

THE SURROGATE MATRIX METHODOLOGY: LOW-COST ASSEMBLY FOR ISOGEOMETRIC ANALYSIS

DANIEL DRZISGA*, BRENDAN KEITH*, AND BARBARA WOHLMUTH*

Abstract. A new methodology in isogeometric analysis (IGA) is presented. This methodology delivers low-cost variable-scale approximations (surrogates) of the matrices which IGA conventionally requires to be computed from element-scale quadrature formulas. To generate surrogate matrices, quadrature must only be performed on certain elements in the computational domain. This, in turn, determines only a subset of the entries in the final matrix. The remaining matrix entries are computed by a simple B-spline interpolation procedure. Poisson’s equation, membrane vibration, plate bending, and Stokes’ flow problems are studied. In these problems, the use of surrogate matrices has a negligible impact on solution accuracy. Because only a small fraction of the original quadrature must be performed, we are able to report beyond a fifty-fold reduction in overall assembly time in the same software. The capacity for even further speed-ups is clearly demonstrated. The implementation used here was achieved by a small number of modifications to the open-source IGA software library GeopDEs. Similar modifications could be made to other present-day software libraries.

Key words. Assembly, surrogate numerical methods, isogeometric analysis, a priori analysis.

1. Introduction. Avoiding unnecessary work is of utmost importance when computing at the frontiers of contemporary research. To frame a workable definition, recall that practical simulations in science and engineering involve a large number of possible sources of error. For instance, we highlight the categories of modeling error, numerical error, and data error, each of which have many subcategories. The total error in a simulation is controlled by the aggregate of each relevant source of error. In this paper, “unnecessary work” — or, more precisely, *over-computation* — is any machine expense used to drive one source of error in a problem far below the total error. It cannot be overstated that removing sources of over-computation can have an outsized influence on the computational cost of getting an accurate solution.

In some instances, circumventing over-computation is the simplest way to accelerate a numerical algorithm. For example, in the use of iterative methods, for both linear and non-linear problems, it has long been acknowledged that *over-solving* a discretized problem is a negligent expense. Relaxing iterative solver errors usually reduces to just adjusting the tolerances naturally built into established algorithms. In other instances, sources of over-computation are less conspicuous and avoiding them requires the development of new algorithms. For example, in the field of uncertainty quantification, it has recently come to light that sampling error can be relaxed — and, in turn, computational cost can be significantly reduced — by the use of a tunable *surrogate response surface* [10, 36].

The focus of this article is the Galerkin form of isogeometric analysis (IGA) [12, 28]. At first sight, in view of the long list of computer methods which rose beforehand, the Galerkin isogeometric method may be seen as a rather paradigmatic approach to the discretization of partial differential equations (PDEs). Indeed, Galerkin IGA methods are little more than finite element methods which employ non-uniform rational B-spline (NURBS) bases [27]. Although it was immediately shown by Hughes et al. [28] that the use of such a basis improves the interoperability between computer-aided design (CAD) and PDE analysis, many other benefits of the IGA approach were also demonstrated early on in the IGA literature. Of particular note, the arbitrary smoothness of NURBS bases generally improves the accuracy per degree of freedom and lends itself to convenient techniques for the discretization of high-order PDEs [27, 30]. It is these and other serendipitous features of IGA which have attributing to its truly meteoric success in modern computational science and engineering research.

*Lehrstuhl für Numerische Mathematik, Fakultät für Mathematik (M2), Technische Universität München, Garching bei München (drzisga@ma.tum.de, keith@ma.tum.de, wohlmuth@ma.tum.de)

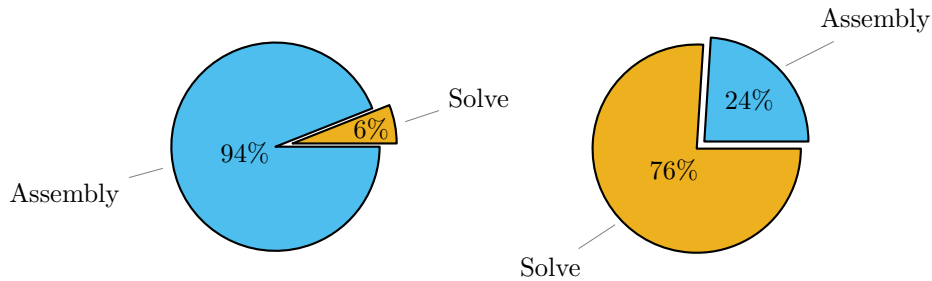


Figure 1.1: Left: Distribution of computational cost with standard IGA. Right: Distribution of computational cost with a surrogate IGA strategy. (Timings taken from the experiment presented in Figures 7.5 and 7.6 for $c = 3$. Both experiments ran on a single compute core and the default MATLAB backslash operator has been used as a solver.)

It is well-established that traditional isogeometric methods face a great computational burden at the point of matrix assembly. This is due, in part, to the large support of the basis functions. Although many other common concerns are naturally alleviated by the IGA paradigm, this particular challenge is clearly evidenced by the expansive literature on quadrature rules and accelerated assembly algorithms [2, 3, 8, 11, 23–26, 29, 33–35, 38]. Indeed, we may further accentuate this remark with the following quote from the 2014 review article [13]:

“...at the moment the assembly of the matrix is the most time-consuming part of isogeometric codes. The development of optimal assembly procedures is an important task required to render isogeometric methods a competitive technology.”

In this article, we present a simple methodology to avoid *over-assembling* matrices in IGA. Roughly speaking, it requires performing quadrature for only a small fraction of the trial and test basis function interactions and then approximating the rest through, for example, interpolation. This leads to a large sparse matrix where the majority of entries have not been computed using any quadrature at all. Usually, such matrices will not coincide with the ones generated by performing quadrature for every non-zero entry (cf. Subsection 5.4), but they can be interpreted as surrogates for those matrices.

The main idea used here was first introduced in the context of first-order finite elements by Bauer et al. in [6]. Thereafter, applications to peta-scale geodynamical simulations were presented in [4, 5] and a theoretical analysis was given in [21]. In the massively parallel applications [4–6], it was natural to work with so-called “macro-meshes” as well as a piecewise polynomial space for resolving the surrogate matrices. This choice was motivated by a low communication cost across the faces of the macro-elements, a convenient cache-aware implementation, and the fact that a hybrid mesh structure allowed for extremely fast evaluation of the three-dimensional polynomials; see [4, 5] for further details. In contrast, the surrogate matrices in this paper are computed using a B-spline interpolation space. With this particular strategy, we demonstrate that the cost of matrix assembly in conventional IGA codes can be reduced by an order of magnitude.

Our approach bears some similarities to the integration by interpolation and lookup (IIL) approach proposed in [33, 34]. In those two works, an integrand factor from the weak form, composed of both the coefficients of the underlying PDE as well as the geometry mapping, is approximated. In this work, the actual entries of the final matrix are shown to be related to a small number of smooth so-called *stencil functions*; instead of a factor in the integrand, it is these stencil functions which are approximated.

An advantage of the IIL approach is that, in theory, it does not require a uniform knot vector assumption (cf. [Section 2](#)). However, in practice, this assumption is necessary in order to obtain compact lookup tables [\[34\]](#). On the other hand, one advantage of our approach is that it can be easily implemented using existing IGA assembly paradigms. Another advantage is that the implementation is identical whether using a B-spline or a NURBS basis.

Before moving on, some other important remarks deserve to be emphasized:

- The methodology we propose for IGA applications is essentially independent of the quadrature rule used at the individual element or basis function level. This lays bare the possibility for it to be used in conjunction with many other cutting edge techniques for accelerated IGA assembly.
- For our new surrogate methods, it would be most efficient if the matrix entries which require quadrature were to be computed based on individual basis function interactions. This leaves out standard element-by-element assembly strategies, but would work well with the control point pairwise method proposed in [\[32\]](#) or, ideally, with a row-by-row approach; e.g., [\[11, 25, 38\]](#). Nevertheless, one should still expect to see significant speed-ups with surrogate methods in standard element-by-element codes, at least for moderate polynomial orders. In order to underscore this fact, we did not develop a stand-alone code. Instead, we implemented our surrogate methods by simply modifying the assembly routines in the open-source library GeoPDEs [\[20, 42\]](#), leaving every other aspect of the code fixed. For illustration, the reader may refer to the left and right sides of [Figure 1.1](#) to compare the relative timings before and after some relatively minor changes were made to this software (cf. [Section 6](#) and [\[22\]](#)). In both cases, the differences in solving time and solution accuracy were negligible. We expect that most other element-by-element IGA codes should be easy to modify in a similar manner.
- Many efficient assembly strategies for IGA see their performance advantage only in the high polynomial order regime. Here, the performance usually grows with each h -refinement. Indeed, at just over one million degrees of freedom, our experiments demonstrate assembly speed-ups beyond *fifty times*, in the exact same code, with a simple second-order NURBS basis (see [Subsection 7.3.3](#)).

