and we focus on two 3D spatial domains $\Omega \subset \mathbb{R}^3$, represented in Figure 4.1b and Figure 4.1c: the cube and the rotated quarter of annulus, respectively. As a comparison, we also consider as preconditioner for CG the Incomplete Cholesky with zero fill-in (IC(0)) factorization of $\mathbf{A}$, that is executed by the Matlab routine `ichol`. Tables 4.1 and 4.2 report the number of iterations and the total solving time, that includes the setup time of the preconditioner. The symbol " * " is used when the construction of the matrix $\mathbf{A}$ or its matrix factors go out-of-memory. We force the execution to be sequential and to use only a single computational thread.

As discussed in the previous section, the matrix-vector products of CG are computed in a matrix-free way using its factors as in (4.3.22). Matrix $\mathbf{A}$ is still assembled in order to use the IC(0) preconditioner. In any case, the assembly times are never included in the reported times.

For simplicity, we consider only the case where $p_s = p_t$. The case where $p_s = p_t + 1$ will lead to a computational cost that is of the same order of the case $p_t = p_s$, as it can be inferred from Section 4.4.4.

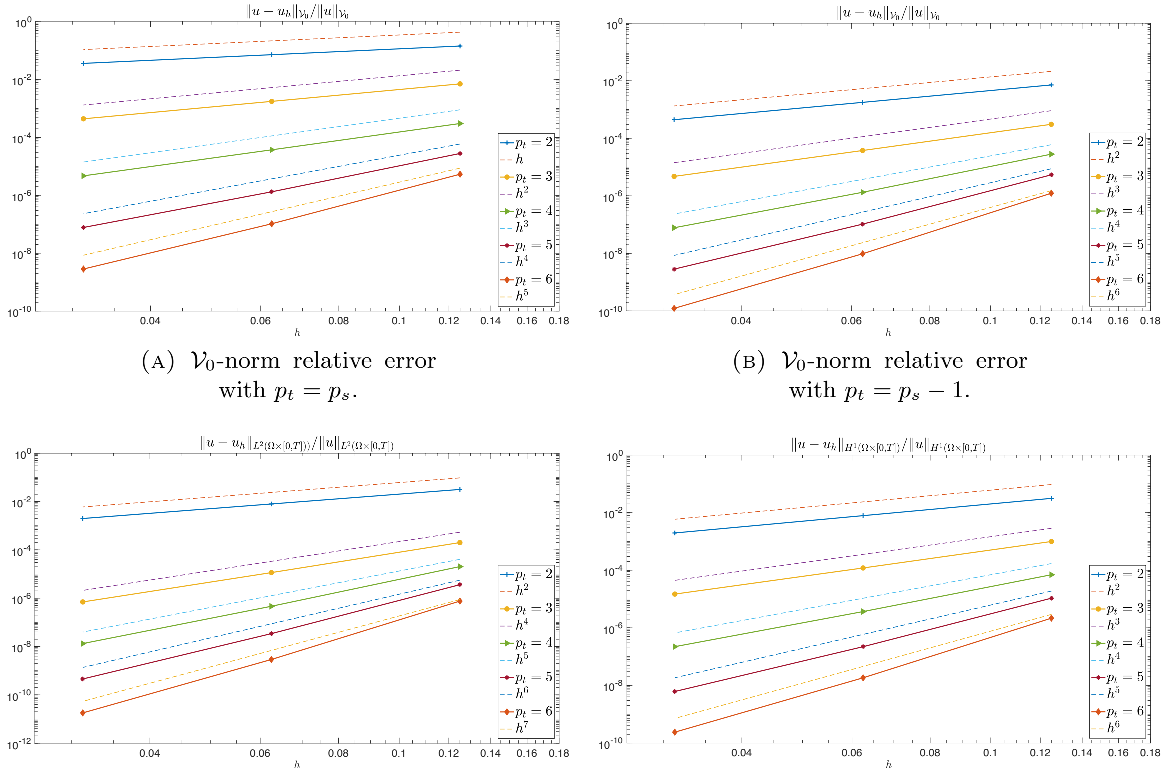


(A) Quarter of annulus.

(B) Cube.

(C) Rotated quarter of annulus.

FIGURE 4.1: Space-time least-squares. Computational domains.



(A) $\mathcal{V}_0$-norm relative error with $p_t = p_s$.

(B) $\mathcal{V}_0$-norm relative error with $p_t = p_s - 1$.

(C) L^2-norm relative error with $p_t = p_s$.

(D) H^1-norm relative error with $p_t = p_s$.

FIGURE 4.2: Space-time least-squares. Relative errors.

Cube. We first consider the domain $\widehat{\Omega} = \Omega = (0,1)^3$ (Figure 4.1b). Note that in this case we have that $[\mathbf{P}]_{i,j} = \widehat{\mathcal{P}}(\widehat{B}_{i,\boldsymbol{p}}, \widehat{B}_{j,\boldsymbol{p}}) = \mathcal{P}(B_{i,\boldsymbol{p}}, B_{j,\boldsymbol{p}})$. We set homogeneous Dirichlet and zero initial boundary conditions and we fix f such that the exact solution is $u = \sin(\pi x)\sin(\pi y)\sin(\pi z)\sin(t)$.

Table 4.1 shows the performance of $\mathbf{P}$ and IC(0) preconditioners in the case $p_t = p_s$. The number of iterations obtained with $\mathbf{P}$ are stable w.r.t p_t and n_{sub}.

Even if the number of iterations of our strategy might be larger than that of IC(0), the overall computational time is significantly lower, up to two orders of magnitude for the problems considered. This is due to the higher setup and application cost of the IC(0) preconditioner.

Rotated quarter of annulus. Finally, we consider as computational domain Ω a quarter of annulus with center in the origin, internal radius 1 and external radius 2, rotated along the axis $\{(x,-1,0) \mid x \in \mathbb{R}\}$ by $\pi/2$ (see Figure 4.1c). Boundary data and forcing function are set such that the exact solution is $u = -(x^2+y^2-1)(x^2+y^2-4)xy^2\sin(z)\sin(t)$.

Table 4.2 shows the results of CG coupled with $\mathbf{P}$, $\mathbf{P}^{\boldsymbol{G}}$ or IC(0) preconditioner. From the spectral estimates of Theorem 4.4, we know that the geometry parametrization $\boldsymbol{G}$, which in this case is not trivial, plays a key-role in the performance of $\mathbf{P}$. This is confirmed by the results of Table 4.2: the number of iterations is higher than the ones obtained in the cube domain, where $\boldsymbol{G}$ is the identity map (see Table 4.1). However, the inclusion of some geometry information, and thus the use of $\mathbf{P}^{\boldsymbol{G}}$ as a preconditioner, improves the performances, as we can see from the middle table of Table 4.2. Moreover, we show that IC(0) is not competitive neither with $\mathbf{P}$ nor with $\mathbf{P}^{\boldsymbol{G}}$, in terms of computational time.

For the last domain, we analyze the percentage of computation time of a $\mathbf{P}^{\boldsymbol{G}}$ application with respect to the overall CG time. The results, reported in Table 4.3, show that the time

	$\mathbf{P}$ + CG $p_t = p_s$ Iterations / Time			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	9 / 0.06	11 / 0.07	11 / 0.18	11 / 0.28
16	11 / 0.27	11 / 0.69	12 / 1.80	12 / 3.80
32	12 / 5.10	12 / 13.37	12 / 27.31	12 / 52.95
64	13 / 100.09	13 / 227.93	13 / 458.86	13 / 924.44
128	13 / 2012.94	13 / 4235.96	$*$	$*$

	IC(0) + CG $p_t = p_s$ Iterations / Time			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	9 / 0.18	7 / 1.69	6 / 14.04	6 / 80.39
16	22 / 5.01	16 / 45.54	12 / 355.99	10 / 1913.90
32	64 / 157.05	$*$	$*$	$*$

TABLE 4.1: Space-time least-squares. Cube domain with $p_t = p_s$. Performance of $\mathbf{P}$+CG (upper table) and of IC(0)+CG (lower table).

spent in the preconditioner application takes only a little amount of the overall solving time. The dominant cost, in this implementation is due to the matrix-vector products of the residual computation, that is the other main operation performed in a CG cycle.

Since we are primarily interested in the preconditioner performance, in Figure 4.3 we report in a log-log scale the computational times required for the setup and for a single application of $\mathbf{P}^{\boldsymbol{G}}$ versus the number of degrees-of-freedom. We see that the setup time is clearly asymptotically proportional to N_{dof}, as expected. Remarkably, the single application time grows slower than the expected theoretical cost $O(N_{dof}^{5/4})$; indeed, it grows almost as the optimal rate $O(N_{dof})$, even for the largest problems tested. As already mentioned, this is likely due to the high efficiency of the BLAS level 3 routines that perform the computational core of the application of the preconditioner.

4.6 Conclusions

In this chapter, we have proposed and studied a least-squares method for the heat equation, that allows us to design an innovative preconditioner in the framework of Isogeometric Analysis. Even though we adopt a global-in-time space-time formulation, based on smooth splines in space and time, the preconditioner $\mathbf{P}$ that we have presented is highly efficient both in terms of FLOPs and memory, thanks to its matrix representation as suitable sum of Kronecker products, leading to a Sylvester-like problem.

The computational cost of the preconditioner setup is at most $O(N_{dof})$ FLOPs while its application is $O(N_{dof}^{1+1/d})$ FLOPs. In our numerical benchmarks the computational time, for serial single-core execution, is in fact close to $O(N_{dof})$, with no dependence on p. The proposed preconditioner $\mathbf{P}$ is indeed robust with respect to the spline degree and its variant, denoted with $\mathbf{P}^{\boldsymbol{G}}$, has a good performance also when the geometry parametrization $\boldsymbol{G}$ of the patch is not trivial.

The storage cost is instead $O(p^d N_s + N_{dof})$, under the reasonable assumption that $n_t^2 \leq C p^d N_s$. We emphasize that is roughly the same storage cost that one would get by discretizing separately in space and time.

	$\mathbf{P}$ + CG $p_t = p_s$ Iterations / Time			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	107 / 0.21	107 / 0.48	114 / 1.17	123 / 2.73
16	126 / 2.56	128 / 6.90	133 / 17.04	135 / 35.17
32	142 / 52.77	143 / 132.24	148 / 292.53	151 / 572.84
64	153 / 1056.21	155 / 2415.23	156 / 4956.68	159 / 9906.33
128	164 / 22106.01	166 / 47539.02	$*$	$*$

	$\mathbf{P^G}$ + CG $p_t = p_s$ Iterations / Time			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	24 / 0.09	24 / 0.13	26 / 0.37	26 / 0.60
16	35 / 0.77	34 / 1.96	33 / 4.62	33 / 9.35
32	42 / 17.03	41 / 39.57	40 / 82.35	41 / 161.73
64	46 / 333.20	44 / 716.03	49 / 1577.55	53 / 3384.08
128	48 / 6767.08	50 / 14814.09	$*$	$*$

	IC(0) + CG $p_t = p_s$ Iterations / Time			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	11 / 0.17	8 / 1.71	7 / 13.96	6 / 80.28
16	29 / 5.52	18 / 45.22	14 / 377.47	11 / 1895.55
32	86 / 185.08	$*$	$*$	$*$

TABLE 4.2: Space-time least-squares. Rotated quarter domain with $p_t = p_s$. Performance of $\mathbf{P}$+CG(upper table), $\mathbf{P^G}$+CG (middle table) and of IC(0)+CG (lower table).

	$\mathbf{P^G}$			
n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$
8	35.86%	20.66%	10.85%	7.05 %
16	17.90%	8.10 %	3.95 %	2.28 %
32	14.25 %	7.35 %	4.05 %	2.49 %
64	17.28 %	8.75 %	4.67 %	2.52 %
128	23.98 %	12.21 %	$*$	$*$

TABLE 4.3: Space-time least-squares. Rotated quarter domain with $p_t = p_s$. Percentage of computational time of the preconditioner $\mathbf{P^G}$ application in the overall CG cycle.

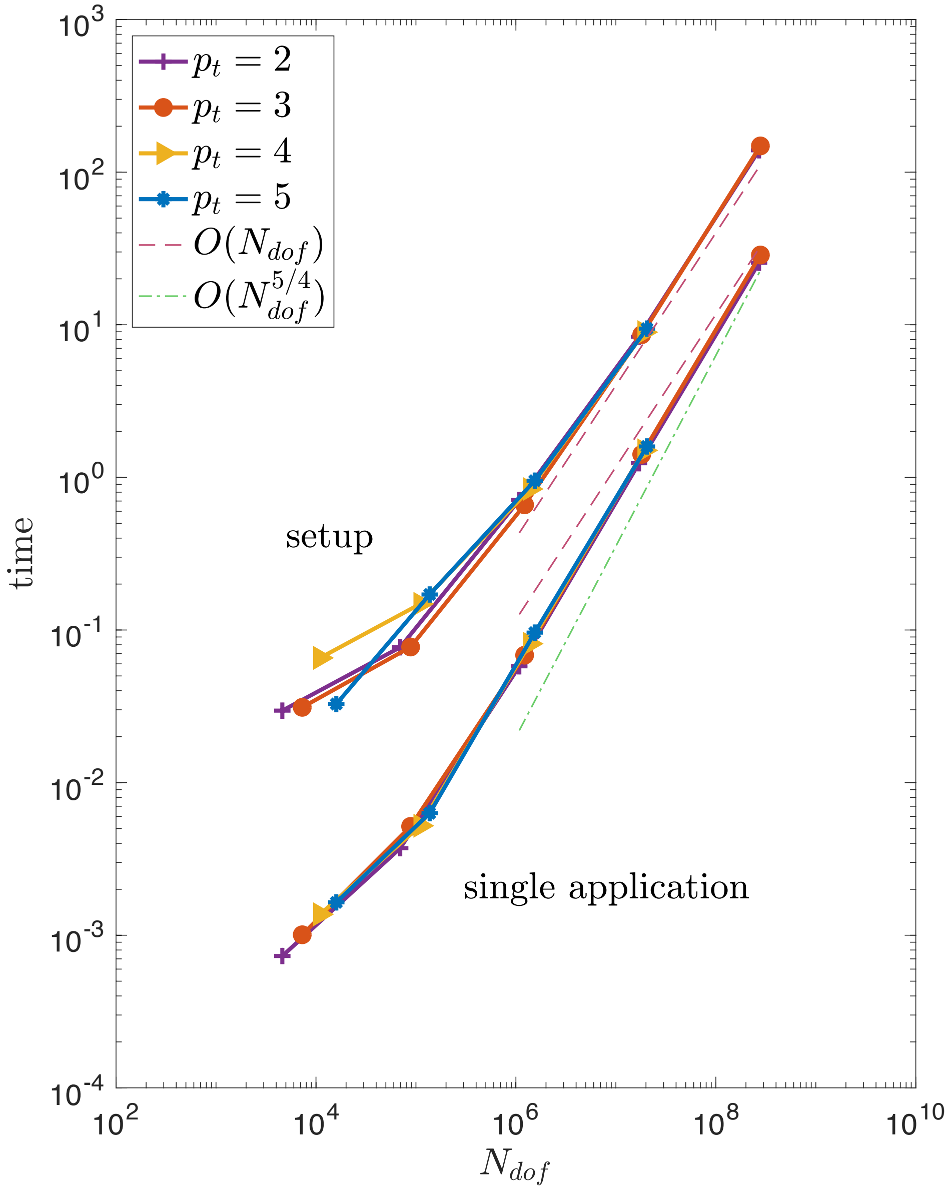


FIGURE 4.3: Space-time least-squares. Rotated quarter domain with $p_t = p_s$. Setup times and single application times of $\mathbf{P}^{\boldsymbol{G}}$.

Our approach could be coupled with a matrix-free idea (see [94]), and this is expected to further improve the efficiency of the overall method. Everything is well-suited for parallelization: even though in this chapter we do not consider parallel implementation, this is a promising research direction for the future.

4.7 Technical results

4.7.1 Smooth approximation of $\widehat{\mathcal{V}}_0$

In this section we prove the density of spaces of smooth functions, with boundary conditions, in suitable Sobolev spaces on the parametric domain. The first result concerns $H_0^1 \cap H^2$.

Lemma 4.6. *Let $Q := \widehat{\Omega} \times (a, b)$ be an open $(d+1)$-dimensional box. Then, the space $C^\infty\left(\overline{Q}\right) \cap H_0^1(Q)$ is dense in $H^2(Q) \cap H_0^1(Q)$.*

Proof. Let $w \in H^2(Q) \cap H_0^1(Q)$ and $g := -\Delta w \in L^2(Q)$. Clearly, w solves, in a weak sense,

$$\begin{cases} -\Delta w = g & \text{in } Q, \\ \quad w = 0 & \text{on } \partial Q. \end{cases}$$

Let $g_n \in C_0^\infty(Q)$ such that $g_n \to g$ in $L^2(Q)$ and let $w_n \in H_0^1(Q)$ be the weak solution of

$$\begin{cases} -\Delta w_n = g_n & \text{in } Q, \\ \quad w_n = 0 & \text{on } \partial Q. \end{cases}$$

Then $w_n \to w$ in $H^2(Q)$. Note that w_n is defined on $\overline{Q}$, vanishes on its boundary ∂Q and is harmonic in a inner neighborhood of ∂Q because g_n has compact support, thus, employing recursively Schwarz reflection (see, e.g., [49, Exercise 9, Section 2.5] and [22, Remarque 10, Section IX.2]) we can extend w_n outside $\overline{Q}$, such that this extension is harmonic in a neighborhood of ∂Q. It follows that $w_n \in C^\infty\left(\overline{Q}\right)$. □

The second result focuses on the space which is needed for our least-squares formulation, that is, H^2 in space and H^1 in time, endowed with homogeneous initial and boundary conditions. This is used to show, in Theorem 4.2, the convergence of our method.

Lemma 4.7. *Let*

$$\widehat{\mathcal{V}}_0 = \left\{ v \in \left[\left(H^2(\widehat{\Omega}) \cap H_0^1(\widehat{\Omega}) \right) \otimes L^2(0,1) \right] \cap \left[L^2(\widehat{\Omega}) \otimes H^1(0,1) \right] \;\middle|\; v = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\}$$

be a Hilbert space endowed with the norm

$$\|v\|_{\widehat{\mathcal{V}}_0}^2 := \int_0^1 \|\Delta v(\cdot, \tau)\|_{L^2(\widehat{\Omega})}^2 \, \mathrm{d}\tau + \int_0^1 \|\partial_\tau v(\cdot, \tau)\|_{L^2(\widehat{\Omega})}^2 \, \mathrm{d}\tau.$$

Then, the space $C^\infty\left([0,1]^{d+1}\right) \cap \widehat{\mathcal{V}}_0$ is dense in $\widehat{\mathcal{V}}_0$.

