Robust mass lumping and outlier removal strategies in isogeometric analysis

Yannis Voet *[1], Espen Sande †[1], and Annalisa Buffa ‡[1]

[1]MNS, Institute of Mathematics, École polytechnique fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland

February 22, 2024

Abstract

Mass lumping techniques are commonly employed in explicit time integration schemes for problems in structural dynamics and both avoid solving costly linear systems with the consistent mass matrix and increase the critical time step. In isogeometric analysis, the critical time step is constrained by so-called "outlier" frequencies, representing the inaccurate high frequency part of the spectrum. Removing or dampening these high frequencies is paramount for fast explicit solution techniques. In this work, we propose robust mass lumping and outlier removal techniques for nontrivial geometries, including multipatch and trimmed geometries. Our lumping strategies provably do not deteriorate (and often improve) the CFL condition of the original problem and are combined with deflation techniques to remove persistent outlier frequencies. Numerical experiments reveal the advantages of the method, especially for simulations covering large time spans where they may halve the number of iterations with little or no effect on the numerical solution.

Keywords: Isogeometric analysis, Explicit dynamics, Mass lumping, Mass scaling, Outlier removal, Trimming

1 Introduction and background

Isogeometric analysis is a discretization technique for solving partial differential equations (PDEs) which relies on spline functions such as B-splines and non-uniform rational B-splines (NURBS) both for parameterizing the geometry and representing the solution [1, 2]. Spline functions used in isogeometric analysis offer distinctive advantages over Lagrange polynomials used in classical finite element discretizations, including exact representation of common geometries and superior approximation properties [3, 4, 5]. In structural dynamics, the advantages of isogeometric analysis were already evidenced in [6, 7, 8] where maximally smooth spline approximations removed the so-called "optical branches" from the discrete spectrum, a typical artifact of classical finite element discretizations. As a matter of fact, nearly all discrete eigenvalues approximate the continuous eigenvalues with high accuracy except for the largest ones. They form noticeable spikes in the upper part of the spectrum and were coined "outlier" eigenvalues for this very reason [6]. These inaccurate eigenvalues grow with the mesh size and spline order [9] thereby severely constraining the critical time step of explicit time integration schemes, often referred to as the Courant–Friedrichs–Lewy (CFL) condition. For example, for undamped dynamical systems, the critical time step of the central difference method is

$$\Delta t_c = \frac{2}{\sqrt{\lambda_n}} \tag{1.1}$$

where λ_n is the largest eigenvalue of the discrete system [10, 11].

Since the advent of isogeometric analysis, much effort has focused on removing the outliers from the discrete spectrum. A nonlinear spline parametrization was first proposed in [6] by uniformly distributing control points by changing the geometry parametrization but defeats the spirit of isogeometric analysis. The method also reportedly undermines the accuracy of the low frequencies and modes [12]. A similar method was later proposed in [13] by constructing "smoothed" knot vectors that are approximations to optimal knot vectors (related to some n-width optimal spline spaces) but suffer from similar drawbacks. In [14] the authors proposed to instead use the n-width optimal spline spaces in [15] to remove outliers, and these spaces were later proven to be outlier-free in [16] without loss of accuracy in the low frequencies. These optimal spaces mimic the true eigenfunctions by imposing certain

*yannis.voet@epfl.ch †espen.sande@epfl.ch ‡annalisa.buffa@epfl.ch

higher-order derivatives to be zero at the boundary. Bases for these spaces were constructed in [17, 15, 18, 16] by using certain symmetry properties of B-splines. A similar strategy was numerically observed to be outlier-free in [12] for the Laplacian, however, some outliers were still observed for the biharmonic problem with the spline spaces proposed in [12]. By mimicking the properties of the n-width optimal spline spaces in [19] outliers for the biharmonic problem were later completely removed in [20]. In both [12] and [20] bases for the considered spline spaces were computed using a similar strategy as the MDB-spline extraction technique developed in [21]. A related strategy to the above was devised in [22] where the authors weakly impose the constraints coming from the true eigenfunctions by penalizing high order derivatives near the boundary. In the multipatch setting such a strategy was also used in [23] by weakly enforcing C^{p-1} continuity at patch interfaces to remove interior outliers. These penalization techniques often take the form of mass scaling, which is also widely employed for classical finite element methods [24, 25, 26]. Rather than completely removing outliers, they strongly mitigate them but also require heuristics or dedicated algorithms for computing the generally unknown penalization parameters. We note that these approaches could be combined to mitigate both boundary and interior outliers resulting from C^0 continuity at patch interfaces. Also in the context of classical finite element methods, an eigenvalue deflation technique was proposed in [27, 28] by explicitly computing the largest eigenvalues and eigenmodes. However, the authors do not discuss the computational overhead in their experiments and applying their method to the global assembled mass matrix is generally infeasible due to the large number of inaccurate high frequencies in classical C^0 finite element methods.

Outlier removal strategies are generally motivated for one-dimensional problems and then extended to higher dimensions via a tensor product construction. This construction, however, inherently limits the applicability of the methods to trivial single-patch geometries and separable coefficient functions. One issue in extending these approaches to nontrivial problems lies in the definition of outliers. For nontrivial geometries, outliers are often smoothened out and the spectrum generally does not feature any spikes, although changes in curvature are sometimes noticeable. Outliers then lose the intrinsic property that characterized them and their identification becomes ambiguous. Straightforwardly applying the constructions proposed in [12, 16, 22] to nontrivial geometries generally does not yield satisfactory results, suggesting that the support of the outlier eigenfunctions might stretch into the domain's interior.

For applications in explicit dynamics, restrictive CFL conditions are not the only issue. Due to the inherent cost of "exactly" solving linear systems with the mass matrix [29, 30], obtaining an easily invertible (preferably diagonal or tridiagonal) mass matrix is paramount. Mass lumping has historically been used for approximating the consistent mass matrix by a diagonal (lumped) mass matrix. Among the scores of methods proposed in the 1970s for classical finite element methods, only a handful are applicable to isogeometric analysis due to the non-interpolatory nature of the basis functions. Among them is the classical row-sum technique [31, 10]. Provably, the row-sum technique does not worsen the CFL condition of the consistent mass [32]. However, for isogeometric analysis, the method performs poorly in higher dimensions and strong numerical evidence suggests it reduces the converge rate to quadratic order, independently of the spline order [6]; a property only proved for 1D problems and low spline orders [6]. Since then much research effort has focused on devising more accurate and potentially high order mass lumping schemes for isogeometric analysis. Cottrell et al. [2] first suggested constructing diagonal mass matrices by using dual basis functions as test functions in a Petrov-Galerkin framework. However, computer implementations come with all sorts of difficulties and initially the idea did not gain much momentum until it was taken up again in [33] with promising results. Since then, there has been a surge of interest. In [34], high order convergence is achieved by combining (approximate) dual basis functions with the row-sum technique. However, the implementation remains technical and ongoing research is focusing on alleviating these issues, in particular related to the imposition of boundary conditions [35]. In another line of research, families of banded and Kronecker product matrices were constructed in [32] by increasing the bandwidth of the row-sum lumped mass matrix and were shown to significantly improve the accuracy. Unfortunately, such improvements are only realized on trivial geometries and are tied to an improvement of the constant instead of the convergence rate.

Mass lumping may sometimes dramatically impact the CFL condition. For trimmed geometries [36], Leidinger [37] first showed that the CFL condition was not affected by small trimmed elements if the mass matrix was lumped. This finding was further supported by the studies in [38, 39] but also raised concerns over the accuracy of the smallest eigenvalues and modes. Devising accurate mass lumping/scaling techniques for trimmed geometries is an ongoing challenge.

Fast solution methods in explicit dynamics usually combine outlier removal, mass scaling and mass lumping in a sometimes ad hoc fashion. In this article, we propose such a strategy for nontrivial geometries and provide a strong mathematical foundation for our method. The article is structured as follows: in Section 3, we first recall the mass lumping techniques devised in [32] and then design new mass lumping techniques for nontrivial single-patch and multipatch problems. Although these techniques provably do not worsen the CFL condition of the original problem, they may not significantly improve it either. Thus, they are combined in Section 4 with outlier removal techniques that deflate the spectrum from persistent outlier eigenvalues and generalize the method proposed in [27, 28]. Contrary to classical C^0 finite elements, isogeometric discretizations typically feature a small number of rapidly increasing eigenvalues towards the end of the spectrum, for which deflation techniques are well-suited. Section 5 gathers some numerical experiments illustrating the theoretical findings and demonstrating the advantages of the method. Finally, conclusions are drawn in Section 6.

2 Model problem and its discretization

In this article, we consider hyperbolic PDEs from structural dynamics. Their simplest instance is the classical wave (or acoustic) equation, which we will select as model problem. Let $\Omega \subset \mathbb{R}^d$ be an open connected domain in d-dimensional space with Lipschitz boundary and let $I = [0, T]$ be the time domain with $T > 0$ denoting the final time. We look for $u\colon \Omega \times [0, T] \to \mathbb{R}$ such that

$$\begin{aligned} \rho(\mathbf{x})\partial_{tt}u(\mathbf{x},t) - \kappa(\mathbf{x})\Delta u(\mathbf{x},t) &= f(\mathbf{x},t) && \text{in } \Omega \times (0,T], \\ u(\mathbf{x},t) &= 0 && \text{on } \partial\Omega \times (0,T], \\ u(\mathbf{x},0) &= u_0(\mathbf{x}) && \text{in } \Omega, \\ \partial_t u(\mathbf{x},0) &= v_0(\mathbf{x}) && \text{in } \Omega, \end{aligned} \tag{2.1}$$

where u_0 and v_0 are some initial displacement and velocity, respectively, ρ and κ are some positive valued coefficient functions and we prescribe homogeneous Dirichlet boundary conditions for simplicity. In a standard Galerkin discretization, we look for an approximation $u_h(.,t)$ of $u(.,t)$ in a finite dimensional subspace V_h and test against all functions in V_h, which leads to solving the semi-discrete problem (see for instance [10, 40])

$$\begin{aligned} M\ddot{\mathbf{u}}(t) + K\mathbf{u}(t) &= \mathbf{f}(t) \qquad \text{for } t \in [0,T], \\ \mathbf{u}(0) &= \mathbf{u}_0, \\ \dot{\mathbf{u}}(0) &= \mathbf{v}_0. \end{aligned} \tag{2.2}$$

where $K, M \in \mathbb{R}^{n\times n}$ are the stiffness and mass matrices, respectively. The time-dependent right-hand side vector $\mathbf{f}(t) \in \mathbb{R}^n$ accounts for the function f and potential non-homogeneous Neumann and Dirichlet boundary conditions. Finally, $\mathbf{u}(t) \in \mathbb{R}^n$ is the coefficient vector of the approximate solution $u_h(\mathbf{x}, t)$ in a basis of V_h. Isogeometric analysis consists in choosing spline functions from computer-aided-design (CAD) such as B-splines both for representing the approximate solution and describing the geometry [1, 2]. Such functions follow a standardized construction in a so-called parametric domain $\hat{\Omega} = (0,1)^d$ before being defined in the physical domain Ω. In dimension $d = 1$, the B-spline basis $\{\hat{B}_i\}_{i=1}^n$ is constructed recursively from a *knot vector* $\Xi := (\xi_1, \dots, \xi_{n+p+1})$, which is a sequence of non-decreasing real values. The integers p and n denote the spline degree and spline space dimension, respectively. A knot vector is called *open* if

$$\xi_1 = \dots = \xi_{p+1} < \xi_{p+2} \leq \dots \leq \xi_n < \xi_{n+1} = \dots = \xi_{n+p+1}.$$

Internal knots of multiplicity $1 \leq m \leq p$ give rise to C^k continuous spline spaces, denoted $\mathcal{S}^k_{p,\Xi}$, where $k = p - m$. Our work primarily focuses on (but is not restricted to) maximally smooth C^{p-1} spaces obtained when the multiplicity of each internal knot is 1; i.e. the so-called isogeometric k-method. In dimension $d \geq 2$, the spline space is defined as a tensor product of univariate spaces, which all follow a similar construction. The degree, space dimension and continuity along each direction are collected in the vectors $\mathbf{p} = (p_1, p_2, \dots, p_d)$, $\mathbf{n} = (n_1, n_2, \dots, n_d)$ and $\mathbf{k} = (k_1, k_2, \dots, k_d)$, respectively, and we denote the resulting spline space $\mathcal{S}^{\mathbf{k}}_{\mathbf{p},\mathbf{\Xi}}$ (where the dependency on the knot vectors $\Xi_1, \dots, \Xi_d$ is specified by $\mathbf{\Xi}$). In dimension $d \geq 2$, it becomes convenient to label basis functions with multi-indices $\mathbf{i} = (i_1, i_2, \dots, i_d)$ which are often identified with "linear" indices in the global numbering. This identification permits a slight abuse of notation when writing

$$\hat{B}_i = \hat{B}_{\mathbf{i}} = \hat{B}_{1i_1}\hat{B}_{2i_2}\dots\hat{B}_{di_d}$$

where $\hat{B}_{lj}$ denotes the jth function in the lth direction and $1 \leq i \leq n := \prod_{l=1}^d n_l$ is a global index which only depends on $\mathbf{i}$ and $\mathbf{n}$. In the isogeometric paradigm, these functions also describe the geometry via the spline parametrization $F\colon \hat{\Omega} \to \Omega$, which maps the parametric domain to the physical domain. Geometries described by such a map are called *single-patch*. The basis functions over the physical domain are then defined as $B_i = \hat{B}_i \circ F^{-1}$ and the spline spaces over the parametric and physical domains are

$$\hat{V}_h = \operatorname{span}\{\hat{B}_{\mathbf{i}}\colon \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}\} \quad \text{and} \quad V_h = \operatorname{span}\{B_{\mathbf{i}}\colon \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}\},$$

respectively, where $\mathbf{1}$ is the vector of all ones and vector inequalities are understood componentwise. For single-patch geometries, the entries of the stiffness and mass matrices are

$$K_{ij} = \int_{\hat{\Omega}} (\nabla\hat{B}_i(\hat{\mathbf{x}}))^T G(\hat{\mathbf{x}}) \nabla\hat{B}_j(\hat{\mathbf{x}}) \quad \text{and} \quad M_{ij} = \int_{\hat{\Omega}} c(\hat{\mathbf{x}})\hat{B}_i(\hat{\mathbf{x}})\hat{B}_j(\hat{\mathbf{x}}) \qquad 1 \leq i,j \leq n \tag{2.3}$$

where $G(\hat{\mathbf{x}}) := \kappa(F(\hat{\mathbf{x}}))|\det(J_F)|(J_F^T J_F)^{-1}$, $c(\hat{\mathbf{x}}) := \rho(F(\hat{\mathbf{x}}))|\det(J_F)|$ and $J_F = J_F(\hat{\mathbf{x}})$ denotes the Jacobian matrix of F. As with any other standard Galerkin method, K and M are both symmetric and while M is positive definite, K is generally only positive semidefinite (unless Dirichlet boundary conditions are prescribed on some portion of the boundary). In isogeometric analysis, M is additionally nonnegative owing to the pointwise nonnegativity of the B-spline basis functions.

NURBS functions enable the exact representation of a broader class of geometries (including conic sections) and lead to similar expressions and properties for the system matrices. However, the range of geometries they may describe is still far too limited for most industrial applications. For complex geometries, it may be necessary to divide the physical domain into N_p subdomains (or patches); i.e.

$$\Omega = \bigcup_{r=1}^{N_p} \Omega_r$$

where each subdomain (or patch) Ω_r is described by its own map $F_r \colon \hat{\Omega} \to \Omega_r$. Thus, a *multipatch* geometry is just a collection of patches. The construction of spline spaces over multipatch geometries is rather straightforward, though the notation becomes more cumbersome due to prolifying indices. Patches in isogeometric analysis are analogous to elements in classical finite element discretizations. Therefore, the assembly of the stiffness and mass matrices for isogeometric multipatch discretizations is analogously expressed as

$$K = \sum_{r=1}^{N_p} R_r^T K_r R_r \quad \text{and} \quad M = \sum_{r=1}^{N_p} R_r^T M_r R_r$$

where K_r and M_r are the local stiffness and mass matrices of the rth patch and R_r maps its local degrees of freedom to global ones.

Despite the additional flexibility of multipatch geometries, they still fall short in describing the highly complex shapes of industrial CAD models [41]. Such models commonly consist in multiple *trimmed* NURBS patches. Trimming is a Boolean operation whereby parts of a geometry are joined, intersected or simply discarded. While these operations change the visualization of the model, they do not change its mathematical description. The analysis on trimmed geometries is particularly challenging for a variety of reasons, including integration on trimmed boundaries, imposition of essential boundary conditions, stability and conditioning issues [36, 42]. Trimming also alters the structure of systems matrices, which heavily impacts the design of assembly algorithms and preconditioning techniques. As a matter of fact, many developments in isogeometric analysis that rely on a tensor product structure are not applicable to trimmed geometries.

For discretizing (2.2) in time, explicit methods are usually preferred for fast dynamic processes such as blasts or impacts due to the physical restriction on the step size. Scores of methods have been proposed in the literature, including the central difference, Wilson-θ and generalized α methods to name just a few. Most of them are commonly included in textbooks [10, 11], which the reader may consult for details. The critical time step depends on the method (see e.g. (1.1) for the central difference method in the undamped case) but in the undamped case all explicit methods require solving a linear system with the mass matrix at least once in each iteration and is the reason for mass lumping, which we describe in the next section.

3 Mass lumping

3.1 Row-sum mass lumping and its generalization

Despite the apparent downgrade in convergence rate, the row-sum technique remains very popular owing to its simplicity and straightforward implementation. We define it through the application of a lumping operator [32].

Definition 3.1 (Lumping operator)**.** Let $B \in \mathbb{R}^{n \times n}$. The lumping operator $\mathcal{L} \colon \mathbb{R}^{n \times n} \to \mathbb{R}^{n \times n}$ is defined as

$$\mathcal{L}(B) = \operatorname{diag}(d_1, \ldots, d_n)$$

where $d_i = \sum_{j=1}^{n} |b_{ij}|$ for $i = 1, \ldots, n$.

Since most mass lumping/scaling methods are defined algebraically as modifications to the consistent mass matrix, it is convenient to introduce an order relation between symmetric matrices.

Definition 3.2 (Loewner partial order)**.** For two symmetric matrices $A, B \in \mathbb{R}^{n \times n}$, we write $A \succeq B$ (respectively $A \succ B$) if $A - B$ is positive semidefinite (respectively positive definite).