In our experiments, we analyze surrogate IGA methods for Poisson’s equation, membrane vibration, plate bending, and Stokes’ flow problems. The Poisson case is analyzed in detail and the additional experiments are provided in order to motivate further study. It is our eventual goal to adapt our methods to a matrix-free framework, similar to what has been used recently in low-order settings [\[4–6, 21\]](#). This would certainly be helpful in order to reach the full potential of IGA in extreme scale computations.

In the next section, we take stock of the majority of mathematical notation used in the remainder of the paper. In [Section 3](#), we introduce the notion of a stencil function in the IGA context. In [Section 4](#), we investigate the accuracy of B-spline interpolation with regard to stencil functions. In [Section 5](#), we use interpolants of the stencil functions (i.e., surrogate stencil functions) to define surrogate matrices for IGA. [Section 6](#) consists of a brief description of our software implementation. A more complete description is provided in [\[22\]](#). In [Sections 7–10](#), we examine surrogate IGA methods for Poisson’s equation, membrane vibration, plate bending, and Stokes’ flow problems, respectively. [Appendix A](#) is included to support some of the analysis carried out in [Section 4](#).

2. Preliminaries. In this section, we lay out the principal mathematical focus and notation of the paper.

2.1. Model problems and notation. Let $\Omega \subseteq \mathbb{R}^n$ be a domain, $n = 2, 3$. Let $V = V(\Omega)$ be a Hilbert space over \mathbb{R} , the field of real numbers, and let V^* denote its topological dual. For historical reasons, we proceed by adopting notation from the h -version of the finite element

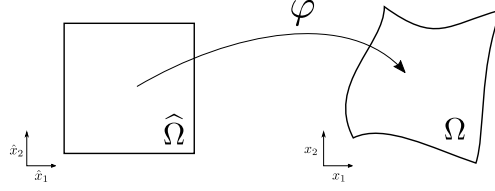


Figure 2.1: Illustration of a smooth transformation from a parametric domain $\hat{\Omega} \subseteq \mathbb{R}^2$ to a physical domain $\Omega \subseteq \mathbb{R}^2$.

method and thus let V_h denote a finite-dimensional subspace of V . Although we also deal with a number of important alternatives (see, e.g., [Sections 8 and 10](#)), we are chiefly interested in the following three problems:

$$(2.1a) \quad \text{Find } u \in V \text{ satisfying} \quad a(u, v) = F(v) \quad \text{for all } v \in V.$$

$$(2.1b) \quad \text{Find } u_h \in V_h \text{ satisfying} \quad a(u_h, v_h) = F(v_h) \quad \text{for all } v_h \in V_h.$$

$$(2.1c) \quad \text{Find } \tilde{u}_h \in V_h \text{ satisfying} \quad \tilde{a}(\tilde{u}_h, v_h) = F(v_h) \quad \text{for all } v_h \in V_h.$$

Here and throughout, $a : V \times V \rightarrow \mathbb{R}$ is a continuous and coercive bilinear form, $\tilde{a} : V_h \times V_h \rightarrow \mathbb{R}$ is an approximation of $a|_{V_h \times V_h}$, hereby deemed the *surrogate* for $a(\cdot, \cdot)$, and $F \in V^*$ is a bounded linear functional.

To keep the exposition simple and to the point, we will assume that the physical domain of every problem $\Omega \subseteq \mathbb{R}^n$ is defined as the image of a single parametric domain $\hat{\Omega} = (0, 1)^n$. This leaves all of our analysis in the single patch geometry setting, $\Omega = \varphi(\hat{\Omega})$, for some diffeomorphism $\varphi : \hat{\Omega} \rightarrow \mathbb{R}^n$ of sufficient regularity; see [Figure 2.1](#). The single patch setting is by no means a necessary assumption. The entirety of the analysis considered here can easily be generalized to the multi-patch setting (cf. [Subsection 3.5](#)). However, in order to stay in the isogeometric setting, we assume that $\varphi(\hat{\mathbf{x}}) = \sum_i c_i \hat{N}_i(\hat{\mathbf{x}})$, where each $c_i \in \mathbb{R}^n$ is a control point vector and each $\hat{N}_i(\hat{\mathbf{x}})$ is a NURBS basis function on the parametric domain $\hat{\Omega}$. Here, NURBS basis functions are defined in the standard way, as described in [Subsection 2.2](#).

For matrices $M \in \mathbb{R}^{l \times m}$, define the max-norm, $\|M\|_{\max} = \max_{i,j} |M_{i,j}|$. For any function $v : \Omega \rightarrow \mathbb{R}$, we will use the notation, $\|v\|_0$, $\|v\|_1$, and $\|v\|_2$, for the canonical $L^2(\Omega)$ -, $H^1(\Omega)$ -, and $H^2(\Omega)$ -norms, respectively. Moreover, if v is smooth, we define its support as $\text{supp}(v) = \{\mathbf{x} \in \Omega : v(\mathbf{x}) \neq 0\}$. When dealing with a domain $\mathcal{D} \subseteq \Omega$, denote the related $L^2(\mathcal{D})$, $H^1(\mathcal{D})$, and $H^2(\mathcal{D})$ norms by $\|v\|_{0,\mathcal{D}}$, $\|v\|_{1,\mathcal{D}}$, and $\|v\|_{2,\mathcal{D}}$, respectively. Denote the space of univariate polynomials of degree at most q by \mathcal{P}_q . Likewise, denote the space of multivariate polynomials of degree at most q , in each Cartesian direction \mathbf{e}_i , by $\mathcal{Q}_q = [\mathcal{P}_q]^n$ and denote $\mathcal{Q}_q(\mathcal{D}) = \{f|_{\mathcal{D}} : f \in \mathcal{Q}_q\}$. We will often deal with Cartesian subdomains $\mathcal{D} = \mathcal{D}_1 \times \dots \times \mathcal{D}_n$. In this case, it is natural to deal with Cartesian-Sobolev seminorms; e.g., $[f]_{W^{r,\infty}(\mathcal{D})} = \sum_{i=1}^n \|D^{r \cdot \mathbf{e}_i} f\|_{L^\infty(\mathcal{D})}$. Note that $[f]_{W^{r,\infty}(\mathcal{D})} \leq |f|_{W^{r,\infty}(\mathcal{D})} = \sum_{|\alpha|=r} \|D^\alpha f\|_{L^\infty(\mathcal{D})}$. All remaining notation will be defined as it arises.

2.2. Cardinal B-splines and NURBS. Let $m \geq 2p + 1$ and $\{b_k\}_{k=1}^m$ be an order p B-spline basis on the unit interval $(0, 1)$. Let $N = m^n$ and let $\{\hat{N}_i\}_{i=1}^N$ be a corresponding NURBS basis on $\hat{\Omega}$. Namely,

$$(2.2) \quad \hat{N}_i(\hat{\mathbf{x}}) = \frac{w_i \hat{B}_i(\hat{\mathbf{x}})}{\sum_j w_j \hat{B}_j(\hat{\mathbf{x}})}, \quad \hat{\mathbf{x}} = (\hat{x}_1, \dots, \hat{x}_n) \in \hat{\Omega},$$

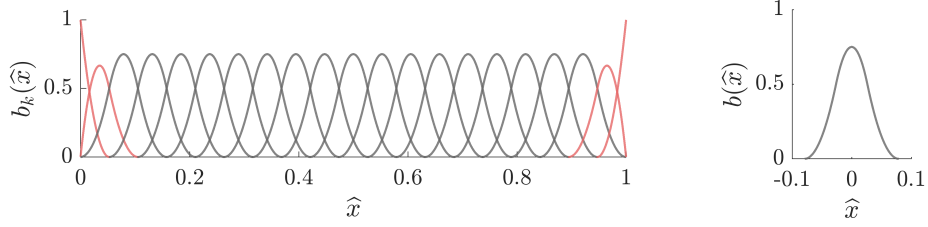


Figure 2.2: Left: 1D B-spline basis functions $\{b_k\}_{k=1}^{21}$ generated by the open uniform knot vector $\Xi^{(2)}$, defined in (2.3). Right: Each gray basis function is equivalent, up to translation, to the function $b(\hat{x})$. The red functions are obviously not.

where each $\widehat{B}_i(\hat{\mathbf{x}}) = b_{i_1}(\hat{x}_1) \cdots b_{i_n}(\hat{x}_n)$ is a multivariate B-spline of uniform order p and each $w_i > 0$ is a fixed weight parameter. Here and from now on, we identify every global index $1 \leq i \leq N$ with a multi-index $\mathbf{i} = (i_1, \dots, i_n)$, $1 \leq i_k \leq m$, through the colexicographical relationship $i = i_1 + (i_2 - 1)m + \cdots + (i_n - 1)m^{n-1}$.