Proof. Consider a given $w \in \widehat{\mathcal{V}}_0$ as the solution of a heat problem on the parametric domain $\widehat{\Omega} \times (0,1) = (0,1)^{d+1}$, with datum $g := (\partial_t w - \Delta w) \in L^2\left((0,1)^{d+1}\right)$, i.e.

$$\begin{cases} \partial_t w - \Delta w = g & \text{in} \quad \widehat{\Omega} \times (0,1), \\ w = 0 & \text{on} \quad \partial\widehat{\Omega} \times (0,1), \\ w = 0 & \text{in} \quad \widehat{\Omega} \times \{0\}. \end{cases} \tag{4.7.1}$$

Let $g_n \in C_0^\infty\left((0,1)^{d+1}\right)$ such that $g_n \to g$ in $L^2\left((0,1)^{d+1}\right)$ and let w_n be the solution of the same heat problem (4.7.1) with datum g_n. Following the proof of Theorem 4.1 applied in $\widehat{\Omega} \times (0,1)$, we get $w_n \to w$ in $\widehat{\mathcal{V}}_0$ while, by [49, Chapter 7, Theorem 6], we also get $w_n \in L^2(\widehat{\Omega}) \otimes H^2(0,1)$.

We use now Lemma 4.6 to approximate w_n. Fix $\delta > 0$ and consider an extension z_n of w_n and $z_n \in \left[\left(H^2(\widehat{\Omega}) \cap H_0^1(\widehat{\Omega})\right) \otimes L^2(0,1+\delta)\right] \cap \left[L^2(\widehat{\Omega}) \otimes H^2(0,1+\delta)\right]$ such that $z_n(\cdot, 1+\delta) = 0$. [2] Now observe that z_n is a function in $H^2\left(\widehat{\Omega} \times (0,1+\delta)\right) \cap H_0^1\left(\widehat{\Omega} \times (0,1+\delta)\right)$, we can then apply Lemma 4.6 to construct a sequence $z_{n,k} \in C^\infty\left([0,1]^d \times [0,1+\delta]\right) \cap H_0^1\left(\widehat{\Omega} \times (1+\delta)\right)$ converging, as $k \to \infty$, to z_n in the $H^2\left(\widehat{\Omega} \times (0,1+\delta)\right)$ norm. The restriction of $z_{n,k}$ to $[0,1]^{d+1}$ belongs to the required space $C^\infty\left([0,1]^{d+1}\right) \cap \widehat{\mathcal{V}}_0$ and the sequence converges (as $k \to \infty$) to z_n in the $H^2((0,1)^{d+1})$ norm, and thus in the $\widehat{\mathcal{V}}_0$-norm. □

4.7.2 A variational formulation equivalent to (4.3.10)–(4.3.11)

In this section, we show that the least-squares space-time functional

$$\mathcal{E}_{LS}(v) := \tfrac{1}{2} \int_0^T \|\partial_t v(\cdot,t) - \Delta v(\cdot,t) - f(\cdot,t)\|^2_{L^2(\Omega)}\, \mathrm{d}t \quad \forall v \in \mathcal{V}_0 \tag{4.7.2}$$

that appears in the minimization problem (4.3.10), coincides with another space-time functional (4.7.3) appearing in the theory of gradient flows and curves of maximal slopes (see e.g., [1, 95]).

First, let us introduce the energy $\mathcal{J} : H_0^1(\Omega) \times [0,T] \to \mathbb{R}$ given by

$$\mathcal{J}(w,t) := \int_\Omega \left(\tfrac{1}{2}|\nabla w(\boldsymbol{x})|^2 - f(\boldsymbol{x},t)w(\boldsymbol{x})\right) \mathrm{d}\Omega$$

and assume, for the sake of simplicity, that $f \in H^1(0,T;L^2(\Omega)) = L^2(\Omega) \otimes H^1(0,T)$. If $w \in H_0^1(\Omega) \cap H_\Delta(\Omega)$ then for all $z \in H_0^1(\Omega)$ and for all $t \in (0,T)$ by Green's formula we have

$$\partial_w \mathcal{J}(w,t)[z] = \int_\Omega \big(- \Delta w(\boldsymbol{x}) - f(\boldsymbol{x},t)\big) z(\boldsymbol{x})\, \mathrm{d}\Omega.$$

Moreover, thanks to the regularity of f we have

$$\partial_t \mathcal{J}(w,t) = -\int_\Omega w(\boldsymbol{x}) \partial_t f(\boldsymbol{x},t)\, \mathrm{d}\Omega.$$

At this point, let us see that the functional $\mathcal{E}_{LS}$ coincides with the following functional defined $\forall v \in \mathcal{V}_0$

$$\begin{aligned}\mathcal{E}(v) := \mathcal{J}(v(\cdot,T),T) + \tfrac{1}{2} \int_0^T \left(\|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)} + \|\Delta v(\cdot,t) + f(\cdot,t)\|^2_{L^2(\Omega)}\right) \mathrm{d}t \\ - \int_0^T \partial_t \mathcal{J}(v(\cdot,t),t)\, \mathrm{d}t. \end{aligned} \tag{4.7.3}$$

[2] The extension is obtained, for instance, in the following way. Consider the null extensions $\widetilde{f}_n$ of f_n in $L^2\left(\widehat{\Omega} \times (0,1+\delta)\right)$. Let $\widetilde{w}_n$ be the solutions of a heat problem (4.7.1) in $\widehat{\Omega} \times (0,1+\delta)$ (note that $\widetilde{w}_n$ is an extension of w_n, by uniqueness, and that $\widetilde{w}_n$ has the same regularity of w). Next, let ϕ be a cut-off function for $(0,1)$ in $(0,1+\delta)$ and let $z_n(\cdot t) = \phi(t)\, \widetilde{w}_n(\cdot,t)$.

For $v \in \mathcal{V}_0$ we know, e.g., by [23, Lemme 3.3], that the energy $t \mapsto \mathcal{J}(v(\cdot,t),t)$ is absolutely continuous and thus

$$\begin{aligned}\mathcal{J}(v(\cdot,T),T) &= \int_0^T \frac{\mathrm{d}}{\mathrm{d}t}\mathcal{J}(v(\cdot,t),t)\,\mathrm{dt} = \int_0^T \left(\partial_w \mathcal{J}(v(\cdot,t),t)[\partial_t v(\cdot,t)] + \partial_t \mathcal{J}(v(\cdot,t),t)\right)\,\mathrm{dt} \\ &= \int_0^T \int_\Omega \big(-\Delta v - f\big)\partial_t v \,\mathrm{d}\Omega\,\mathrm{dt} - \int_0^T \int_\Omega v\,\partial_t f\,\mathrm{d}\Omega\,\mathrm{dt}\,.\end{aligned}$$

Then, we can re-write the least-squares functional (4.7.2) as follows:

$$\begin{aligned}\mathcal{E}_{LS}(v) &= \tfrac{1}{2}\int_0^T \Big(\|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)} + \|\Delta v(\cdot,t) + f(\cdot,t)\|^2_{L^2(\Omega)}\Big)\,\mathrm{dt} - \int_0^T \int_\Omega (\Delta v + f)\partial_t v\,\mathrm{d}\Omega\,\mathrm{dt} \\ &= \mathcal{J}(v(\cdot,T),T) + \tfrac{1}{2}\int_0^T \Big(\|\partial_t v(\cdot,t)\|^2_{L^2(\Omega)} + \|\Delta v(\cdot,t) + f(\cdot,t)\|^2_{L^2(\Omega)}\Big)\,\mathrm{dt} \\ &\quad - \int_0^T \partial_t \mathcal{J}(v(\cdot,t),t)\,\mathrm{dt} = \mathcal{E}(v).\end{aligned}$$

As a consequence, the representation (4.3.21) in the discrete space $\mathcal{V}_{h,0}$ holds also in the space $\mathcal{V}_0$, moreover, the bilinear form (4.3.12) turns out to be the Euler-Lagrange equation of the functional (4.7.3).

4.8 Some classical results

To help the reader going through the proofs of theorems, lemmas and propositions of Section 4.3.1 and Section 4.3.2, we report in this section some of the classical results we have used. For the sake of easiness, we decide to rewrite the original results with the notations of the present chapter. For the original statements, we refer to the corresponding works.

4.8.1 Results from [73]

We report the definition that we need for the following lemma, always referring to the original work [73] for more details and for an accurate definition of quadratic form and its proper values.

Definition 4.1 (Piecewise smooth surface with curvature bounded from below by the number K (pag. 161))**.** *A surface S is said to be a piecewise smooth surface with curvature bounded from below by the number K if the following two properties are satisfied:*

1. *S is piecewise smooth with non null interior angles;*
2. *for almost every $x_0 \in S$ there is a plane tangent to S and the equation in local Cartesian coordinates of the neighborhood of x_0 has the form $y_n = \omega(y_1, \dots, y_{n-1})$, with ω two times differentiable (the axis y_n is in the direction of the exterior normal derivative to S at x_0 while the axes $y_1, \dots, y_{n-1}$ are in the plane tangent to S at x_0). We also require that the eigenvalues $\mu_1(x_0), \dots, \mu_{n-1}(x_0)$ of the matrix W with entries defined as $[W]_{k,l} = \frac{\partial^2 \omega}{\partial y_k \partial y_l}$ for $k, l = 1, \dots, n-1$ evaluated at the point x_0 are bounded from below by a non negative constant $K \in \mathbb{R}$ as $\sup_{k=1,\dots,n-1; x_0 \in S} \{\mu_k(x_0)\} \leq -K$.*

The lemma below is used in the proof of Proposition 4.1 and of Lemma 4.1.

Lemma 4.8 (Lemme 11.1 of Chapitre III)**.** *Let $\Omega \subset \mathbb{R}^n$ have piecewise bounded boundary with curvature bounded from below by the number K.Then, it holds*

$$\|u\|_{H^2(\Omega)} \leq c(\|\Delta u\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}) \quad \forall u \in H^1_0(\Omega) \cap H^2(\Omega)$$

where the constant c depends only on Ω.

4.8.2 Results from [59]

Theorem 3.2.1.2 of [59], that states the regularity of the solution of a Poisson problem in a convex domain, is reported below. We use this theorem in the proof of Lemma 4.1.

Theorem 4.5 (Theorem 3.2.1.2)**.** *Let $\Omega \subset \mathbb{R}^n$ be a convex, bounded and open subset of $\mathbb{R}^n$. Then for each $f \in L^2(\Omega)$, there exist a unique $u \in H^2(\Omega)$ solution of*

$$\begin{cases} -\Delta u = f & \text{in} \quad \Omega, \\ u = 0 & \text{on} \ \ \partial\Omega. \end{cases} \tag{4.8.1}$$

4.8.3 Results from [49]

We first report the definition of weak solution for the heat equation, as it is defined in [49], then we recall the theorem that we employ in the proof of Theorem 4.1. In the following, as usually, we denote by $H^{-1}(\Omega)$ the dual space of $H^1_0(\Omega)$.

Definition 4.2 (Weak solution of the parabolic problem of in Chapter 7)**.** *A function $u \in H^1_0(\Omega) \otimes L^2(0,T)$ with $\partial_t u \in H^{-1}(\Omega) \otimes L^2(0,T)$ is a weak solution of the problem*

$$\begin{cases} \partial_t u - \Delta u \ = \ f & \text{in} \quad \ \Omega \times (0,T), \\ u \ = \ 0 & \text{on} \ \ \partial\Omega \times (0,T), \\ u \ = \ 0 & \text{in} \quad \ \Omega \times \{0\}. \end{cases} \tag{4.8.2}$$

with $f(\cdot,t) \in L^2(\Omega)$ for almost every $t \in [0,T]$ if the following two conditions are satisfied

- $\int_\Omega \partial_t u(\cdot,t)v \ \mathrm{d}\Omega + \int_\Omega \nabla u(\cdot,t)\nabla v \ \mathrm{d}\Omega = \int_\Omega f(\cdot,t)v \ \mathrm{d}\Omega \ \forall v \in H^1_0(\Omega)$ *and almost every* $t \in [0,T]$,
- $u(\cdot,0) = 0$ *almost everywhere in* Ω.

We now report the first statement of [49, Theorem 5, Chapter 7].

Theorem 4.6 (Theorem 5 in Chapter 7)**.** *Let $f \in L^2(\Omega)\otimes L^2(0,T)$. Suppose that $u \in H^1_0(\Omega)\otimes L^2(0,T)$ with $\partial_t u \in H^{-1}(\Omega)\otimes L^2(0,T)$ is the weak solution of* (4.8.2) *as defined in Definition 4.2. Then we have that $u \in (H^2(\Omega) \otimes L^2(0,T)) \cap (H^1_0(\Omega) \otimes L^\infty(0,T)) \cap (L^2(\Omega) \otimes H^1(0,T))$. Moreover it holds*

$$\|u\|_{H^2(\Omega)\otimes L^2(0,T)} + \|u\|_{L^2(\Omega)\otimes H^1(0,T)} + \|u\|_{H^1_0(\Omega)\otimes L^\infty(0,T)} \leq C\|f\|_{L^2(\Omega\times(0,T))},$$

where C is a constant that depends only on Ω and T.

4.8.4 Results from [23]

We report below the lemma of [23], in the simplified setting of this chapter, that allows us to prove Lemma 4.3.

Lemma 4.9 (Lemme 3.3)**.** *Let $u \in ((H^1_0(\Omega) \cap H^2(\Omega)) \otimes L^2(0,T)) \cap (L^2(\Omega) \otimes H^1(0,T))$. Then the function $t \mapsto \frac{1}{2}\int_\Omega |\nabla u(\cdot,t)|^2 \ \mathrm{d}\Omega$ is absolutely continue in $[0,T]$.*

Chapter 5

The heat equation: Galerkin method

In this chapter we focus again on the heat equation but, differently than the previous chapter, we consider a space-time Galerkin isogeometric discretization. We focus in particular on the plain Galerkin space-time method, whose well-posedness has been studied, for finite element discretizations and for the heat equation, in the recent papers [103] and [104]. As already seen in the previous chapter, a key issue, when adopting smooth approximation in space and time, is the design of an efficient solver for the space-time system, which is inherently global. This is indeed the aim of the work presented in this chapter. Exploiting the tensor product structure of the spline basis and assuming that the spatial domain does not change with time, the linear system has the structure

$$\gamma W_t \otimes M_s + \nu M_t \otimes K_s, \tag{5.0.1}$$

where W_t is given by the Galerkin discretization of the time derivative, K_s is given by the discretization of the Laplacian in the spatial variables, M_t and M_s are "mass matrices" in time and space, respectively, and $\gamma, \nu > 0$ are constants of the problem. Adopting an iterative solver, we do not need to form the matrix (5.0.1) (observe that the cost of formation of the matrices in (5.0.1) is comparable to the cost of forming a steady-state diffusion matrix) but there is the need of an efficient preconditioning strategy. The main contribution of the work presented in this chapter of the thesis is the construction of a preconditioner for (5.0.1) generalizing the classical fast diagonalization method [77]. Indeed the fast diagonalization, as other fast solvers for (5.0.1), would require the eigendecomposition of the pencil (W_t, M_t) which is numerically unstable. We circumvent this difficulty by introducing an ad-hoc factorization of the time matrices which allows to design a solver conceptually similar to the fast diagonalization method. The computational cost of the setup of the resulting preconditioner is $O(N_{dof})$ floating-point operations (FLOPs) while its application is $O(N_{dof}^{1+1/d})$ FLOPs, where d is the number of spatial dimensions and N_{dof} denotes the total number of degrees-of-freedom (assuming, for simplicity, to have the same number of degrees-of-freedom in time and in each spatial direction). Our numerical benchmarks show that the computing time (serial and single-core execution) is close to optimality, that is, proportional to N_{dof}. The preconditioner is robust with respect to the polynomial degree. Furthermore, our approach is optimal in terms of memory requirement: denoting by N_s the total number of degrees-of-freedom in space, the storage cost is $O(p^d N_s + N_{dof})$. We also remark that global space-time methods in principle facilitate the full parallelization of the solver, see [42, 54, 72]. A comparison between the L^2 least-squares variational formulation and the related preconditioner of the previous chapter and the Galerkin formulation and the related preconditioner of the current chapter is carried out in the numerical experiments, showing the higher efficiency of the plain Galerkin method.

The outline of this chapter is as follows. In Section 5.1 we recall the notations for the univariate and multivariate spline spaces while in Section 5.2 we present the isogeometric spaces we will use for the discretization. The model problem is introduced in Section 5.3 and in Section 5.4 we define the preconditioner and we discuss its application. We present the numerical results assessing the performance of the proposed preconditioner in Section 5.5. Finally, in the last section we draw some conclusions and we highlight some future research directions.

5.1 Notations and main assumptions for the spline spaces

In this section we summarize the notations and the assumptions for the univariate and multivariate spline spaces that we employ in the rest of the chapter.

We consider functions that depend on d spatial variables and the time variable. Therefore we introduce $d+1$ univariate knot vectors $\Xi_l := \{\xi_{l,1} \leq \cdots \leq \xi_{l,m_l+p_l+1}\}$ for $l = 1, \ldots, d$ and $\Xi_t := \{\xi_{t,1} \leq \cdots \leq \xi_{t,m_t+p_t+1}\}$. For the definition of univariate B-splines in each parametric direction we always refer to Section 2.1.1. Let $\boldsymbol{p}$ be the vector that contains the degree indexes, i.e. $\boldsymbol{p} := (\boldsymbol{p_s}, p_t)$, where $\boldsymbol{p_s} := (p_s, \ldots, p_s) \in \mathbb{N}^d$, that is, we assume to have the same polynomial degree in all spatial directions. Let h_s be the maximal meshsize in all spatial knot vectors and let h_t be the meshsize of the time knot vector. We assume that the following quasi-uniformity condition on the knot vectors holds.