The Loewner partial order is a natural choice in our context since it is the matrix equivalent of bounding bilinear forms. Indeed, if M and $\tilde{M}$ are some matrices stemming from the symmetric bilinear forms $b, \tilde{b} \colon V_h \times V_h \to \mathbb{R}$, respectively, then

$$\tilde{b}(u_h, u_h) \geq b(u_h, u_h) \quad \forall u_h \in V_h \iff \tilde{M} \succeq M.$$

When the construction of $\tilde{M}$ is completely algebraic, we usually do not have an explicit representation of $\tilde{b}(u_h, u_h)$, but we might still bound it by understanding the relation between $\tilde{M}$ and M in the Loewner ordering. The argument has already been used to good effect in [32] where it was shown that for two symmetric positive definite matrices $A, B \in \mathbb{R}^{n \times n}$, the generalized eigenvalues of $(A, \mathcal{L}(B))$ are always smaller or equal to those of (A, B), which is a consequence of the fact that $\mathcal{L}(B) \succeq B$ [32, Corollary 3.10]. The authors in [32] used eigenvalue bounds to prove the result but there exist multiple short and elegant ways of reaching the same conclusion. For instance, if B is nonnegative and one thinks of it as a weighted adjacency matrix, then $\mathcal{L}(B)$ is the degree matrix and $\mathcal{L}(B) - B$ is its graph Laplacian, which plays a prominent role in spectral clustering techniques [43]. A direct computation then shows that $\mathbf{x}^T(\mathcal{L}(B) - B)\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n}(x_i - x_j)^2 b_{ij} \geq 0$ for any $\mathbf{x} \in \mathbb{R}^n$. This technique will be employed later in our paper.

Unfortunately, in the context of isogeometric analysis, the row-sum technique performs poorly in higher dimensions, even for moderate spline degrees. Therefore, in [32], the authors considered increasing the bandwidth in order to improve the accuracy, while still underestimating the generalized eigenvalues of (A, B). It led to defining a finite partially ordered sequence of banded matrices with increasing bandwidth

$$\mathcal{L}(B) = P_1 \succeq P_2 \succeq \cdots \succeq P_{n-1} \succeq P_n = B$$

where P_1 coincides with the usual row-sum lumped mass matrix, P_2 is tridiagonal, P_3 is pentadiagonal and so forth. This strategy was then generalized to higher dimensions for Kronecker product matrices by lumping the factor matrices defining the Kronecker product.

3.2 Block mass lumping

In general, for nontrivial geometries and coefficients, the mass matrix cannot be expressed as a Kronecker product and the techniques described in [22, 16, 12] are not straightforwardly applicable. In [32], the authors suggested computing the nearest Kronecker product approximation and substituting it to the consistent mass. However, the loss of accuracy induced by the approximation depends on the singular value decay of a reordered matrix and is independent of the discretization parameters. Not only may the approximation be rather crude but combining it with mass lumping also does not give a theoretical guarantee of improving the CFL, although it was observed in all practically relevant cases. Both issues will be addressed in this section. The key observation is that, although the mass matrix generally cannot be expressed as a Kronecker product, it still inherits some favorable structure from the tensor product basis functions which we will exploit for designing algebraic mass lumping techniques. We will first describe this structure in detail for the single-patch case and then propose a block generalization of the methods presented in [32]. From there, the multipatch case naturally follows.

Thanks to the local support property of the basis functions commonly used in isogeometric analysis (e.g. B-splines and NURBS), maximally smooth discretizations of 1D problems lead to banded matrices, where the bandwidth is the spline degree p [44]. For higher dimensional problems, the tensor product construction of the basis functions leads to hierarchical banded matrices that were defined inductively in [44]. Without loss of generality, the framework introduced in this section assumes a lexicographical type labeling of the degrees of freedom, which may always be recovered after a suitable reordering of the system matrices.

Definition 3.3 (1-level banded matrix)**.** A matrix $B \in \mathbb{R}^{n \times n}$ is called *1-level banded* (or simply *banded*) with bandwidth b if

$$|i - j| > b \implies b_{ij} = 0 \quad i, j = 1, \dots, n.$$

Definition 3.4 (d-level banded matrix)**.** A block matrix $\mathcal{B} \in \mathbb{R}^{n_1 n_2 \dots n_d \times n_1 n_2 \dots n_d}$ partitioned as

$$\mathcal{B} = \begin{pmatrix} B_{1,1} & \cdots & B_{1,n_1} \\ \vdots & \ddots & \vdots \\ B_{n_1,1} & \cdots & B_{n_1,n_1} \end{pmatrix}$$

is called *d-level banded* with bandwidths $(b_1, b_2, \dots, b_d)$ if each block $B_{i,j} \in \mathbb{R}^{n_2 \dots n_d \times n_2 \dots n_d}$ is $(d-1)$-level banded with bandwidths $(b_2, \dots, b_d)$ and

$$|i - j| > b_1 \implies B_{i,j} = 0 \quad i, j = 1, \dots, n_1.$$

This hierarchical notion of bandedness is related to the standard notion of bandedness through the bandwidths and block sizes. In the rest of the paper we will denote $r_k = \prod_{j=k+1}^{d} n_j$ the size of the matrices on the kth hierarchical level (with $k < d$). On the finest level, the "blocks" reduce to scalars and we set $r_d = 1$.

Lemma 3.5. Let $\mathcal{B} \in \mathbb{R}^{n_1 n_2 \dots n_d \times n_1 n_2 \dots n_d}$ be d-level banded with bandwidths $\mathbf{b} = (b_1, b_2, \dots, b_d)$ and block sizes $\mathbf{r} = (r_1, r_2, \dots, r_d)$. Then $\mathcal{B}$ has bandwidth

$$\mathbf{b} \cdot \mathbf{r} = \sum_{i=1}^{d} b_i r_i. \tag{3.1}$$

Proof. The proof is by induction on the dimension d. For $d = 1$, $\mathcal{B}$ is 1-level banded (i.e. banded) with bandwidth b_d and the property (3.1) holds since $r_d = 1$. We now verify the property for dimension d assuming it holds for dimension $d - 1$. Let $\mathcal{B}$ be d-level banded with bandwidths $\mathbf{b} = (b_1, b_2, \dots, b_d)$. Then, by definition,

$$\mathcal{B} = \begin{pmatrix} B_{1,1} & \cdots & B_{1,n_1} \\ \vdots & \ddots & \vdots \\ B_{n_1,1} & \cdots & B_{n_1,n_1} \end{pmatrix}$$

and each block $B_{i,j}$ is $(d-1)$-level banded with bandwidths $(b_2, \dots, b_d)$ and $B_{i,j} = 0$ if $|i - j| > b_1$. Thus, the bandwidth of $\mathcal{B}$ is given by the sum of $b_1 r_1$ and the bandwidth of $B_{i,j}$. Since $B_{i,j}$ is $(d-1)$-level banded, the induction hypothesis completes the proof. □

Definitions 3.3 and 3.4 only provide a description of the sparsity. In order to build a working framework, they must be complemented with Definitions 3.6 and 3.7 that describe both symmetry and spectral properties.

Definition 3.6. The sets of symmetric positive semidefinite (SPSD) and symmetric positive definite (SPD) matrices of size n are defined, respectively, as

$$\mathcal{S}_n = \{B \in \mathbb{R}^{n \times n} \colon B = B^T, B \succeq 0\} \quad \text{and} \quad \mathcal{S}_n^+ = \{B \in \mathbb{R}^{n \times n} \colon B = B^T, B \succ 0\}.$$

The next definition provides a hierarchical generalization.

Definition 3.7. For block matrices $\mathcal{B} \in \mathbb{R}^{n_1 n_2 \dots n_d \times n_1 n_2 \dots n_d}$ partitioned as

$$\mathcal{B} = \begin{pmatrix} B_{1,1} & \cdots & B_{1,n_1} \\ \vdots & \ddots & \vdots \\ B_{n_1,1} & \cdots & B_{n_1,n_1} \end{pmatrix}$$

where $B_{i,j} \in \mathbb{R}^{n_2 \dots n_d \times n_2 \dots n_d}$, we define the sets

$$\mathcal{S}_{(n_1,n_2,\dots,n_d)} = \{\mathcal{B} \in \mathcal{S}_n \colon B_{i,j} \in \mathcal{S}_{(n_2,\dots,n_d)}\} \quad \text{and} \quad \mathcal{S}^+_{(n_1,n_2,\dots,n_d)} = \{\mathcal{B} \in \mathcal{S}_n^+ \colon B_{i,j} \in \mathcal{S}_{(n_2,\dots,n_d)}\}$$

where $n = \prod_{i=1}^{d} n_i$.

Given a vector $\mathbf{n} = (n_1, n_2, \dots, n_d)$, we will often denote the corresponding sets $\mathcal{S}_{\mathbf{n}}$ and $\mathcal{S}_{\mathbf{n}}^+$, respectively. Clearly, if $\mathcal{B} \in \mathcal{S}_{\mathbf{n}}$ (or $\mathcal{S}_{\mathbf{n}}^+$) is d-level banded with bandwidths $\mathbf{b} = (b_1, b_2, \dots, b_d)$, then $\mathbf{b} \le \mathbf{n} - \mathbf{1}$ componentwise, where $\mathbf{1}$ is the vector of all ones.

Lemma 3.8. For any vector $\mathbf{n} = (n_1, n_2, \dots, n_d) \in \mathbb{N}^d$,

$$\begin{aligned} &\mathcal{S}_{(n_1,n_2,\dots,n_d)} \subseteq \mathcal{S}_{(n_1,n_2,\dots,n_{d-1},r_{d-1})} \subseteq \mathcal{S}_{(n_1,n_2,\dots,n_{d-2},r_{d-2})} \subseteq \cdots \subseteq \mathcal{S}_{(n_1,r_1)} \subseteq \mathcal{S}_n, \\ &\mathcal{S}^+_{(n_1,n_2,\dots,n_d)} \subseteq \mathcal{S}^+_{(n_1,n_2,\dots,n_{d-1},r_{d-1})} \subseteq \mathcal{S}^+_{(n_1,n_2,\dots,n_{d-2},r_{d-2})} \subseteq \cdots \subseteq \mathcal{S}^+_{(n_1,r_1)} \subseteq \mathcal{S}^+_n. \end{aligned}$$

Proof. We prove the inclusions from left to right. Let $\mathcal{B} \in \mathcal{S}_{(n_1,n_2,\dots,n_d)}$. By definition, on the $(d-1)$th hierarchical level the matrices are in $\mathcal{S}_{n_d} = \mathcal{S}_{r_{d-1}}$ and the first inclusion trivially follows. On the $(d-2)$th level the matrices are in $\mathcal{S}_{(n_{d-1},r_{d-1})} \subseteq \mathcal{S}_{r_{d-2}}$. Thus, $\mathcal{B} \in \mathcal{S}_{(n_1,n_2,\dots,n_{d-2},r_{d-2})}$. We then repeatedly apply the same argument by moving up the hierarchical structure and realizing that at level k (with $1 \le k \le d-2$) the matrices are in $\mathcal{S}_{(n_{k+1},r_{k+1})} \subseteq \mathcal{S}_{r_k}$. The proof of the second statement is completely analogous. □

Thus, d-level banded matrices in $\mathcal{S}_{\mathbf{n}}$ (or $\mathcal{S}_{\mathbf{n}}^+$) have a hierarchical block banded structure with SPSD blocks. The following lemma shows that the isogeometric mass matrix falls in this category.

Lemma 3.9. Let $\mathcal{M} \in \mathbb{R}^{n_1 \dots n_d \times n_1 \dots n_d}$ be a d-dimensional isogeometric single-patch mass matrix with associated dimensions vector $\mathbf{n} = (n_1, n_2, \dots, n_d)$. Then

1. $\mathcal{M}$ is d-level banded,
2. $\mathcal{M} \in \mathcal{S}^+_{\mathbf{n}}$.

Proof. We prove the two statements below.

1. The key observation is noticing that the mass matrices $\mathcal{M}$ and $\hat{\mathcal{M}}$ in the physical and parametric domains, respectively, have the same sparsity pattern and therefore the same hierarchical block bandedness. Indeed, their entries are defined as (see (2.3))
$$\mathcal{M}_{ij} = \int_{\hat{\Omega}} c(\hat{\mathbf{x}})\hat{B}_i(\hat{\mathbf{x}})\hat{B}_j(\hat{\mathbf{x}}) \quad \text{and} \quad \hat{\mathcal{M}}_{ij} = \int_{\hat{\Omega}} \hat{B}_i(\hat{\mathbf{x}})\hat{B}_j(\hat{\mathbf{x}}).$$
From the positivity of the B-spline (or NURBS) basis and the fact that $c(\hat{\mathbf{x}}) := \rho(F(\hat{\mathbf{x}}))|\det(J_F(\hat{\mathbf{x}}))| > 0$, it follows that $\mathcal{M}_{ij} = 0 \iff \hat{\mathcal{M}}_{ij} = 0$. Thus, $\mathcal{M}$ and $\hat{\mathcal{M}}$ have the same sparsity pattern. Moreover, since $\hat{\mathcal{M}}$ is the mass matrix in the parametric domain
$$\hat{\mathcal{M}} = \bigotimes_{i=1}^{d} \hat{M}_i$$
where $\hat{M}_i \in \mathbb{R}^{n_i \times n_i}$ is banded with bandwidth b_i for $i = 1, \dots, d$. Thus, by definition, $\hat{\mathcal{M}}$ is d-level banded with bandwidths $(b_1, b_2, \dots, b_d)$ and consequently so is $\mathcal{M}$.
2. We start at the top of the hierarchical structure and work our way downward. Firstly, since the mass matrix is symmetric positive definite $\mathcal{M} \in \mathcal{S}^+_n$. Secondly, it may be written as
$$\mathcal{M} = \begin{pmatrix} M_{1,1} & \cdots & M_{1,n_1} \\ \vdots & \ddots & \vdots \\ M_{n_1,1} & \cdots & M_{n_1,n_1} \end{pmatrix} \tag{3.2}$$
where $M_{i,j} \in \mathbb{R}^{r_1 \times r_1}$. We will show that $M_{i,j} \in \mathcal{S}_{r_1}$ for all $i, j = 1, \dots n_1$. Let $\hat{\mathbf{B}}_l \in \mathbb{R}^{n_l}$ denote the vector of basis functions $\{\hat{B}_{li}\}_{i=1}^{n_l}$ along the lth direction in the parametric domain. The matrix $M_{i,j}$ is then given by
$$M_{i,j} = \int_{\hat{\Omega}} \underbrace{c(\hat{\mathbf{x}})\hat{B}_{1i}\hat{B}_{1j}}_{g_{ij}} \bigotimes_{l=2}^{d} \hat{\mathbf{B}}_l\hat{\mathbf{B}}_l^T = \int_{\hat{\Omega}} g_{ij} \bigotimes_{l=2}^{d} \hat{\mathbf{B}}_l\hat{\mathbf{B}}_l^T. \tag{3.3}$$
From (3.3), $M_{i,j}$ is evidently symmetric. Moreover, thanks to the pointwise nonnegativity of the basis functions, $g_{ij} \geq 0$ and consequently, for any vector $\mathbf{x} \in \mathbb{R}^{r_1}$,
$$\mathbf{x}^T M_{i,j}\mathbf{x} = \int_{\hat{\Omega}} g_{ij}(\mathbf{x}^T \bigotimes_{l=2}^{d} \hat{\mathbf{B}}_l)^2 = \int_{\hat{\Omega}} g_{ij} v^2 \geq 0 \tag{3.4}$$
where $v = \mathbf{x}^T \bigotimes_{l=2}^{d} \hat{\mathbf{B}}_l$ is a function in a finite element subspace. Thus, (3.3) and (3.4) together show that $M_{i,j} \in \mathcal{S}_{r_1}$. Finally, since $\mathcal{M} \in \mathcal{S}^+_n$ and $M_{i,j} \in \mathcal{S}_{r_1}$ for all $i, j = 1, \dots n_1$, then $\mathcal{M} \in \mathcal{S}^+_{(n_1, r_1)}$ by definition. We now repeat the same argument by first showing that each $M_{i,j}$ can itself be expressed as a block matrix similarly to (3.2) with blocks of size $r_2 \times r_2$. By repeating the arguments in (3.3) and (3.4) one easily shows that each of these blocks is in $\mathcal{S}_{r_2}$ and consequently $M_{i,j} \in \mathcal{S}_{(n_2, r_2)}$. Finally, moving up one level, we deduce that $\mathcal{M} \in \mathcal{S}^+_{(n_1, n_2, r_2)}$. By recursively applying the same arguments, we finally prove that $\mathcal{M} \in \mathcal{S}^+_{\mathbf{n}}$.

□

Remark 3.10. For single-patch isogeometric discretizations, the dimensions vector $\mathbf{n}$ corresponds to the number of basis functions along each parametric direction and is always uniquely determined by the number of subdivisions, order, smoothness and boundary conditions. Moreover, for maximally smooth discretizations, the bandwidths are equal to the spline degrees such that $\mathbf{b} = \mathbf{p}$.

The definitions above also accommodate vector-valued PDEs such as linear elasticity. In this context, the mass matrix is a $(d+1)$-level banded matrix of bandwidths $(0, b_1, \dots, b_d)$ (i.e. a block diagonal matrix). Moreover, if each component of the solution is discretized using the same scalar spaces, then $\mathcal{M} \in \mathcal{S}^+_{(d,\mathbf{n})}$, where $\mathbf{n}$ is the dimensions vector for a scalar problem.

We stress that Lemma 3.9 is a sole consequence of the tensor product construction of the basis functions and their pointwise nonnegativity and does not depend on the geometry mapping. In particular, it shows that the isogeometric mass matrix is *not only* symmetric positive definite, but actually enjoys additional structure. For instance $\mathcal{S}^+_{(n_1,n_2)}$, typically encountered for 2-dimensional discretizations, is the set of SPD block matrices with SPSD blocks. This structure is key to extending mass lumping techniques to nontrivial problems in dimension $d \geq 2$. We now define the block analogue of the lumping operator introduced in [32].

Definition 3.11 (Block lumping operator)**.** Let $\mathcal{B} \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$ be a block matrix partitioned as

$$\mathcal{B} = \begin{pmatrix} B_{1,1} & \cdots & B_{1,n_1} \\ \vdots & \ddots & \vdots \\ B_{n_1,1} & \cdots & B_{n_1,n_1} \end{pmatrix}$$

where $B_{i,j} \in \mathbb{R}^{n_2 \times n_2}$. The block lumping operator $\mathcal{L}$ is defined as

$$\mathcal{L}(\mathcal{B}) = \operatorname{diag}(D_1, \ldots, D_{n_1}) := \begin{pmatrix} D_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_{n_1} \end{pmatrix}$$

where $D_i = \sum_{j=1}^{n_1} B_{i,j}$ for $i = 1, \ldots, n_1$.

Whereas the (scalar) lumping operator in Definition 3.1 returns a diagonal matrix, the block lumping operator returns a block diagonal matrix. We establish some useful consequences of this definition for the sets $\mathcal{S}_{\mathbf{n}}$ and $\mathcal{S}^+_{\mathbf{n}}$.

Lemma 3.12. For any vector $\mathbf{n} = (n_1, n_2, \ldots, n_d) \in \mathbb{N}^d$,

$$\mathcal{L}(\mathcal{S}_{\mathbf{n}}) \subseteq \mathcal{S}_{\mathbf{n}} \quad \text{and} \quad \mathcal{L}(\mathcal{S}^+_{\mathbf{n}}) \subseteq \mathcal{S}^+_{\mathbf{n}}.$$

Proof. Let $\mathcal{B} \in \mathcal{S}_{\mathbf{n}}$. The result for $d = 1$ is obvious from Definition 3.1. Now assume that $d \geq 2$ and let $\mathcal{L}(\mathcal{B})$ be constructed following Definition 3.11. The proof simply follows from the stability of $\mathcal{S}_{\mathbf{n}}$ under addition: since $B_{i,j} \in \mathcal{S}_{(n_2,\ldots,n_d)}$ for all $i, j = 1, \ldots n_1$, then

$$D_i = \sum_{j=1}^{n_1} B_{i,j} \in \mathcal{S}_{(n_2,\ldots,n_d)}.$$

Since $\mathcal{L}(\mathcal{B})$ is block diagonal with SPSD blocks, it is itself SPSD and $\mathcal{L}(\mathcal{B}) \in \mathcal{S}_{\mathbf{n}}$. The proof of the second statement is completely analogous (noting that for matrices $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$ all diagonal blocks and diagonal sub-blocks down the hierarchy are positive definite). □

The next lemma is the block generalization of [32, Lemma 3.9].