Generally, a univariate B-spline basis $\{b_k\}_{k=1}^m$ is defined by an ordered multiset, or *knot vector*, $\Xi = \{\xi_1, \dots, \xi_{m+p+1}\}$. In this paper, we deal only with *open uniform knot vectors*; i.e., $\xi_1, \dots, \xi_{p+1} = 0$, $\xi_{m+1}, \dots, \xi_{m+p+1} = 1$, and $\xi_{k+1} - \xi_k = \frac{1}{m-p}$, otherwise. The quantity $h = \max_{1 \leq k \leq m-1} |\xi_{k+1} - \xi_k| = \frac{1}{m-p}$ will be an important parameter for us, which we hereby refer to as the *mesh size*. Clearly, we could consider NURBS spaces with different orders p_1, \dots, p_n in each Cartesian direction [28, 37]. In order to simplify the exposition, we avoid this complication.

For an explicit example of an open uniform knot vector, consider

$$(2.3) \quad \Xi^{(p)} = \underbrace{\{0, \dots, 0\}}_{p \text{ times}} \cup \left\{ \frac{k}{19} \right\}_{k=0}^{19} \cup \underbrace{\{1, \dots, 1\}}_{p \text{ times}}.$$

The corresponding $p = 2$ B-spline basis is depicted in Figure 2.2. Observe that all but four of the basis functions (highlighted in red) are identical, up to an equally spaced set of translations. These functions are called *cardinal B-splines* [39–41].

Let $\tilde{x}^{(k)} = (k - \frac{p+1}{2}) \cdot h$, for each $k = p+1, \dots, m-p$. In general, there are always $m - 2p$ univariate cardinal B-spline basis functions which can each be expressed $b_k(\hat{x}) = b(\hat{x} - \tilde{x}^{(k)})$, for some function, $b(\hat{x})$, centered at the origin with support in $(-\frac{p+1}{2} \cdot h, \frac{p+1}{2} \cdot h)$ (see, e.g., Figure 2.2). The $\tilde{x}^{(k)}$ correspond to the midpoints of each function b_k . Likewise, there are $(m - 2p)^n$ multivariate cardinal B-splines. That is $\widehat{B}_i(\hat{\mathbf{x}}) = \widehat{B}(\hat{\mathbf{x}} - \tilde{\mathbf{x}}_i)$, where $\tilde{\mathbf{x}}_i = (\tilde{x}^{(i_1)}, \dots, \tilde{x}^{(i_n)})$ and $\widehat{B}(\hat{\mathbf{x}}) = b(\hat{x}_1) \cdots b(\hat{x}_n)$. For future reference, define the set of all such $\tilde{\mathbf{x}}_i$ as $\tilde{\mathbb{X}}$. Also, notice that the ratio of cardinal B-splines basis functions to total B-spline basis functions quickly tends to unity, $(\frac{m-2p}{m})^n \rightarrow 1$, as m increases.

3. Surrogate matrices: Exploiting basis structure. Equations (2.1b) and (2.1c), respectively, induce two related matrix equations,

$$(3.1) \quad \mathbf{A}\mathbf{u} = \mathbf{f} \quad \text{and} \quad \tilde{\mathbf{A}}\tilde{\mathbf{u}} = \mathbf{f},$$

for basis function coefficients $\mathbf{u}, \tilde{\mathbf{u}} \in \mathbb{R}^N$. As mentioned previously, the key idea in this paper is constructing the majority of the surrogate stiffness matrix $\tilde{\mathbf{A}}$ via *interpolation* of the true stiffness matrix \mathbf{A} . In this section, we first describe exactly what is meant by this statement and then demonstrate how isogeometric analysis makes it possible.

3.1. Stencil functions. Recall (2.1b). Generally, every function $v_h \in V_h$ can be identified with a unique function on the domain $\widehat{\Omega}$ through a suitable pushforward operator φ_* . Namely, $v_h = \varphi_* \widehat{v}_h$. Define \widehat{V}_h be the set of all such \widehat{v}_h , which is a discrete space in the parametric domain $\widehat{\Omega}$. Accordingly, the bilinear form $a : V_h \times V_h \rightarrow \mathbb{R}$ can be identified with a parametric domain bilinear form $\widehat{a} : \widehat{V}_h \times \widehat{V}_h \rightarrow \mathbb{R}$ in such a way that $a(w_h, v_h) = \widehat{a}(\widehat{w}_h, \widehat{v}_h)$, for all $\widehat{w}_h, \widehat{v}_h \in \widehat{V}_h$.

Let $\{\phi_i\} = \{\varphi_* \widehat{\phi}_i\}$ be a basis for V_h with $\{\widehat{\phi}_i\}$ the corresponding basis for \widehat{V}_h . The fundamental observation in the surrogate matrix methodology now follows. If, $\widehat{\phi}$ is some fixed reference function and, for a set of indices i, j , $\widehat{\phi}_i(\widehat{\mathbf{x}}) = \widehat{\phi}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_i)$ and $\widehat{\phi}_j(\widehat{\mathbf{x}}) = \widehat{\phi}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_j)$, then

$$(3.2) \quad \widehat{a}(\widehat{\phi}_j, \widehat{\phi}_i) = \widehat{a}(\widehat{\phi}(\cdot - \widehat{\mathbf{x}}_j), \widehat{\phi}(\cdot - \widehat{\mathbf{x}}_i)) := \Phi(\widehat{\mathbf{x}}_j, \widehat{\mathbf{x}}_i).$$

Here, in the rightmost equality, the definition of a new scalar-valued function $\Phi(\cdot, \cdot)$ has been made, wherein any dependence on the mesh size h has been implicitly assumed. This function, $\Phi(\widehat{\mathbf{x}}_j, \widehat{\mathbf{x}}_i)$, may also be expressed in terms of $\widehat{\mathbf{x}}_i$ and a translation $\boldsymbol{\delta} = \widehat{\mathbf{x}}_j - \widehat{\mathbf{x}}_i$. In this alternative characterization, after denoting $\mathbf{A}_{ij} = a(\phi_j, \phi_i)$, we may write

$$(3.3) \quad \mathbf{A}_{ij} = \Phi_{\boldsymbol{\delta}}(\widehat{\mathbf{x}}_i)$$

and $\boldsymbol{\delta}$ may be treated as a parameter. We define $\Phi_{\boldsymbol{\delta}}(\widehat{\mathbf{x}}_i) = \Phi(\widehat{\mathbf{x}}_i + \boldsymbol{\delta}, \widehat{\mathbf{x}}_i) = \Phi(\widehat{\mathbf{x}}_j, \widehat{\mathbf{x}}_i)$.