Assumption 5.1. *There exists $0 < \alpha \leq 1$, independent of h_s and h_t, such that each non-empty knot span $(\xi_{l,i}, \xi_{l,i+1})$ fulfils $\alpha h_s \leq \xi_{l,i+1} - \xi_{l,i} \leq h_s$ for $1 \leq l \leq d$ and each non-empty knot-span $(\xi_{t,i}, \xi_{t,i+1})$ fulfils $\alpha h_t \leq \xi_{t,i+1} - \xi_{t,i} \leq h_t$.*

We then introduce the univariate spline spaces $\widehat{\mathcal{S}}^{p_s}_{h_s}$ and $\widehat{\mathcal{S}}^{p_t}_{h_t}$. The multivariate B-spline is defined as

$$\widehat{B}_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\eta}, \tau) := \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta}) \widehat{b}_{i_t,p_t}(\tau),$$

where

$$\widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta}) := \widehat{b}_{i_1,p_s}(\eta_1) \ldots \widehat{b}_{i_d,p_s}(\eta_d), \tag{5.1.1}$$

$\boldsymbol{i_s} := (i_1, \ldots, i_d)$, $\boldsymbol{i} := (\boldsymbol{i_s}, i_t)$ and $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_d)$. The corresponding spline space is denoted as

$$\widehat{\mathcal{S}}^{\boldsymbol{p}}_{h} := \operatorname{span}\left\{ \widehat{B}_{\boldsymbol{i},\boldsymbol{p}} \;\middle|\; i_k = 1, ..., m_k \text{ for } k = 1, \ldots, d; i_t = 1, \ldots, m_t \right\},$$

where $h := \max\{h_s, h_t\}$. We have $\widehat{\mathcal{S}}^{\boldsymbol{p}}_{h} = \widehat{\mathcal{S}}^{\boldsymbol{p_s}}_{h_s} \otimes \widehat{\mathcal{S}}^{p_t}_{h_t} = \widehat{\mathcal{S}}^{p_s}_{h_s} \otimes \cdots \otimes \widehat{\mathcal{S}}^{p_s}_{h_s} \otimes \widehat{\mathcal{S}}^{p_t}_{h_t}$, where $\widehat{\mathcal{S}}^{\boldsymbol{p_s}}_{h_s} := \operatorname{span}\left\{ \widehat{B}_{\boldsymbol{i_s},\boldsymbol{p_s}}(\boldsymbol{\eta}) \;\middle|\; i_k = 1, ..., m_k \text{ for } k = 1, \ldots, d \right\}$.

We need the following assumptions on the regularity of the splines.

Assumption 5.2. *We assume that $p_t, p_s \geq 1$ and that $\widehat{\mathcal{S}}^{\boldsymbol{p_s}}_{h_s} \subset C^0(\widehat{\Omega})$ and $\widehat{\mathcal{S}}^{p_t}_{h_t} \subset C^0((0,1))$.*

5.2 Isogeometric spaces

Even if similar to the previous chapter, in order to have a self-contained part, we define the isogeometric space-time spaces that we will need in the following.

The space-time computational domain that we consider is $\Omega \times (0, T)$, where $\Omega \subset \mathbb{R}^d$ and $T > 0$ is the final time. We make the following assumptions.

Assumption 5.3. *We assume that Ω is parametrized by $\boldsymbol{F} : \widehat{\Omega} \to \Omega$, with $\boldsymbol{F} \in \left[\widehat{\mathcal{S}}^{\boldsymbol{p_s}}_{h_s}\right]^d$.*

Assumption 5.4. *We assume that $\boldsymbol{F}^{-1}$ has piecewise bounded derivatives of any order.*

We define $\boldsymbol{x} = (x_1, \dots, x_d) := \boldsymbol{F}(\boldsymbol{\eta})$ and $t := T\tau$. Then space-time domain is given by the parametrization $\boldsymbol{G} : \widehat{\Omega} \times (0,1) \to \Omega \times (0,T)$, such that $\boldsymbol{G}(\boldsymbol{\eta}, \tau) := (\boldsymbol{F}(\boldsymbol{\eta}), T\tau) = (\boldsymbol{x}, t)$.

We introduce the spline space with initial and boundary conditions, in parametric coordinates, as

$$\widehat{\mathcal{X}}_h := \left\{ \widehat{v}_h \in \widehat{\mathcal{S}}_h^{\boldsymbol{p}} \;\middle|\; \widehat{v}_h = 0 \text{ on } \partial\widehat{\Omega} \times (0,1) \text{ and } \widehat{v}_h = 0 \text{ on } \widehat{\Omega} \times \{0\} \right\}.$$

We also have that $\widehat{\mathcal{X}}_h = \widehat{\mathcal{X}}_{h_s} \otimes \widehat{\mathcal{X}}_{h_t}$, where

$$\widehat{\mathcal{X}}_{h_s} := \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_s}^{\boldsymbol{p_s}} \;\middle|\; \widehat{w}_h = 0 \text{ on } \partial\widehat{\Omega} \right\} = \operatorname{span}\left\{ \widehat{b}_{i_1,p_s} \dots \widehat{b}_{i_d,p_s} \;\middle|\; i_k = 2, \dots, m_k - 1;\; k = 1, \dots, d \right\},$$

$$\widehat{\mathcal{X}}_{h_t} := \left\{ \widehat{w}_h \in \widehat{\mathcal{S}}_{h_t}^{p_t} \;\middle|\; \widehat{w}_h(0) = 0 \right\} = \operatorname{span}\left\{ \widehat{b}_{i_t,p_t} \;\middle|\; i_t = 2, \dots, m_t \right\}.$$

By introducing a colexicographical reordering of the basis functions, we can write

$$\widehat{\mathcal{X}}_{h_s} = \operatorname{span}\left\{ \widehat{b}_{i_1,p_s} \dots \widehat{b}_{i_d,p_s} \;\middle|\; i_k = 1, \dots, n_{s,k};\; k = 1, \dots, d \right\} = \operatorname{span}\left\{ \widehat{B}_{i,\boldsymbol{p_s}} \;\middle|\; i = 1, \dots, N_s \right\},$$

$$\widehat{\mathcal{X}}_{h_t} = \operatorname{span}\left\{ \widehat{b}_{i,p_t} \;\middle|\; i = 1, \dots, n_t \right\}$$

and then

$$\widehat{\mathcal{X}}_h = \operatorname{span}\left\{ \widehat{B}_{i,\boldsymbol{p}} \;\middle|\; i = 1, \dots, N_{dof} \right\}, \tag{5.2.2}$$

where $n_t := m_t - 1$, $n_{s,k} := m_k - 2$, $N_s := \prod_{k=1}^d n_{s,k}$, $N_{dof} := N_s n_t$.

Finally, the isogeometric space we consider is the isoparametric push-forward of (5.2.2) through the geometric map $\boldsymbol{G}$, i.e.

$$\mathcal{X}_h := \operatorname{span}\left\{ B_{i,\boldsymbol{p}} := \widehat{B}_{i,\boldsymbol{p}} \circ \boldsymbol{G}^{-1} \;\middle|\; i = 1, \dots, N_{dof} \right\}. \tag{5.2.3}$$

We also have that $\mathcal{X}_h = \mathcal{X}_{h_s} \otimes \mathcal{X}_{h_t}$, where

$$\mathcal{X}_{h_s} := \operatorname{span}\left\{ B_{i,\boldsymbol{p_s}} := \widehat{B}_{i,\boldsymbol{p_s}} \circ \boldsymbol{F}^{-1} \;\middle|\; i = 1, \dots, N_s \right\},\ \mathcal{X}_{h_t} := \operatorname{span}\left\{ b_{i,p_t} := \widehat{b}_{i,p_t}(\cdot/T) \;\middle|\; i = 1, \dots, n_t \right\}.$$

5.3 Parabolic model problem and its discretization

5.3.1 Space-time variational formulation

Our model problem is the heat equation: we look for a solution u such that

$$\left\{ \begin{array}{rcll} \gamma \partial_t u - \nabla \cdot (\nu \nabla u) & = & f & \text{in} \quad \Omega \times (0,T), \\ u & = & 0 & \text{on} \quad \partial\Omega \times [0,T], \\ u & = & u_0 & \text{in} \quad \Omega \times \{0\}, \end{array} \right. \tag{5.3.1}$$

where $\Omega \subset \mathbb{R}^d$, T is the final time, $\gamma > 0$ is the heat capacity constant and $\nu > 0$ is the thermal conductivity constant. We assume that $f \in L^2(0,T; H^{-1}(\Omega))$ and that $u_0 \in L^2(\Omega)$. This last assumption guarantees the existence of a lifting $\bar{u}_0$ of u_0 such that $\bar{u}_0 \in L^2(0,T; H_0^1(\Omega)) \cap H^1(0,T; H^{-1}(\Omega))$, see [49]. We introduce the Hilbert spaces

$$\mathcal{X} := \left\{ v \in L^2(0,T; H_0^1(\Omega)) \cap H^1(0,T; H^{-1}(\Omega)) \mid v(\boldsymbol{x}, 0) = 0 \right\} \quad \text{and} \quad \mathcal{Y} := L^2(0,T; H_0^1(\Omega)),$$

endowed with the following norms

$$\|v\|^2_{\mathcal{X}} := \frac{\gamma^2}{\nu}\|\partial_t v\|^2_{L^2(0,T;H^{-1}(\Omega))} + \nu\|v\|^2_{L^2(0,T;H^1_0(\Omega))} \quad \text{and} \quad \|v\|^2_{\mathcal{Y}} := \nu\|v\|^2_{L^2(0,T;H^1_0(\Omega))},$$

respectively. The variational formulation of (5.3.1) reads:

$$\text{Find } \bar{u} \in \mathcal{X} \text{ such that } \mathcal{A}(\bar{u}, v) = \mathcal{F}_0(v) := \mathcal{F}(v) - \mathcal{A}(\bar{u}_0, v) \quad \forall v \in \mathcal{Y}, \tag{5.3.2}$$

where the bilinear form $\mathcal{A}(\cdot,\cdot)$ and the linear form $\mathcal{F}(\cdot)$ are defined $\forall v \in \mathcal{X}$ and $\forall w \in \mathcal{Y}$ as

$$\mathcal{A}(v, w) := \int_0^T \int_\Omega (\gamma\partial_t v\, w + \nu\nabla v \cdot \nabla w)\, \mathrm{d}\Omega\, \mathrm{d}t \quad \text{and} \quad \mathcal{F}(w) := \int_0^T \int_\Omega f\, w\, \mathrm{d}\Omega\, \mathrm{d}t.$$

Then, the solution u of (5.3.1) is $u := \bar{u} + \bar{u_0}$. The well-posedness of the variational formulation above is a classical result, see for example [103].

5.3.2 Space-time Galerkin approximation

Let $\mathcal{X}_h \subset \mathcal{X}$ be the isogeometric space defined in (5.2.3). We consider the following Galerkin method for (5.3.2):

$$\text{Find } u_h \in \mathcal{X}_h \text{ such that } \mathcal{A}(u_h, v_h) = \mathcal{F}_0(v_h) \quad \forall v_h \in \mathcal{X}_h. \tag{5.3.3}$$

Following [103], let $N_h : L^2(0, T; H^{-1}(\Omega)) \to \mathcal{X}_h$ be the discrete Newton potential operator: given $\phi \in L^2(0, T; H^{-1}(\Omega))$ then $N_h\phi \in \mathcal{X}_h$ fulfills

$$\int_0^T \int_\Omega \nu\nabla(N_h\phi) \cdot \nabla v_h\, \mathrm{d}\Omega\, \mathrm{d}t = \gamma \int_0^T \int_\Omega \phi\, v_h\, \mathrm{d}\Omega\, \mathrm{d}t \quad \forall v_h \in \mathcal{X}_h.$$

Thus, we define the norm in $\mathcal{X}_h$ as

$$\|w\|^2_{\mathcal{X}_h} := \nu\|N_h(\partial_t w)\|^2_{L^2(0,T;H^1_0(\Omega))} + \nu\|w\|^2_{L^2(0,T;H^1_0(\Omega))}.$$

The stability and the well-posedness of the formulation (5.3.3) are guaranteed by a straightforward extension to IgA of [103, Equation (2.7)], [103, Theorem 3.1] and [103, Theorem 3.2].

Proposition 5.1. *It holds*

$$\mathcal{A}(w, v) \le \sqrt{2}\|w\|_{\mathcal{X}}\|v\|_{\mathcal{Y}} \quad \forall w \in \mathcal{X} \text{ and } \forall v \in \mathcal{Y},$$

and

$$\|w_h\|_{\mathcal{X}_h} \le 2\sqrt{2} \sup_{v_h \in \mathcal{X}_h} \frac{\mathcal{A}(w_h, v_h)}{\|v_h\|_{\mathcal{Y}}} \quad \forall w_h \in \mathcal{X}_h.$$

Theorem 5.1. *There exists a unique solution $u_h \in \mathcal{X}_h$ to the discrete problem* (5.3.3). *Moreover, it holds*

$$\|u - u_h\|_{\mathcal{X}_h} \le 5 \inf_{w_h \in \mathcal{X}_h} \|u - w_h\|_{\mathcal{X}},$$

where $u \in \mathcal{X}$ is the solution of (5.3.2).

We have then the following a-priori estimate for h-refinement.

Theorem 5.2. *Let q be an integer such that $1 \le q \le \min\{p_s, p_t\} + 1$. If $u \in \mathcal{X} \cap H^1(0, T; H^q(\Omega)) \cap H^q(0, T; H^1(\Omega))$ is the solution of* (5.3.2) *and $u_h \in \mathcal{X}_h$ is the solution* *of* (5.3.3)*, then it holds*

$$\|u - u_h\|_{\mathcal{X}_h} \leq C\sqrt{\frac{\gamma^2}{\nu} + \nu}\left(h_s^{q-1}\|u\|_{H^1(0,T;H^q(\Omega))} + h_t^{q-1}\|u\|_{H^q(0,T;H^1(\Omega))}\right) \tag{5.3.4}$$

where C is independent of h_s, h_t, γ, ν and u.

Proof. We use the approximation estimates of the isogeometric spaces from [11]. We report here only the main steps, since the proof is similar to the one of Proposition 4.4 in Chapter 4 (see also [80, Proposition 4]).

Let $u \in \mathcal{X} \cap H^1(0,T;H^q(\Omega)) \cap H^q(0,T;H^1(\Omega))$. Let $\Pi_h u$ be a suitable projection of u in $\mathcal{X}_h$, based on the construction of [11]. We have the a-priori bounds

$$\begin{aligned}\|\partial_t(u - \Pi_h u)\|_{L^2(0,T;H^{-1}(\Omega))} &\leq C_1\|\partial_t(u - \Pi_h u)\|_{L^2(0,T;L^2(\Omega))} \\ &\leq C_2\left(h_s^{q-1}\|u\|_{H^1(0,T;H^{q-1}(\Omega))} + h_t^{q-1}\|u\|_{H^q(0,T;L^2(\Omega))}\right),\end{aligned}$$

and also

$$\|u - \Pi_h u\|_{L^2(0,T;H_0^1(\Omega))} \leq C_3\left(h_s^{q-1}\|u\|_{L^2(0,T;H^q(\Omega))} + h_t^{q-1}\|u\|_{H^{q-1}(0,T;H^1(\Omega))}\right).$$

Therefore, we get

$$\|u - \Pi_h u\|_{\mathcal{X}}^2 \leq C_4\left(\frac{\gamma^2}{\nu} + \nu\right)\left[h_s^{2(q-1)}\|u\|^2_{H^1(0,T;H^q(\Omega))} + h_t^{2(q-1)}\|u\|^2_{H^q(0,T;H^1(\Omega))}\right]$$

which gives (5.3.4) thanks to Theorem 5.1. The constants C_1, C_2, C_3 and C_4 above are independent of h_s, h_t, γ, ν and u. □

Remark 5.1. *The constants in the estimates of Proposition 5.1 and of Theorem 5.1 can be improved by considering a different norm in the functional space $\mathcal{X}$, i.e. by choosing $|||v|||^2 := \|v\|^2_{\mathcal{X}} + \gamma\|v(T)\|^2_{L^2(\Omega)}$, as remarked in [103, 104].*

5.3.3 Discrete system

The linear system associated to (5.3.3) is

$$\mathbf{A}\mathbf{u} = \mathbf{b}, \tag{5.3.5}$$

where $[\mathbf{A}]_{i,j} = \mathcal{A}(B_{j,\boldsymbol{p}}, B_{i,\boldsymbol{p}})$ and $[\mathbf{b}]_i = \mathcal{F}_0(B_{i,\boldsymbol{p}})$. The tensor-product structure of the isogeometric space (5.2.3) allows to write the system matrix $\mathbf{A}$ as sum of Kronecker products of matrices as

$$\mathbf{A} = \gamma W_t \otimes M_s + \nu M_t \otimes K_s, \tag{5.3.6}$$

where for $i, j = 1, \dots, n_t$

$$[W_t]_{i,j} = \int_0^T b'_{j,p_t}(t)\, b_{i,p_t}(t)\,\mathrm{d}t \quad \text{and} \quad [M_t]_{i,j} = \int_0^T b_{i,p_t}(t)\, b_{j,p_t}(t)\,\mathrm{d}t, \tag{5.3.7a}$$

while for $i, j = 1, \dots, N_s$

$$[K_s]_{i,j} = \int_\Omega \nabla B_{i,\boldsymbol{p}_s}(\boldsymbol{x}) \cdot \nabla B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\,\mathrm{d}\Omega \quad \text{and} \quad [M_s]_{i,j} = \int_\Omega B_{i,\boldsymbol{p}_s}(\boldsymbol{x})\, B_{j,\boldsymbol{p}_s}(\boldsymbol{x})\,\mathrm{d}\Omega. \tag{5.3.7b}$$

5.4 Preconditioner definition and application

We introduce, for the system (5.3.5), the preconditioner

$$[\widehat{\mathbf{A}}]_{i,j} := \widehat{\mathcal{A}}(\widehat{B}_{j,\boldsymbol{p}}, \widehat{B}_{i,\boldsymbol{p}}),$$

where

$$\widehat{\mathcal{A}}(\widehat{v},\widehat{w}) := \int_0^1 \int_{\widehat{\Omega}} (\gamma \partial_t \widehat{v}\, \widehat{w} + \nu \nabla \widehat{v} \cdot \nabla \widehat{w})\, \mathrm{d}\widehat{\Omega}\, \mathrm{d}\tau \quad \forall \widehat{v}, \widehat{w} \in \widehat{\mathcal{X}}_h.$$