Lemma 3.13. Let $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$ with $\mathbf{n} = (n_1, \ldots, n_d) \in \mathbb{N}^d$ and $d \geq 2$. Then,

$$\mathcal{L}(\mathcal{B}) \succeq \mathcal{B}.$$

Proof. Let $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$,

$$\mathcal{B} = \begin{pmatrix} B_{1,1} & \cdots & B_{1,n_1} \\ \vdots & \ddots & \vdots \\ B_{n_1,1} & \cdots & B_{n_1,n_1} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{n_1} \end{pmatrix}.$$

Then, using the fact that $B_{i,j} = B^T_{j,i} = B_{j,i}$ and $B_{i,j} \succeq 0$ for all $i, j = 1, \ldots, n_1$,

$$\begin{aligned} \mathbf{x}^T(\mathcal{L}(\mathcal{B}) - \mathcal{B})\mathbf{x} &= \sum_{i=1}^{n_1} \mathbf{x}_i^T \left(\sum_{j=1}^{n_1} B_{i,j} \right) \mathbf{x}_i - \sum_{i,j=1}^{n_1} \mathbf{x}_i^T B_{i,j} \mathbf{x}_j \\ &= \frac{1}{2} \left(\sum_{i=1}^{n_1} \mathbf{x}_i^T \left(\sum_{j=1}^{n_1} B_{i,j} \right) \mathbf{x}_i - 2 \sum_{i,j=1}^{n_1} \mathbf{x}_i^T B_{i,j} \mathbf{x}_j + \sum_{j=1}^{n_1} \mathbf{x}_j^T \left(\sum_{i=1}^{n_1} B_{j,i} \right) \mathbf{x}_j \right) \\ &= \frac{1}{2} \sum_{i,j=1}^{n_1} \mathbf{x}_i^T B_{i,j} \mathbf{x}_i - 2\mathbf{x}_i^T B_{i,j} \mathbf{x}_j + \mathbf{x}_j^T B_{i,j} \mathbf{x}_j \\ &= \frac{1}{2} \sum_{i,j=1}^{n_1} (\mathbf{x}_i - \mathbf{x}_j)^T B_{i,j} (\mathbf{x}_i - \mathbf{x}_j) \geq 0, \end{aligned}$$

which proves that $\mathcal{L}(\mathcal{B}) - \mathcal{B} \succeq 0$. □

Remark 3.14. Interestingly, Lemma 3.13 also holds for the larger set of symmetric block matrices with SPSD blocks. Moreover, denoting $\mathbf{e}$ the vector of all ones, $(1, \mathbf{e})$ is an eigenpair of $(\mathcal{B}, \mathcal{L}(\mathcal{B}))$ regardless of whether $\mathcal{B}$ is nonnegative. For nonnegative matrices and $d = 1$, our results simply reduce to those of [32].

In [32], the authors considered the matrix splitting $B = D_i + R_i$, where D_i consists of all super and sub-diagonals strictly smaller than i and R_i is the remainder. Lumped matrices were then defined by lumping the remainder R_i and adding it to D_i. The block lumped matrices introduced in Definition 3.15 are the block analogue of those constructed in [32] and feature blockwise operations instead of entrywise operations.

Definition 3.15 (Block lumped matrices)**.** Let $\mathcal{B} \in \mathcal{S}_{\mathbf{n}}^{+}$ with $\mathbf{n} \in \mathbb{N}^d$ and $d \geq 2$ and consider the matrix splitting $\mathcal{B} = \mathcal{D}_i + \mathcal{R}_i$ where $\mathcal{D}_i$ consists of all super and sub block diagonals strictly smaller than i and $\mathcal{R}_i$ is the remainder. We define the sequence of matrices $\mathcal{P}_i = \mathcal{D}_i + \mathcal{L}(\mathcal{R}_i)$ for $i = 1, \dots, n_1$. In particular, we observe that $\mathcal{P}_1 = \mathcal{L}(\mathcal{B})$ and $\mathcal{P}_{n_1} = \mathcal{B}$.

By construction, $\mathcal{P}_i$ only reduces the highest hierarchical level of bandedness: if $\mathcal{B}$ is d-level banded with bandwidths $(b_1, b_2, \dots, b_d)$, then $\mathcal{P}_i$ (with $i \leq b_1 + 1$) is d-level banded with bandwidths $(i-1, b_2, \dots, b_d)$. Figure 3.1 shows an example for a 2-level banded matrix $\mathcal{P}_2$ with bandwidths $(1, 3)$ (block tridiagonal matrix) constructed from a 2-level banded matrix $\mathcal{B}$ with bandwidths $(3, 3)$ (block septadiagonal matrix). The next theorem provides a generalization of [32, Theorem 3.21]. From now on, we will always assume that the eigenvalues are ordered increasingly.

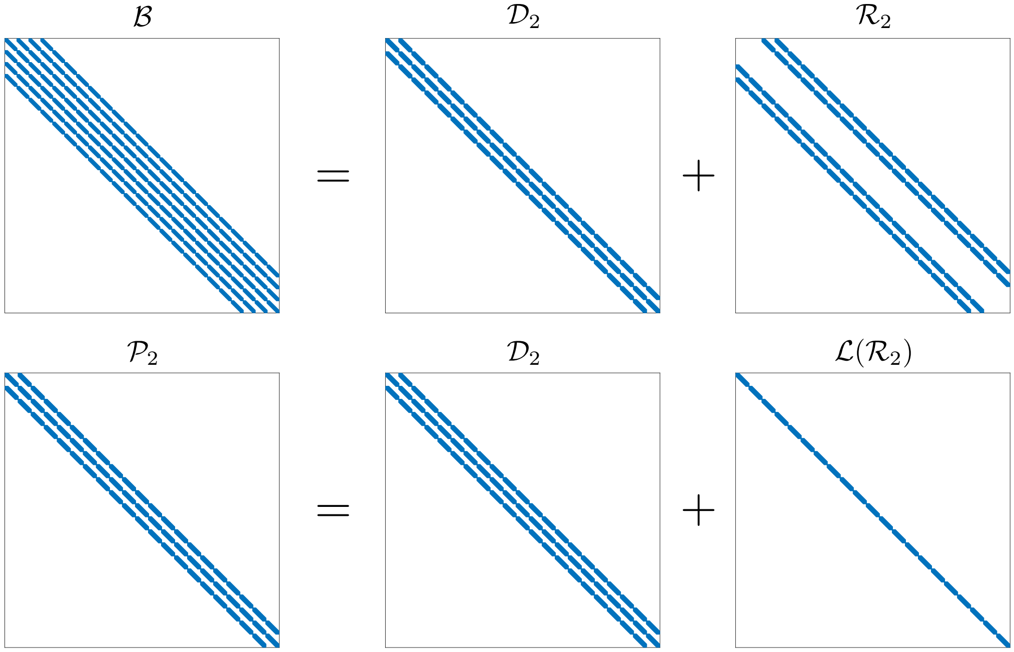


Figure 3.1: Block tridiagonal matrix $\mathcal{P}_2$ constructed from a block septadiagonal matrix $\mathcal{B}$

Theorem 3.16. Let $\mathcal{B} \in \mathcal{S}_{\mathbf{n}}^{+}$ with $\mathbf{n} = (n_1, \dots, n_d) \in \mathbb{N}^d$ and $d \geq 2$. Then, the sequence of matrices $\{\mathcal{P}_i\}_{i=1}^{n_1}$ constructed from $\mathcal{B}$ according to Definition 3.15 satisfies the following properties:

1. $\Lambda(\mathcal{B}, \mathcal{P}_i) \subset (0, 1]$ for all $i = 1, \dots, n_1$,
2. $\lambda_k(\mathcal{B}, \mathcal{P}_i) \leq \lambda_k(\mathcal{B}, \mathcal{P}_{i+1})$ for all k and any given $i = 1, \dots, n_1 - 1$,
3. $\lambda_n(\mathcal{B}, \mathcal{P}_i) = 1$ for all $i = 1, \dots, n_1$.

Proof. The proof is analogous to [32, Theorem 3.21] using Lemma 3.13 and Remark 3.14. □

The proof arguments of Theorem 3.16 also show that the block lumped matrices satisfy

$$\mathcal{L}(\mathcal{B}) = \mathcal{P}_1 \succeq \mathcal{P}_2 \succeq \dots \succeq \mathcal{P}_{n_1 - 1} \succeq \mathcal{P}_{n_1} = \mathcal{B}. \tag{3.5}$$

This ordering implies that for a matrix $\mathcal{A} \in \mathcal{S}_n$,

$$\lambda_k(\mathcal{A}, \mathcal{L}(\mathcal{B})) = \lambda_k(\mathcal{A}, \mathcal{P}_1) \leq \lambda_k(\mathcal{A}, \mathcal{P}_2) \leq \dots \leq \lambda_k(\mathcal{A}, \mathcal{P}_{n_1 - 1}) \leq \lambda_k(\mathcal{A}, \mathcal{P}_{n_1}) = \lambda_k(\mathcal{A}, \mathcal{B}) \qquad 1 \leq k \leq n.$$

Clearly, block diagonal matrices such as $\mathcal{P}_1$ are very appealing given that linear systems can be solved in parallel for each block. The block tridiagonal case can still be treated efficiently by a block forward elimination and backward substitution algorithm; see [45, Section 4.5.1] for the details.

3.3 Hierarchical mass lumping

Mass lumping can also be applied down the hierarchical structure in many different ways. The following lemma sets the foundation and generalizes Lemma 3.12.

Lemma 3.17. If $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$ with $\mathbf{n} = (n_1, \dots, n_d) \in \mathbb{N}^d$, then $\mathcal{P}_k \in \mathcal{S}^+_{\mathbf{n}}$ for all $k = 1, \dots, n_1$.

Proof. The case $d = 1$ was already proved in [32, Theorem 3.21]. For $d \geq 2$, by definition $\mathcal{P}_k = \mathcal{D}_k + \mathcal{L}(\mathcal{R}_k)$ and is partitioned similarly to (3.2) with blocks $P_{i,j}$ for $i, j = 1, \dots, n_1$. Since $\mathcal{S}_{\mathbf{n}}$ is closed under addition, $P_{i,j} \in S_{(n_2,\dots,n_d)}$ for all $i, j = 1, \dots, n_1$. Moreover, $\mathcal{P}_k \succeq \mathcal{B} \succ 0$ implies that $\mathcal{P}_k \in \mathcal{S}^+_n$ and consequently $\mathcal{P}_k \in S^+_{\mathbf{n}}$. □

Although $\mathcal{P}_1 = \mathcal{L}(\mathcal{B})$ is block diagonal, for high-dimensional problems the size of each block may still be quite large. However, following Lemma 3.17, $\mathcal{P}_1 \in \mathcal{S}^+_{\mathbf{n}}$, which suggests recursively applying the lumping operator on its diagonal blocks, which are in $\mathcal{S}^+_{(n_2,\dots,n_d)}$. On the second to last level, all diagonal blocks are in $\mathcal{S}^+_{n_d}$. At this stage, if $\mathcal{B}$ is nonnegative, applying the standard row-sum technique results in the standard row-sum lumped mass matrix. As we progress down the hierarchical structure, the number of diagonal blocks increases but their size decreases. Indeed, on the kth level, the number of diagonal blocks is $q_k = \prod_{j=1}^k n_j$ and their size is $r_k = \prod_{j=k+1}^d n_j$ such that for all k the product $q_k r_k = n$ is the size of the full matrix. The following definition formalizes the procedure.

Definition 3.18 (Hierarchical lumped matrices)**.** Let $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$ with $\mathbf{n} = (n_1, \dots, n_d) \in \mathbb{N}^d$ and $d \geq 2$. Set $\mathcal{H}_1 = \mathcal{L}(\mathcal{B})$ and let $\mathcal{H}_k$ for $1 \leq k \leq d-1$ be such that

$$\mathcal{H}_k = \operatorname{diag}(D_{k,1}, \dots, D_{k,q_k})$$

where $q_k = \prod_{j=1}^k n_j$. Then, $\mathcal{H}_{k+1}$ is defined from $\mathcal{H}_k$ as

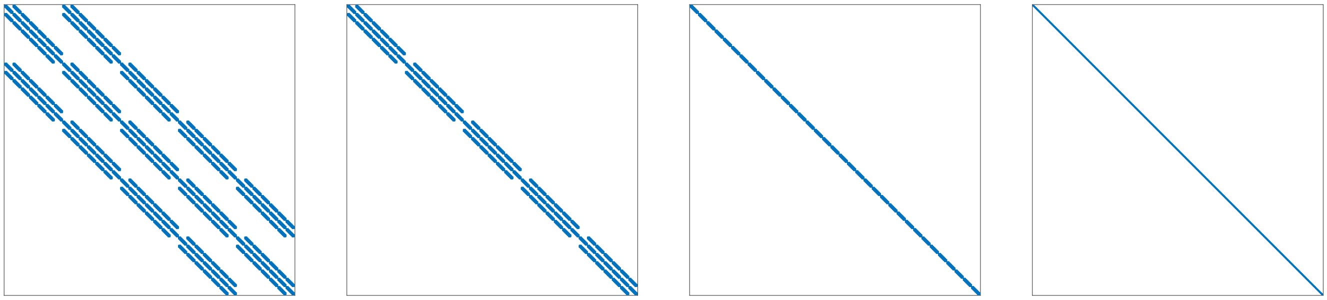
$$\mathcal{H}_{k+1} = \operatorname{diag}(\mathcal{L}(D_{k,1}), \dots, \mathcal{L}(D_{k,q_k})).$$

Figure 3.2, for example, shows the sparsity pattern of a matrix $\mathcal{B} \in \mathcal{S}^+_{(n_1,n_2,n_3)}$ together with its hierarchical lumped mass matrices $\mathcal{H}_k$ for $k = 1, 2, 3$. By construction, hierarchical mass lumping reduces the bandwidth down the hierarchical structure: if $\mathcal{B}$ is d-level banded with bandwidths $(b_1, b_2, \dots, b_d)$, then $\mathcal{H}_k$ is d-level banded with bandwidths $(0, \dots, 0, b_{k+1}, \dots, b_d)$. Similarly to (3.5), hierarchical lumped matrices also satisfy an order relation.

Corollary 3.19. Let $\mathcal{B} \in \mathcal{S}^+_{\mathbf{n}}$ with $\mathbf{n} \in \mathbb{N}^d$ and $d \geq 2$. Then, the sequence of matrices $\{\mathcal{H}_k\}_{k=1}^d$ constructed from $\mathcal{B}$ according to Definition 3.18 satisfies

$$\mathcal{H}_d \succeq \mathcal{H}_{d-1} \succeq \cdots \succeq \mathcal{H}_1.$$

Proof. The proof is an obvious consequence of Lemma 3.13. □



(a) $\mathcal{B}$ (b) $\mathcal{H}_1$ (c) $\mathcal{H}_2$ (d) $\mathcal{H}_3$

Figure 3.2: Sparsity patterns

3.4 Multipatch mass lumping

We recall that in the multipatch setting, the stiffness and mass matrices are expressed as

$$\mathcal{K} = \sum_{r=1}^{N_p} R_r^T \mathcal{K}_r R_r \quad \text{and} \quad \mathcal{M} = \sum_{r=1}^{N_p} R_r^T \mathcal{M}_r R_r$$

where N_p is the number of patches, $\mathcal{K}_r$ and $\mathcal{M}_r$ are the local stiffness and mass matrices of the rth patch and R_r maps its local degrees of freedom to global ones. Since $\mathcal{M}_r$ are single-patch mass matrices, it motivates the following definition of multipatch mass lumping.

Definition 3.20 (Multipatch lumped matrices)**.** Let $\mathcal{B} = \sum_{r=1}^{N_p} R_r^T \mathcal{B}_r R_r$ be a multipatch matrix, where $\mathcal{B}_r \in \mathcal{S}_{\mathbf{n}}^+$ for all $r = 1, \dots, N_p$. We define $\mathcal{P}_{\mathbf{i}} = \sum_{r=1}^{N_p} R_r^T \mathcal{P}_{r,i_r} R_r$ as a multipatch lumped matrix, where $\mathcal{P}_{r,i_r}$ is constructed from $\mathcal{B}_r$ following Definition 3.15 and $\mathbf{i} = (i_1, \dots, i_{N_p})$ is a multi-index.

For notational convenience, we will assume that the discretization parameters are identical for each patch such that we may choose $i_r = i$ for all patches $r = 1, \dots, N_p$ and simply denote $\mathcal{P}_i$ the resulting multipatch lumped mass matrix. Although this notation conflicts with the single-patch case, it will always be clear from the context whether $\mathcal{P}_i$ refers to a single-patch or multipatch lumped mass matrix. The next lemma generalizes our previous findings to the multipatch case.

Lemma 3.21. Let $\mathcal{B} = \sum_{r=1}^{N_p} R_r^T \mathcal{B}_r R_r$, where $\mathcal{B}_r \in \mathcal{S}_{\mathbf{n}}^+$ for all $r = 1, \dots, N_p$. Then the sequence of matrices $\{\mathcal{P}_i\}_{i=1}^{n_1}$ constructed from $\mathcal{B}$ following Definition 3.20 satisfies

$$\mathcal{P}_1 \succeq \mathcal{P}_2 \succeq \cdots \succeq \mathcal{P}_{n_1 - 1} \succeq \mathcal{P}_{n_1} = \mathcal{B}.$$

Proof. First recall that for any symmetric matrices $A, B \in \mathbb{R}^{n \times n}$ and any $V \in \mathbb{R}^{n \times m}$, if $A \succeq B$, then $V^T A V \succeq V^T B V$ [46, Theorem 7.7.2(a)]. The result then immediately follows since for any $1 \leq r \leq N_p$ and any index $1 \leq i < n_1$,

$$\begin{aligned} & \mathcal{P}_{r,i} \succeq \mathcal{P}_{r,i+1}, && \text{see (3.5)} \\ \implies & R_r^T \mathcal{P}_{r,i} R_r \succeq R_r^T \mathcal{P}_{r,i+1} R_r, && \\ \implies & \mathcal{P}_i = \sum_{r=1}^{N_p} R_r^T \mathcal{P}_{r,i} R_r \succeq \sum_{r=1}^{N_p} R_r^T \mathcal{P}_{r,i+1} R_r = \mathcal{P}_{i+1}. && \end{aligned}$$

The statement then follows from an inductive application of the previous inequality. □

For high-dimensional problems, it might be useful to resort to hierarchical mass lumping techniques on the single-patch level, as described in Section 3.3. There is an obvious analogue of Definition 3.20 and Lemma 3.21 for such cases.