For a fixed number of translations $\boldsymbol{\delta}$, these so-called *stencil functions*, $\Phi_{\boldsymbol{\delta}}(\cdot)$, can be identified with the majority of entries in many IGA stiffness matrices. In many circumstance, each $\Phi_{\boldsymbol{\delta}}(\cdot)$ is smooth and may, therefore, be interpolated after only being evaluated at small number of points in the parametric domain $\widehat{\mathbf{x}}_i \in \widehat{\Omega}$. After denoting the interpolants — i.e., the *surrogate stencil functions* — by $\widetilde{\Phi}_{\boldsymbol{\delta}}(\cdot)$, we simply define

$$(3.4) \quad \widetilde{\mathbf{A}}_{ij} = \widetilde{\Phi}_{\boldsymbol{\delta}}(\widehat{\mathbf{x}}_i).$$

REMARK 3.1. *In some cases, the stencil functions $\Phi_{\boldsymbol{\delta}}$ are themselves polynomials (see, e.g., Proposition 5.1 and Corollaries 8.1 and 10.1). Therefore, if polynomial interpolation of sufficiently high order is used, the true stiffness matrix be generated exactly — i.e., $\widetilde{\Phi}_{\boldsymbol{\delta}} = \Phi_{\boldsymbol{\delta}}$ and thus $\widetilde{\mathbf{A}} = \mathbf{A}$, up to round-off error — in significantly less time than with a traditional assembly algorithm. Otherwise, in many scenarios, a sufficiently accurate approximation of the stiffness matrix $\widetilde{\mathbf{A}} \approx \mathbf{A}$ will be generated.*

3.2. B-spline basis functions. Fix $V = H^1(\Omega)$, $\varphi : \widehat{\Omega} \rightarrow \Omega$, $\widehat{V}_h = \text{span}\{\widehat{B}_i\}$, and, accordingly, $V_h = \text{span}\{B_i\}$, where each $B_i = \widehat{B}_i \circ \varphi^{-1}$. Consider the bilinear form $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx$. It is easy to verify that $a(\cdot, \cdot)$ pulls back to

$$(3.5) \quad \widehat{a}(\widehat{w}, \widehat{v}) = \int_{\widehat{\Omega}} \widehat{\nabla} \widehat{w}(\widehat{\mathbf{x}})^{\top} K(\widehat{\mathbf{x}}) \widehat{\nabla} \widehat{v}(\widehat{\mathbf{x}}) \, d\widehat{\mathbf{x}}, \quad \text{where} \quad K = \frac{D\varphi^{-1} D\varphi^{-\top}}{|\det(D\varphi^{-1})|},$$

with arguments $\widehat{w}, \widehat{v} \in \widehat{V} = H^1(\widehat{\Omega})$.

Recall Subsection 2.2. Assume that $\{\widehat{B}_i\}$ is generated by an open uniform knot vector Ξ with $p > 0$ fixed. Obviously, $\mathbf{A}_{ij} = \widehat{a}(\widehat{B}_j, \widehat{B}_i)$. In the cardinal B-spline setting, $\widehat{B}_i(\widehat{\mathbf{x}}) = \widehat{B}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_i)$ and $\widehat{B}_j(\widehat{\mathbf{x}}) = \widehat{B}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_j)$. Therefore, by a simple change of variables,

$$(3.6) \quad \mathbf{A}_{ij} = \int_{\widehat{\Omega}} \widehat{\nabla} \widehat{B}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_i)^{\top} K(\widehat{\mathbf{x}}) \widehat{\nabla} \widehat{B}(\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_j) \, d\widehat{\mathbf{x}} = \int_{\widehat{\omega}_{\boldsymbol{\delta}}} \widehat{\nabla} \widehat{B}(\widehat{\mathbf{y}})^{\top} K(\widehat{\mathbf{x}}_i + \widehat{\mathbf{y}}) \widehat{\nabla} \widehat{B}_{\boldsymbol{\delta}}(\widehat{\mathbf{y}}) \, d\widehat{\mathbf{y}},$$

where $\boldsymbol{\delta} = \widehat{\mathbf{x}}_j - \widehat{\mathbf{x}}_i$, $\widehat{B}_{\boldsymbol{\delta}}(\widehat{\mathbf{y}}) = \widehat{B}(\widehat{\mathbf{y}} - \boldsymbol{\delta})$, and $\widehat{\omega}_{\boldsymbol{\delta}} = \text{supp}(\widehat{B}) \cap \text{supp}(\widehat{B}_{\boldsymbol{\delta}})$.

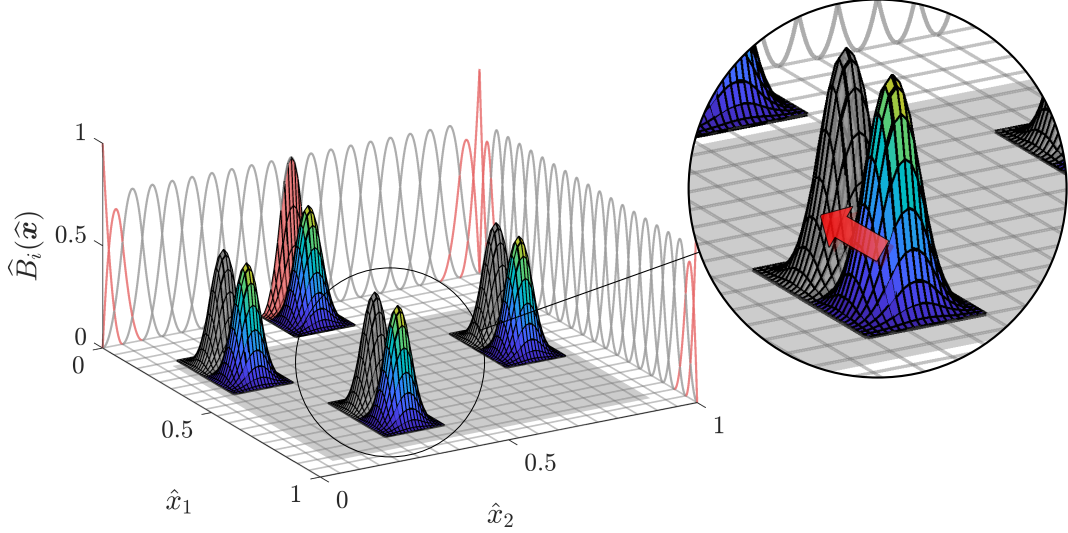


Figure 3.1: Second-order cardinal B-spline basis functions $\hat{B}_i(\hat{\mathbf{x}}) = \hat{B}(\hat{\mathbf{x}} - \tilde{\mathbf{x}}_i)$ (color gradient). After translation by $\boldsymbol{\delta} = \begin{pmatrix} -2h \\ 0 \end{pmatrix}$ (represented by the red arrow), most of these basis functions are equal to another cardinal basis function $\hat{B}_j(\hat{\mathbf{x}}) = \hat{B}_i(\hat{\mathbf{x}} - \boldsymbol{\delta})$ (gray). For every such function, $\tilde{\mathbf{x}}_i \in \tilde{\Omega}_\delta$ (this subset is shaded in gray on the mesh). Clearly, this property does not hold for every cardinal basis function, as indicated by the cardinal B-spline neighboring the boundary and the nearby (non-cardinal) basis function (red).

There is a natural correspondence between the density of the matrix \mathbf{A} and the set of translations $\boldsymbol{\delta}$ such that $\hat{\omega}_\delta \neq \emptyset$. Consequently, the cardinality of the set of relevant translations, $\mathcal{D} = \{\boldsymbol{\delta} = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i : \hat{\omega}_\delta \neq \emptyset\}$, is fixed for all sufficiently large m . Namely, $|\mathcal{D}| = (2p + 1)^n$. Now, for each $\boldsymbol{\delta} \in \mathcal{D}$, we may define the stencil function

$$(3.7) \quad \Phi_\delta(\tilde{\mathbf{x}}) = \int_{\hat{\omega}_\delta} \hat{\nabla} \hat{B}(\hat{\mathbf{y}})^\top K(\tilde{\mathbf{x}} + \hat{\mathbf{y}}) \hat{\nabla} \hat{B}_\delta(\hat{\mathbf{y}}) d\hat{\mathbf{y}}.$$

Let $\text{conv}(\tilde{\mathbf{X}})$ denote the convex hull of $\tilde{\mathbf{X}}$. Such functions are defined at any point $\tilde{\mathbf{x}} \in \text{conv}(\tilde{\mathbf{X}})$ where $\tilde{\mathbf{x}} + \boldsymbol{\delta} \in \text{conv}(\tilde{\mathbf{X}})$. Ultimately, this means that the domain of Φ_δ , which we will denote by $\tilde{\Omega}_\delta$, depends on $\boldsymbol{\delta}$ (see, e.g., [Figure 3.1](#)). Clearly, we always have $\mathbf{0} \in \mathcal{D}$. The reader may easily verify that $\tilde{\Omega}_0 = \text{conv}(\tilde{\mathbf{X}}) = \bigcup_{\boldsymbol{\delta} \in \mathcal{D}} \tilde{\Omega}_\delta$ and $\tilde{\Omega}_\delta + \boldsymbol{\delta} \subseteq \tilde{\Omega}_0$, for each $\boldsymbol{\delta} \in \mathcal{D}$.