We have again

$$\widehat{\mathbf{A}} = \gamma \widehat{W_t} \otimes \widehat{M_s} + \nu \widehat{M_t} \otimes \widehat{K}_s, \tag{5.4.1}$$

where $\widehat{W_t}$, $\widehat{K}_t$, $\widehat{K}_s$ and $\widehat{M_s}$ are the equivalent of (5.3.7a) and (5.3.7b), respectively, in the parametric domain, i.e. for $i,j = 1,\dots,n_t$

$$[\widehat{W_t}]_{i,j} = \int_0^1 \widehat{b}'_{j,p_t}(\tau)\, \widehat{b}_{i,p_t}(\tau)\, \mathrm{d}\tau \quad \text{and} \quad [\widehat{M_t}]_{i,j} = \int_0^1 \widehat{b}_{i,p_t}(\tau) \widehat{b}_{j,p_t}(t)\, \mathrm{d}\tau, \tag{5.4.2a}$$

while for $i,j = 1,\dots,N_s$

$$[\widehat{K}_s]_{i,j} = \int_{\widehat{\Omega}} \nabla \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{\eta}) \cdot \nabla \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{\eta})\, \mathrm{d}\widehat{\Omega} \quad \text{and} \quad [\widehat{M_s}]_{i,j} = \int_{\widehat{\Omega}} \widehat{B}_{i,\boldsymbol{p}_s}(\boldsymbol{\eta})\ \widehat{B}_{j,\boldsymbol{p}_s}(\boldsymbol{\eta})\, \mathrm{d}\widehat{\Omega}. \tag{5.4.2b}$$

Thanks to the definition of the spline spaces in the parametric domain (5.1.1), the spatial matrices (5.4.2b) have the following structure

$$\widehat{K}_s = \sum_{k=1}^{d} \widehat{M_d} \otimes \dots \otimes \widehat{M}_{k+1} \otimes \widehat{K}_k \otimes \widehat{M}_{k-1} \otimes \dots \otimes \widehat{M_1} \quad \text{and} \quad \widehat{M_s} = \widehat{M_d} \otimes \dots \otimes \widehat{M_1}, \tag{5.4.3}$$

where for $k = 1,\dots,d$ and for $i,j = 1,\dots,n_{s,k}$

$$[\widehat{K}_k]_{i,j} := \int_0^1 \widehat{b}'_{i,p_s}(\eta_k) \widehat{b}'_{j,p_s}(\eta_k) \mathrm{d}\eta_k \quad \text{and} \quad [\widehat{M_k}]_{i,j} := \int_0^1 \widehat{b}_{i,p_s}(\eta_k) \widehat{b}_{j,p_s}(\eta_k) \mathrm{d}\eta_k.$$

The efficient application of the proposed preconditioner, that is, the solution of a system with matrix $\widehat{\mathbf{A}}$, should exploit the structure highlighted above. When the pencils $(\widehat{W_t}, \widehat{M_t})$, $(\widehat{K}_1, \widehat{M_1}), \dots, (\widehat{K}_d, \widehat{M_d})$ admit a stable generalized eigendecomposition, a possible approach is the fast diagonalization (FD) method, see [40] and [77] for details. We will see in Section 5.4.1 that the spatial pencils $(\widehat{K}_1, \widehat{M_1}), \dots, (\widehat{K}_d, \widehat{M_d})$ admit a stable diagonalization, but this is not the case of $(\widehat{W_t}, \widehat{M_t})$, that needs a special treatment as explained in Section 5.4.2.

5.4.1 Stable factorization of the pencils $(\widehat{K}_i, \widehat{M_i})$ $i = 1,\dots,d$

The spatial stiffness and mass matrices $\widehat{K}_i$ and $\widehat{M_i}$ are symmetric and positive definite. Thus, the pencils $(\widehat{K}_i, \widehat{M_i})$ for $i = 1,\dots,d$ admit the generalized eigendecomposition

$$\widehat{K}_i U_i = \widehat{M_i} U_i \Lambda_i \tag{5.4.4}$$

where the matrices U_i contain in each column the $\widehat{M_i}$-orthonormal generalized eigenvectors, and Λ_i are diagonal matrices whose entries contain the generalized eigenvalues. Therefore we have for $i = 1,\dots,d$ the factorizations

$$U_i^T \widehat{K}_i U_i = \Lambda_i \quad \text{and} \quad U_i^T \widehat{M_i} U_i = \mathbb{I}_{n_{s,i}}, \tag{5.4.5}$$

where $\mathbb{I}_{n_{s,i}}$ denotes the identity matrix of dimension $n_{s,i} \times n_{s,i}$. The stability of the decomposition (5.4.5) is expressed by the condition number of the eigenvector matrix. In particular $U_i^T \widehat{M}_i U_i = \mathbb{I}_{n_{s,i}}$ implies that

$$\kappa_{\widehat{M}_i}(U_i) := \|U_i\|_{\widehat{M}_i} \|U_i^{-1}\|_{\widehat{M}_i} = 1,$$

where $\|\cdot\|_{\widehat{M}_i}$ is the norm induced by the vector norm $\|\boldsymbol{v}\|_{\widehat{M}_i} := \left(\boldsymbol{v}^T \widehat{M}_i \boldsymbol{v}\right)^{1/2}$ for $\boldsymbol{v} \in \mathbb{R}^{n_{s,i}}$. Furthermore,

$$\kappa_2(U_i) := \|U_i\|_2 \|U_i^{-1}\|_2 = \sqrt{\kappa_2(\widehat{M}_i)},$$

where $\|\cdot\|_2$ is the norm induced by the Euclidean vector norm. The condition number $\kappa_2(\widehat{M}_i)$ has been studied in [53] and it does not depend on n_{sub} but it depends on the polynomial degree. Indeed, we report in Table 5.1 the behavior of $\kappa_2(U_i)$ that exhibits a dependence only on the degree p_s, but stays moderately low for all low polynomial degrees that are in the range of interest.

n_{sub}	$p_s = 2$	$p_s = 3$	$p_s = 4$	$p_s = 5$	$p_s = 6$	$p_s = 7$	$p_s = 8$
32	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$
64	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$
128	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$
256	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$
512	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$
1024	$2.7 \cdot 10^0$	$4.5 \cdot 10^0$	$7.6 \cdot 10^0$	$1.3 \cdot 10^1$	$2.1 \cdot 10^1$	$3.5 \cdot 10^1$	$5.7 \cdot 10^1$

TABLE 5.1: $\kappa_2(U_i)$ for different polynomial degree p_s and number of dyadic subdivisions n_{sub}.

5.4.2 Stable factorization of the pencil $(\widehat{W}_t, \widehat{M}_t)$

Numerical instability of the eigendecomposition

While $\widehat{M}_t$ is symmetric, $\widehat{W}_t$ is neither symmetric nor skew-symmetric. Indeed

$$[\widehat{W}_t]_{i,j} + [\widehat{W}_t]_{j,i} = \int_0^1 \widehat{b}'_{j,p_t}(t)\, \widehat{b}_{i,p_t}(\tau)\; \mathrm{d}\tau + \int_0^1 \widehat{b}'_{i,p_t}(\tau)\, \widehat{b}_{j,p_t}(\tau)\; \mathrm{d}\tau = \widehat{b}_{i,p_t}(1)\; \widehat{b}_{j,p_t}(1) \qquad (5.4.6)$$

where $\widehat{b}_{i,p_t}(1)\; \widehat{b}_{j,p_t}(1)$ vanishes for all $i = 1, \ldots, n_t - 1$ or $j = 1, \ldots, n_t - 1$. A numerical computation of the generalized eigendecomposition of the pencil $(\widehat{W}_t, \widehat{M}_t)$, that is

$$\widehat{W}_t U = \widehat{M}_t U \Lambda_t, \qquad (5.4.7)$$

where Λ_t is the diagonal matrix of the generalized complex eigenvalues and U is the complex matrix whose columns are the generalized eigenvectors (with normalization w.r.t. the $\|\cdot\|_{\widehat{M}_t}$-norm), reveals that the eigenvectors are far from $\widehat{M}_t$-orthogonality, i.e. the matrix $U^* \widehat{M}_t U$ is not diagonal. As seen in Table 5.2 and Table 5.3, the numerically computed condition numbers $\kappa_2(U)$ and $\kappa_{\widehat{M}_t}(U)$ are large and grow exponentially with respect to the degree p_t and the level of mesh refinement, in contrast to the spatial case (see Table 5.1).

n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$	$p_t = 6$	$p_t = 7$	$p_t = 8$
32	$8.9 \cdot 10^2$	$3.0 \cdot 10^4$	$5.0 \cdot 10^4$	$3.4 \cdot 10^5$	$3.1 \cdot 10^6$	$4.2 \cdot 10^7$	$7.0 \cdot 10^8$
64	$4.4 \cdot 10^3$	$2.6 \cdot 10^5$	$5.0 \cdot 10^5$	$5.4 \cdot 10^6$	$8.9 \cdot 10^7$	$3.1 \cdot 10^9$	$2.0 \cdot 10^{10}$
128	$2.3 \cdot 10^4$	$1.2 \cdot 10^6$	$5.8 \cdot 10^6$	$1.0 \cdot 10^8$	$3.0 \cdot 10^9$	$6.4 \cdot 10^{11}$	$1.3 \cdot 10^{12}$
256	$1.2 \cdot 10^5$	$9.4 \cdot 10^6$	$7.6 \cdot 10^7$	$2.1 \cdot 10^9$	$1.2 \cdot 10^{11}$	$1.2 \cdot 10^{13}$	$2.1 \cdot 10^{13}$
512	$7.0 \cdot 10^5$	$8.3 \cdot 10^7$	$1.1 \cdot 10^9$	$4.9 \cdot 10^{10}$	$4.5 \cdot 10^{12}$	$3.6 \cdot 10^{13}$	$4.9 \cdot 10^{12}$
1024	$4.1 \cdot 10^6$	$8.0 \cdot 10^8$	$1.9 \cdot 10^{10}$	$1.3 \cdot 10^{12}$	$9.6 \cdot 10^{12}$	$1.4 \cdot 10^{12}$	$5.6 \cdot 10^{12}$

TABLE 5.2: $\kappa_2(U)$ for different degree p_t and number of dyadic subdivisions n_{sub}.

n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$	$p_t = 6$	$p_t = 7$	$p_t = 8$
32	$1.8 \cdot 10^3$	$7.7 \cdot 10^4$	$1.3 \cdot 10^5$	$6.3 \cdot 10^5$	$4.1 \cdot 10^6$	$3.6 \cdot 10^7$	$4.3 \cdot 10^8$
64	$9.9 \cdot 10^3$	$7.9 \cdot 10^5$	$1.5 \cdot 10^6$	$1.3 \cdot 10^7$	$1.5 \cdot 10^8$	$3.6 \cdot 10^9$	$1.4 \cdot 10^{10}$
128	$5.5 \cdot 10^4$	$4.0 \cdot 10^6$	$2.1 \cdot 10^7$	$3.1 \cdot 10^8$	$6.8 \cdot 10^9$	$1.1 \cdot 10^{12}$	$1.1 \cdot 10^{12}$
256	$3.2 \cdot 10^5$	$3.3 \cdot 10^7$	$3.3 \cdot 10^8$	$8.6 \cdot 10^9$	$3.5 \cdot 10^{11}$	$2.3 \cdot 10^{13}$	$2.8 \cdot 10^{13}$
512	$1.8 \cdot 10^6$	$3.1 \cdot 10^8$	$5.6 \cdot 10^9$	$2.5 \cdot 10^{11}$	$1.9 \cdot 10^{13}$	$1.6 \cdot 10^{14}$	$9.3 \cdot 10^{12}$
1024	$1.1 \cdot 10^7$	$3.1 \cdot 10^9$	$1.0 \cdot 10^{11}$	$8.6 \cdot 10^{12}$	$5.6 \cdot 10^{13}$	$6.0 \cdot 10^{12}$	$6.1 \cdot 10^{12}$

TABLE 5.3: $\kappa_{\widehat{M}_t}(U)$ for different degree p_t and number of dyadic subdivisions n_{sub}.

These tests clearly indicate a numerical instability when computing the generalized eigendecomposition of $(\widehat{W}_t, \widehat{M}_t)$. Similar instabilities have also been highlighted in [63].

Construction of the stable factorization

The analysis above motivates the search of a different but stable factorization of the pencil $(\widehat{W}_t, \widehat{M}_t)$. We look now for a factorization of the form

$$\widehat{W}_t U_t = \widehat{M}_t U_t \Delta_t, \tag{5.4.8}$$

where Δ_t is a complex matrix with non-zero entries allowed on the diagonal, on the last row and on the last column only. We also require that U_t fulfils the orthogonality condition

$$U_t^* \widehat{M}_t U_t = \mathbb{I}_{n_t}. \tag{5.4.9}$$

From (5.4.8)–(5.4.9) we then obtain the factorizations

$$U_t^* \widehat{W}_t U_t = \Delta_t \quad \text{and} \quad U_t^* \widehat{M}_t U_t = \mathbb{I}_{n_t}. \tag{5.4.10}$$

With this aim, we look for U_t as follows:

$$U_t := \begin{bmatrix} \mathring{U}_t & \mathbf{k} \\ \mathbf{0}^T & \rho \end{bmatrix} \tag{5.4.11}$$

where $\mathring{U}_t \in \mathbb{C}^{(n_t-1)\times(n_t-1)}$, $\mathbf{k} \in \mathbb{C}^{n_t-1}$, $\rho \in \mathbb{C}$ and where $\mathbf{0} \in \mathbb{R}^{n_t-1}$ denotes the null vector. In order to guarantee the non-singularity of U_t, we further impose $\rho \neq 0$. Accordingly, we split the time matrices $\widehat{W_t}$ and $\widehat{M_t}$ as

$$\widehat{W_t} = \begin{bmatrix} \mathring{W}_t & \mathbf{w} \\ -\mathbf{w}^T & \omega \end{bmatrix} \quad \text{and} \quad \widehat{M_t} = \begin{bmatrix} \mathring{M}_t & \mathbf{m} \\ \mathbf{m}^T & \mu \end{bmatrix}, \tag{5.4.12}$$

where we have defined

$$\omega := [\widehat{W_t}]_{n_t,n_t}, \qquad \mu := [\widehat{M_t}]_{n_t,n_t},$$

$$[\mathbf{w}]_i = [\widehat{W_t}]_{i,n_t} \quad \text{and} \quad [\mathbf{m}]_i = [\widehat{M_t}]_{i,n_t} \quad \text{for} \quad i = 1, \dots, n_t - 1,$$

$$[\mathring{W}_t]_{i,j} = [\widehat{W_t}]_{i,j} \quad \text{and} \quad [\mathring{M}_t]_{i,j} = [\widehat{M_t}]_{i,j} \quad \text{for} \quad i, j = 1, \dots, n_t - 1. \tag{5.4.13}$$

Recalling (5.4.6), we observe that $\mathring{W}_t$ is skew-symmetric and, since $\mathring{M}_t$ is symmetric, we can write the eigendecomposition of the pencils $(\mathring{W}_t, \mathring{M}_t)$:

$$\mathring{W}_t \mathring{U}_t = \mathring{M}_t \mathring{U}_t \mathring{\Lambda}_t \quad \text{with} \quad \mathring{U}_t^* \mathring{M}_t \mathring{U}_t = \mathbb{I}_{n_t-1}, \tag{5.4.14}$$

where $\mathring{U}_t$ contains the complex generalized eigenvectors and $\mathring{\Lambda}_t$ is the diagonal matrix of the generalized eigenvalues, that are pairs of complex conjugate pure imaginary numbers plus, eventually, the eigenvalue zero. From (5.4.11)–(5.4.12), it follows

$$U_t^* \widehat{M_t} U_t = \begin{bmatrix} \mathbb{I}_{n_t-1} & \mathring{U}_t^* \mathring{M}_t \mathbf{k} + \mathring{U}_t^* \mathbf{m} \rho \\ \mathbf{k}^* \mathring{M}_t \mathring{U}_t + \rho^* \mathbf{m}^T \mathring{U}_t & [\mathbf{k}^* \rho^*] \, \widehat{M_t} \begin{bmatrix} \mathbf{k} \\ \rho \end{bmatrix} \end{bmatrix},$$

where for the top-left block we have used (5.4.14).

The orthogonality condition in (5.4.9) holds if and only if $\mathbf{k}$ and ρ fulfil the two conditions:

$$\mathring{U}_t^* \mathring{M}_t \mathbf{k} + \mathring{U}_t^* \mathbf{m} \rho = \mathbf{0}, \tag{5.4.15a}$$

$$[\mathbf{k}^* \rho^*] \, \widehat{M_t} \begin{bmatrix} \mathbf{k} \\ \rho \end{bmatrix} = 1. \tag{5.4.15b}$$

In order to calculate $\mathbf{k}$ and ρ, we first find $\mathbf{v} \in \mathbb{C}^{n_t-1}$ such that

$$\mathring{M}_t \mathbf{v} = -\mathbf{m}; \tag{5.4.16}$$

then normalize the vector $\begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}$ w.r.t. the $\| \cdot \|_{\widehat{M_t}}$-norm to get

$$\begin{bmatrix} \mathbf{k} \\ \rho \end{bmatrix} := \frac{\begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix}}{\left([\mathbf{v}^* \; 1] \widehat{M_t} \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} \right)^{\frac{1}{2}}}$$

that fulfils (5.4.15a)–(5.4.15b). Finally, we get (5.4.8) by defining

$$\Delta_t := U_t^* \widehat{W_t} U_t = \begin{bmatrix} \mathring{\Lambda}_t & \mathbf{l} \\ -\mathbf{l}^* & \sigma \end{bmatrix}, \tag{5.4.17}$$

where $\mathbf{l} := \mathring{U}_t^* \begin{bmatrix} \mathring{W}_t & \mathbf{w} \end{bmatrix} \begin{bmatrix} \mathbf{k} \\ \rho \end{bmatrix}$ and $\sigma := [\mathbf{k}^* \rho^*]\, \widehat{W}_t \begin{bmatrix} \mathbf{k} \\ \rho \end{bmatrix}$. Note that matrix (5.4.17) has an arrowhead structure.

To assess the stability of the new decomposition (5.4.10), we compute the condition numbers $\kappa_2(U_t)$ for dyadically refined uniform knot spans and different degrees. Thanks to (5.4.9), we have $\kappa_2(U_t) = \sqrt{\kappa_2(\widehat{M}_t)}$. The results, reported in Table 5.4, show that the condition numbers $\kappa_2(U_t)$ are uniformly bounded w.r.t. the mesh refinement, they grow with respect to the polynomial degree but they are moderately small for all the degrees of interest. As a consequence of (5.4.9), we also have that $\kappa_{\widehat{M}_t}(U_t) = 1$. We conclude that the factorization (5.4.10) for the time pencil $(\widehat{W}_t, \widehat{M}_t)$ is stable.

n_{sub}	$p_t = 2$	$p_t = 3$	$p_t = 4$	$p_t = 5$	$p_t = 6$	$p_t = 7$	$p_t = 8$
32	$3.2 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$
64	$3.3 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$
128	$3.3 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$
256	$3.3 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$
512	$3.3 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$
1024	$3.3 \cdot 10^0$	$5.2 \cdot 10^0$	$8.3 \cdot 10^0$	$1.3 \cdot 10^1$	$2.2 \cdot 10^1$	$3.6 \cdot 10^1$	$5.9 \cdot 10^1$

TABLE 5.4: $\kappa_2(U_t)$ for different degree p_t and number of dyadic subdivisions n_{sub}.