The purpose of mass lumping is first and foremost to reduce the block bandedness of the mass matrix and guarantee a CFL condition that cannot be worse than the original one. However, generally speaking, mass lumping does not significantly improve the CFL while it might undermine the accuracy. Multipatch problems are not exempt and the issue already originates on the single-patch level, before local matrices are merged into the global one. This merging process is identical to the assembly procedure of classical finite element methods, which is not surprising given the analogy between patches and elements. Thus, the proof of the following lemma is well-known (see e.g. [47, 48]), but is repeated for the sake of completeness and notational consistency.

Lemma 3.22. Let $\mathcal{A} = \sum_{r=1}^{N_p} R_r^T \mathcal{A}_r R_r$ and $\mathcal{B} = \sum_{r=1}^{N_p} R_r^T \mathcal{B}_r R_r$, where $\mathcal{A}_r \in \mathcal{S}_n$ and $\mathcal{B}_r \in \mathcal{S}_n^+$ for all $r = 1, \dots, N_p$. Then,

$$\min_r \lambda_{\min}(\mathcal{A}_r, \mathcal{B}_r) \leq \lambda_{\min}(\mathcal{A}, \mathcal{B}), \qquad \lambda_{\max}(\mathcal{A}, \mathcal{B}) \leq \max_r \lambda_{\max}(\mathcal{A}_r, \mathcal{B}_r).$$

Proof. We first note that the assembly of global multipatch matrices can be written out more compactly as

$$\mathcal{A} = \sum_{r=1}^{N_p} R_r^T \mathcal{A}_r R_r = \mathsf{R}^T \mathsf{A} \mathsf{R}, \qquad \mathcal{B} = \sum_{i=1}^{N_p} R_r^T \mathcal{B}_r R_r = \mathsf{R}^T \mathsf{B} \mathsf{R}$$

where $\mathsf{R}^T = [R_1^T, \dots, R_{N_p}^T]$, $\mathsf{A} = \operatorname{diag}(\mathcal{A}_1, \dots, \mathcal{A}_{N_p})$ and $\mathsf{B} = \operatorname{diag}(\mathcal{B}_1, \dots, \mathcal{B}_{N_p})$. Consequently, by the Courant-Fischer theorem [46, Theorem 4.2.6],

$$\lambda_{\min}(\mathcal{A}, \mathcal{B}) = \min_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T \mathsf{R}^T \mathsf{A} \mathsf{R} \mathbf{x}}{\mathbf{x}^T \mathsf{R}^T \mathsf{B} \mathsf{R} \mathbf{x}} = \min_{\substack{\mathbf{y} \in \mathcal{V} \\ \mathbf{y} \neq \mathbf{0}}} \frac{\mathbf{y}^T \mathsf{A} \mathbf{y}}{\mathbf{y}^T \mathsf{B} \mathbf{y}} \geq \min_{\mathbf{y} \neq \mathbf{0}} \frac{\mathbf{y}^T \mathsf{A} \mathbf{y}}{\mathbf{y}^T \mathsf{B} \mathbf{y}} = \min_r \lambda_{\min}(\mathcal{A}_r, \mathcal{B}_r).$$

where $\mathcal{V}$ is the space spanned by the columns of R. Similarly,

$$\lambda_{\max}(\mathcal{A}, \mathcal{B}) = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}^T \mathsf{R}^T \mathsf{A} \mathsf{R} \mathbf{x}}{\mathbf{x}^T \mathsf{R}^T \mathsf{B} \mathsf{R} \mathbf{x}} = \max_{\substack{\mathbf{y} \in \mathcal{V} \\ \mathbf{y} \neq \mathbf{0}}} \frac{\mathbf{y}^T \mathsf{A} \mathbf{y}}{\mathbf{y}^T \mathsf{B} \mathbf{y}} \leq \max_{\mathbf{y} \neq \mathbf{0}} \frac{\mathbf{y}^T \mathsf{A} \mathbf{y}}{\mathbf{y}^T \mathsf{B} \mathbf{y}} = \max_r \lambda_{\max}(\mathcal{A}_r, \mathcal{B}_r).$$

□

The upper bound of Lemma 3.22 may be quite tight and is an incentive for acting on the single-patch matrices before assembling the global ones.

3.5 Solving linear systems with the lumped mass matrix

In the single-patch case, linear systems with the lumped mass matrices are conveniently solved using sparse Cholesky factorizations; i.e. $P = LL^T$, where L is a lower triangular matrix. In isogeometric analysis, the fill-in of the Cholesky factor is well described by the *envelope* of the matrix [49].

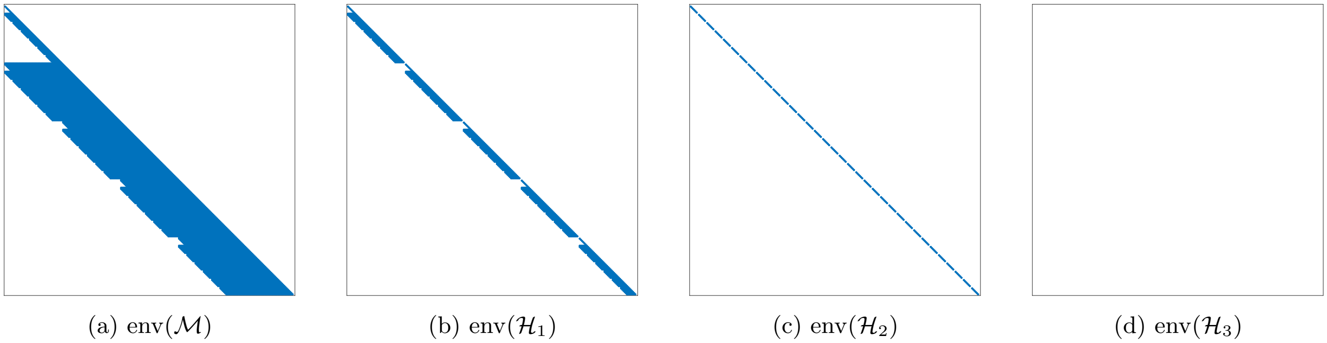
Definition 3.23. The envelope of a matrix $B \in \mathbb{R}^{n\times n}$ is defined as

$$\text{env}(B) = \{(i,j) \colon 1 \leq i \leq n,\ J_i(B) \leq j < i\}, \quad J_i(B) = \min\{j \colon 1 \leq j \leq i,\ b_{ij} \neq 0\}.$$

In words, $J_i(B)$ is simply the index of the first nonzero entry in the ith row of the lower triangular part of B. The envelopes of the consistent mass and lumped mass matrices of Figure 3.2 are shown in Figure 3.3. It is well-known that any fill-in of the Cholesky factor can only occur within the envelope [50]. For this reason, many techniques for minimizing the fill-in are based on minimizing the envelope of a permuted matrix. The mass lumping techniques discussed in this work reduce the bandwidth and envelope of the consistent mass, thereby significantly accelerating sparse direct solvers. Indeed, a direct application of Lemma 3.5 shows that the bandwidth of $\mathcal{H}_k$ is

$$(0, \dots, 0, b_{k+1}, \dots, b_d) \cdot (r_1, \dots, r_k, r_{k+1}, \dots, r_d) = \sum_{i=k+1}^{d} b_i r_i.$$

In comparison to the bandwidth of $\mathcal{B}$, the bandwidth of $\mathcal{H}_k$ suppresses the first k largest contributors to the sum. For multi-dimensional problems, it is a compelling argument for first reducing the bandwidth at the top of the hierarchy and then working our way downward.


(a) env($\mathcal{M}$)

(b) env($\mathcal{H}_1$)

(c) env($\mathcal{H}_2$)

(d) env($\mathcal{H}_3$)

Figure 3.3: Envelope of the matrices in Figure 3.2

While solving linear systems with the lumped mass matrix in the single-patch case is relatively straightforward, the multipatch case deserves some more explanations. The multipatch lumped mass matrix (after potentially a symmetric permutation) is a generalized saddle point matrix [51], expressed as

$$\mathcal{P} = \begin{pmatrix} D & C \\ C^T & X \end{pmatrix} \quad \text{with} \quad D = \text{diag}(D_1, \dots, D_{N_p}).$$

We consider the linear system

$$\begin{pmatrix} D & C \\ C^T & X \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

After block Gaussian elimination, we solve the upper block triangular system

$$\begin{pmatrix} D & C \\ 0 & S \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \tilde{\mathbf{g}} \end{pmatrix} \tag{3.6}$$

where $S = X - C^T D^{-1} C$ is the Schur complement of D in $\mathcal{P}$ and $\tilde{\mathbf{g}} = \mathbf{g} - C^T D^{-1}\mathbf{f}$. Contrary to the consistent mass matrix (which features a similar structure), the Schur complement of the lumped mass matrix can be formed explicitly and cheaply owing to its sparsity and simple block diagonal structure. Once the Schur complement is formed, which is done once and for all, (3.6) can be solved by backward substitution.

4 Outlier removal

4.1 Deflation techniques

Mass lumping generally mitigates but does not completely eliminate outlier frequencies from the spectrum. Thus, it is usually combined with dedicated outlier removal techniques. Unfortunately, the methods described in [14, 22, 16, 12] are only applicable to highly structured problems rarely met in practical applications. Moreover, numerical experiments show that predefined penalization terms barely help remove outliers and must instead be tailored to the specific problem at hand. For this reason, we present in this section a robust and algebraic outlier removal technique based on low-rank perturbations. Our strategy consists in deflating the spectrum from its largest eigenvalues, while preserving the smallest ones. Of course, this choice of scaling assumes that the dynamics are completely resolved by the low-frequency part of the spectrum, which is often (nearly) the case. We first recall some preliminary results, providing the theoretical foundation of the method.

Lemma 4.1 ([52, Theorem VI.1.15])**.** Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices with B positive definite. Then, all generalized eigenvalues of (A, B) are real and there exists an invertible matrix $U \in \mathbb{R}^{n\times n}$ such that

$$U^T A U = D, \qquad U^T B U = I,$$

where $D = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ is a real diagonal matrix containing the eigenvalues.

Definition 4.2 (Scaled matrix pencil)**.** Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices with B positive definite and f, g be two functions defined on the spectrum of (A, B). The scaled pencil $(\bar{A}, \bar{B})$ is defined as

$$\begin{aligned}\bar{A} &= A + V f(D_2) V^T,\\ \bar{B} &= B + V g(D_2) V^T,\end{aligned}$$

where $V = BU_2 \in \mathbb{R}^{n\times r}$, with $U_2 = [\mathbf{u}_{n-r+1}, \dots, \mathbf{u}_n]$ the matrix formed by the last r B-orthonormal eigenvectors of (A, B) and $D_2 = \mathrm{diag}(\lambda_{n-r+1}, \dots, \lambda_n) \in \mathbb{R}^{r\times r}$ the diagonal matrix formed by the last r eigenvalues with $r \ll n$.

The next theorem shows that this definition provides the desired scaling.

Theorem 4.3 (Deflation of matrix pencils)**.** Let $A, B \in \mathbb{R}^{n\times n}$ be symmetric matrices with B positive definite and $(\bar{A}, \bar{B})$ be the scaled pencil introduced in Definition 4.2. Then,

- The eigenvectors of (A, B) and $(\bar{A}, \bar{B})$ are the same.
- The eigenvalues of $(\bar{A}, \bar{B})$ are given by:

$$\bar{\lambda}_{i_k} = \begin{cases} \lambda_k & \text{for } k = 1, \dots, n-r,\\ \frac{\lambda_k + f(\lambda_k)}{1+g(\lambda_k)} & \text{for } k = n-r+1, \dots, n.\end{cases}$$

Proof. We first note that the matrices $\bar{A}$ and $\bar{B}$ can be rewritten as

$$\begin{aligned}\bar{A} &= A + BU\,\mathrm{diag}(0, f(D_2))U^T B,\\ \bar{B} &= B + BU\,\mathrm{diag}(0, g(D_2))U^T B,\end{aligned}$$

where $\mathrm{diag}(0, f(D_2)), \mathrm{diag}(0, g(D_2)) \in \mathbb{R}^{n\times n}$ are the block diagonal matrices obtained by appending zeros to $f(D_2)$ and $g(D_2)$, respectively, and $U = (U_1, U_2)$ is the matrix of eigenvectors. Verifying that $\mathbf{u}_i$ is an eigenvector of $(\bar{A}, \bar{B})$ for $i = 1, \dots, n$ is straightforward. Moreover, since the matrix pencils $(\bar{A}, \bar{B})$ and $(U^T\bar{A}U, U^T\bar{B}U)$ are equivalent [53, Chapter 15],

$$\Lambda(\bar{A}, \bar{B}) = \Lambda(U^T\bar{A}U, U^T\bar{B}U) = \Lambda(D + \mathrm{diag}(0, f(D_2)), I + \mathrm{diag}(0, g(D_2))) = \{\lambda_k\}_{k=1}^{n-r} \cup \left\{\frac{\lambda_k + f(\lambda_k)}{1+g(\lambda_k)}\right\}_{k=n-r+1}^{n},$$

where the second equality follows from Lemma 4.1. □

Remark 4.4. In numerical linear algebra, *deflation* refers to the removal of unwanted eigenvalues. The result of Theorem 4.3 is analogous to deflation "by substraction", which originated from the early work of Hotelling [54]. The reader may refer to [53, 55] for an overview of deflation techniques.

The previous theorem allows to map the largest eigenvalues of (A, B) to virtually any real number. However, the transformation must be carefully chosen such that it does not reduce too much the outlier frequencies. Indeed, since the eigenvectors are not affected by the transformation, if outlier frequencies are mapped to low frequencies, their spurious eigenvectors will artificially enter the solution, which might have disastrous consequences for the dynamics. We give below some suitable choices for the functions f and g that avoid this issue.

1. Set $f(\lambda) = \lambda_{n-r} - \lambda$ and $g(\lambda) = 0$.
2. Set $f(\lambda) = 0$ and $g(\lambda) = \frac{\lambda}{\lambda_{n-r}} - 1$.
3. More generally, choose any function $g(\lambda)$ (defined on the spectrum of (A, B)) and set $f(\lambda) = \lambda_{n-r}(1+g(\lambda))-\lambda$.

In the first case, a negative semidefinite perturbation is added to the stiffness matrix while in the second case, a positive semidefinite perturbation is added to the mass matrix. The latter has already been proposed in [27, 28] as a mass scaling strategy (referred therein as *spectral scaling* and *mass tailoring*, respectively). The nature of the perturbation in the third case depends on the specific choice of functions. In the remaining part of the article, we will confine ourselves to the choices of f and g listed above, which all lead to the same transformed eigenvalues, given by

$$\bar{\lambda}_k = \begin{cases} \lambda_k & \text{for } k = 1, \dots, n-r, \\ \lambda_{n-r} & \text{for } k = n-r+1, \dots, n. \end{cases}$$

The result, graphically illustrated in Figure 4.1, consists in shaving off the upper part of the spectrum. Note that in our context λ_{n-r} is the largest "regular" (or non-outlier) eigenvalue. In principle, it could be replaced with a cutoff value, as suggested in [27, 28], to avoid computing an additional eigenvalue. However, choosing λ_{n-r} preserves the eigenvalue numbers and prevents spurious eigenfunctions from moving to the lower to intermediate frequency part of the spectrum. Thus, we prefer computing this additional eigenvalue, which in practice barely introduces any overhead.

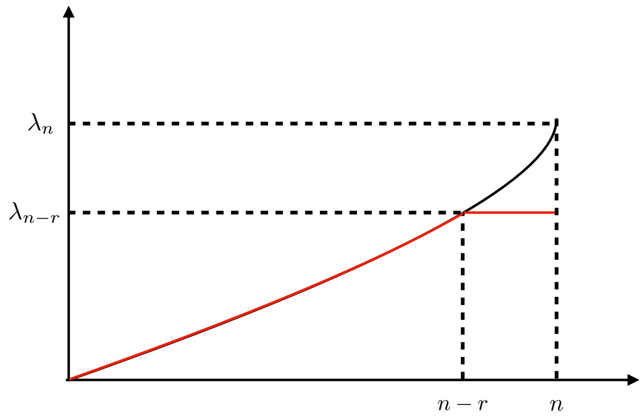


Figure 4.1: Truncation of the largest eigenvalues

Clearly, once the $r+1$ largest eigenpairs have been computed, the increase in critical time step is known. For instance, for the central difference method in the undamped case (see (1.1))

$$\frac{\bar{\Delta t}_c}{\Delta t_c} = \sqrt{\frac{\lambda_n}{\lambda_{n-r}}}.$$

On the one hand, Theorem 4.3 circumvents the lack of robustness from predefined perturbations terms. On the other hand, the cost for its implementation is also much higher since it involves a few eigenpairs, which must be computed on a case by case basis. We first focus on the rationale of our method and discuss its computational cost more thoroughly in Section 4.2.

As we have seen, we may deflate the spectrum by either perturbing the mass, stiffness or both. In order to align ourselves with common practice, which tends to only modify the mass matrix, we propose a two-step lumping-scaling strategy: we first approximate the mass matrix with one of the lumping strategies proposed in Section 3 (or, alternatively, any suitable ad hoc mass lumping technique) and then scale the lumped mass matrix to remove persistent outliers. Note that the scaled mass matrix is generally completely dense and must obviously never be formed explicitly (in contrast to the method proposed in [27]). Instead, it is represented implicitly by only storing the lumped mass matrix and the terms V and $g(D_2)$ defining the low-rank perturbation. Moreover, since the perturbation is low-rank, the scaled mass matrix may be easily inverted thanks to the Woodbury matrix identity [56] leading to

$$\bar{B}^{-1} = B^{-1} - U_2(g(D_2)^{-1} + I_r)^{-1}U_2^T. \tag{4.1}$$

Since $g(D_2)$ is diagonal (and nonsingular), $(g(D_2)^{-1} + I_r)^{-1}$ can be formed explicitly. Thus, solving a linear system with the scaled mass matrix only requires solving a linear system with the lumped mass matrix and computing a matrix-vector multiplication with a low-rank matrix. While the former uses standard techniques, as described in Section 3.5, the latter only requires $O(rn)$ additional flops. Thus, if the perturbation's rank is relatively small, these matrix-vector multiplications do not introduce any significant overhead. Due to repeated matrix-vector multiplications with the stiffness matrix in the time stepping scheme, the cost incurred by instead scaling the stiffness matrix is the same and eventually the choice is just a matter of taste. The method described herein is very general and could even be beneficial for practitioners using the consistent mass. Indeed, the scaling does not effect the smallest eigenpairs and therefore preserves the higher order convergence characterizing the consistent mass.

Nevertheless, the strategy may seem rather impractical given that it requires explicit knowledge of the outlier eigenvalues and associated eigenvectors, whose number grows under mesh refinement [12]. Surprisingly, this aspect was completely neglected in earlier works [27, 28]. Several arguments underpin our strategy. Firstly, in contract to classical C^0 finite elements for which similar methods were proposed, maximally smooth C^{p-1} spline discretizations feature far fewer outlier eigenvalues, as outlined in the Appendix. Secondly, although practitioners often rule out (approximate) eigenvalue computations as prohibitively expensive, as we will discuss in the next subsection, the workload for computing a few of the largest eigenpairs with the Lanczos method is very similar to performing a few iterations of an explicit algorithm for dynamical simulations. Thus, it might be worthwhile spending a few iterations to remove outliers if we might save up on hundreds of iterations later on, especially for long-time simulations. The Lanczos method is the state-of-the-art solver for sparse symmetric generalized eigenvalue problems. It is briefly summarized in the next section to support our argument.