3.3. NURBS basis functions. The principal difference between the treatment of a NURBS basis $\{\hat{N}_i\}$ and the related B-spline basis $\{\hat{B}_i\}$ is that a NURBS basis cannot be assumed to have the translation invariance property which leads directly to [\(3.2\)](#). Fortunately, as we now demonstrate, this property is not entirely necessary to define a useful stencil function.

Define $W(\hat{\mathbf{x}}) = \sum_j w_j \hat{B}_j(\hat{\mathbf{x}})$, where $\{w_j\}$ are the weight parameters appearing in [\(2.2\)](#). It is known that $W(\hat{\mathbf{x}})$ is unchanged under mesh refinements. Therefore, employing a similar change of variables argument as used in [\(3.6\)](#), it holds that

$$(3.8) \quad \mathbf{A}_{ij} = \hat{a}(\hat{N}_j, \hat{N}_i) = w_i w_j \int_{\hat{\omega}_\delta} \hat{\nabla} \left(\frac{\hat{B}(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}}_i + \hat{\mathbf{y}})} \right)^\top K(\tilde{\mathbf{x}}_i + \hat{\mathbf{y}}) \hat{\nabla} \left(\frac{\hat{B}_\delta(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}}_i + \hat{\mathbf{y}})} \right) d\hat{\mathbf{y}},$$

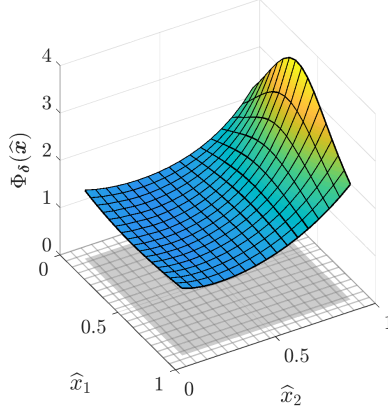


Figure 3.2: The graph of a stencil function $\Phi_{\delta} : \tilde{\Omega}_{\delta} \rightarrow \mathbb{R}$ defined by (3.9). The corresponding physical geometry Ω is depicted in Figure 7.1 (a). In this case, $\delta = \mathbf{0} \cdot h$ and $p = 2$. The domain $\tilde{\Omega}_{\mathbf{0}}$ is shaded gray. For further details, see Subsection 7.3.

where, as in (3.6), $\delta = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$. At this point, it may be natural to divide by $w_i w_j$ and define the stencil function $\Phi_{\delta}(\tilde{\mathbf{x}})$ from the resulting expression on the right-hand side of (3.8). Instead, we pause to consider the regularity of $W(\hat{\mathbf{x}})$.

Recall (3.4). Since each $\hat{B}_i(\hat{\mathbf{x}})$ is only piecewise polynomial, it is clear that, in general, $W \notin C^q(\hat{\Omega})$, for any $q \geq p$. This fact could significantly limit the accuracy of an interpolant $\tilde{\Phi}_{\delta} = \Pi\Phi_{\delta}$ which we may wish to construct. Therefore, we restrict our attention to $W \in C^q(\hat{\Omega})$, where $q \geq p$. It turns out that this set of functions, $\text{span}\{\hat{B}_i\} \cap C^q(\hat{\Omega})$, is equal to the polynomial space $\mathcal{Q}_p(\hat{\Omega})$. Moreover, restricting to the subset $\tilde{\Omega}_{\mathbf{0}}$, each weight parameter can be expressed as $w_i = w(\tilde{\mathbf{x}}_i)$, where $w(\hat{\mathbf{x}})$ is a polynomial in $\mathcal{Q}_p(\hat{\Omega}_{\mathbf{0}})$. (See Appendix A for details.) Therefore, for any $W \in \mathcal{Q}_p(\hat{\Omega})$, we may define

$$(3.9) \quad \Phi_{\delta}(\tilde{\mathbf{x}}) = w(\tilde{\mathbf{x}})w(\tilde{\mathbf{x}} + \delta) \int_{\tilde{\omega}_{\delta}} \hat{\nabla} \left(\frac{\hat{B}(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}} + \hat{\mathbf{y}})} \right)^{\top} K(\tilde{\mathbf{x}} + \hat{\mathbf{y}}) \hat{\nabla} \left(\frac{\hat{B}_{\delta}(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}} + \hat{\mathbf{y}})} \right) d\hat{\mathbf{y}},$$

for each $\tilde{\mathbf{x}} \in \tilde{\Omega}_{\delta}$ and $\delta \in \mathcal{D}$. Clearly, $A_{ij} = \Phi_{\delta}(\tilde{\mathbf{x}}_i)$ for each corresponding $\delta \in \mathcal{D}$. An illustration of a stencil function coming from an IGA discretization with a NURBS basis is presented in Figure 3.2.

REMARK 3.2. *Notably, when $W = 1$, then $w = 1$ also. Therefore, (3.9) is consistent with (3.7). Obviously, in practice, neither of these expressions needs to be used in order to evaluate $\Phi_{\delta}(\cdot)$ at any point $\tilde{\mathbf{x}}_i$. Indeed, since $\Phi_{\delta}(\tilde{\mathbf{x}}_i) = A_{ij}$, any existing IGA code already has a mechanism to compute $\Phi_{\delta}(\tilde{\mathbf{x}}_i)$ using quadrature (cf. Section 6). Nevertheless, these expressions are important for analysis.*

3.4. Symmetric bilinear forms. When $a(w, v) = a(v, w)$, for all $w, v \in V$, a translational symmetry is induced on the set of corresponding stencil functions. Indeed, it is simple to see that $\Phi(\tilde{\mathbf{x}}_j, \tilde{\mathbf{x}}_i) = \Phi(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ and, therefore,

$$\Phi_{\delta}(\tilde{\mathbf{x}}_i) = \Phi(\tilde{\mathbf{x}}_i + \delta, \tilde{\mathbf{x}}_i) = \Phi(\tilde{\mathbf{x}}_j - \delta, \tilde{\mathbf{x}}_j) = \Phi_{-\delta}(\tilde{\mathbf{x}}_j).$$

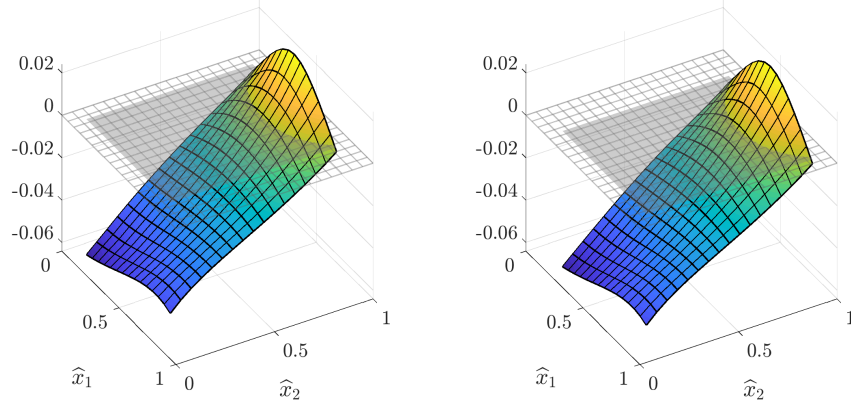


Figure 3.3: Left and right, respectively: Graphs of stencil functions $\Phi_{\delta}(\tilde{\mathbf{x}})$ for $\delta = \begin{pmatrix} 2h \\ 0 \end{pmatrix}$ and $\delta = \begin{pmatrix} -2h \\ 0 \end{pmatrix}$. The associated domains $\tilde{\Omega}_{\delta}$ are shaded in gray (cf. Figure 3.1). Notice the translational symmetry $\Phi_{\delta}(\tilde{\mathbf{x}}) = \Phi_{-\delta}(\tilde{\mathbf{x}} + \delta)$.

A similar conclusion can be drawn in the NURBS scenario above. Figure 3.3 presents a visual comparison of two stencil functions, Φ_{δ} and $\Phi_{-\delta}$, generated by an isogeometric NURBS basis and the corresponding symmetric bilinear form (3.5).