5.4.3 Preconditioner application by the extended fast diagonalization method

The application of the preconditioner involves the solution of the linear system

$$\widehat{\mathbf{A}}\mathbf{s} = \mathbf{r}, \tag{5.4.18}$$

where $\widehat{\mathbf{A}}$ has the structure (5.4.1). We are able to efficiently solve system (5.4.18) by extending the fast diagonalization method. The starting points, that are involved in the setup of the preconditioner, are the following ones:

- for the pencils $(\widehat{K}_i, \widehat{M}_i)$ for $i = 1, \dots, d$ we have the factorizations (5.4.5);
- for the pencil $(\widehat{W}_t, \widehat{M}_t)$ we have the factorization (5.4.10).

Then, by defining $U_s := U_d \otimes \cdots \otimes U_1$ and $\Lambda_s := \sum_{i=1}^{d} \mathbb{I}_{n_{s,d}} \otimes \cdots \otimes \mathbb{I}_{n_{s,i+1}} \otimes \Lambda_i \otimes \mathbb{I}_{n_{s,i-1}} \otimes \cdots \otimes \mathbb{I}_{n_{s,1}}$, we have for the matrix $\widehat{\mathbf{A}}$ the factorization

$$\widehat{\mathbf{A}} = \left(U_t^* \otimes U_s^T\right)^{-1} \left(\gamma \Delta_t \otimes \mathbb{I}_{N_s} + \nu \mathbb{I}_{n_t} \otimes \Lambda_s\right) \left(U_t \otimes U_s\right)^{-1}. \tag{5.4.19}$$

Note that the second factor in (5.4.19) has the block-arrowhead structure

$$\gamma \Delta_t \otimes \mathbb{I}_{N_s} + \nu \mathbb{I}_{n_t} \otimes \Lambda_s = \begin{bmatrix} H_1 & & & B_1 \\ & \ddots & & \vdots \\ & & H_{n_t-1} & B_{n_t-1} \\ -B_1^* & \dots & -B_{n_t-1}^* & H_{n_t} \end{bmatrix} \tag{5.4.20}$$

where H_i and $B_i$are diagonal matrices defined as

$$H_i := \gamma [\Lambda_t]_{ii} \mathbb{I}_{N_s} + \nu \Lambda_s \quad \text{and} \quad B_i := \gamma [\mathbf{l}]_i \mathbb{I}_{N_s} \quad \text{for} \quad i = 1, \dots, n_t - 1,$$

$$H_{n_t} := \gamma\sigma\mathbb{I}_{N_s} + \nu\Lambda_s.$$

The matrix (5.4.20) has the following easy-to-invert block LU decomposition

$$\gamma\Delta_t \otimes \mathbb{I}_{N_s} + \nu\mathbb{I}_{n_t} \otimes \Lambda_s = \begin{bmatrix} \mathbb{I}_{N_s} & & & \\ & \ddots & & \\ & & \mathbb{I}_{N_s} & \\ -B_1^* H_1^{-1} & \dots & -B_{n_t-1}^* H_{n_t-1}^{-1} & \mathbb{I}_{N_s} \end{bmatrix} \begin{bmatrix} H_1 & & & B_1 \\ & \ddots & & \vdots \\ & & H_{n_t-1} & B_{n_t-1} \\ & & & S \end{bmatrix} \tag{5.4.21}$$

where $S := H_{n_t} + \sum_{i=1}^{n_t-1} B_i^* H_i^{-1} B_i$ is a diagonal matrix.

Summarising, the solution of (5.4.18) can be computed by the following algorithm.

Algorithm 4 Extended FD

```
1: Setup: Compute the factorizations (5.4.5) and (5.4.10).
2: Application: Compute $\widetilde{\mathbf{s}} = (U_t^* \otimes U_s^T)\mathbf{s}$.
3:              Compute $\widetilde{\mathbf{q}} = (\gamma\Delta_t \otimes \mathbb{I}_{N_s} + \nu\mathbb{I}_{n_t} \otimes \Lambda_s)^{-1}\widetilde{\mathbf{s}}$.
4:              Compute $\mathbf{r} = (U_t \otimes U_s)\,\widetilde{\mathbf{q}}$.
```

5.4.4 Inclusion of the geometry and coefficient information in the preconditioner

The preconditioner (5.4.1) does not incorporate any information on the geometry parametrization $\boldsymbol{G}$. Thus, the performance of $\widehat{\mathbf{A}}$ may depend on the geometry map: we see this trend in the numerical tests of Section 5.5 and, in particular, in the upper tables of Table 5.5 and Table 5.7. However, we can generalize (5.4.1) by including in the time matrices $\widehat{W_t}$ and $\widehat{M_t}$ and in the univariate spatial matrices $\widehat{K_i}, \widehat{M_i}$ for $i = 1, \dots, d$ a suitable approximation of $\boldsymbol{G}$, without increasing the asymptotic computational cost. A similar approach has been used also in [81] for the Stokes problem (see also Section 3.5.3) and in [80] for a least-squares formulation of the heat equation (see also Section 4.4.3). We briefly give an overview of this strategy.

Referring to Section 5.2 for the notation of the basis functions, we rewrite the entries of the system matrix (5.3.5) in the parametric domain as

$$\begin{aligned} [\mathbf{A}]_{i,j} &= \mathcal{A}(B_{j,\boldsymbol{p}}, B_{i,\boldsymbol{p}}) \\ &= \gamma \int_0^1 \int_{\widehat{\Omega}} \tfrac{1}{T}\partial_\tau \widehat{B}_{j,\boldsymbol{p}} \widehat{B}_{i,\boldsymbol{p}} |\det(J_{\boldsymbol{G}})| \, \mathrm{d}\widehat{\Omega} \, \mathrm{d}\tau + \int_0^1 \int_{\widehat{\Omega}} \nu (\nabla \widehat{B}_{j,\boldsymbol{p}})^T J_{\boldsymbol{G}}^{-1} J_{\boldsymbol{G}}^{-T} \nabla \widehat{B}_{i,\boldsymbol{p}} |\det(J_{\boldsymbol{G}})| \, \mathrm{d}\widehat{\Omega} \, \mathrm{d}\tau \\ &= \int_0^1 \int_{\widehat{\Omega}} \left[(\nabla \widehat{B}_{j,\boldsymbol{p}})^T \;\; \partial_\tau \widehat{B}_{j,\boldsymbol{p}} \right] \boldsymbol{C} \left[(\nabla \widehat{B}_{i,\boldsymbol{p}})^T \;\; \widehat{B}_{i,\boldsymbol{p}} \right]^T \mathrm{d}\widehat{\Omega} \, \mathrm{d}\tau, \end{aligned} \tag{5.4.22}$$

where

$$\boldsymbol{C}(\boldsymbol{\eta}, \tau) := \begin{bmatrix} \nu J_{\boldsymbol{F}}^{-1} J_{\boldsymbol{F}}^{-T} |\det(J_{\boldsymbol{F}})| T & \\ & \gamma |\det(J_{\boldsymbol{F}})| \end{bmatrix}$$

and where we used that $B_{i,\boldsymbol{p}} = \widehat{B}_{i,\boldsymbol{p}} \circ \boldsymbol{G}^{-1}$, $B_{j,\boldsymbol{p}} = \widehat{B}_{j,\boldsymbol{p}} \circ \boldsymbol{G}^{-1}$ and $|\det(J_{\boldsymbol{G}})| = T|\det(J_{\boldsymbol{F}})|$. The construction of the preconditioner is based on the following approximation of the diagonal entries only of $\boldsymbol{C}$:

$$[\boldsymbol{C}(\boldsymbol{\eta}, \tau)]_{k,k} \approx [\widetilde{\boldsymbol{C}}(\boldsymbol{\eta}, \tau)]_{k,k} := \varphi_1(\eta_1) \dots \varphi_{k-1}(\eta_{k-1}) \Phi_k(\eta_k) \varphi_{k+1}(\eta_{k+1}) \dots \varphi_d(\eta_d) \varphi_{d+1}(\tau) \qquad k = 1, \dots, d, \tag{5.4.23a}$$

$$[\boldsymbol{C}(\boldsymbol{\eta},\tau)]_{d+1,d+1} \approx [\widetilde{\boldsymbol{C}}(\boldsymbol{\eta},\tau)]_{d+1,d+1} := \varphi_1(\eta_1)\dots\varphi_d(\eta_d)\Phi_{d+1}(\tau). \tag{5.4.23b}$$

We interpolate the functions $\widetilde{\boldsymbol{C}}_{k,k}$ in (5.4.23) by piecewise constants in each element and we build the univariate factors φ_k and Φ_k by using the separation of variables algorithm detailed in the Appendix A. The computational cost of the approximation above is proportional to the number of elements, that, when using smooth B-splines, is almost equal to N_{dof} and it is independent of p_s and p_t and thus negligible in the whole iterative strategy.

Then we define

$$[\widetilde{\mathbf{A}}]_{i,j} := \int_0^1\int_{\widehat{\Omega}} \left[(\nabla\widehat{B}_{j,\boldsymbol{p}})^T \quad \partial_\tau\widehat{B}_{j,\boldsymbol{p}}\right] \widetilde{\boldsymbol{C}} \left[(\nabla\widehat{B}_{i,\boldsymbol{p}})^T \quad \widehat{B}_{i,\boldsymbol{p}}\right]^T \mathrm{d}\widehat{\Omega}\,\mathrm{d}\tau.$$

The previous matrix maintains the same Kronecker structure as (5.4.1):

$$\widetilde{\mathbf{A}} = \widetilde{W_t}\otimes\widetilde{M_s} + \widetilde{M_t}\otimes\widetilde{K}_s, \tag{5.4.24}$$

where for $i,j = 1,\dots,n_t$

$$[\widetilde{W_t}]_{i,j} := \int_0^1 \Phi_{d+1}(\tau)\widehat{b}'_{j,p_t}(\tau)\,\widehat{b}_{i,p_t}(\tau)\,\mathrm{d}\tau \quad \text{and} \quad [\widetilde{M_t}]_{i,j} := \int_0^1 \varphi_{d+1}(\tau)\widehat{b}_{i,p_t}(\tau)\widehat{b}_{j,p_t}(t)\,\mathrm{d}\tau,$$

$$\widetilde{K}_s := \sum_{k=1}^{d} \widetilde{M_d}\otimes\dots\otimes\widetilde{M_{k+1}}\otimes\widetilde{K}_k\otimes\widetilde{M_{k-1}}\otimes\dots\otimes\widetilde{M_1}, \qquad \widetilde{M_s} := \widetilde{M_d}\otimes\dots\otimes\widetilde{M_1},$$

and where for $k = 1,\dots,d$ and for $i,j = 1,\dots,n_{s,k}$ we define

$$[\widetilde{K}_k]_{i,j} := \int_0^1 \Phi_k(\eta_k)\widehat{b}'_{i,p_s}(\eta_k)\widehat{b}'_{j,p_s}(\eta_k)\mathrm{d}\eta_k \quad \text{and} \quad [\widetilde{M_k}]_{i,j} := \int_0^1 \varphi_k(\eta_k)\widehat{b}_{i,p_s}(\eta_k)\widehat{b}_{j,p_s}(\eta_k)\mathrm{d}\eta_k.$$

We remark that the application of (5.4.24) can still be performed by Algorithm 4. Finally, we apply a diagonal scaling on $\widetilde{\mathbf{A}}$ and we define the preconditioner as

$$\widehat{\mathbf{A}}^{\boldsymbol{G}} := \mathbf{D}^{\frac{1}{2}}\widetilde{\mathbf{A}}\mathbf{D}^{\frac{1}{2}} \tag{5.4.25}$$

where $[\mathbf{D}]_{i,i} := [\mathbf{A}]_{i,i}/[\widetilde{\mathbf{A}}]_{i,i}$.

Remark 5.2. *We remark that when γ and ν do not depend on time, it holds*

$$W_t = \widehat{W_t} \quad \textit{and} \quad M_t = T\widehat{M_t}$$

and we can set explicitly $\widetilde{W_t} = W_t$ and $\widetilde{M_t} = M_t$. However, as in our numerical tests we consider a more general framework in which γ and ν depend on time, we have presented the more general strategy above, that allows to incorporate in $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ possible non-constant coefficients.

5.4.5 Computational cost and memory consumption of the linear solver

The linear system (5.3.5) is neither positive definite nor symmetric, and we choose GMRES as linear solver. In GMRES, the orthogonalization of the basis of the Krylov subspace makes the computational cost nonlinear with respect to the number of iterations. However, as long as this number is not too high, at each iteration the two dominant costs are the application of the preconditioning strategy and the computation of the residual. We assume, for simplicity that for $i = 1,\dots,d$ the matrices $\widehat{K}_i$, $\widehat{M_i}$ and $\widetilde{K}_i$, $\widetilde{M_i}$ have dimensions $n_s\times n_s$ and that the matrices

$\widehat{W}_t$, $\widehat{M}_t$ and $\widetilde{W}_t$, $\widetilde{M}_t$ have dimensions $n_t \times n_t$. Thus the total number of degrees-of-freedom is $N_{dof} = N_s n_t = n_s^d n_t$.

The setup of $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{A}}^{\mathbf{G}}$ includes the operations performed in Step 1 of Algorithm 4, i.e. d spatial eigendecompositions, that have a total cost of $O(dn_s^3)$ FLOPs, and the factorization of the time matrices. The computational cost of the latter is the sum of the cost of the eigendecomposition (5.4.14) and of the cost of the solution of the linear system (5.4.16), yielding a cost of $O(n_t^3)$ FLOPs. Then, the total cost of the space and time factorizations is $O(dn_s^3 + n_t^3)$ FLOPs. Note that, if $n_t = O(n_s)$., this cost is optimal for $d = 2$ and negligible for $d = 3$. The setup cost of $\widehat{\mathbf{A}}^{\mathbf{G}}$ includes also the the construction of the diagonal matrix $\mathbf{D}$, that has a negligible cost, and the computation of the $2(d + 1)$ approximations $\varphi_1, \dots, \varphi_{d+1}$ and $\Phi_1, \dots, \Phi_{d+1}$ in (5.4.23), that, as mentioned in Section 5.4.4, has the optimal cost of $O(N_{dof})$ FLOPs. We remark that the setup of the preconditioners has to be performed only once, since the matrices involved do not change during the iterative procedure.

The application of the preconditioner is performed by Steps 2-4 of Algorithm 4. Exploiting (2.2.5), Step 2 and Step 4 costs $4(dn_s^{d+1}n_t + n_t^2 n_s^d) = 4N_{dof}(dn_s + n_t)$ FLOPs. The use of the block LU decomposition (5.4.21) makes the cost for Step 3 equal to $O(N_{dof})$ FLOPs.

In conclusion, the total cost of Algorithm 4 is $4N_{dof}(dn_s + n_t) + O(N_{dof})$ FLOPs. The non-optimal dominant cost of Step 2 and Step 4 is determined by the dense matrix-matrix products. However, these operations are usually implemented on modern computers in a very efficient way. For this reason, in our numerical tests, the overall serial computational time grows almost as $O(N_{dof})$, see Figure 5.3 in Section 5.5.

The other dominant computational cost in a GMRES iteration is the cost of the residual computation, that is the multiplication of the matrix $\mathbf{A}$ with a vector. This multiplication is done by exploiting the special structure (5.3.6), that allows a matrix-free approach and the use of formula (2.2.5). Note in particular that we do not need to compute and to store the whole matrix $\mathbf{A}$, but only its time and spatial factors. Since the time matrices M_t and W_t are banded with a band of width $2p_t + 1$ and the spatial matrices K_s and M_s have roughly $N_s(2p_s + 1)^d$ nonzero entries, we have that the computational cost of a single matrix-vector product is $6N_{dof}[(2p_s + 1)^d + 2p_t + 1] \approx 6N_{dof}(2p + 1)^d = O(N_{dof}p^d)$ FLOPs, if we assume $p = p_s \approx p_t$. The numerical experiments reported in Table 5.6 of Section 5.5 show that the dominant cost in the iterative solver is represented by the residual computation. This is a typical behaviour of the FD-based preconditioning strategies, see [80, 81, 93].

We now investigate the memory consumption. For the preconditioner we have to store the eigenvector spatial matrices $U_1, \dots, U_d$, the time matrix U_t and the block-arrowhead matrix (5.4.20). The memory required is roughly

$$n_t^2 + dn_s^2 + 2N_{dof}.$$

For the system matrix, we have to store the time factors M_t and W_t and the spatial factors M_s and K_s. Thus the memory required is roughly

$$2(2p_t + 1)n_t + 2(2p_s + 1)^d N_s \approx 4p_t n_t + 2^{d+1}p_s^d N_s.$$

As for the least-squares case [80], we conclude that, in terms of memory requirement, our approach is very attractive w.r.t. other approaches, e.g. the ones obtained by discretizing in space and in time separately. For example if we assume $d = 3$, $p_t \approx p_s = p$ and $n_t^2 \leq Cp^3 N_s$, then the total memory consumption is $O(p^3 N_s + N_{dof})$, that is equal to the sum of the memory needed to store the Galerkin matrices associated to spatial variables and the memory needed to store the solution of the problem.

We remark that we could avoid storing the factors of $\mathbf{A}$ by using the matrix-free approach of [94]. The memory and the computational cost of the iterative solver would significantly improve, both for the setup and the matrix-vector multiplications. However, we do not pursue this strategy, as it is beyond the scope of this paper.