4.2 Eigenvalue and eigenvector computations

The Lanczos method for generalized eigenproblems can be derived from the one for standard eigenproblems, after transforming the generalized eigenproblem to standard form; e.g. via the Cholesky factorization of B. However, further transformations are needed for computational efficiency and numerical stability. A basic version is presented in Algorithm 4.1.

```
   Input: Symmetric matrix pair $(A, B)$ with $B$ positive definite, starting vector $\mathbf{b}$, number of iterations $m$
   Output: $m$ approximate eigenpairs of $(A, B)$
1: Set $\mathbf{v}_0 = 0$, $\mathbf{v}_1 = \mathbf{b}/\|\mathbf{b}\|_B$, $\beta_1 = 0$
2: for $j = 1, 2, \cdots, m$ do
3:     $\mathbf{v} = A\mathbf{v}_j$
4:     $\alpha_j = (\mathbf{v}, \mathbf{v}_j)$
5:     $\mathbf{w} = B^{-1}\mathbf{v} - \alpha_j \mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$
6:     $\beta_{j+1} = \sqrt{(\mathbf{v}, \mathbf{w})}$
7:     $\mathbf{v}_{j+1} = \mathbf{w}/\beta_{j+1}$
8: end for
```

Algorithm 4.1: Lanczos method [55, Algorithm 9.1]

The vectors $\mathbf{v}_j$ computed during the course of the iterations are stored along the columns of the matrix V_m; i.e. $V_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$, which by construction forms a B-orthonormal basis for the Krylov subspace

$$\mathcal{K}_m(C, \mathbf{b}) = \operatorname{span}(\mathbf{b}, C\mathbf{b}, C^2\mathbf{b}, \dots, C^{m-1}\mathbf{b}),$$

where $C = B^{-1}A$ and $\mathbf{b}$ is the starting vector, usually randomly chosen. The coefficients α_j and β_j are stored in the symmetric tridiagonal matrix

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_m \\ & & \beta_m & \alpha_m \end{pmatrix}.$$

Rewriting line 5 in matrix form leads to the famous Lanczos decomposition

$$AV_m = BV_mT_m + \beta_{m+1}B\mathbf{v}_{m+1}\mathbf{e}_m^T.$$

From the B-orthonormality of V_m, we deduce that

$$V_m^T A V_m = T_m, \quad V_m^T B V_m = I_m.$$

Following the Rayleigh-Ritz procedure [53, 55], an approximate eigenpair (called Ritz pair) is defined as $(\lambda, \mathbf{u}) = (\mu, V_m\mathbf{q})$, where $(\mu, \mathbf{q})$ is an exact eigenpair of the much smaller tridiagonal matrix T_m.

Remark 4.5. Due to the propagation of round-off errors, a reorthogonalization procedure is necessary to restore, at least occasionally, the B-orthonormality of the Krylov basis. Implementations of the Lanczos method are further supplemented with restarting procedures that truncate the Krylov basis to avoid its prohibitive growth and potential storage issues. Common guidelines recommend using a basis size of $m = 2k$, where k is the number of desired eigenpairs [57]. Finally, convergence checks are also implemented to serve as stopping criterion. We have voluntarily left aside all those advanced topics to focus on the essential. The interested reader may refer to the extensive literature for a detailed discussion; e.g. [57, 58, 53, 55].

Although we have implemented an eigensolver from scratch for the sake of writing this paper, it is absolutely not necessary for applying the techniques presented herein. Several efficient implementations are available in software packages and all the user has to worry about is supplying an algorithm for computing matrix-vector multiplications with the stiffness matrix and solving linear systems with the (lumped) mass matrix, which depends on the nature of the problem (e.g. single-patch, multipatch,...). If the number of iterations m remains relatively small, which is ensured through restarting procedures, each iteration of the Lanczos method costs nearly as much as an iteration of the central difference method. Moreover, the Lanczos method is known to converge very fast to eigenvalues that are well separated from the rest of the spectrum [59, 53, 55], which is precisely a distinctive feature of outliers. However, the time span of the simulation must be sufficiently large to amortize the cost of computing outlier eigenpairs. In general, the shorter the simulation, the smaller the number of outliers we can afford computing. Finally, although our method scales down outlier frequencies, it does not remove the corresponding outlier modes that might negatively impact the solution.

The deflation procedure proposed in this section is very general and can in principle be applied to any type of problem, including nontrivial (multipatch) geometries.

In the multipatch setting, it is possible to locally scale the single-patch system matrices, similarly to local (elementwise) mass scaling techniques [27, 60]. Indeed, Lemma 3.22 reveals that the largest eigenvalues of $(\mathcal{K}, \mathcal{P}_i)$ could be controlled by the largest eigenvalues of $(\mathcal{K}_r, \mathcal{P}_{r,i})$ and suggests a scaling strategy directly targeting the origin of the issue: at the patch level. While this strategy is generally cheaper than scaling $(\mathcal{K}, \mathcal{P}_i)$ globally, it has three shortcomings: firstly, the assembly into global matrices generally effects the smallest eigenvalues; secondly it cannot remove outliers introduced by the C^0 coupling of patch interfaces and finally, since the inverse of the global mass matrix *is not* given by the assembly of the local inverses, we cannot directly apply (4.1) locally. To resolve the third issue, we suggest locally scaling the stiffness matrix instead. Denoting $\bar{\mathcal{K}}_r$ the locally scaled stiffness matrix and recalling that the perturbation is negative semidefinite (see Section 4.1), $\bar{\mathcal{K}}_r \preceq \mathcal{K}_r$ and

$$\bar{\mathcal{K}} := \sum_{r=1}^{N_p} R_r^T \bar{\mathcal{K}}_r R_r \preceq \sum_{r=1}^{N_p} R_r^T \mathcal{K}_r R_r = \mathcal{K}.$$

Consequently, $\lambda_k(\bar{\mathcal{K}}, \mathcal{P}_i) \leq \lambda_k(\mathcal{K}, \mathcal{P}_i) \leq \lambda_k(\mathcal{K}, \mathcal{M})$. Thus, our strategy essentially boils down to computing a few of the largest eigenpairs of a sequence of generalized eigenproblems on single patches, which is also well suited for parallel computations. We will better assess the numerical properties of this method in Section 5.

Remark 4.6. For some special cases it is yet far more advantageous to exploit the structure of the problem. In particular, for problems featuring a Kronecker product structure, all outliers can be removed by separately scaling the 1D factor matrices. For maximally smooth spline discretizations of 1D problems, the number of outliers only depends on the spline order, differential operator and type of boundary conditions [12, 16]. Consequently, the number of eigenvalues computed scales *linearly* with the dimension and does not depend on the mesh size. For instance, based on the upper bounds provided in [16], for a uniform C^{p-1} discretization of the Laplace on the hypercube $(0, 1)^d$ with homogeneous Dirichlet boundary conditions, only $d(p-1)$ eigenpairs are required to remove $O(n^{d-1}(p-1))$ outliers, where n is the dimension of the univariate spline space [12]. This method also offers some advantages over the ones presented in [12, 16]. Firstly, its implementation is straightforward: it does not require a change of basis, the standard B-spline basis is sufficient. Secondly, it seems more robust than the method presented in [12], which generally stumbles on domains with curved boundaries, even if the mass matrix is a Kronecker product.

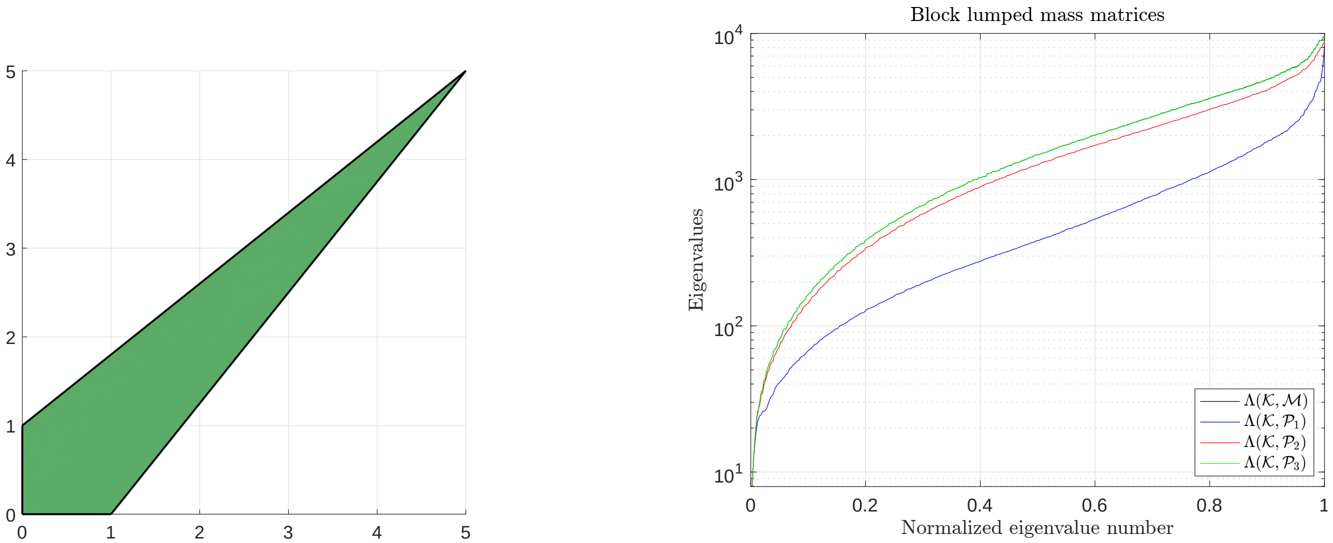
5 Numerical experiments

This section gathers a few numerical experiments designed to verify our theoretical results and demonstrate the usefulness of our strategies in the context of explicit dynamics. All experiments in this section are done using GeoPDEs [61], a software package for isogeometric analysis.

5.1 Single-patch geometries

Example 5.1. We consider a cubic discretization of the 2D Laplace on two nontrivial single-patch domains shown in Figures 5.1a and 5.2a: a stretched square and a quarter of a plate with a hole, represented by a (near) singular NURBS patch. Here, $\mathcal{M} \in \mathcal{S}^+_{(n_1,n_2)}$ and we construct the first three lumped mass matrices of the sequence. The spectrum of $(\mathcal{K}, \mathcal{M})$ and $(\mathcal{K}, \mathcal{P}_i)$, for $i = 1, 2, 3$, is shown in Figures 5.1b and 5.2b for the stretched square and the plate with a hole, respectively. As predicted, the generalized eigenvalues of the matrix pairs $(\mathcal{K}, \mathcal{P}_i)$ monotonically converge to the eigenvalues of $(\mathcal{K}, \mathcal{M})$ from below for increasing values of i. This property holds for all eigenvalues, including the "outliers", now characterized by a sharp but smooth increase of the spectrum rather than a stepwise increase. For this reason, the distinction between "outlier" and "regular" eigenvalues is rather ambiguous. For nontrivial problems, "outlier" eigenvalues are merely large eigenvalues, which can be removed using deflation techniques such as those presented in Section 4.1. In order to assess the practical gains of the procedure, we solve the wave equation on the plate geometry shown in Figure 5.2a over the time span $T = [0, 6]$ with the manufactured solution $u(x, y, t) = xy(x + 4)(y - 4)(x^2 + y^2 - 1)(2 + \sin(2\pi t))$. The numerical solutions are computed with the central difference method using the critical time step (1.1) multiplied by a safeguarding factor of 0.85. The results are shown in Figure 5.3 for a small time ($t = 0.65$) and a larger time ($t = 2.65$). Figure 5.4 represents the evolution of the L^2 error over time. As one could expect, increasing the block bandwidth improves the accuracy of the lumping techniques.

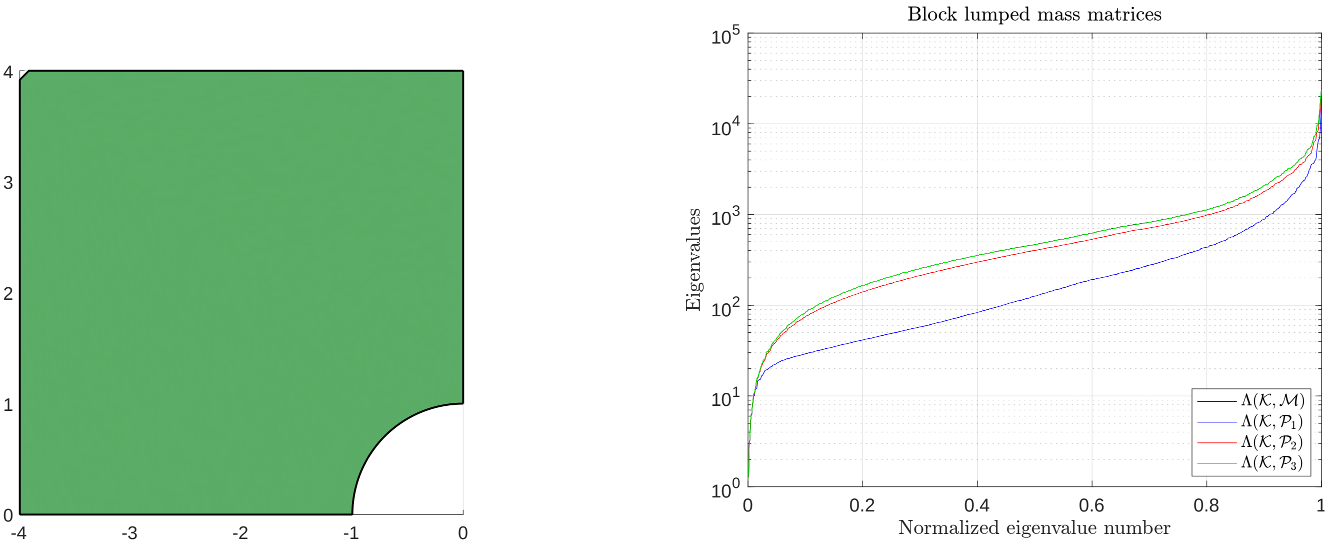
Figure 5.5 shows the ratio $(N_s + N_i)/N_w$ for the block diagonal matrix $\mathcal{P}_1$, where N_s and N_w are the number of iterations with and without scaling, respectively, and N_i is the number of iterations needed by the eigensolver for computing the scaling. The ratio is computed for ranks $r = 10, 20, 40$ and simulations ranging over increasingly larger time spans. The horizontal line for $r = 0$ indicates the absence of scaling and serves as comparison. A ratio strictly larger than 1 indicates a deficit: the saving due to the scaling could not offset the cost for computing it. This situation commonly occurs for short-time simulations. However, the workload for computing a few eigenpairs is quickly amortized over longer simulations and may eventually save more than 50% iterations. In general, one should favor smaller ranks for shorter simulations and larger ranks for longer simulations. Moreover, we could verify over the range of ranks tested that the scaling did not have any significant effect on the quality of the solution.