3.5. The multi-patch setting. In the multi-patch setting, the physical domain Ω is partitioned into a finite number of disjoint subdomains $\bar{\Omega} = \bigcup_{k=1}^L \bar{\Omega}^{(k)}$. We may assume that each such domain or *patch* $\Omega^{(k)}$, as they are usually called, can be identified with a common parametric domain $\hat{\Omega}$, through a unique NURBS mapping $\Omega^{(k)} = \varphi^{(k)}(\hat{\Omega})$. In this case, one may define a separate set of stencil functions $\Phi_{\delta}^{(k)}(\tilde{\mathbf{x}}) : \tilde{\Omega}_{\delta} \rightarrow \mathbb{R}$, for each index k . The forthcoming analysis is immediately applicable to this setting, but treating it outright would require unnecessarily complicated notation. We also did not consider this setting in our numerical experiments.

3.6. Edge-based and face-based stencil functions. It is also possible to define stencil functions, in addition to those above, which exploit lower-dimensional translational symmetries. For instance, many functions in the basis $\{\hat{N}_i\}$ are only equivalent under translations parallel to a given edge in the parametric domain $\hat{\Omega}$. Likewise, a new family of stencil functions could be constructed for each edge or face in a patch $\Omega^{(k)}$. We do not pursue this observation here.

4. Surrogate matrices: Interpolation of stencil functions. Recall (3.3) and (3.4). A surrogate matrix $\tilde{\mathbf{A}} \approx \mathbf{A}$ will be useful to us if: (1) each stencil function Φ_{δ} has high regularity and, therefore, the high-order approximation $\Pi\Phi_{\delta} \approx \Phi_{\delta}$ will have high-order accuracy; and (2) each $\tilde{\Phi}_{\delta} = \Pi\Phi_{\delta}$ can be computed and evaluated fast. In this section, we focus on the former of these two requirements. Particulars on our implementation are withheld until Section 6.

As a simplifying accommodation, we perform all of our analysis on the largest subset of $\hat{\Omega}$ where every stencil function is defined. That is, $\tilde{\Omega} = \bigcap_{\delta \in \mathcal{D}} \tilde{\Omega}_{\delta}$. A simple computation shows that

$$\tilde{\Omega} = \left[\frac{3p+1}{2(m-p)}, 1 - \frac{3p+1}{2(m-p)} \right]^n \subsetneq \hat{\Omega}.$$

All of the results in this section can be reformulated with $\tilde{\Omega}$ replaced by $\tilde{\Omega}_\delta$. However, this generalization also requires a different operator Π for each $\delta \in \mathcal{D}$; see, e.g., [21].

4.1. B-spline interpolation. Let $\{\tilde{B}_j\}$ be a degree $q \geq 0$ multivariate B-spline basis on $\tilde{\Omega}$ with the quasi-uniform knot vector $\tilde{\Xi} = \tilde{\Xi}_1 \times \cdots \times \tilde{\Xi}_n$ and define $S_q(\tilde{\Xi}) = \text{span}\{\tilde{B}_j\}$. We will refer to each $\tilde{\xi}_j \in \tilde{\Xi}$, $\mathbf{j} = (j_1, \dots, j_n)$, as a *sampling point* and define a new length scale parameter, hereby referred to as the *sampling length*, $H = \max_{|j|=1, i} \{\|\tilde{\xi}_{i+j} - \tilde{\xi}_i\|_{\max} : \tilde{\xi}_{i+j} \in \tilde{\Xi}\}$.

In isogeometric analysis, the geometry determines the basis used in PDE discretization. When constructing the surrogate stencil functions $\tilde{\Phi}_\delta$, we are primarily interested in using a basis of higher order q than the underlying spatial discretization p . Generally, such a choice $q > p$ is desirable because it will allow us to guarantee that the discretization error in a standard IGA method will dominate the error actually attributed to using a surrogate (cf. Subsection 7.3.2).

In this paper, we only consider constructing B-spline interpolants $\tilde{\Phi}_\delta = \Pi_H \Phi_\delta$, where Π_H is a stable local interpolation operator onto the space $S_q(\tilde{\Xi})$. Various global interpolants could also be considered, as well as sparse grid interpolants [9] and least-squares projections (cf. [21]). We see no benefit in using a NURBS basis to approximate Φ_δ , even when a NURBS mapping $\varphi : \tilde{\Omega} \rightarrow \Omega$ defines the physical domain. The following lemma follows directly from [14, Theorem 4.2].

LEMMA 4.1. *For every bounded projection $\Pi_H : C^0(\tilde{\Omega}) \rightarrow S_q(\tilde{\Xi})$, with $\|\Pi_H\| < C_0$ for some H -independent constant C_0 , it holds that*

$$\|f - \Pi_H f\|_{L^\infty(\tilde{\Omega})} \leq C_1 H^{q+1} [f]_{W^{q+1, \infty}(\tilde{\Omega})}, \quad \text{for all } f \in W^{q+1, \infty}(\tilde{\Omega}),$$

where C_1 is a constant depending only on q , $\tilde{\Omega}$, and $\|\Pi_H\|$.

Figure 4.1 presents graphs of stencil functions, $\Phi_\delta(\tilde{\mathbf{x}})$ and their surrogates $\tilde{\Phi}_\delta(\tilde{\mathbf{x}})$ obtained by B-spline interpolation for various δ . The stencil functions are generated by an isogeometric NURBS basis and the symmetric bilinear form (3.5).

REMARK 4.1. *The norm $\|\Pi_H\|$ can be greatly influenced by the distribution of the sampling points $\tilde{\xi}_j \in \tilde{\Xi}$. In all of our experiments, we kept $\tilde{\Xi} \subseteq \tilde{\Omega} \cap \tilde{\mathbb{X}}$. This convenient choice delivered good results. When we wish to underscore the convention $\tilde{\Xi} \subseteq \tilde{\Omega} \cap \tilde{\mathbb{X}}$, we will denote the sampling points in $\tilde{\Xi}$ by $\tilde{\mathbf{x}}_i^s$. Moreover, from now on, dependence on the subset $\tilde{\Omega}$ will not be stated since $\tilde{\Omega} \rightarrow \hat{\Omega}$ as $h \rightarrow 0$ and the parametric domain $\hat{\Omega} = (0, 1)^n$ is always fixed.*

4.2. Regularity of the stencil functions. Define $\tilde{\Phi}_\delta = \Pi_H \Phi_\delta$, where Π_H is any projection satisfying the assumptions of Lemma 4.1. In this subsection, we present an essential theorem on the error in the class of surrogate stencil functions defined in (3.9). In order to expedite our presentation, we only prove Theorem 4.2 here under an assumption which directly relates to the B-spline basis scenario (3.7). The general proof is given in Appendix A.

THEOREM 4.2. *Let $\Phi_\delta : \tilde{\Omega} \rightarrow \mathbb{R}$ be defined by (3.9) and assume that $\varphi : \hat{\Omega} \rightarrow \Omega$ induces a coefficient tensor $K \in [W^{q+1, \infty}(\hat{\Omega})]^{n \times n}$. If $W \in \mathcal{Q}_p(\hat{\Omega})$, then there exists a constant C_2 , depending only on p , q , $\|\Pi_H\|$, and φ , such that*

$$\|\Phi_\delta - \tilde{\Phi}_\delta\|_{L^\infty(\tilde{\Omega})} \leq C_2 h^{n-2} H^{q+1} \quad \text{for each } \delta \in \mathcal{D}.$$

Moreover, if $W(\hat{\mathbf{x}}) = 1$, then $C_2 \leq C C_1 [K]_{W^{q+1, \infty}(\hat{\Omega})}$, for some C depending only on p .