Remark 5.3. *For a better computational efficiency, we use a real-arithmetic version of Algorithm 4: we replace* $\widetilde{\Lambda_t}$ *in* (5.4.17) *by a block diagonal matrix where each pair of generalized eigenvalues* $i\lambda_j$ *and* $-i\lambda_j$ *is replaced by a diagonal block*

$$\begin{bmatrix} 0 & \lambda_j \\ -\lambda_j & 0 \end{bmatrix}$$

and we set

$$H_j := \begin{bmatrix} \nu\Lambda_s & \gamma\lambda_j\mathbb{I}_{n_s} \\ -\gamma\lambda_j\mathbb{I}_{n_s} & \nu\Lambda_s \end{bmatrix} \qquad \textit{and} \qquad B_j := \gamma\left[[\boldsymbol{l}]_{2(j-1)+1}\mathbb{I}_{N_s}, \quad [\boldsymbol{l}]_{2(j-1)+2}\mathbb{I}_{N_s}\right]^T.$$

Note that the computational cost of Step 3 in Algorithm 4 does not change, as we have

$$H_j^{-1} := \begin{bmatrix} \frac{1}{\nu}\Lambda_s^{-1} - \frac{\gamma^2}{\nu^2}\lambda_j^2\Lambda_s^{-1}\left(\nu\Lambda_s + \frac{\gamma^2}{\nu}\lambda_j^2\Lambda_s^{-1}\right)^{-1}\Lambda_s^{-1} & -\frac{\gamma}{\nu}\lambda_j\Lambda_s^{-1}\left(\nu\Lambda_s + \frac{\gamma^2}{\nu}\lambda_j^2\Lambda_s^{-1}\right)^{-1} \\ \frac{\gamma}{\nu}\lambda_j\left(\nu\Lambda_s + \frac{\gamma^2}{\nu}\lambda_j^2\Lambda_s^{-1}\right)^{-1}\Lambda_s^{-1} & \left(\nu\Lambda_s + \frac{\gamma^2}{\nu}\lambda_j^2\Lambda_s^{-1}\right)^{-1} \end{bmatrix}.$$

5.5 Numerical results

In this section we first present the numerical experiments that assess the convergence behavior of the Galerkin approximation and then we analyze the performance of the preconditioners.

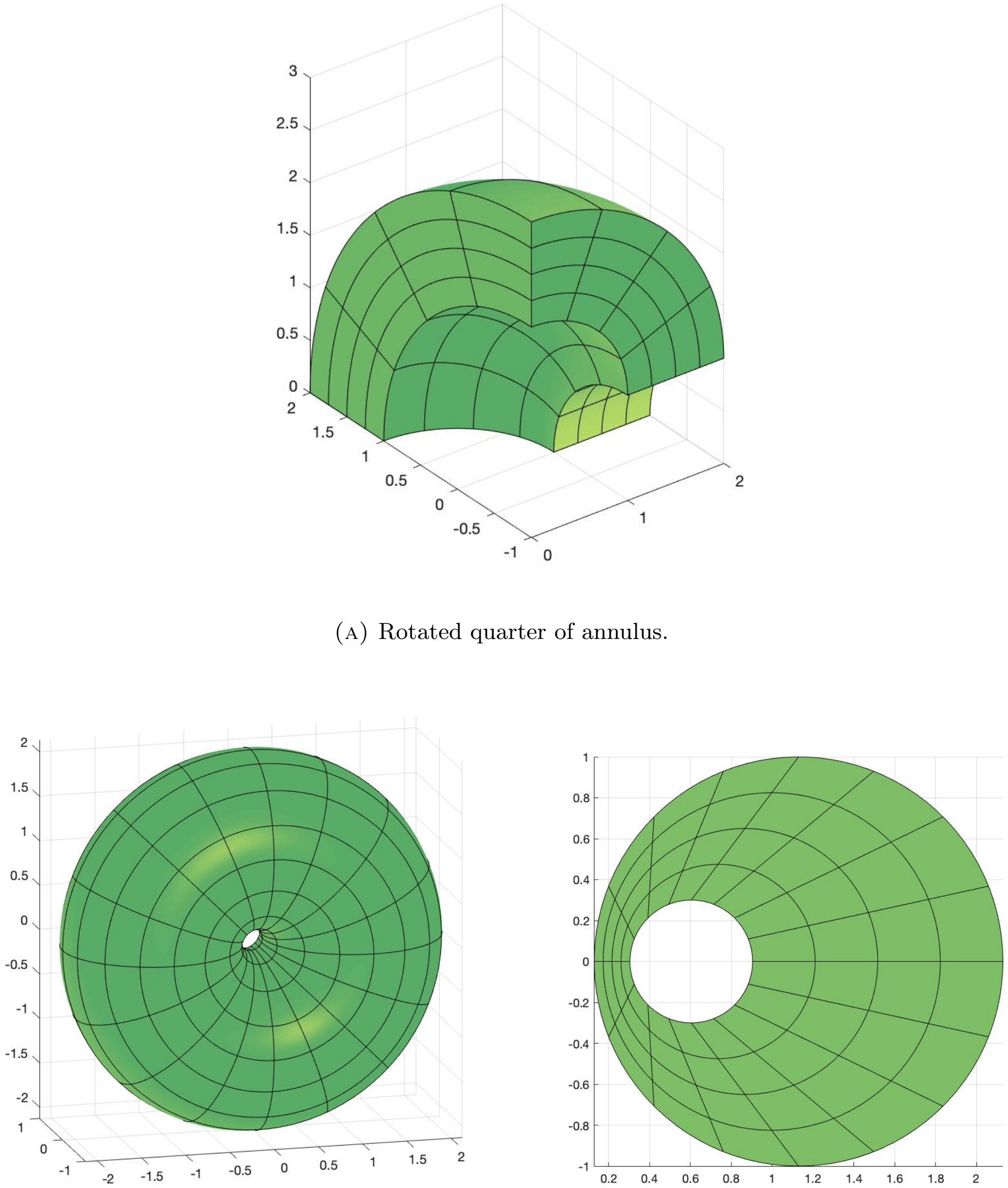
The tests are performed with Matlab R2015a and GeoPDEs toolbox [111]. We consider only sequential executions and we force the use of a single computational thread in a Intel Core i7-5820K processor, running at 3.30 GHz and with 64 GB of RAM. We use the `eig` Matlab function to compute the generalized eigendecompositions present in Step 1 of Algorithm 4, while Tensorlab toolbox [102] is employed to perform the multiplications with Kronecker matrices occurring in Step 2 and Step 4. The linear system is solved by GMRES without restart, with tolerance equal to 10^{-8} and with the null vector as initial guess in all tests.

We consider the same mesh-size in space and in time, by setting $h_s = h_t =: h$, and we denote the number of subdivisions in each parametric direction by n_{sub}. We use splines of maximal continuity allowed and of the same degree both in space and in time, i.e. we set $p_t = p_s =: p$.

5.5.1 Orders of convergence

We consider as spatial computational domain Ω a rotated quarter of annulus, represented in Figure 5.1a: we rotate by $\frac{\pi}{2}$ a quarter of annulus with center in the origin, internal radius 1 and external radius 2 along the axis$\{(x, -1, 0) \mid x \in \mathbb{R}\}$. Dirichlet and initial boundary conditions are set such that $u(x, y, z, t) = -(x^2 + y^2 - 1)(x^2 + y^2 - 4)xy^2 \sin(t)\sin(z)$ is the exact solution with constants $\nu = \gamma = 1$.

In Figure 5.2a we represent the relative errors in $L^2(0, T; H_0^1(\Omega)) \cap H^1(0, T; L^2(\Omega))$ norm, an easily computable upper bound of $\|\cdot\|_{\mathcal{X}_h}$, for polynomial degrees $p = 1, 2, 3, 4, 5$. The rates of convergence are optimal, i.e. of order $O(h^p)$, consistent with the a-priori estimate (5.3.4). Even if this case is not covered by theoretical results, we also compute the relative errors in $L^2(0, T; L^2(\Omega))$ norm: the orders of convergence are still optimal, that is of order $O(h^{p+1})$, as Figure 5.2b shows.

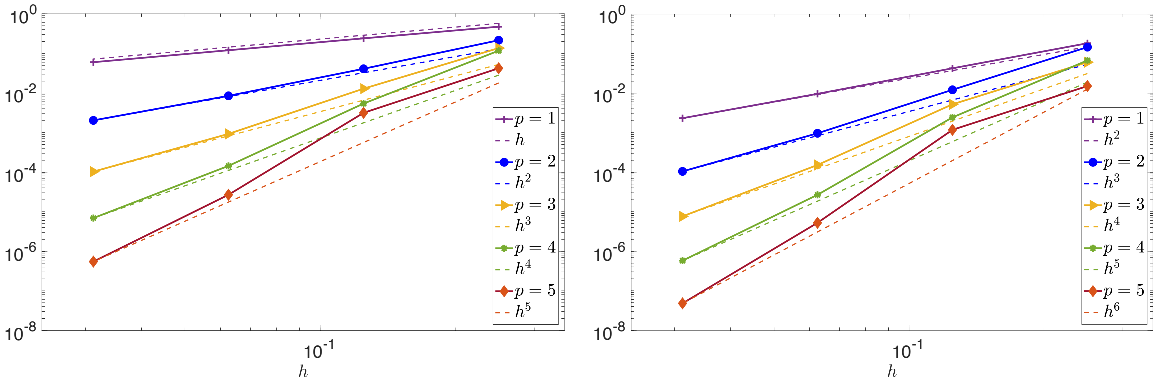


(A) Rotated quarter of annulus.

(B) Hollow torus.

(C) Section of the hollow torus.

FIGURE 5.1: Space-time Galerkin. Computational domains.



(A) $L^2(0,T;H_0^1(\Omega)) \cap H^1(0,T;L^2(\Omega))$ norm relative errors.

(B) $L^2(0,T;L^2(\Omega))$ norm relative errors.

FIGURE 5.2: Space-time Galerkin. Relative errors.

5.5.2 Performance of the preconditioner

In this section we present the performance our preconditioner in two computational domains: a rotated quarter of annulus and a hollow torus.

The symbol "$*$" denotes that the construction of the matrix factors of $\mathbf{A}$ (see (5.3.6)) goes out of memory, while the symbol "$**$" indicates that the dimension of the Krylov subspace is too high and there is not enough memory to store all GMRES iterates. We remark that in all the tables the total solving time of the iterative strategies includes also the setup time of the considered preconditioner.

Rotated quarter of annulus. We consider again as spatial computational domain Ω the rotated quarter of annulus of Figure 5.1a and the same exact solution, initial and boundary data as in Section 5.5.1. We analyze the performance of both $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{A}}^{\boldsymbol{G}}$. The maximum dimension of the Krylov subspace is set equal to 100 for both the preconditioners up to $n_{sub} = 64$. We are able to reach convergence and to perform the tests with $\widehat{\mathbf{A}}^{\boldsymbol{G}}$, $n_{sub} = 128$ and $p = 1, 2, 3$ by setting the maximum Krylov subspace dimension equal to 25. In Table 5.5 we report the number of iterations and the total solving time of GMRES preconditioned with $\widehat{\mathbf{A}}$ (upper table) and $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ (middle table). The non-trivial geometry clearly affects the performance of $\widehat{\mathbf{A}}$, but, when we include some information on the parametrization by using $\widehat{\mathbf{A}}^{\boldsymbol{G}}$, the number of iterations is more than halved and it is stable w.r.t. p and n_{sub}. Moreover, the computational times are one order of magnitude lower for the highest degrees and n_{sub}.

Finally, we analyze with more details the performance of $\widehat{\mathbf{A}}^{\boldsymbol{G}}$. First, we consider the percentage of time spent in the application of $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ in one GMRES iteration. The results, reported in Table 5.6, clearly show that the dominant cost consists of the matrix-vector multiplications, while the application of the preconditioner takes a small percentage of the total computational time, for example less than 10% for polynomial degree 5 and $n_{sub} = 32$ or $n_{sub} = 64$. In Figure 5.3 we report the setup time and the single application time of $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ w.r.t. the number of degrees of freedom. As expected, the setup time is proportional to $O(N_{dof})$. What is more interesting is that the application time grows slower than $O(N_{dof}^{5/4})$, i.e. the FLOPS counting, and it is almost proportional to $O(N_{dof})$: this may be explained by the fact that the memory access is the dominant cost due to the high-efficiency of CPU operations, in our case implemented in Matlab Tensorlab [102].

	$\widehat{\mathbf{A}}$ Iterations / Time				
n_{sub}	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
8	34 / 0.20	37 / 0.21	42 / 0.42	46 / 0.63	50 / 1.13
16	43 / 1.15	46 / 1.65	50 / 3.42	54 / 5.80	57 / 11.87
32	50 / 22.75	53 / 31.10	57 / 54.02	61 / 96.06	64 / 184.84
64	57 / 586.73	60 / 764.26	67 / 1254.81	67 / 1858.55	71 / 3188.51
128	**	**	**	*	*

	$\widehat{\mathbf{A}}^{\boldsymbol{G}}$ Iterations / Time				
n_{sub}	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
8	11 / 0.06	12 / 0.09	12 / 0.11	13 / 0.18	14 / 0.29
16	13 / 0.26	14 / 0.52	14 / 1.18	14 / 1.44	15 / 3.85
32	15 / 4.73	15 / 6.76	15 / 12.67	15 / 21.47	16 / 40.54
64	16 / 107.24	16 / 135.74	18 / 249.27	16 / 370.31	17 / 695.44
128	17 / 2623.57	17 / 3105.76	17 / 5614.10	*	*

Table 5.5: Space-time Galerkin. Revolved quarter domain. Performance of $\widehat{\mathbf{A}}$ (upper table) and $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ (lower table).

n_{sub}	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
8	73.02 %	79.24 %	66.62 %	46.94 %	33.73 %
16	68.10 %	46.13 %	30.06 %	17.63 %	11.27 %
32	53.09 %	33.34 %	20.44 %	13.06 %	8.19 %
64	54.71 %	32.46 %	20.20 %	12.52 %	7.31 %
128	54.12 %	33.53 %	18.89 %	*	*

Table 5.6: Space-time Galerkin. Percentage of computing time of $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ in one GMRES iteration for the rotated quarter domain.

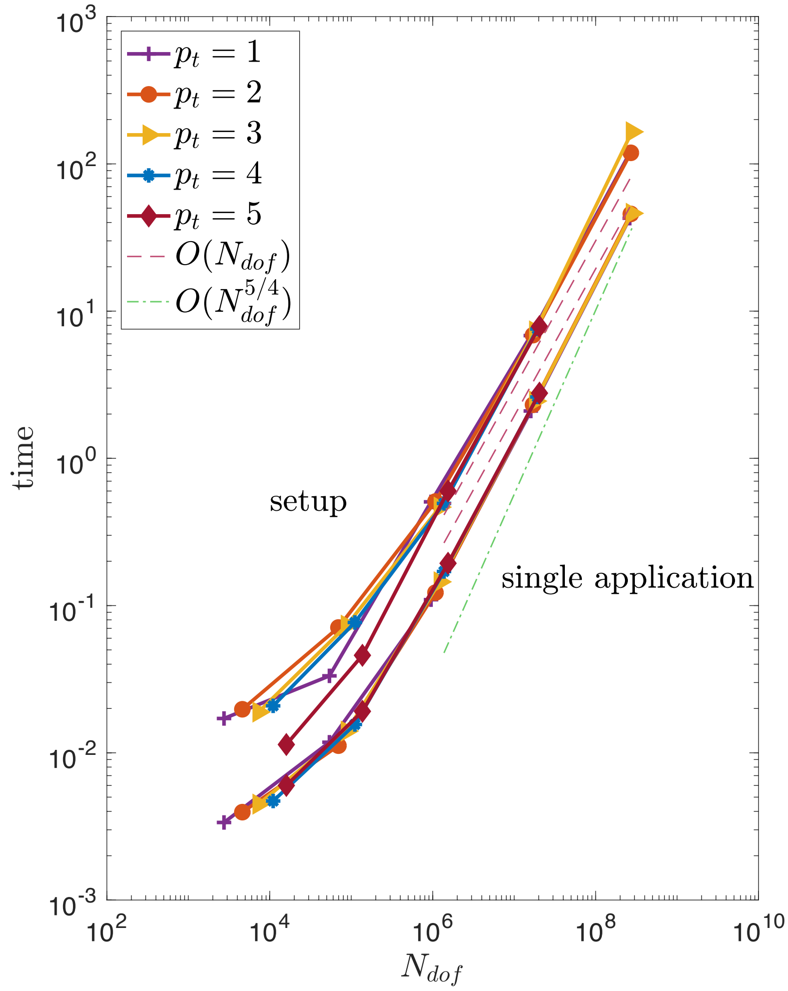


FIGURE 5.3: Space-time Galerkin. Setup time and single application time of $\widehat{\mathbf{A}}^{\boldsymbol{G}}$ in the rotated quarter domain.

	$\widehat{\mathbf{A}}$ Iterations / Time				
n_{sub}	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
8	32 / 0.49	70 / 0.79	101 / 2.02	128 / 5.83	156 / 14.48
16	98 / 5.83	121 / 10.54	149 / 26.13	167 / 57.27	177 / 128.68
32	143 / 122.28	165 / 236.47	177 / 400.79	193 / 746.28	197 / 1230.60
64	165 / 3657.33	168 / 4733.98	175 / 6596.99	179 / 15894.01	184 / 20215.23

	$\widehat{\mathbf{A}}^{\mathbf{G}}$ Iterations / Time				
n_{sub}	$p=1$	$p=2$	$p=3$	$p=4$	$p=5$
8	14 / 0.30	15 / 0.50	19 / 0.71	20 / 1.11	23 / 1.98
16	18 / 0.87	19 / 1.66	21 / 2.79	23 / 5.77	25 / 14.12
32	22 / 8.88	24 / 16.08	25 / 29.66	26 / 61.22	27 / 114.93
64	26 / 207.70	27 / 303.33	28 / 495.29	29 / 1118.44	30 / 1923.20

TABLE 5.7: Space-time Galerkin. Hollow torus domain. Performance of $\widehat{\mathbf{A}}$ (upper table) and $\widehat{\mathbf{A}}^{\mathbf{G}}$ (lower table).

Hollow torus. We consider a torus with a hole (Figure 5.1b) that is obtained by revolving an eccentric annulus (Figure 5.1c) along the y axis. For this problem we consider $\gamma = 1$ and a separable in space and time, non-constant diffusion coefficient ν. Precisely, we choose

$$\nu(x,y,z,t) = \left\{1 + 50\left[1 + \cos\left(\frac{t}{2\pi}\right)\right]\right\}\left\{1 + \frac{99}{2}\left[1 + \frac{1}{\left(1+\frac{x^2}{z^2}\right)^{\frac{1}{2}}}\right]\right\}.$$

The initial data and right-hand side are defined such that $u(x,y,z,t) := \sin(\pi x)\sin(\pi y)\sin(\pi z)\sin(\pi t)$ is the exact solution. In this case, we replace ν in (5.4.1) with its integral mean $\frac{1}{T|\Omega|}\int_0^T\int_\Omega \nu(x,y,z,t)\ \mathrm{d}\Omega\ \mathrm{d}t$. In Table 5.7 we compare the performance of $\widehat{\mathbf{A}}$ (upper table) and $\widehat{\mathbf{A}}^{G}$ (lower table): the inclusion of the information about the geometry parametrization and ν significantly reduces the number of iterations and the computational times.