(a) Geometry

(b) Spectrum

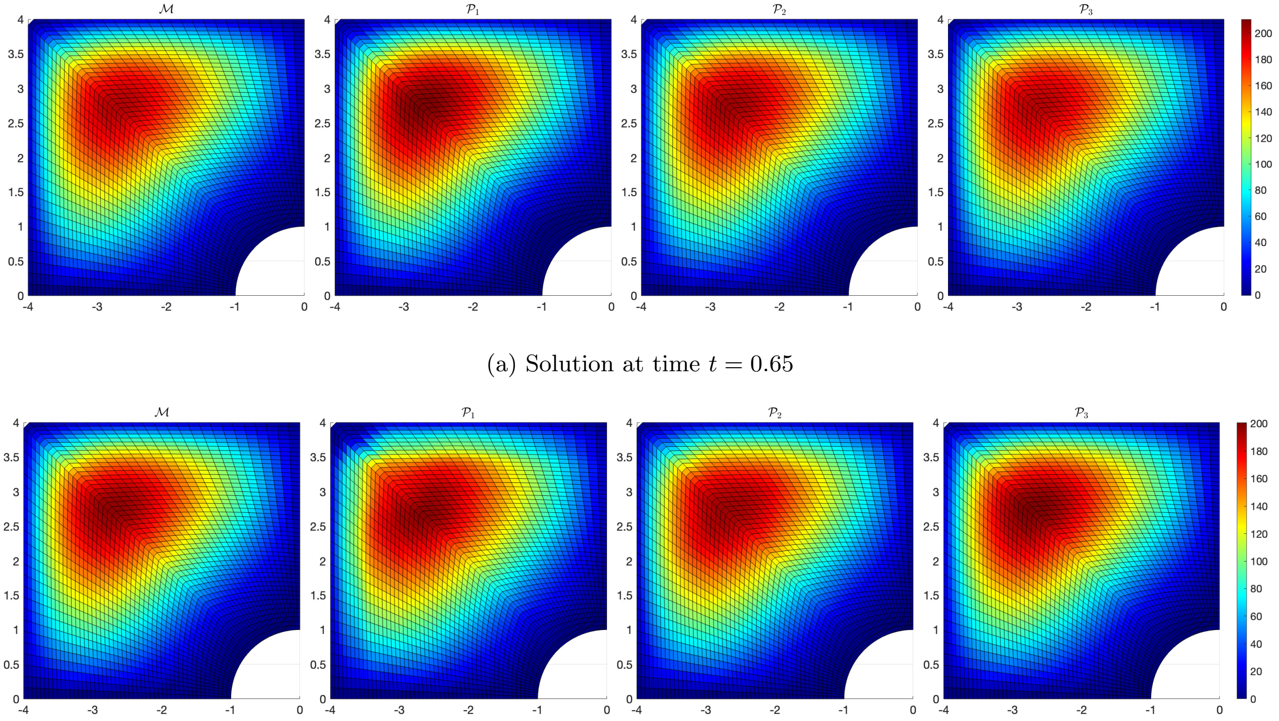
Figure 5.1: Stretched square



(a) Geometry

(b) Spectrum

Figure 5.2: Quarter of a plate with a hole



(a) Solution at time $t = 0.65$

(b) Solution at time $t = 2.65$

Figure 5.3: Numerical solutions for the plate geometry

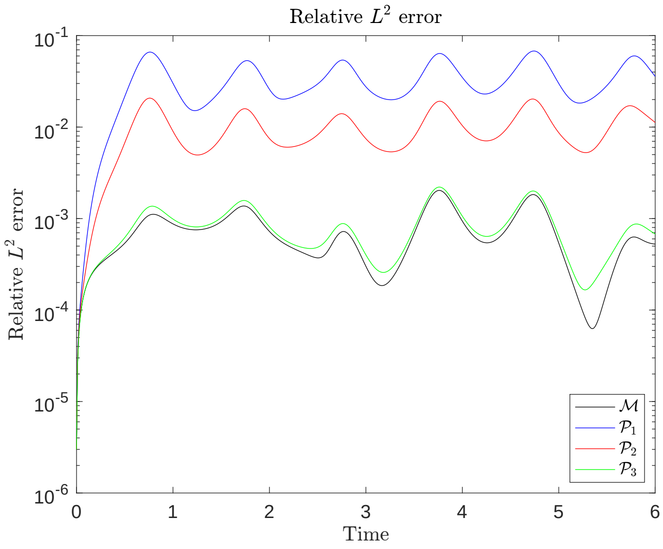


Figure 5.4: Relative L^2 error for the plate geometry

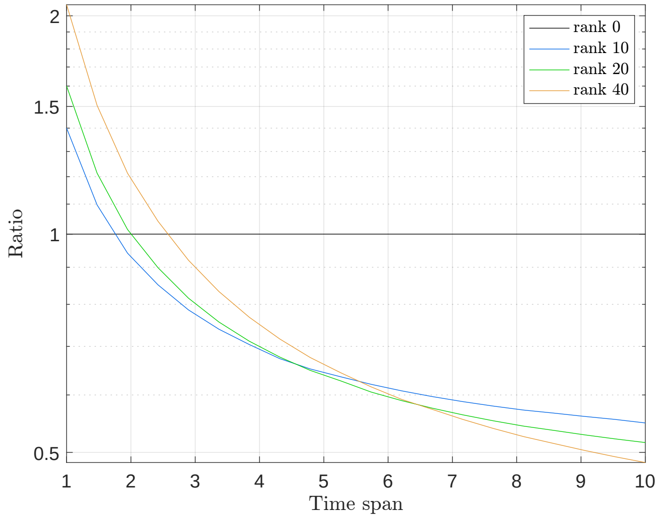
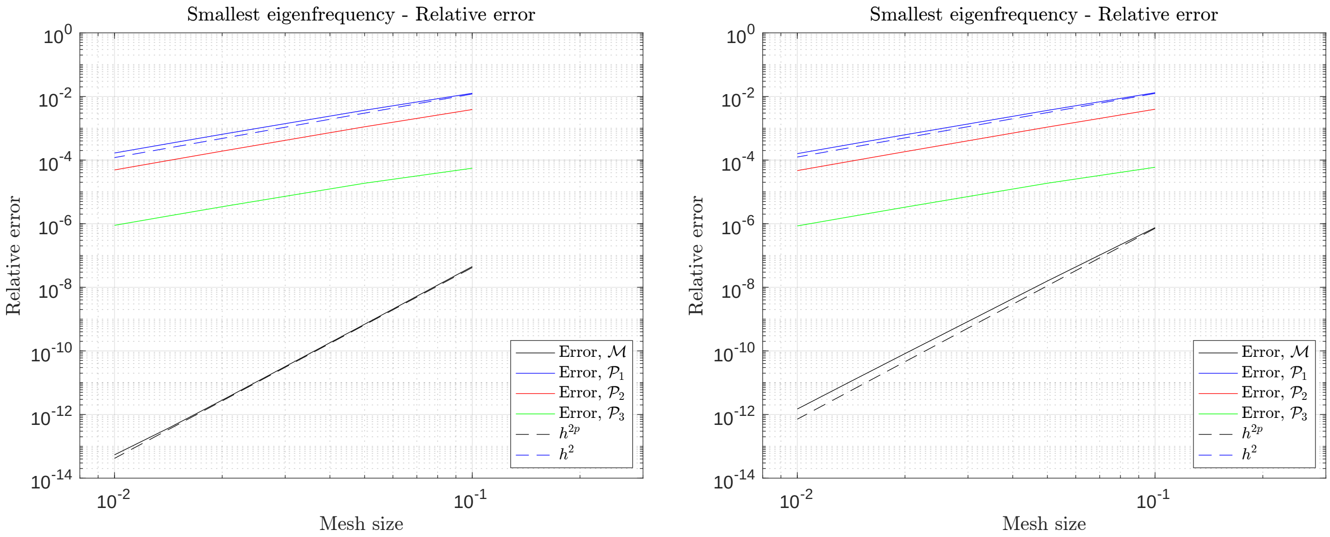


Figure 5.5: Ratio of number of iterations for the scaled and unscaled methods using $\mathcal{P}_1$

Remark 5.2. It is important to bear in mind that mass lumping generally depends on the labeling of the parametric directions, which is specific to each software. GeoPDEs, for instance, uses a reverse labeling; i.e. labels 1, 2 and 3 are associated to the z, y and x directions, respectively. It might sometimes be advantageous to reorder the system matrices and relabel the parametric directions but we have not experimented with it.

Example 5.3 (Convergence test)**.** We now check the convergence of the smallest eigenfrequency for the approximations introduced in Section 3. We consider two test cases designed to evaluate the effect of a coefficient function and geometry mapping. The first one is the unit square with a non-separable (but continuous) density function $\rho(x) = |\sin(xy)| + x + y + 1$ and the second one is the quarter of a plate with a hole (see Figure 5.2a). Figures 5.6a and 5.6b show the relative error $\frac{\omega_1 - \omega_{h,1}}{\omega_1}$ for the first eigenfrequency and a cubic discretization. Due to the lack of closed form solutions, the reference eigenfrequency ω_1 is a high order approximation computed with the consistent mass on a very fine mesh. The smallest eigenfrequency of $(\mathcal{K}, \mathcal{M})$ converges at the expected rate of $2p$, while the smallest eigenfrequency of $(\mathcal{K}, \mathcal{P}_i)$ converges at a reduced quadratic rate. This observation is in agreement with the well-known fact that the row-sum technique converges at a reduced quadratic rate, independently of the spline order [6].



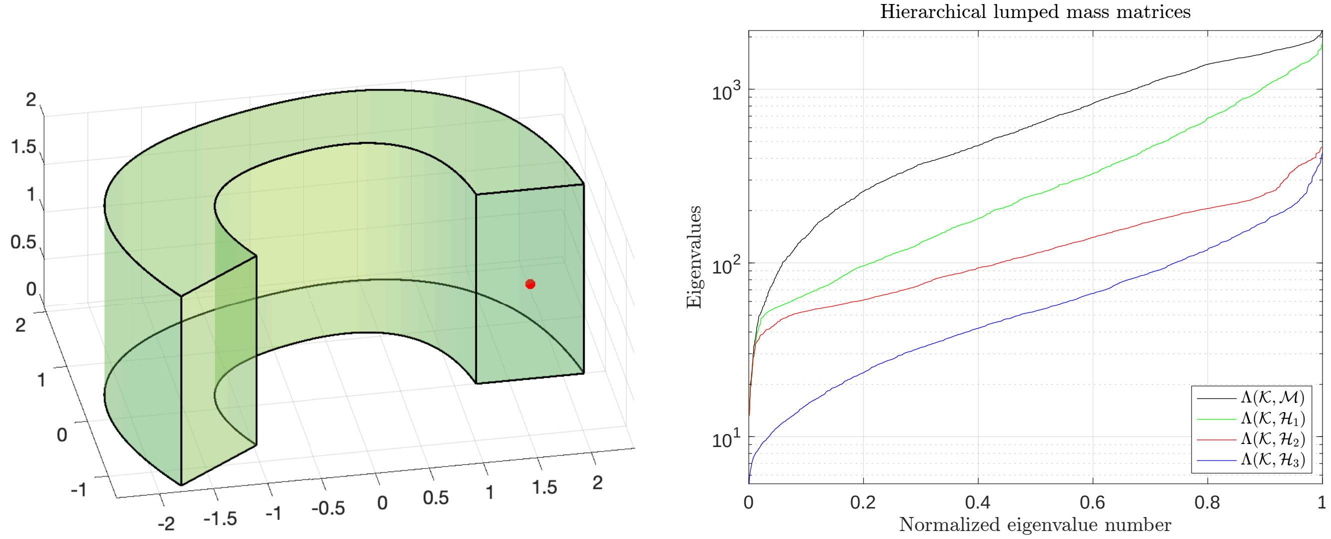
(a) Unit square with a non-separable density function

(b) Quarter of a plate with a hole

Figure 5.6: Relative error $\frac{\omega_1 - \omega_{h,1}}{\omega_1}$

Example 5.4 (Hierarchical mass lumping)**.** Now we consider a quadratic discretization of the 3D Laplace on the magnet domain shown in Figure 5.7a and test the hierarchical lumped mass matrices described in Section 3.3. Their sparsity pattern is shown in Figure 5.8 together with the consistent mass for $N = 6$ subdivisions in each parametric direction. Hierarchical mass lumping leads to a significant reduction of the bandwidth and number of nonzero entries, which drastically speeds up sparse direct solvers. For assessing the performance of mass lumping in explicit dynamics, we solve a sequence of 1000 linear systems with the consistent mass $\mathcal{M}$ and hierarchical lumped mass matrices $\mathcal{H}_k$ for $k = 1, 2, 3$ on increasingly fine meshes. The solver relies on sparse Cholesky factorizations computed on the reordered matrices using nested dissection. According to Table 5.1, mass lumping may save several orders of magnitude of computing time, even for relatively small systems. We also noticed that the reordering only slightly reduced the number of nonzero entries in the Cholesky factors and was not the main driver for the enhanced performance. Table 5.2 shows the computing time for a simulation spanning 50 seconds. While Table 5.1 only accounts for the effect of the linear system solver, Table 5.2 additionally accounts for the saving in the number of time steps thanks to the increase of the critical time step computed with (1.1).

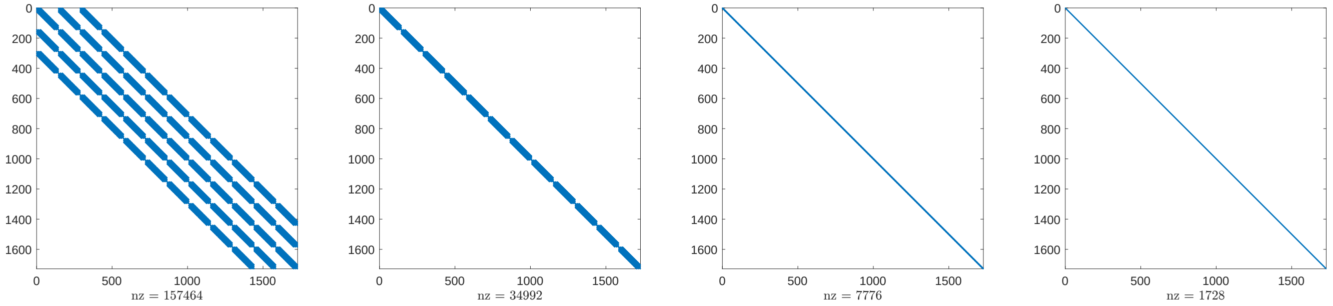
The impact of hierarchical mass lumping on the accuracy of the solution is (partly) determined by the generalized eigenpairs of $(\mathcal{K}, \mathcal{H}_k)$. The eigenvalues are shown in Figure 5.7b alongside those of $(\mathcal{K}, \mathcal{M})$ for $N = 6$ subdivisions. Interestingly, $\mathcal{H}_2$ seems much more accurate than $\mathcal{H}_3$ and yet barely increases its associated CFL condition. This encouraging result indicates that improved accuracy is possible with only a marginal increase in computational cost. Moreover, similarly to Example 5.3, we verified that hierarchical mass lumping delivered second order convergent eigenvalues.



(a) Geometry. The red dot identifies the measurement point in Figure 5.9

(b) Spectrum

Figure 5.7: Magnet



(a) $\mathcal{M}$

(b) $\mathcal{H}_1$

(c) $\mathcal{H}_2$

(d) $\mathcal{H}_3$

Figure 5.8: Sparsity patterns

		Time [s]			
N	Size	$\mathcal{M}$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{H}_3$
3	216	0.02	0.008	0.006	0.005
6	1728	0.56	0.07	0.03	0.02
9	5832	3.64	0.33	0.11	0.07
12	13824	13.5	0.97	0.27	0.17
15	27000	36.38	3.22	0.53	0.33

Table 5.1: System size and computing times (in seconds) for solving a sequence of 1000 linear systems with the consistent mass and hierarchical lumped mass matrices. $N = h^{-1}$ denotes the number of subdivisions in each parametric direction.

		Time [s]				Number of time steps			
N	Size	$\mathcal{M}$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{H}_3$	$\mathcal{M}$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{H}_3$
3	216	0.012	0.004	0.002	0.001	578	530	270	257
6	1728	0.67	0.08	0.02	0.01	1167	1069	539	518
9	5832	6.37	0.54	0.09	0.06	1756	1608	814	785
12	13824	31.7	2.06	0.29	0.18	2345	2148	1090	1052
15	27000	106.3	6.46	0.73	0.45	2935	2689	1367	1320

Table 5.2: System size, computing times (in seconds) and number of time steps for simulating 50 seconds with the consistent mass and hierarchical lumped mass matrices. $N = h^{-1}$ denotes the number of subdivisions in each parametric direction.

We consider now a more realistic situation by solving a linear elasticity problem on the magnet domain of Figure 5.7a. Homogeneous Dirichlet boundary conditions are prescribed on its base and homogeneous Neumann boundary conditions on its side faces. A slowly oscillating traction force, given by

$$\boldsymbol{\tau}(\mathbf{x},t) = \begin{pmatrix} 0 \\ 0 \\ -q\sin(\frac{8\pi t}{T}) \end{pmatrix}$$

is applied on its top face, where $q = 20$ MPa is the pressure's magnitude and $T = 10^{-2}$ s is the final time. We assume the magnet is made out of steel (elastic modulus $E = 207$ GPa, Poisson's ratio $\nu = 0.3$ and density $\rho = 7800$ kg/m^3). The problem is discretized in space using quadratic B-splines with $N = 6$ subdivisions in each parametric direction and approximated in time using the central difference method. The critical time step, given by (1.1), and multiplied by a safeguarding factor of 0.85 leads to 2305, 2213, 1107 and 1057 time steps for $\mathcal{M}$, $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$, respectively. Figure 5.9 compares the vertical component of the displacement at the point identified by a red dot in Figure 5.7a. Whereas the row-sum lumped mass matrix $\mathcal{H}_3$ induces a significant error, the hierarchical lumped mass matrices $\mathcal{H}_2$ and $\mathcal{H}_1$ provide a much better approximation. Moreover, the computing times for solving linear systems scaled similarly as those reported in Table 5.2 for $N = 6$. These two observations make a compelling case for $\mathcal{H}_2$ as it (nearly) provides the accuracy of $\mathcal{H}_1$ but at the cost of $\mathcal{H}_3$.

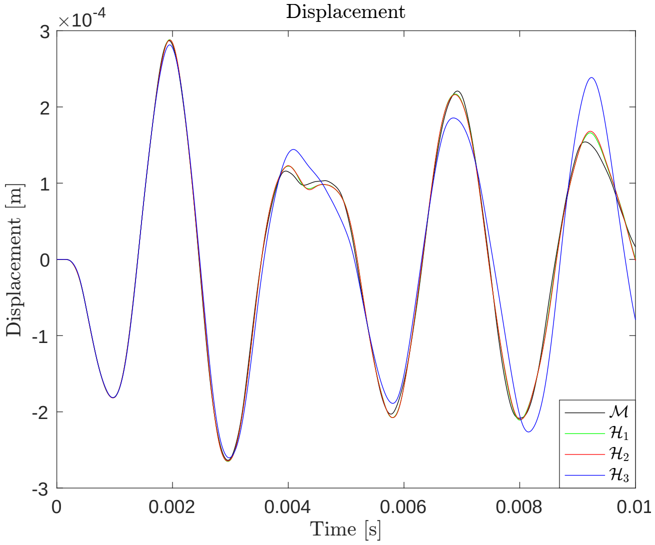
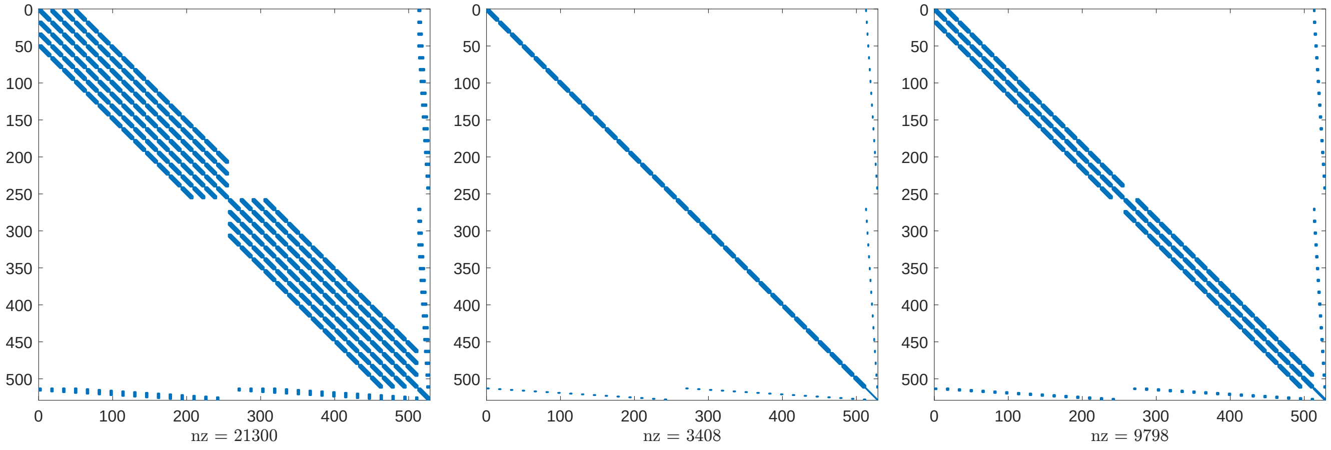


Figure 5.9: Vertical displacement at the point identified by a red dot in Figure 5.7a

5.2 Multipatch geometries

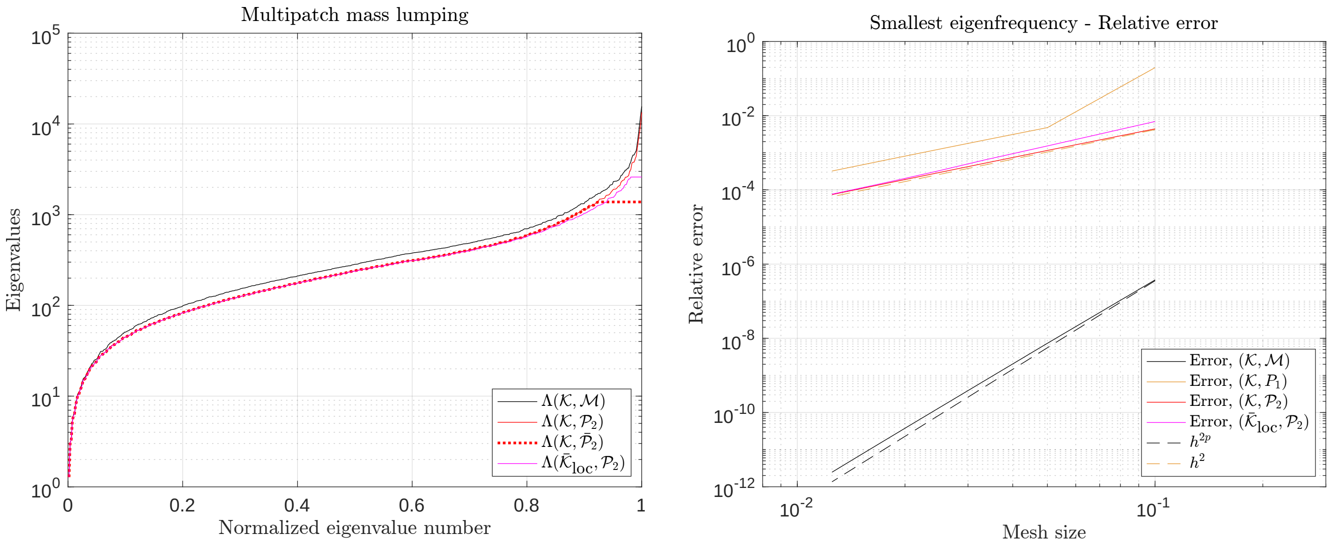
Example 5.5. We consider a cubic discretization of the Laplace on the quarter of a plate with a hole, as shown in Figure 5.2a, and split into two patches to remove sources of singularity. The sparsity patterns of the consistent mass and lumped mass matrices $\mathcal{P}_i$ for $i = 1, 2$ are shown in Figure 5.10 for 15 subdivisions in each parametric direction and each patch. Mass lumping for multipatch problems does not completely remove the coupling but instead focuses on reducing the bandwidth of the diagonal blocks in order to easily form the Schur complement. As explained in Section 4, we try scaling the local stiffness matrices before assembling them into a global matrix.