Proof of Theorem 4.2, under the assumption $W(\hat{\mathbf{x}}) = 1$. For every multi-index $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$, with each $0 \leq \alpha_i \leq q + 1$, it holds that

$$D^\alpha \Phi_\delta(\tilde{\mathbf{x}}) = \int_{\hat{\omega}_\delta} \widehat{\nabla} \widehat{B}(\hat{\mathbf{y}})^\top D^\alpha K(\tilde{\mathbf{x}} + \hat{\mathbf{y}}) \widehat{\nabla} \widehat{B}_\delta(\hat{\mathbf{y}}) d\hat{\mathbf{y}}.$$

Therefore, $\|D^\alpha \Phi_\delta\|_{L^\infty(\tilde{\Omega})} \leq \|D^\alpha K\|_{L^\infty(\hat{\Omega})} \|\widehat{\nabla} \widehat{B} \cdot \widehat{\nabla} \widehat{B}_\delta\|_{L^1(\hat{\omega}_\delta)} \leq Ch^{n-2} \|D^\alpha K\|_{L^\infty(\hat{\Omega})}$ and, moreover, $\Phi_\delta \in W^{q+1, \infty}(\tilde{\Omega})$. The result follows from Lemma 4.1 using $f = \Phi_\delta$. \square

REMARK 4.2. *Observe that $[K]_{W^{q+1, \infty}(\hat{\Omega})} = 0$ iff $K \in [\mathcal{Q}_q(\hat{\Omega})]^{n \times n}$. Generally, due to the definition of K appearing in (3.5), this assumption can only be expected to be satisfied when the geometry map $\boldsymbol{\varphi}$ is affine. Nevertheless, if $W = 1$ as well, then Theorem 4.2 would imply that $\tilde{\Phi}_\delta = \Phi_\delta \in \mathcal{Q}_q(\tilde{\Omega})$. This is a useful reproduction property which can help to verify the implementation. It also manifests in stencil functions defined from more complicated bilinear forms than (3.5). The reproduction of stencil functions is described in a general scenario in Subsection 5.4.*

5. Surrogate matrices: Preserving structure. In this section, we discuss a number of different surrogate matrix definitions. Under both definitions, we also show that the surrogate matrices may actually fully reproduce the very matrices they approximate. Note that these definitions may also be employed for general matrices and not only for a stiffness matrix. If not stated otherwise, in the following subsections we assume that the matrix \mathbf{A} emanates from a discretization of a general bilinear form.

5.1. General surrogate matrices. In Subsection 3.1, the definition $\tilde{\mathbf{A}}_{ij} = \tilde{\Phi}_\delta(\tilde{\mathbf{x}}_i)$ was presented, with $\boldsymbol{\delta} = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$. However, this is valid exclusively for the special indices $1 \leq i, j \leq N$ with a cardinal B-spline associated to them. Of course, we may always compute the remaining components of the surrogate matrix $\tilde{\mathbf{A}}$ directly, using element-wise quadrature, as done in standard IGA assembly algorithms. Therefore, we propose the following definition:

$$(5.1a) \quad \tilde{\mathbf{A}}_{ij} = \begin{cases} \tilde{\Phi}_\delta(\tilde{\mathbf{x}}_i) & \text{if } \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\Omega}, \\ \mathbf{A}_{ij} & \text{otherwise.} \end{cases}$$

This is a convenient definition which essentially can be used for constructing a surrogate matrix from any general bilinear form $a(\cdot, \cdot)$. Typically, this definition is used for general non-symmetric matrices, e.g., for the divergence matrices \mathbf{B} arising in the discretization of Stokes' flow (cf. Section 10). However, (5.1a) can easily be improved when the bilinear form is symmetric or has a well-known kernel.

5.2. Symmetry. If the bilinear form is symmetric, (5.1a) does not guarantee that the corresponding surrogate stiffness matrix $\tilde{\mathbf{A}}$ will be symmetric. In order to enforce symmetry, it is convenient to just include the action of copying \mathbf{A}_{ij} into \mathbf{A}_{ji} , for all $i > j$. Therefore, we propose the following symmetric surrogate matrix definition:

$$(5.1b) \quad \tilde{\mathbf{A}}_{ij} = \begin{cases} \tilde{\Phi}_\delta(\tilde{\mathbf{x}}_i) & \text{if } \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\Omega} \text{ and } i \leq j, \\ \tilde{\mathbf{A}}_{ji} & \text{if } \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\Omega} \text{ and } i > j, \\ \mathbf{A}_{ij} & \text{otherwise.} \end{cases}$$

With the definition, note that only $\frac{(2p+1)^n + 1}{2}$ surrogate stencil functions need to be computed. We employ this definition in the construction of surrogate mass matrices \mathbf{M} which are symmetric but do not have a kernel (cf. Section 8).

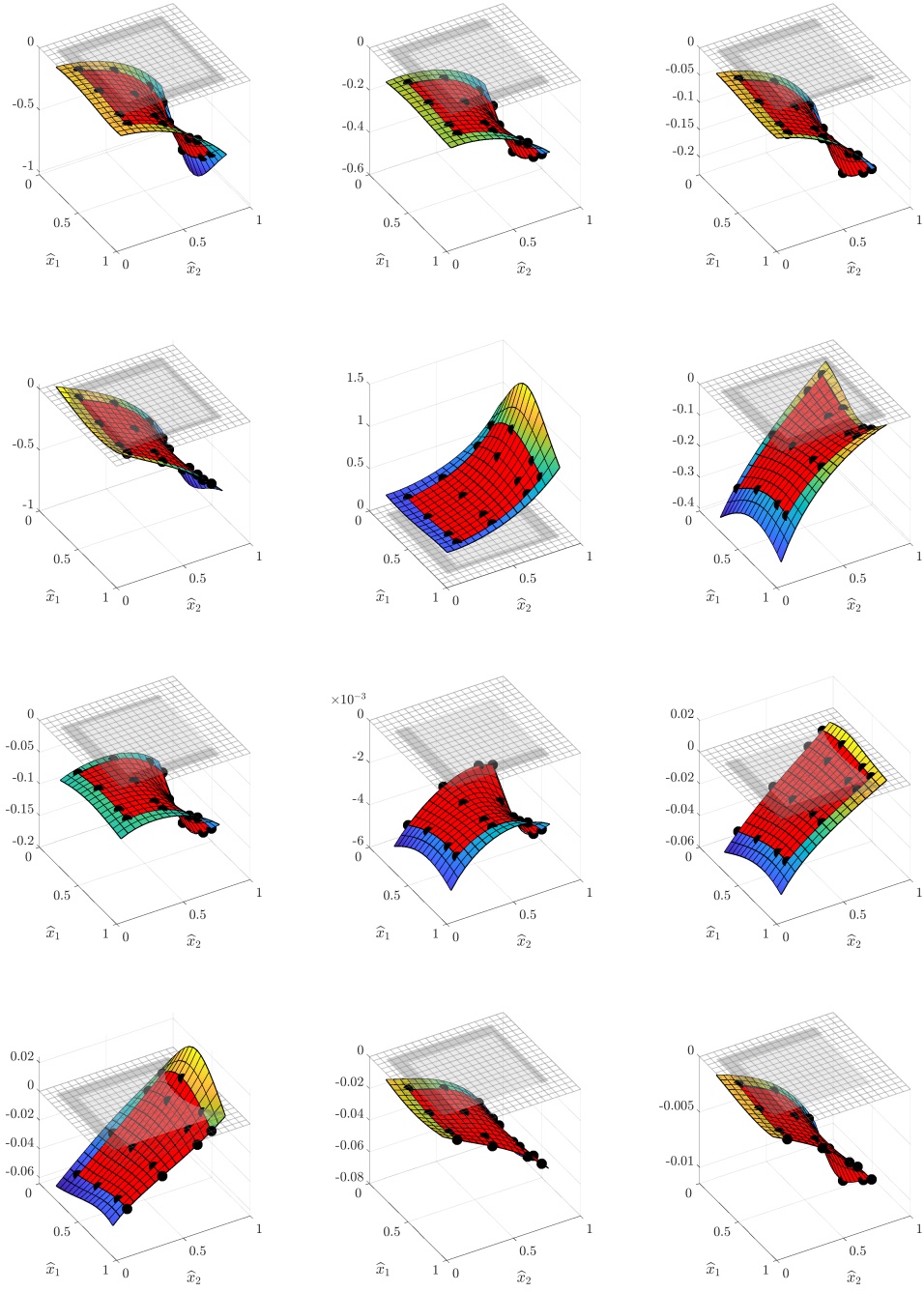


Figure 4.1: Graphs of the stencil functions $\Phi_\delta(\tilde{\mathbf{x}})$ (color gradient) and their surrogates $\tilde{\Phi}_\delta(\tilde{\mathbf{x}})$ (red) for the various δ required in populating the upper diagonal of the stiffness matrix A_{ij} defined in (5.1). The sampling points $\tilde{\mathbf{x}}_i^s \in \tilde{\Xi} \subset \tilde{\Omega} \cap \tilde{\mathbb{X}}$ used to construct the interpolant are depicted with black dots. The subsets $\Omega \subseteq \Omega_\delta$ are highlighted in light gray and the remaining subsets $\Omega_\delta \setminus \Omega$ are depicted in a darker gray.