5.6 Comparison between least-squares and Galerkin approaches

In this section we want to compare the least-squares solver of Chapter 4 with the Galerkin approach of the present chapter. The least-squares formulation requires basis functions with a higher regularity and degree than the Galerkin ones, at least for spatial functions. Indeed, if for the Galerkin formulation we only need C^0 continuity and $p_s \geq 1$ (cfr. Assumption 5.2), for the least-squares method we request C^1 smoothness and $p_s \geq 2$ (cfr. Assumption 4.2). However, this is not a problem in the framework of the k-method.

The least-squares linear system is symmetric and positive definite and thus preconditioned CG linear solver can be employed. On the contrary, the Galerkin linear-system does not have these properties and thus we use preconditioned GMRES. Differently then for CG, with GMRES all the iterates have to be stored and the memory consumption is higher.

The design of the least-squares preconditioner does not require any special techniques and the FD method can be used straightforwardly for its application (see Section 4.4). On the other side, the time matrices in the Galerkin linear system do not have a stable eigendecomposition and we need to build an extension of the FD method (see Section 5.4), that, however, has the same computational cost as the standard FD method.

If we look at practical examples, we can compare the number of iterations and the computational times in the the rotated quarter of annulus test. We focus on the variant of the preconditioners that incorporates some information of the geometry parametrization. First, we see from Figure 4.3 and Figure 5.3, that the setup cost and application cost of the preconditioners are asymptotically the same for both formulations. Then, we consider the middle table of Table 4.2 in Chapter 4 and the lower table of Table 5.5. For the least squares solver, the number of iterations is more than doubled and the computational times are three times higher than the number of iterations and computational times of the Galerkin preconditioner.

To conclude, even if the least-squares formulation yields to a symmetric positive definite linear system and makes easier the use of the FD method in the preconditioning strategy, the Galerkin formulation with the related preconditioner gives better performances in practical experiments.

5.7 Conclusions

In this work we proposed a preconditioner suited for a space-time Galerkin isogeometric discretization of the heat equation. Our preconditioner $\widehat{\mathbf{A}}$ is represented by a suitable sum of Kronecker products of matrices, that makes the computational cost of its construction (setup) and application, as well as the storage cost, very appealing. In particular the application of the preconditioner, inspired by the fast diagonalization technique, exploits an ad-hoc factorization of the time matrices. The preconditioner cost seen in numerical tests, for a serial single core execution, is almost equal to $O(N_{dof})$ and does not depend on the polynomial degree.

At the same time, the storage cost is roughly the same that we would have by discretizing separately in space and in time, if we assume $n_t \leq Cp^d N_s$. Indeed, in this case the memory used for the whole iterative solver is $O(p^d N_s + N_{dof})$.

The coupling with a matrix-free approach [94] will lead to a significant improvement of the solver strategy. Our method is also suited for parallelization and this will be an interesting future direction study.

Chapter 6

Conclusions

In this thesis we developed efficient solvers for linear systems arising in the isogeometric discretization of two kind of problems: the Stokes system and the heat equation.

The basis of all our preconditioning strategies was the fast diagonalization method, a fast solver that has been efficiently employed in IgA in [93] for the construction of a solving method for the Poisson system. The preoconditioning matrix was obtained by discretizing the Poisson equation in the parametric domain and considering constant coefficients. Its application through the FD method revealed to be very fast.

We wanted to employ a similar idea to design preconditioners for the isogeometric discretization of Stokes system and the heat equation: we built the preconditioning matrices by discretizing the considered PDE (or a simplification of it) in the parametric domain in such a way that their application could be efficiently performed by the FD method. We provided spectral estimates that assured the good behavior of our preconditioners when used in a iterative solver, as CG or MINRES. Moreover, we went further this simple idea by creating a strategy, based on a separation of variables algorithm, that allowed to incorporate in the basic versions of our preconditioners some information on the geometry parametrization and the coefficients of the PDE. The overall asymptotic computational cost, which, in practical applications, is proportional to the number of degrees of freedom, was not increased.

The first problem we considered is the Stokes stationary system, discretized either with isogeometric Taylor-Hood or Raviart-Thomas elements. We proposed three kind of preconditioners for the resulting saddle-point linear system: block diagonal, block triangular and constrained. Theoretical results were supported by numerical experiments, that also demonstrated the better performances of our preconditioners with respect to the more classical Incomplete Cholesky based one.

The second PDE that we studied was the heat equation. We focused on space-time discretizations, that allowed to exploit the high regularity and continuity of isogeometric basis functions. We considered two kind of formulations. The first one was based on the least-squares principle and provided a linear system that was symmetric and positive definite. We proved a-priori error estimates that guaranteed the good convergence of the method. We proposed preconditioners whose application could be straightforwardly and efficiently done with the FD method. Numerical tests confirmed the efficiency of the proposed preconditioning strategies and the superiority of our approach with respect to a classical Incomplete Cholesky preconditioner. The other formulation yielded to a plain Galerkin space-time method. The straightforward use of the FD method for the application of the designed preconditioners was not possible, in that case. We circumvented this problem, by introducing an ad-hoc factorization of the matrices that allowed us to develop a solver that was conceptually similar to the FD method and had the same computational cost. We also provided comparisons of the performances of the two formulations and related preconditioners.

The FD method was a key ingredient in the development of all of our preconditioning strategies. We believe that it has the potential to be part of efficient solvers designed for other kind of PDEs or in non-overlapping domain decomposition area. Moreover, the combination with matrix-free approaches would lead to a significant improvement of the performances. Furthermore, the parallelization of the solver, especially for space-time methods, could be an important future development.

Appendix A

Separation of variables algorithm

The basic version of our preconditioners is built by discretizing the PDE or a simplification of it in the parametric domain $\widehat{\Omega}$ and considering constant coefficients. Numerical tests of Section 3.6, Section 4.5 and Section 5.5 confirm that the geometry parametrization clearly affects the performance of the preconditioning strategy. To overcome this issue, we have proposed an improved version of each preconditioner that incorporates in the univariate factors coefficients containing some information on the geometry and on the PDE without losing the tensor-product structure.

An important part of this process is the approximation of N multivariate functions $c_k(\boldsymbol{\eta})$ for $k = 1, \ldots, N$ by the product of univariate factors as

$$c_k(\boldsymbol{\eta}) \approx \mu_1(\eta_1) \ldots \mu_{k-1}(\eta_{k-1})\omega_k(\eta_k)\mu_{k+1}(\eta_{k+1}) \ldots \mu_N(\eta_N) \quad k = 1, \ldots, N, \tag{A.0.1}$$

where $\boldsymbol{\eta} \in \mathbb{R}^N$. We describe here the procedure for the general case while specific details for each PDE will be given at the end of the section. After approximating each function by piecewise constants, (A.0.1) becomes

$$\left[\mathfrak{C}^{(k)}\right]_{i_1,\ldots,i_N} \approx [\boldsymbol{\mu}^{(1)}]_{i_1} \cdots [\boldsymbol{\mu}^{(k-1)}]_{i_{k-1}}[\boldsymbol{\omega}^{(k)}]_{i_k}[\boldsymbol{\mu}^{(k+1)}]_{i_{k+1}} \cdots [\boldsymbol{\mu}^{(N)}]_{i_N}, \tag{A.0.2}$$

where, denoting by $\mathbb{R}_+$ the set of strictly positive real numbers, the tensors $\mathfrak{C}^{(k)} \in \mathbb{R}_+^{n_1 \times \cdots \times n_N}$ are given and $\boldsymbol{\mu}^{(k)}, \boldsymbol{\omega}^{(k)} \in \mathbb{R}_+^{n_k}$, $k = 1, \ldots, N$, are unknown vectors to be computed.

In order to compute the approximation (A.0.2), we aim at finding $\boldsymbol{\mu}^{(k)}, \boldsymbol{\omega}^{(k)} \in \mathbb{R}_+^{n_k}$ for $k = 1, \ldots, N$, that minimize the functional

$$\left[\boldsymbol{\chi}^{(k)}, \boldsymbol{\psi}^{(k)}\right]_{k=1,\ldots,N} \longmapsto \max_{\substack{i_k=1,\ldots,n_k;\\ k=1,\ldots,N}} \left\{ \left| \log \left(\frac{[\mathfrak{C}^{(k)}]_{i_1,\ldots,i_N}}{[\boldsymbol{\chi}^{(1)}]_{i_1} \cdots [\boldsymbol{\chi}^{(k-1)}]_{i_{k-1}}[\boldsymbol{\psi}^{(k)}]_{i_k}[\boldsymbol{\chi}^{(k+1)}]_{i_{k+1}} \cdots [\boldsymbol{\chi}^{(N)}]_{i_N}} \right) \right| \right\}.$$

Equivalently, we look for $\boldsymbol{\mu}^{(k)}, \boldsymbol{\omega}^{(k)} \in \mathbb{R}_+^{n_k}$ for $k = 1, \ldots, N$, such that the minimum and maximum values of the ratio

$$\frac{[\mathfrak{C}^{(k)}]_{i_1,\ldots,i_N}}{[\boldsymbol{\mu}^{(1)}]_{i_1} \cdots [\boldsymbol{\mu}^{(k-1)}]_{i_{k-1}}[\boldsymbol{\omega}^{(k)}]_{i_k}[\boldsymbol{\mu}^{(k+1)}]_{i_{k+1}} \cdots [\boldsymbol{\mu}^{(N)}]_{i_N}}, \qquad i_k = 1, \ldots, n_k;\ k = 1, \ldots, N,$$

are as close as possible to 1 (in the logarithmic sense).

Algorithm 5 computes an approximate solution of the above optimization problem. This algorithm generalizes the one used in [113] which is focused on the case of two variables, i.e.

it computes the approximations

$$\left[\mathfrak{C}^{(1)}\right]_{i_1,i_2} \approx [\boldsymbol{\omega}^{(1)}]_{i_1}[\boldsymbol{\mu}^{(2)}]_{i_2}, \qquad \left[\mathfrak{C}^{(2)}\right]_{i_1,i_2} \approx [\boldsymbol{\mu}^{(1)}]_{i_1}[\boldsymbol{\omega}^{(2)}]_{i_2}.$$

Note that in this case the two approximation problems are completely decoupled, so they can be solved independently. As in [113], in all our tests we set $maxit = 2$.

In the case of Stokes system of Chapter 3, we have to compute three approximations as (A.0.2), one for each diagonal block of the preconditioner P_V (see (3.5.1)). Thus, for $l = 1, 2, 3$, we set $N = 3$ and, referring to Section 3.5.3 for the notations, we have $c_k(\boldsymbol{\eta}) = [\boldsymbol{C}_l]_{k,k}(\boldsymbol{\eta})$. We construct $\mathfrak{C}^{(k)} \in \mathbb{R}_+^{n_1\times n_2\times n_3}$ by interpolating the functions c_k directly at the quadrature points, whose number in each parametric direction is equal to n_1, n_2, n_3, respectively.

For the space-time least-squares formulation of the heat problem of Chapter 4 and for the space-time Galerkin formulation of Chapter 5 we consider d spatial variables and the time: we set $N := d + 1$ and $\eta_{d+1} := \tau$. The functions c_k in (A.0.1) that need to be approximated correspond to the functions defined in (4.4.7) and (5.4.23) for the least-squares and Galerkin formulations, respectively. In both case, $n_1, \dots, n_d$ are the number of elements in each spatial direction and n_{d+1} the number of elements in time, and we construct $\mathfrak{C}^{(k)} \in \mathbb{R}_+^{n_1\times\cdots\times n_{d+1}}$ by interpolating c_k in the element barycenters. The approximation at the quadrature points required to construct the univariate factors in (4.4.9) and (4.4.10) is then recovered by interpolation.

Algorithm 5 Separation of variables

```
Initialize $\boldsymbol{\mu}^{(l)} = \boldsymbol{\omega}^{(l)} = \mathbf{1}_{n_l}$ for $l = 1, \dots, N$.
for $iter = 1 \dots maxit$ do
    for $k = 1, \dots, N$ do
        Compute $\mathfrak{V}^{(k)} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ s.t.
            $\left[\mathfrak{V}^{(k)}\right]_{i_1,\dots,i_{d+1}} = \frac{[\mathfrak{C}^{(k)}]_{i_1,\dots,i_N}}{[\boldsymbol{\mu}^{(1)}]_{i_1} \cdots [\boldsymbol{\mu}^{(k-1)}]_{i_{k-1}} [\boldsymbol{\mu}^{(k+1)}]_{i_{k+1}} \cdots [\boldsymbol{\mu}^{(N)}]_{i_N}}.$
        for $j = 1, \dots, n_k$ do
            Compute $m = \min \left\{ \mathfrak{V}^{(k)}_{i_1,\dots,i_{k-1},j,i_{k+1},\dots i_N} \mid i_l = 1, \dots, n_l; l = 1, \dots, N \text{ and } l \neq k \right\}.$
            Compute $M = \max \left\{ \mathfrak{V}^{(k)}_{i_1,\dots,i_{k-1},j,i_{k+1},\dots i_N} \mid i_l = 1, \dots, n_l; l = 1, \dots, N \text{ and } l \neq k \right\}.$

            Update $[\boldsymbol{\omega}^{(k)}]_j = \sqrt{mM}.$
        end for
    end for
    for $k = 1, \dots, N$ do
        for $l = 1, \dots, N$ do
            if $l \neq k$ then
                Compute $\mathfrak{W}^{(k,l)} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ s.t.
                    $\left[\mathfrak{W}^{(k,l)}\right]_{i_1,\dots,i_N} = \frac{[\mathfrak{C}^{(k)}]_{i_1,\dots,i_N} [\boldsymbol{\mu}^{(l)}]_{i_l}}{[\boldsymbol{\mu}^{(1)}]_{i_1} \cdots [\boldsymbol{\mu}^{(k-1)}]_{i_{k-1}} [\boldsymbol{\omega}^{(k)}]_{i_k} [\boldsymbol{\mu}^{(k+1)}]_{i_{k+1}} \cdots [\boldsymbol{\mu}^{(N)}]_{i_N}}.$
            end if
        end for
        Compute $\mathfrak{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ s.t.
            $[\mathfrak{Y}]_{i_1,\dots,i_{n_N}} = \min \left\{ [\mathfrak{W}^{(k,l)}]_{i_1,\dots,i_{n_N}} \mid l = 1, \dots, N \text{ and } l \neq k \right\}$
        Compute $\mathfrak{Z} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ s.t.
            $[\mathfrak{Z}]_{i_1,\dots,i_{n_N}} = \max \left\{ [\mathfrak{W}^{(k,l)}]_{i_1,\dots,i_{n_N}} \mid l = 1, \dots, N \text{ and } l \neq k \right\}$
        for $j = 1, \dots, n_k$ do
            Compute $m = \min \left\{ [\mathfrak{Y}]_{i_1,\dots,i_{k-1},j,i_{k+1},\dots i_N} \mid i_l = 1, \dots, n_l; l = 1, \dots, N \text{ and } l \neq k \right\}.$

            Compute $M = \max \left\{ [\mathfrak{Z}]_{i_1,\dots,i_{k-1},j,i_{k+1},\dots i_N} \mid i_l = 1, \dots, n_l; l = 1, \dots, N \text{ and } l \neq k \right\}.$

            Update $[\boldsymbol{\mu}^{(k)}]_j = \sqrt{mM}.$
        end for
    end for
end for
```

Bibliography

[1] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient flows: in metric spaces and in the space of probability measures.* Springer Science & Business Media, 2008.

[2] P. Antolin, A. Buffa, F. Calabro, M. Martinelli, and G. Sangalli. "Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization". In: *Computer Methods in Applied Mechanics and Engineering* 285 (2015), pp. 817–828.

[3] J.-P. Aubin. *Applied functional analysis.* Translated from the French by Carole Labrousse, With exercises by Bernard Cornet and Jean-Michel Lasry. John Wiley & Sons, New York-Chichester-Brisbane, 1979, pp. xv+423. ISBN: 0-471-02149-0.

[4] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. "Isogeometric analysis: approximation, stability and error estimates for h-refined meshes". In: *Math. Mod. and Meth. Appl. Sc.* 16.07 (2006), pp. 1031–1090.

[5] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. "Isogeometric analysis using T-splines". In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 229–263.

[6] Y. Bazilevs, V. M. Calo, J. A. Cottrell, T. J. R. Hughes, A. Reali, and G. Scovazzi. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows". In: *Computer methods in applied mechanics and engineering* 197.1-4 (2007), pp. 173–201.

[7] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. "Analysis-suitable T-splines of arbitrary degree: definition, linear independence and approximation properties". In: *Mathematical Models and Methods in Applied Sciences* 23.11 (2013), pp. 1979–2003.

[8] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. "BDDC preconditioners for isogeometric analysis". In: *Mathematical Models and Methods in Applied Sciences* 23.06 (2013), pp. 1099–1142.

[9] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. "Isogeometric Schwarz preconditioners for linear elasticity systems". In: *Comput. Methods Appl. Mech. Engrg.* 253 (2013), pp. 439–454. ISSN: 0045-7825.

[10] L. Beirão da Veiga, D. Cho, L. F. Pavarino, and S. Scacchi. "Overlapping Schwarz methods for isogeometric analysis". In: *SIAM J. Numer. Anal.* 50.3 (2012), pp. 1394–1416.

[11] L. Beirão da Veiga, D. Cho, and G. Sangalli. "Anisotropic NURBS approximation in isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 209 (2012), pp. 1–11.

[12] L. Beirão da Veiga, L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. "Isogeometric BDDC preconditioners with deluxe scaling". In: *SIAM Journal on Scientific Computing* 36.3 (2014), A1118–A1139. ISSN: 1064-8275.

[13] B. C. Bell and K. S. Surana. “A space–time coupled p-version least-squares finite element formulation for unsteady fluid dynamics problems”. In: *International journal for numerical methods in engineering* 37.20 (1994), pp. 3545–3569.

[14] B. C. Bell and K. S. Surana. “A space-time coupled p-version least squares finite element formulation for unsteady two-dimensional Navier–Stokes equations”. In: *International journal for numerical methods in engineering* 39.15 (1996), pp. 2593–2618.

[15] M. Benzi, G. H. Golub, and J. Liesen. “Numerical solution of saddle point problems”. In: *Acta numerica* 14 (2005), pp. 1–137.

[16] P. B. Bochev and M. D. Gunzburger. *Least-squares finite element methods.* Vol. 166. Springer Science & Business Media, 2009.