For clarity, we denote $\bar{\mathcal{K}}_{\text{loc}}$ the locally scaled stiffness matrix. Given its heuristic nature, this scaling strategy might effect the smallest eigenvalues. Nevertheless, Figure 5.11a, obtained for a rank 20 patchwise scaling, reveals that this effect is very mild. By comparing the results with a rank 40 global mass scaling, we notice that the local scaling method removes fewer outliers. This is expected given that it cannot remove interior outliers, arising from the C^0 coupling of patch interfaces. The convergence test carried out in Figure 5.11b further indicates that the smallest eigenfrequency converges at a second order rate, as it does in the purely lumped mass case (using P_1). For the sake of clarity, we have only reported the results for $\mathcal{P}_2$ but they seem to hold more generally.



(a) $\mathcal{M}$ (b) $\mathcal{P}_1$ (c) $\mathcal{P}_2$

Figure 5.10: Sparsity patterns

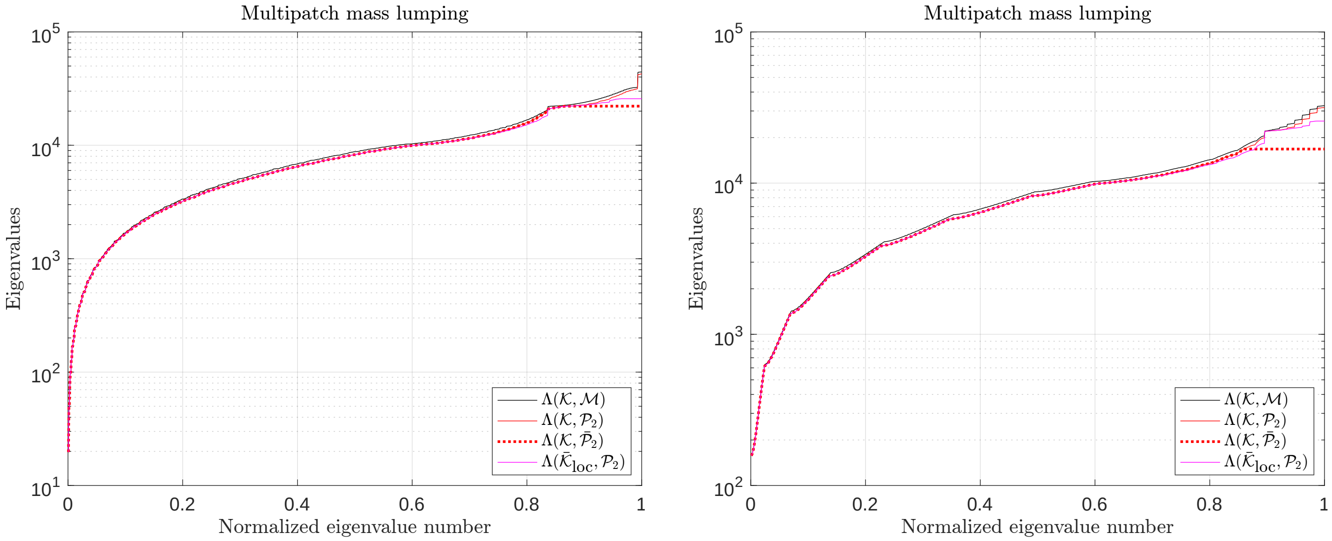


(a) Spectrum

(b) Relative error $\frac{\omega_1-\omega_{h,1}}{\omega_1}$ for the consistent mass and lumped mass with and without scaling

Figure 5.11: Spectrum and convergence test

Example 5.6. This experiment aims at better understanding how patch configurations effect the scaling outcome. Our benchmarks consist of the unit square split into a 4×4 grid of patches and a rectangle of length 4 and width 0.25 split into a 1×16 grid. Both configurations are discretized using quadratic B-splines and $N=8$ subdivisions in each direction and each patch. The size and number of patches of both configurations are the same (16 patches of size 0.25×0.25) but are coupled very differently. Although homogeneous Dirichlet boundary conditions are prescribed along the entire boundary in both cases, the first one features a number of interior patches and consequently more degrees of freedom. Figures 5.12a and 5.12b compare for both configurations a local patchwise scaling of rank 10 with a global scaling of rank 160. Surprisingly, our local scaling strategy performs poorly even for the second, weakly coupled, configuration. In both cases, the global scaling is again more efficient at removing outliers. However, this strategy is also more expensive than our local (sequential) patchwise scaling, as shown in Tables 5.13a and 5.13b for increasingly fine meshes. The global scaling strategy is slightly faster on the second configuration, probably owing to the smaller system size. Nevertheless, the outcome generally depends on the patch configuration, which effects the spectral properties of the system matrices and consequently the convergence speed of the Lanczos method. Therefore, a direct comparison of global and local scaling is not trivial, even while neglecting potential for parallelism. A more thorough study is needed before drawing definite conclusions.



(a) Unit square split in a 4 × 4 grid of patches

(b) Rectangle of length 4 and width 0.25 split into a 1 × 16 grid of patches

Figure 5.12: Comparison of global and local scaling for different patch configurations

		Time [s]	
N	Size	Global	Local
4	361	0.04	0.03
8	1225	0.22	0.06
12	2601	0.48	0.07
16	4489	0.77	0.12
20	6889	1.40	0.16

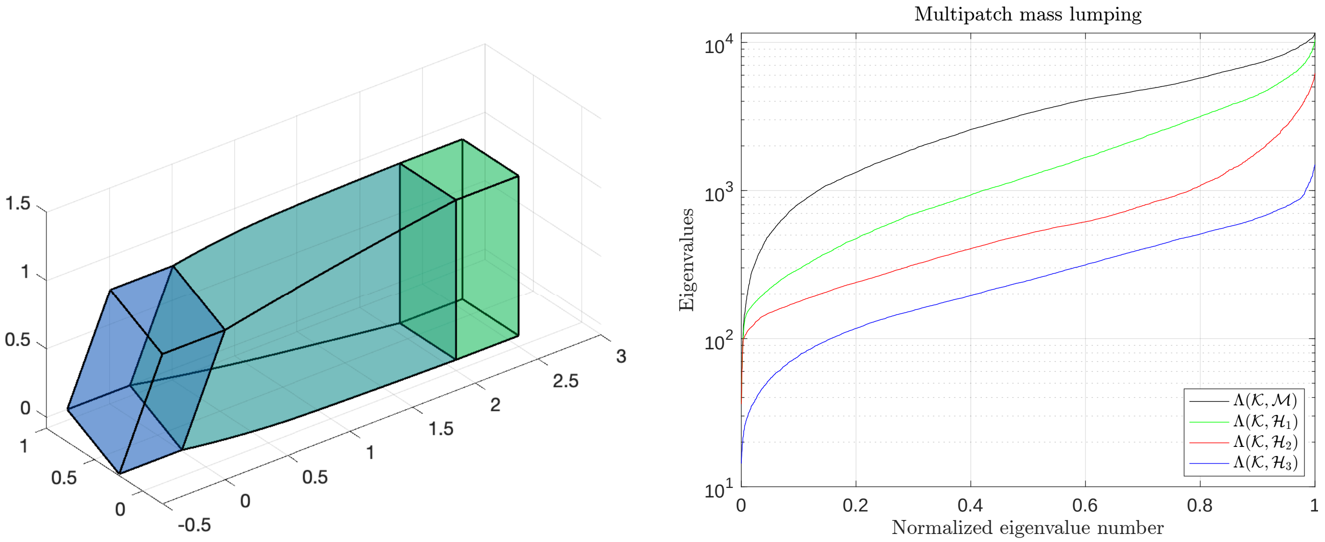
(a) Unit square split into a 4 × 4 grid of patches.

		Time [s]	
N	Size	Global	Local
4	316	0.02	0.03
8	1144	0.14	0.09
12	2484	0.38	0.12
16	4336	0.83	0.13
20	6700	1.29	0.17

(b) Rectangle of length 4 and width 0.25 split into a 1 × 16 grid of patches.

Figure 5.13: System size and computing times (in seconds) for the global and local (sequential) scaling strategies. $N = h^{-1}$ denotes the number of subdivisions in each parametric direction and each patch.

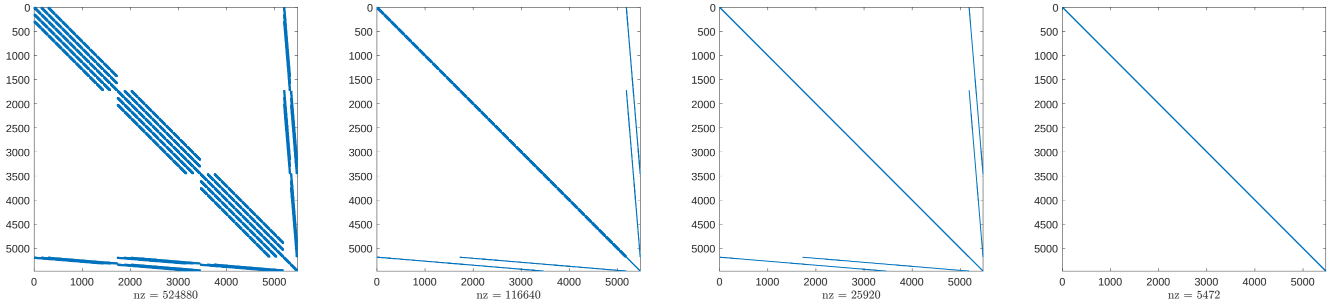
Example 5.7. As we have noted in Section 3.4, any suitable mass lumping technique may be applied patchwise before assembling the global multipatch lumped mass matrix. In dimension $d \geq 3$, hierarchical mass lumping stands out as the natural candidate. When the context is clear, we do not distinguish the single-patch matrices from their multipatch counterpart. We solve the Laplace eigenvalue problem on the 3-patch twisted box geometry shown in Figure 5.14a, discretized with quadratic B-splines and $N = 6$ subdivisions in each parametric direction and each patch. The spectrum of $(\mathcal{K}, \mathcal{M})$ and $(\mathcal{K}, \mathcal{H}_k)$ for $k = 1, 2, 3$ is shown in Figure 5.14b and confirm the improved accuracy on the low-frequency part of the spectrum with respect to the row-sum technique (i.e. $\mathcal{H}_3$). Similarly to Example 5.4, hierarchical mass lumping yields a drastic reduction of the bandwidth and number of nonzero entries, as shown in Figure 5.15.



(a) Geometry. Different colors identify different patches

(b) Spectrum

Figure 5.14: Twisted box



(a) $\mathcal{M}$

(b) $\mathcal{H}_1$

(c) $\mathcal{H}_2$

(d) $\mathcal{H}_3$

Figure 5.15: Sparsity patterns

Example 5.8. In this example, the Laplace eigenvalue problem with pure Neumann boundary conditions is solved on a shifted and rotated square, as shown in Figure 5.16. The physical domain is embedded in a regular unit square, which is trimmed to fit the domain's boundaries. We test our block lumping strategies, combined with a rank 80 scaling, and compare them to the standard row-sum technique. System matrices for trimmed domains generally do not feature a Kronecker structure, not even in their sparsity. However, by padding them with zero entries, they may be embedded in larger matrices whose structure allows applying the block lumping operator. Once the operator has been applied, they are trimmed back to their original size by removing the artificial rows and columns. Thus, our lumping strategies also apply to trimmed geometries. The spectrum (after discarding the zero eigenvalue) is reported in Figures 5.17a and 5.17b for $p = 2$ and $p = 3$, respectively. It is well-known that small trimmed elements heavily deteriorate the conditioning of the system matrices (among other issues) [42, 62]. Beyond a certain threshold, the computations altogether are no longer accurate. Our rotation angle, while still being unfavorable, avoids such extreme situations. Nevertheless, one should exercise caution when computing eigenvalues of matrix pairs (A, B) with a heavily ill-conditioned matrix B. In our numerical experiments, we have computed the eigenvalues of the equivalent matrix pair (DAD, DBD), where $D = \operatorname{diag}(d_1, \ldots, d_n)$ with $d_i = 1/\sqrt{b_{ii}}$ for $i = 1, \ldots, n$ is a Jacobi preconditioner for B [62]. In the context of trimming, the conditioning of DBD is generally orders of magnitude better than the one of B, which improves the stability of eigensolvers.

As shown in Figures 5.17a and 5.17b, the outlier eigenvalues for this problem are particularly pronounced, as evidenced by the sharp change of curvature in the spectrum. Although the row-sum technique is most effective at improving the CFL condition, it is also most inaccurate, even for moderate spline degrees. Our block lumping method drastically improves the accuracy but also leads to more restrictive CFL conditions. However, combining it with scaling strongly mitigates this effect. The fast increase of the outlier eigenvalues also speeds up the convergence of the Lanczos method, which in turn improves the efficiency of the method for explicit dynamics.

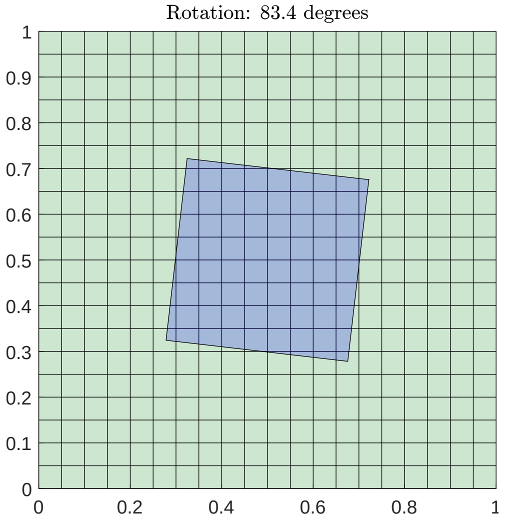
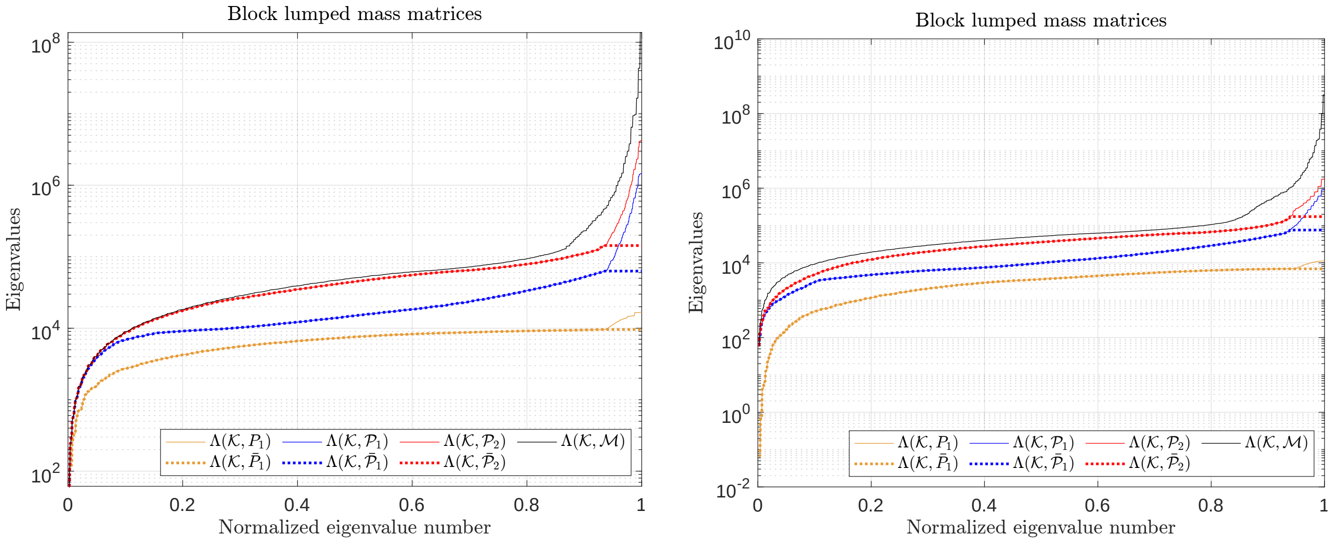


Figure 5.16: Shifted and rotated square



(a) Spectrum for $p = 2$

(b) Spectrum for $p = 3$

Figure 5.17: Spectrum

6 Conclusion

In this article, we have proposed robust mass lumping and outlier removal techniques for nontrivial isogeometric discretizations, including multipatch and trimmed geometries. Our mass lumping techniques provably do not deteriorate the CFL condition of the original problem and oftentimes improve it, thereby extending the methods proposed in [32] to more realistic settings. For a significant increase in step size, we suggest purging persistent outliers by deflating the spectrum. Contrary to existing outlier removal techniques, this method only relies on standard eigensolvers, whose cost per iteration is comparable to explicit time integration methods and whose efficiency is enhanced by the large eigenvalue gaps characterizing outliers. Numerical experiments have shown that the cost for computing a few eigenpairs is rapidly amortized by the subsequent increase in critical time step.

Acknowledgments

The second author kindly acknowledges the support of the SNSF through the project "Smoothness in Higher Order Numerical Methods" n. TMPFP2_209868 (SNSF Swiss Postdoctoral Fellowship 2021).
The third author kindly acknowledges the support of SNSF through the project "PDE tools for analysis-aware geometry processing in simulation science" n. 200021_215099.