[17] J. Bonilla and S. Badia. “Maximum-principle preserving space–time isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 354 (2019), pp. 422–440.

[18] A. Bressan. “Isogeometric regular discretization for the Stokes problem”. In: *IMA journal of numerical analysis* 31.4 (2010), pp. 1334–1356.

[19] A. Bressan. “Some properties of LR-splines”. In: *Computer Aided Geometric Design* 30.8 (2013), pp. 778–794.

[20] A. Bressan and G. Sangalli. “Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique”. In: *IMA Journal of Numerical Analysis* 33.2 (2012), pp. 629–651.

[21] A. Bressan and S. Takacs. “Sum factorization techniques in Isogeometric Analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 352 (2019), pp. 437–460.

[22] H. Brezis. *Analyse fonctionnelle. Théorie et applications.* Collection Mathématiques Appliquées pour la Maîtrise. Masson, Paris, 1983, pp. xiv+234. ISBN: 2-225-77198-7.

[23] H. Brézis. “Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert”. In: 5 (1973). North-Holland Mathematics Studies, No. 5. Notas de Matemática (50).

[24] J. C. Bruch Jr and G. Zyvoloski. “Transient two-dimensional heat conduction problems solved by the finite element method”. In: *International Journal for Numerical Methods in Engineering* 8.3 (1974), pp. 481–494.

[25] A. Buffa, C. De Falco, and G. Sangalli. “Isogeometric analysis: stable elements for the 2D Stokes equation”. In: *International Journal for Numerical Methods in Fluids* 65.11-12 (2011), pp. 1407–1422.

[26] A. Buffa and C. Giannelli. “Adaptive isogeometric methods with hierarchical splines: error estimator and convergence”. In: *Mathematical Models and Methods in Applied Sciences* 26.01 (2016), pp. 1–25.

[27] A. Buffa, H. Harbrecht, A. Kunoth, and G. Sangalli. “BPX-preconditioning for isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 265 (2013), pp. 63–70.

[28] A. Buffa, J. Rivas, G. Sangalli, and R. Vázquez. “Isogeometric discrete differential forms in three dimensions”. In: *SIAM Journal on Numerical Analysis* 49.2 (2011), pp. 818–844.

[29] A. Buffa, G. Sangalli, and C. Schwab. “Exponential convergence of the hp version of isogeometric analysis in 1D”. In: *Spectral and High Order Methods for Partial Differential Equations-ICOSAHOM 2012.* Springer, 2014, pp. 191–203.

[30] F. Calabrò, G. Sangalli, and M. Tani. “Fast formation of isogeometric Galerkin matrices by weighted quadrature”. In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 606–622.

[31] P. G. Ciarlet. *Mathematical elasticity. Vol. I.* Vol. 20. Studies in Mathematics and its Applications. Three-dimensional elasticity. North-Holland Publishing Co., Amsterdam, 1988, pp. xlii+451. ISBN: 0-444-70259-8.

[32] C. Coley, J. Benzaken, and J. A. Evans. “A geometric multigrid method for isogeometric compatible discretizations of the generalized Stokes and Oseen problems”. In: *arXiv preprint arXiv:1705.09282* (2017).

[33] N. Collier, L. Dalcin, D. Pardo, and V. M. Calo. “The cost of continuity: performance of iterative solvers on isogeometric finite elements”. In: *SIAM Journal on Scientific Computing* 35.2 (2013), A767–A784.

[34] N. Collier, D. Pardo, L. Dalcin, M. Paszynski, and V. M. Calo. “The cost of continuity: a study of the performance of isogeometric finite elements using direct solvers”. In: *Computer Methods in Applied Mechanics and Engineering* 213 (2012), pp. 353–361.

[35] A. M. Côrtes, Alvaro L. G. A. Coutinho, Lisandro Dalcin, and Victor M Calo. “Performance evaluation of block-diagonal preconditioners for the divergence-conforming B-spline discretization of the Stokes system”. In: *Journal of Computational Science* 11 (2015), pp. 123–136.

[36] A. M. Côrtes, L Dalcin, A. F. Sarmiento, N Collier, and Victor M Calo. “A scalable block-preconditioning strategy for divergence-conforming B-spline discretizations of the Stokes problem”. In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 839–858.

[37] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[38] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. “Isogeometric analysis of structural vibrations”. In: *Computer methods in applied mechanics and engineering* 195.41-43 (2006), pp. 5257–5296.

[39] C. De Boor. *A practical guide to splines (revised edition)*. Applied Mathematical Sciences. Berlin: Springer, 2001.

[40] M. O. Deville, P. F. Fischer, and E. H. Mund. *High-order methods for incompressible fluid flow*. Cambridge University Press, 2002.

[41] M. Donatelli, C. Garoni, C. Manni, S. Serra-Capizzano, and H. Speleers. “Robust and optimal multi-iterative techniques for IgA Galerkin linear systems”. In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 230–264.

[42] C. A. Dorao and H. A. Jakobsen. “A parallel time–space least-squares spectral element solver for incompressible flow problems”. In: *Applied mathematics and computation* 185.1 (2007), pp. 45–58.

[43] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics & Scientific Computation, 2014.

[44] J. A. Evans, Y. Bazilevs, I. Babuška, and T. J. R. Hughes. “n-Widths, sup–infs, and optimality ratios for the k-version of the isogeometric finite element method”. In: *Computer Methods in Applied Mechanics and Engineering* 198.21-26 (2009), pp. 1726–1741.

[45] J. A. Evans and T. J. R. Hughes. “Isogeometric divergence-conforming B-splines for the Darcy–Stokes–Brinkman equations”. In: *Mathematical Models and Methods in Applied Sciences* 23.04 (2013), pp. 671–741.

[46] J. A. Evans and T. J. R. Hughes. "Isogeometric divergence-conforming B-splines for the steady Navier–Stokes equations". In: *Mathematical Models and Methods in Applied Sciences* 23.08 (2013), pp. 1421–1478.

[47] J. A. Evans and T. J. R. Hughes. "Isogeometric divergence-conforming B-splines for the unsteady Navier–Stokes equations". In: *Journal of Computational Physics* 241 (2013), pp. 141–167.

[48] J. A. Evans and Thomas J. R. Hughes. "Explicit trace inequalities for isogeometric analysis and parametric hexahedral finite elements". In: *Numerische Mathematik* 123.2 (2013), pp. 259–290.

[49] L. C. Evans. *Partial Differential equations.* Berlin: American Mathematical Society, 2010.

[50] C. Farhat and F. X. Roux. "A method of finite element tearing and interconnecting and its parallel solution algorithm". In: *Internat. J. Numer. Methods Engrg.* 32.6 (1991), pp. 1205–1227. ISSN: 0029-5981.

[51] I. Fried. "Finite-element analysis of time-dependent phenomena." In: *AIAA Journal* 7.6 (1969), pp. 1170–1173.

[52] K. P. S. Gahalaut, J. K. Kraus, and S. K. Tomar. "Multigrid methods for isogeometric discretization". In: *Computer Methods in Applied Mechanics and Engineering* 253 (2013), pp. 413–425.

[53] K. P. S. Gahalaut, S. K. Tomar, and C. Douglas. "Condition number estimates for matrices arising in NURBS based isogeometric discretizations of elliptic partial differential equations". In: *arXiv preprint arXiv:1406.6808* (2014).

[54] M. J. Gander. "50 years of time parallel time integration". In: *Multiple Shooting and Time Domain Decomposition Methods.* Springer, 2015, pp. 69–113.

[55] C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, and J. Špeh. "THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 299 (2016), pp. 337–365.

[56] C. Giannelli, B. Jüttler, and H. Speleers. "THB-splines: The truncated basis for hierarchical splines". In: *Computer Aided Geometric Design* 29.7 (2012), pp. 485–498.

[57] A. Greenbaum, V. Pták, and Z. Strakoš. "Any nonincreasing convergence curve is possible for GMRES". In: *SIAM journal on matrix analysis and applications* 17.3 (1996), pp. 465–469.

[58] P. P. Grinevich and M. A. Olshanskii. "An iterative method for the Stokes-type problem with variable viscosity". In: *SIAM Journal on Scientific Computing* 31.5 (2009), pp. 3959–3978.

[59] P. Grisvard. *Elliptic problems in nonsmooth domains.* Vol. 69. SIAM, 2011.

[60] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems.* Vol. 49. 1. NBS Washington, DC, 1952.

[61] C. Hofer. "Parallelization of continuous and discontinuous Galerkin dual–primal isogeometric tearing and interconnecting methods". In: *Computers & Mathematics with Applications* 74.7 (2017), pp. 1607–1625.

[62] C. Hofer and U. Langer. "Dual-primal isogeometric tearing and interconnecting solvers for multipatch dG-IgA equations". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 2–21. ISSN: 0045-7825.

[63] C. Hofer, U. Langer, M. Neumuüller, and R. Schneckenleitner. "Parallel and robust preconditioning for space-time isogeometric analysis of parabolic evolution problems". In: *SIAM Journal on Scientific Computing* 41.3 (2019), A1793–A1821.

[64] C. Hofreither and S. Takacs. "Robust multigrid for isogeometric analysis based on stable splittings of spline spaces". In: *SIAM Journal on Numerical Analysis* 55.4 (2017), pp. 2004–2024.

[65] C. Hofreither, S. Takacs, and W. Zulehner. *A Robust Multigrid Method for Isogeometric Analysis using Boundary Correction.* Tech. rep. 33. NFN, 2015.

[66] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39 (2005), pp. 4135–4195.

[67] T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. "A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations". In: *Computer Methods in Applied Mechanics and Engineering* 73.2 (1989), pp. 173–189.

[68] K. A. Johannessen, T. Kvamsdal, and T. Dokken. "Isogeometric analysis using LR B-splines". In: *Computer Methods in Applied Mechanics and Engineering* 269 (2014), pp. 471–514.

[69] C. Keller, N. I. M. Gould, and A. J. Wathen. "Constraint preconditioning for indefinite linear systems". In: *SIAM J. Matrix Anal. Appl.* 21.4 (2000), pp. 1300–1317. ISSN: 0895-4798.

[70] S. K. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. "IETI–isogeometric tearing and interconnecting". In: *Computer Methods in Applied Mechanics and Engineering* 247 (2012), pp. 201–215.

[71] T. G. Kolda and B. W. Bader. "Tensor decompositions and applications". In: *SIAM review* 51.3 (2009), pp. 455–500.

[72] A. M. Kvarving and E. M. Rønquist. "A fast tensor-product solver for incompressible fluid flow in partially deformed three-dimensional domains: Parallel implementation". In: *Computers & Fluids* 52 (2011), pp. 22–32.

[73] O. A. Ladyzhenskaja and N. N. Ural'ceva. *Équations aux dérivées partielles de type elliptique.* Dunod, 1968.

[74] U. Langer, S. E. Moore, and M. Neumüller. "Space-time isogeometric analysis of parabolic evolution problems". In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 342 –363.

[75] U. Langer, M. Neumüller, and I. Toulopoulos. "Multipatch space-time isogeometric analysis of parabolic diffusion problems". In: *International Conference on Large-Scale Scientific Computing.* Springer. 2017, pp. 21–32.

[76] G. Loli, M. Montardini, G. Sangalli, and M. Tani. "Space-time Galerkin isogeometric method and efficient solver for parabolic problem". In: *arXiv e-prints*, arXiv:1909.07309 (2019).

[77] R. E. Lynch, J. R. Rice, and D. H. Thomas. "Direct solution of partial difference equations by tensor product methods". In: *Numerische Mathematik* 6.1 (1964), pp. 185–199.

[78] A. Mantzaflaris, B. Jüttler, B. N. Khoromskij, and U. Langer. "Low rank tensor methods in Galerkin-based isogeometric analysis". In: *Comput. Methods Appl. Mech. Engrg.* 316 (2017), pp. 1062–1085.

[79] A. Mantzaflaris, F. Scholz, and I. Toulopoulos. “Low-rank space-time decoupled isogeometric analysis for parabolic problems with varying coefficients”. In: *Computer Methods in Applied Mathematics* (2018).

[80] M. Montardini, M. Negri, G. Sangalli, and M. Tani. “Space-time least-squares isogeometric method and efficient solver for parabolic problems”. In: *Mathematics of Computation* (accepted for publication).

[81] M. Montardini, G. Sangalli, and M. Tani. “Robust isogeometric preconditioners for the Stokes system based on the Fast Diagonalization method”. In: *Computer Methods in Applied Mechanics and Engineering* 338 (2018), pp. 162 –185.

[82] S. Morganti, F. Auricchio, D. J. Benson, F. I. Gambarin, S. Hartmann, T. J. R. Hughes, and A. Reali. “Patient-specific isogeometric structural analysis of aortic valve closure”. In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 508–520.

[83] M. F. Murphy, G. H. Golub, and A. J. Wathen. “A note on preconditioning for indefinite linear systems”. In: *SIAM J. Sci. Comput.* 21.6 (2000), pp. 1969–1972.

[84] H. Nguyen and J. Reynen. “A space-time least-square finite element scheme for advection-diffusion equations”. In: *Computer Methods in Applied Mechanics and Engineering* 42.3 (1984), pp. 331–342.

[85] J. T. Oden. “A general theory of finite elements. I. Topological considerations”. In: *International Journal for Numerical Methods in Engineering* 1.2 (1969), pp. 205–221.

[86] J. T. Oden. “A general theory of finite elements. II. Applications”. In: *International Journal for Numerical Methods in Engineering* 1.3 (1969), pp. 247–259.

[87] C. C. Paige and M. A. Saunders. “Solution of Sparse Indefinite Systems of Linear Equations”. In: *SIAM journal on numerical analysis* 12.4 (1975), pp. 617–629.

[88] L. F. Pavarino and S. Scacchi. “Isogeometric block FETI-DP preconditioners for the Stokes and mixed linear elasticity systems”. In: *Computer Methods in Applied Mechanics and Engineering* 310 (2016), pp. 694–710.

[89] L. F. Pavarino, S. Scacchi, O. B. Widlund, and S. Zampini. “Isogeometric BDDC deluxe preconditioners for linear elasticity”. In: *Mathematical Models and Methods in Applied Sciences* 28.7 (2018), pp. 1337–1370. ISSN: 0218-2025.

[90] L. Piegl and W. Tiller. *The NURBS book*. Springer, 1997.

[91] Y. Saad. *Iterative methods for sparse linear systems*. Second. SIAM, 2003, pp. xviii+528. ISBN: 0-89871-534-2.

[92] Y. Saad and M. H. Schultz. “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”. In: *SIAM Journal on scientific and statistical computing* 7.3 (1986), pp. 856–869.

[93] G. Sangalli and M. Tani. “Isogeometric preconditioners based on fast solvers for the Sylvester equation”. In: *SIAM Journal on Scientific Computing* 38.6 (2016), A3644–A3671.

[94] G. Sangalli and M. Tani. “Matrix-free weighted quadrature for a computationally efficient isogeometric k-method”. In: *Computer Methods in Applied Mechanics and Engineering* 338 (2018), pp. 117 –133. ISSN: 0045-7825.

[95] F. Santambrogio. “{Euclidean, metric, and Wasserstein} gradient flows: an overview”. In: *Bulletin of Mathematical Sciences* 7.1 (2017), pp. 87–154.

[96] L. Schumaker. *Spline functions: basic theory*. Cambridge University Press, 2007.

[97] C. Schwab and R. Stevenson. “Space-time adaptive wavelet methods for parabolic evolution problems”. In: *Mathematics of Computation* 78.267 (2009), pp. 1293–1318.

[98] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. "Local refinement of analysis-suitable T-splines". In: *Computer Methods in Applied Mechanics and Engineering* 213 (2012), pp. 206–222.

[99] F. Shakib and T. J. R. Hughes. "A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms". In: *Computer Methods in Applied Mechanics and Engineering* 87.1 (1991), pp. 35–58.

[100] D. Silvester and A. Wathen. "Fast iterative solution of stabilised Stokes systems Part II: Using general block preconditioners". In: *SIAM Journal on Numerical Analysis* 31.5 (1994), pp. 1352–1367.

[101] V. Simoncini. "Computational methods for linear matrix equations". In: *SIAM Review* 58.3 (2016), pp. 377–441.

[102] L. Sorber, M. Van Barel, and L. De Lathauwer. "Tensorlab v2. 0". In: *Available online, URL: www.tensorlab.net* (2014).

[103] O. Steinbach. "Space-time finite element methods for parabolic problems". In: *Computational methods in applied mathematics* 15.4 (2015), pp. 551–566.

[104] R. Stevenson and J. Westerdiep. "Stability of Galerkin discretizations of a mixed space-time variational formulation of parabolic evolution equations". In: *arXiv:1902.06279* (2019).

[105] G. Strang and G. J. Fix. *An analysis of the finite element method.* Vol. 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[106] S. Takacs. "Robust multigrid methods for isogeometric discretizations of the Stokes equations". In: *International Conference on Domain Decomposition Methods* (2017), pp. 511–520.

[107] K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. "Space–time fluid mechanics computation of heart valve models". In: *Computational Mechanics* 54.4 (2014), pp. 973–986.

[108] K. Takizawa, T. E. Tezduyar, Y. Otoguro, T. Terahara, T. Kuraishi, and H. Hattori. "Turbocharger flow computations with the space–time isogeometric analysis (ST-IGA)". In: *Computers & Fluids* 142 (2017), pp. 15–20.

[109] K. Takizawa, T. E. Tezduyar, and T. Terahara. "Ram-air parachute structural and fluid mechanics computations with the space–time isogeometric analysis (ST-IGA)". In: *Computers & Fluids* 141 (2016), pp. 191–200.

[110] K. Takizawa, T. E. Tezduyar, T. Terahara, and T. Sasaki. "Heart valve flow computation with the space–time slip interface topology change (ST-SI-TC) method and isogeometric analysis (IGA)". In: *Biomedical Technology.* Springer, 2018, pp. 77–99.

[111] R. Vázquez. "A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0". In: *Computers & Mathematics with Applications* 72.3 (2016), pp. 523–554.

[112] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. "A hierarchical approach to adaptive local refinement in isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 200.49-52 (2011), pp. 3554–3567.

[113] E. L. Wachspress. "Generalized ADI preconditioning". In: *Computers & mathematics with applications* 10.6 (1984), pp. 457–461.

[114] A. Wathen and D. Silvester. "Fast iterative solution of stabilised Stokes systems. Part I: Using simple diagonal preconditioners". In: *SIAM Journal on Numerical Analysis* 30.3 (1993), pp. 630–649.