A Appendix

It was shown in [4] that maximally smooth spline spaces provide better approximation per degree of freedom than C^0 finite element spaces in almost all cases of practical interest. In this section we shall see how this improved approximation guarantees that there are fewer highly inaccurate outlier modes in the case of smooth spline approximations than for C^0 FEM for any elliptic PDE. In this section we use the standard Sobolev spaces

$$H^r(\Omega) := \{u \in L^2(\Omega) : \partial_1^{\alpha_1} \cdots \partial_d^{\alpha_d} u \in L^2(\Omega),\, 1 \leq \alpha_1 + \cdots + \alpha_d \leq r,\, \alpha_i \geq 0,\, i = 1, \ldots, d\},$$

with corresponding norms $\| \cdot \|_{H^r}$ given by

$$\|u\|_{H^r}^2 = \sum_{0 \leq \alpha_1 + \cdots + \alpha_d \leq r} \|\partial_1^{\alpha_1} \cdots \partial_d^{\alpha_d} u\|_{L^2}^2.$$

Furthermore, $H^{-1}(\Omega)$ is the usual dual space to $H_0^1(\Omega) = \{u \in H^1 : u|_{\partial\Omega} = 0\}$ with corresponding dual norm $\|\cdot\|_{H^{-1}}$. For simplicity, we will only consider a second-order elliptic problem with zero Dirichlet boundary conditions. The variational form of such a PDE can be stated as: given $f \in H^{-1}(\Omega)$ find $u \in H_0^1(\Omega)$ such that

$$a(u, v) = (f, v) \qquad \forall v \in H_0^1(\Omega). \tag{A.1}$$

The problem is well-posed if

$$\begin{aligned} \|u\|_{H^1}^2 &\lesssim a(u, u) && \forall u \in H_0^1(\Omega), \\ a(u, v) &\lesssim \|u\|_{H^1} \|v\|_{H^1} && \forall u, v \in H_0^1(\Omega). \end{aligned} \tag{A.2}$$

Here we use the notation $a \lesssim b$ as shorthand for the inequality $a \leq Cb$ for some constant $C > 0$ independent of u. For a well-posed PDE the solution u in (A.1) satisfies

$$\|u\|_{H^1} \lesssim \|f\|_{H^{-1}}. \tag{A.3}$$

If the coefficients in $a(\cdot, \cdot)$ are sufficiently smooth and the domain Ω is either convex or its boundary $\partial\Omega$ is also sufficiently smooth then it follows from the elliptic regularity theorem that (A.3) can be improved to

$$\|u\|_{H^2} \lesssim \|f\|_{L^2}, \tag{A.4}$$

see [63, Chapter 6.3] and [64, Chapter 3.2] for the details. In fact, for any nonnegative integer m, if the coefficients in $a(\cdot, \cdot)$ are C^{m+1} and the boundary $\partial\Omega$ is C^{m+2} then we have the estimate [63, Chapter 6.3]

$$\|u\|_{H^{m+2}} \lesssim \|f\|_{H^m}. \tag{A.5}$$

Let us now consider the eigenvalue problem: find $\mu_j \in \mathbb{R}$ and $u_j \in H_0^1(\Omega)$, $j = 1, 2, \ldots$, such that

$$a(u_j, v) = \mu_j (u_j, v) \qquad \forall v \in H_0^1(\Omega). \tag{A.6}$$

As explained in Section 2, problem (A.6) can be discretized using B-splines to obtain the problem: find $\lambda_j \in \mathbb{R}$ and $\mathbf{u}_j \in \mathbb{R}^n$, $j = 1, 2, \ldots, n$, such that

$$K\mathbf{u}_j = \lambda_j M \mathbf{u}_j. \tag{A.7}$$

To simplify the analysis we assume that $p = p_1 = \ldots = p_d$, $k = k_1 = \ldots = k_d$ and that the mesh is uniform. Consider the (pushforward) L^2 projection $\Pi_{p,n}^k$ onto the (pushforward) spline space $\mathcal{S}_{\mathbf{p},\mathbf{\Xi}}^{\mathbf{k}}$. This projection is stable in $H^1(\Omega)$ since the mesh is uniform. For a non-uniform mesh a different projection should be considered in the following analysis (ideally, the so-called Ritz projection); see [16] for the details in the case of the Laplacian in one space dimension. It follows from the min-max theory of Strang and Fix [65] that the error between the discrete eigenvalue λ_j in (A.7) and the true eigenvalue μ_j in (A.6) is bounded by the error $\|u_j - \Pi_{p,n}^k u_j\|_{L^2}$, where u_j is the corresponding eigenfunction in (A.6).

Following [5] we define the constant $C_{p,k,r}$ as follows. If $k = p - 1$, we let

$$C_{p,p-1,r} := \left(\frac{1}{\pi}\right)^r$$

and if $k \leq p - 2$, we let

$$C_{p,k,r} := \begin{cases} \left(\dfrac{1}{2}\right)^r \left(\dfrac{1}{\sqrt{(p-k)(p-k+1)}}\right)^r, & k \geq r - 2, \\[2ex] \left(\dfrac{1}{2}\right)^r \left(\dfrac{1}{\sqrt{(p-k)(p-k+1)}}\right)^{k+1} \sqrt{\dfrac{(p+1-r)!}{(p-1+r-2k)!}}, & k < r - 2. \end{cases}$$

For any $u \in H^r(\Omega)$ it is shown in [5] that we have the explicit error estimate

$$\|u - \Pi^k_{p,n} u\|_{L^2} \leq C_{\mathrm{Geo}} C_{p,k,r} h^r \|u\|_{H^r}, \quad r \leq p+1 \tag{A.8}$$

where the constant C_{Geo} only depends on the geometry maps F_i, $i = 1, \dots, N_p$ and is explicitly given in [5]. In classical finite element methods the smoothness $k = 0$ and in this case the constant $C_{p,0,r}$ satisfies, for $p \geq 2$, the following inequalities [5, Remark 3]

$$C_{p,0,r} = \left(\frac{1}{2\sqrt{p(p+1)}} \right)^r \leq \left(\frac{1}{2p} \right)^r, \quad r = 1, 2$$

and

$$C_{p,0,r} = \left(\frac{1}{2} \right)^r \left(\frac{1}{\sqrt{p(p+1)}} \right) \sqrt{\frac{(p+1-r)!}{(p-1+r)!}} \leq \left(\frac{e}{4p} \right)^r, \quad r > 2.$$

It follows from the spline dimension formula that the mesh size $h \sim d(p-k)/n$ and thus $C_{p,k,r} h^r \sim C_{p,k,r} (d(p-k)/n)^r$. More explicitly, with the upper bounds previously obtained:

- For $r = 1, 2$,
$$C_{p,k,r} \frac{d(p-k)}{n} \lesssim \begin{cases} \left(\frac{d}{2n}\right)^r & \text{if } k = 0, \\ \left(\frac{d}{\pi n}\right)^r & \text{if } k = p-1. \end{cases}$$

- For $r > 2$,
$$C_{p,k,r} \frac{d(p-k)}{n} \lesssim \begin{cases} \left(\frac{ed}{4n}\right)^r & \text{if } k = 0, \\ \left(\frac{d}{\pi n}\right)^r & \text{if } k = p-1. \end{cases}$$

Although these estimates are only upper bounds, the difference between the C^0 and C^{p-1} cases is readily appreciated. For instance, for $p = 3$ and $r = 4$ the upper bound is about 20 times smaller in the C^{p-1} case than in the C^0 case. The reader may refer to [5, Figs. 1 and 2] for a graphical comparison of the constants for different values of p and r. In fact, by using a lower bound on the best approximation constant in the C^0 case it is shown in [4] that the maximally smooth approximation constant $C_{p,p-1,r}$ becomes exponentially better than the best achievable approximation constant for $k = 0$ as the degree $p \geq 3$ and r increase.

Theorem A.1. Let n be the dimension of $\mathcal{S}^{\mathbf{k}}_{\mathbf{p},\boldsymbol{\Xi}}$. For any $j = 1, \dots, n$ let u_j be the jth eigenfunction of (A.6) with corresponding eigenvalue μ_j. Then, for all $0 \leq k \leq p-1$, we have

$$\frac{\|u_j - \Pi^k_{p,n} u_j\|_{L^2}}{\|u_j\|_{L^2}} \leq C_{\mathrm{PDE}} C_{\mathrm{Geo}} C_{p,k,1} h \sqrt{\mu_j}. \tag{A.9}$$

Proof. Let $u = u_j$ in (A.8) with $r = 1$ and use (A.2) together with $a(u_j, u_j) = \mu_j \|u_j\|^2_{L^2}$. □

From our previous discussion, it follows that for fixed n and r and for a given tolerance, maximally smooth splines allow to approximate a larger fraction of the eigenfunctions (and eigenvalues) than C^0 finite element spaces. Moreover, if the PDE satisfies elliptic regularity then the error estimate in (A.9) can be further improved.

Theorem A.2. Let $r \leq p+1$ and assume the coefficients in $a(\cdot, \cdot)$ are C^{r-1} and the boundary $\partial\Omega$ is C^r. Let n be the dimension of $\mathcal{S}^{\mathbf{k}}_{\mathbf{p},\boldsymbol{\Xi}}$. For any $j = 1, \dots, n$ let u_j be the jth eigenfunction of (A.6) with corresponding eigenvalue μ_j. Then, for all $0 \leq k \leq p-1$, we have

$$\frac{\|u_j - \Pi^k_{p,n} u_j\|_{L^2}}{\|u_j\|_{L^2}} \leq C_{\mathrm{PDE}} C_{\mathrm{Geo}} C_{p,k,r} h^r \mu_j^{r/2},$$

Proof. If r is even then iterate the elliptic regularity result in (A.5) with $f = \mu_j u_j$, $r/2$ times and use (A.8). If r is odd then additionally use the argument of Theorem A.1 once. □

The improved approximation for maximally smooth splines compared with C^0 finite element spaces will only get better as r increases in Theorem A.2, and we are guaranteed good approximation of a larger fraction of the eigenfunctions and eigenvalues than for C^0 FEM.

References

[1] T. J. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering 194 (39-41) (2005) 4135–4195.

[2] J. A. Cottrell, T. J. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.

[3] Y. Bazilevs, L. Beirao da Veiga, J. A. Cottrell, T. J. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes, Mathematical Models and Methods in Applied Sciences 16 (07) (2006) 1031–1090.

[4] A. Bressan, E. Sande, Approximation in FEM, DG and IGA: a theoretical comparison, Numerische Mathematik 143 (2019) 923–942.

[5] E. Sande, C. Manni, H. Speleers, Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis, Numerische Mathematik 144 (4) (2020) 889–929.

[6] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. Hughes, Isogeometric analysis of structural vibrations, Computer methods in applied mechanics and engineering 195 (41-43) (2006) 5257–5296.

[7] T. J. Hughes, A. Reali, G. Sangalli, Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS, Computer methods in applied mechanics and engineering 197 (49-50) (2008) 4104–4124.

[8] T. J. Hughes, J. A. Evans, A. Reali, Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems, Computer Methods in Applied Mechanics and Engineering 272 (2014) 290–320.

[9] D. Gallistl, P. Huber, D. Peterseim, On the stability of the Rayleigh–Ritz method for eigenvalues, Numerische Mathematik 137 (2017) 339–351.

[10] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[11] K.-J. Bathe, Finite element procedures, Klaus-Jurgen Bathe, 2006.

[12] R. R. Hiemstra, T. J. Hughes, A. Reali, D. Schillinger, Removal of spurious outlier frequencies and modes from isogeometric discretizations of second-and fourth-order problems in one, two, and three dimensions, Computer Methods in Applied Mechanics and Engineering 387 (2021) 114115.

[13] J. Chan, J. A. Evans, Multi-patch discontinuous Galerkin isogeometric analysis for wave propagation: Explicit time-stepping and efficient mass matrix inversion, Computer Methods in Applied Mechanics and Engineering 333 (2018) 22–54.

[14] E. Sande, C. Manni, H. Speleers, Sharp error estimates for spline approximation: Explicit constants, n-widths, and eigenfunction convergence, Mathematical Models and Methods in Applied Sciences 29 (06) (2019) 1175–1205.

[15] M. S. Floater, E. Sande, Optimal Spline Spaces for L^2 n-Width Problems with Boundary Conditions, Constructive Approximation 50 (2019) 1–18.

[16] C. Manni, E. Sande, H. Speleers, Application of optimal spline subspaces for the removal of spurious outliers in isogeometric discretizations, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114260.

[17] S. Takacs, T. Takacs, Approximation error estimates and inverse inequalities for B-splines of maximum smoothness, Mathematical Models and Methods in Applied Sciences 26 (07) (2016) 1411–1445.

[18] J. Sogn, S. Takacs, Robust multigrid solvers for the biharmonic problem in isogeometric analysis, Computers & Mathematics with Applications 77 (1) (2019) 105–124.

[19] M. S. Floater, E. Sande, Optimal spline spaces of higher degree for L^2 n-widths, Journal of Approximation Theory 216 (2017) 1–15.

[20] C. Manni, E. Sande, H. Speleers, Outlier-free spline spaces for isogeometric discretizations of biharmonic and polyharmonic eigenvalue problems, Computer Methods in Applied Mechanics and Engineering (2023) 116314.

[21] D. Toshniwal, H. Speleers, R. R. Hiemstra, C. Manni, T. J. Hughes, Multi-degree B-splines: Algorithmic computation and properties, Computer Aided Geometric Design 76 (2020) 101792.

[22] Q. Deng, V. M. Calo, A boundary penalization technique to remove outliers from isogeometric analysis on tensor-product meshes, Computer Methods in Applied Mechanics and Engineering 383 (2021) 113907.

[23] T.-H. Nguyen, R. R. Hiemstra, S. K. Stoter, D. Schillinger, A variational approach based on perturbed eigenvalue analysis for improving spectral properties of isogeometric multipatch discretizations, Computer Methods in Applied Mechanics and Engineering 392 (2022) 114671.

[24] R. W. Macek, B. H. Aubert, A mass penalty technique to control the critical time increment in explicit dynamic finite element analyses, Earthquake engineering & structural dynamics 24 (10) (1995) 1315–1331.

[25] L. Olovsson, K. Simonsson, M. Unosson, Selective mass scaling for explicit finite element analyses, International Journal for Numerical Methods in Engineering 63 (10) (2005) 1436–1445.

[26] S. K. Stoter, T.-H. Nguyen, R. R. Hiemstra, D. Schillinger, Variationally consistent mass scaling for explicit time-integration schemes of lower-and higher-order finite element methods, Computer Methods in Applied Mechanics and Engineering 399 (2022) 115310.

[27] A. Tkachuk, M. Bischoff, Local and global strategies for optimal selective mass scaling, Computational Mechanics 53 (6) (2014) 1197–1207.

[28] J. A. González, K. Park, Large-step explicit time integration via mass matrix tailoring, International Journal for Numerical Methods in Engineering 121 (8) (2020) 1647–1664.

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

[30] N. Collier, L. Dalcin, D. Pardo, V. M. Calo, The cost of continuity: performance of iterative solvers on isogeometric finite elements, SIAM Journal on Scientific Computing 35 (2) (2013) A767–A784.

[31] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The finite element method: its basis and fundamentals, Elsevier, 2005.

[32] Y. Voet, E. Sande, A. Buffa, A mathematical theory for mass lumping and its generalization with applications to isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 410 (2023) 116033.

[33] C. Anitescu, C. Nguyen, T. Rabczuk, X. Zhuang, Isogeometric analysis for explicit elastodynamics using a dual-basis diagonal mass formulation, Computer Methods in Applied Mechanics and Engineering 346 (2019) 574–591.

[34] T.-H. Nguyen, R. R. Hiemstra, S. Eisenträger, D. Schillinger, Towards higher-order accurate mass lumping in explicit isogeometric analysis for structural dynamics, arXiv preprint arXiv:2305.12916 (2023).

[35] R. R. Hiemstra, T.-H. Nguyen, S. Eisentrager, W. Dornisch, D. Schillinger, Higher order accurate mass lumping for explicit isogeometric methods based on approximate dual basis functions, arXiv preprint arXiv:2310.13379 (2023).

[36] B. Marussig, T. J. Hughes, A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects, Archives of computational methods in engineering 25 (2018) 1059–1127.

[37] L. Leidinger, Explicit isogeometric B-Rep analysis for nonlinear dynamic crash simulations, Ph.D. thesis, Technische Universität München (2020).

[38] L. Coradello, Accurate isogeometric methods for trimmed shell structures., Ph.D. thesis, École polytechnique fédérale de Lausanne (2021).

[39] S. K. Stoter, S. C. Divi, E. H. van Brummelen, M. G. Larson, F. de Prenter, C. V. Verhoosel, Critical time-step size analysis and mass scaling by ghost-penalty for immersogeometric explicit dynamics, Computer Methods in Applied Mechanics and Engineering 412 (2023) 116074.

[40] A. Quarteroni, Numerical models for differential problems, Vol. 2, Springer, 2009.

[41] L. Leidinger, M. Breitenberger, A. Bauer, S. Hartmann, R. Wüchner, K.-U. Bletzinger, F. Duddeck, L. Song, Explicit dynamic isogeometric b-rep analysis of penalty-coupled trimmed nurbs shells, Computer Methods in Applied Mechanics and Engineering 351 (2019) 891–927.

[42] F. de Prenter, C. V. Verhoosel, E. H. van Brummelen, M. G. Larson, S. Badia, Stability and conditioning of immersed finite element methods: analysis and remedies, Archives of Computational Methods in Engineering (2023) 1–40.

[43] U. Von Luxburg, A tutorial on spectral clustering, Statistics and computing 17 (2007) 395–416.

[44] C. Hofreither, A black-box low-rank approximation algorithm for fast matrix assembly in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 333 (2018) 311–330.

[45] G. H. Golub, C. F. Van Loan, Matrix computations, JHU press, 2013.

[46] R. A. Horn, C. R. Johnson, Matrix analysis, Cambridge university press, 2012.

[47] T. J. Hughes, K. S. Pister, R. L. Taylor, Implicit-explicit finite elements in nonlinear transient analysis, Computer Methods in Applied Mechanics and Engineering 17 (1979) 159–182.

[48] A. J. Wathen, Realistic eigenvalue bounds for the Galerkin mass matrix, IMA Journal of Numerical Analysis 7 (4) (1987) 449–457.

[49] T. A. Davis, S. Rajamanickam, W. M. Sid-Lakhdar, A survey of direct methods for sparse linear systems, Acta Numerica 25 (2016) 383–566.

[50] T. A. Davis, Direct methods for sparse linear systems, SIAM, 2006.

[51] M. Benzi, G. H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta numerica 14 (2005) 1–137.

[52] G. W. Stewart, J.-g. Sun, Matrix perturbation theory, Computer science and scientific computing, Academic Press, 1990.

[53] B. N. Parlett, The symmetric eigenvalue problem, SIAM, 1998.

[54] H. Hotelling, Some new methods in matrix calculation, The Annals of Mathematical Statistics 14 (1) (1943) 1–34.

[55] Y. Saad, Numerical methods for large eigenvalue problems: revised edition, SIAM, 2011.

[56] M. A. Woodbury, Inverting modified matrices, Tech. rep., Department of Statistics, Princeton University (1950).

[57] G. W. Stewart, A Krylov–Schur algorithm for large eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (3) (2002) 601–614.

[58] K. Wu, H. Simon, Thick-restart Lanczos method for large symmetric eigenvalue problems, SIAM Journal on Matrix Analysis and Applications 22 (2) (2000) 602–616.

[59] Y. Saad, On the rates of convergence of the Lanczos and the block-Lanczos methods, SIAM Journal on Numerical Analysis 17 (5) (1980) 687–706.

[60] S. Eisenträger, L. Radtke, W. Garhuom, S. Löhnert, A. Düster, D. Juhre, D. Schillinger, An eigenvalue stabilization technique for immersed boundary finite element methods in explicit dynamics, arXiv preprint arXiv:2310.11935 (2023).

[61] R. Vázquez, A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0, Computers & Mathematics with Applications 72 (3) (2016) 523–554.

[62] F. de Prenter, C. V. Verhoosel, G. J. van Zwieten, E. H. van Brummelen, Condition number analysis and preconditioning of the finite cell method, Computer Methods in Applied Mechanics and Engineering 316 (2017) 297–327.

[63] L. C. Evans, Partial differential equations, Vol. 19, American Mathematical Society, 2022.

[64] P. Grisvard, Elliptic problems in nonsmooth domains, SIAM, 2011.

[65] G. Strang, G. J. Fix, An analysis of the finite element method, 2nd Edition, Wellesley-Cambridge Press, 2008.