5.3. Preserving the kernel. Recall that $\sum N_i(\mathbf{x}) = 1$. In the situation $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx$, we have that $a(1, w) = a(w, 1) = 0$, for all $w \in H^1(\Omega)$. Therefore, $1 \in V_h = \text{span}\{N_i\}$ and, moreover, $\mathbf{A}\mathbf{v}_1 = 0$, where $\mathbf{v}_1 = (1, \dots, 1)^\top$. Clearly, neither definition (5.1a) nor definition (5.1b), will guarantee that $\tilde{\mathbf{A}}\mathbf{v}_1 = 0$. Therefore, we will pose the following symmetric kernel-preserving definition:

$$(5.1c) \quad \tilde{\mathbf{A}}_{ij} = \begin{cases} \tilde{\Phi}_{\delta}(\tilde{\mathbf{x}}_i) & \text{if } \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\Omega} \text{ and } i < j, \\ \tilde{\mathbf{A}}_{ji} & \text{if } \tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \tilde{\Omega} \text{ and } i > j, \\ \mathbf{A}_{ij} & \text{in all other cases where } i \neq j \\ -\sum_{k \neq i} \tilde{\mathbf{A}}_{ik} & \text{if } i = j, \end{cases}$$

With this definition, the reader may readily verify that $\tilde{\mathbf{A}}\mathbf{v}_1 = 0$ and $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}^\top$.

REMARK 5.1. *Two important comments are in order. First, in a matrix-free setting, where memory copying cannot be performed efficiently, one can actually design a set of surrogate stencil functions $\tilde{\Phi}_{\delta}$ which preserves the symmetry of the stiffness matrix (see, e.g., [21, Remark 3.4]). Second, in general, it is difficult to generalize the row sum trick in (5.1c), in a way which preserves symmetry, when the bilinear form $a(\cdot, \cdot)$ has a multi-dimensional kernel (cf. Section 9).*

5.4. Polynomial reproduction. Until now, we have focused, almost entirely, on the analysis of surrogate stiffness matrices which derive from the bilinear form generated by Poisson's equation. Clearly, the methodology presented above can be applied to other settings as well. The general scenario we are interested in is when $a(\cdot, \cdot)$ in (2.1a) can be expressed in the parametric domain as

$$(5.2a) \quad \hat{a}(\hat{w}, \hat{v}) = \int_{\hat{\Omega}} G(\hat{\mathbf{x}}, \hat{w}(\hat{\mathbf{x}}), \hat{v}(\hat{\mathbf{x}})) \, d\hat{\mathbf{x}} \quad \text{for all } \hat{w}, \hat{v} \in \hat{V},$$

where, for all smooth \hat{w}, \hat{v} ,

$$(5.2b) \quad G(\hat{\mathbf{x}}, \hat{w}(\hat{\mathbf{y}}), \hat{v}(\hat{\mathbf{y}})) = 0, \quad \text{whenever } \hat{\mathbf{y}} \notin \text{supp}(\hat{w}) \cap \text{supp}(\hat{v}).$$

We now consider the general coefficient matrix \mathbf{A} and a cardinal B-spline basis $\{\hat{B}_i\}$ (cf. Subsection 3.2). Invoking (5.2b), a simple change of variables leads us to

$$\mathbf{A}_{ij} = \hat{a}(\hat{B}_j, \hat{B}_i) = \int_{\hat{\omega}_{\delta}} G(\tilde{\mathbf{x}}_i + \hat{\mathbf{y}}, \hat{B}_{\delta}(\hat{\mathbf{y}}), \hat{B}(\hat{\mathbf{y}})) \, d\hat{\mathbf{y}},$$

where, as before, $\delta = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$ and $\hat{\omega}_{\delta} = \text{supp}(\hat{B}) \cap \text{supp}(\hat{B}_{\delta})$. Using the techniques put forth in Subsection 3.3, this expression can easily be generalized for cardinal NURBS bases with polynomial weight functions $W(\hat{\mathbf{x}})$. However, considering only the case of a cardinal B-spline basis, if $G(\cdot, \cdot, \cdot)$ is a $\mathcal{Q}_p(\hat{\Omega})$ polynomial in its first argument, we have the following reproduction property (cf. Remark 4.2).

PROPOSITION 5.1. *Assume that (5.2a) and (5.2b) hold. For all $\hat{\mathbf{x}} \in \tilde{\Omega}$, define*

$$(5.3) \quad \Phi_{\delta}(\tilde{\mathbf{x}}) = \int_{\hat{\omega}_{\delta}} G(\tilde{\mathbf{x}} + \hat{\mathbf{y}}, \hat{B}_{\delta}(\hat{\mathbf{y}}), \hat{B}(\hat{\mathbf{y}})) \, d\hat{\mathbf{y}}.$$

If $G(\cdot, \hat{\mathbf{y}}, \hat{\mathbf{y}}) \in \mathcal{Q}_p(\hat{\Omega})$, for every $\hat{\mathbf{y}} \in \hat{\Omega}$, then $\Phi_{\delta} \in \mathcal{Q}_q(\tilde{\Omega})$. Moreover, taking (5.1a) as the definition of the surrogate $\tilde{\mathbf{A}}$, it holds that $\tilde{\mathbf{A}} = \mathbf{A}$.

Proof. It suffices to show that $\tilde{\Phi}_\delta = \Phi_\delta$, for all $\delta \in \mathcal{D}$. Let $\alpha = (\alpha_1, \dots, \alpha_n)$ be a multi-index, $\tilde{\mathbf{x}}^\alpha = \tilde{x}_1^{\alpha_1} \dots \tilde{x}_n^{\alpha_n}$. By assumption, we may express $G(\tilde{\mathbf{x}}, \hat{B}_\delta(\hat{\mathbf{y}}), \hat{B}(\hat{\mathbf{y}})) = \sum_{i=1}^n \sum_{\alpha_i \leq p} c_\alpha(\hat{\mathbf{y}}) \tilde{\mathbf{x}}^\alpha$, where each coefficient function $c_\alpha(\hat{\mathbf{y}})$ has support only in $\hat{\omega}_\delta$. Moreover, if $\tilde{\mathbf{x}} + \hat{\mathbf{y}} \in \hat{\Omega}$, then

$$G(\tilde{\mathbf{x}} + \hat{\mathbf{y}}, \hat{B}_\delta(\hat{\mathbf{y}}), \hat{B}(\hat{\mathbf{y}})) = \sum_{i=1}^n \sum_{\alpha_i \leq p} c_\alpha(\hat{\mathbf{y}}) (\tilde{\mathbf{x}} + \hat{\mathbf{y}})^\alpha$$

is clearly an equal degree polynomial in the $\tilde{\mathbf{x}}$ -variable. The proof is completed noting that the integral in (5.3) is performed in the $\hat{\mathbf{y}}$ -variable over the set $\hat{\omega}_\delta$ and $\hat{\omega}_\delta \subseteq \hat{\Omega}$, for every $\delta \in \mathcal{D}$. \square

6. Surrogate matrices: Faster assembly with existing software. All the examples in this paper were implemented using the GeoPDEs package for Isogeometric Analysis in MATLAB and Octave [20, 42]. This package provides a framework for implementing and testing new isogeometric methods for the solution of partial differential equations. We reused most of the original low-level functions and only had to make changes to some high-level assembly functions. A detailed explanation of the code modifications and extensions is provided in [22]. Additionally, a reference implementation with example code is available in the git repository [1]. Nonetheless, we give here a short explanation of the implementation in GeoPDEs. In particular, for the Poisson problem, we modified `op_gradu_gradv_tp` using the following strategy:

First, the sampling length H and the sampling points need to be specified. For this purpose, we introduce the sampling parameter $M \in \mathbb{N}$ which relates the small scale h to the coarse scale H via $H = M \cdot h$. Starting from the first point $\tilde{\mathbf{x}}_i \in \hat{\Omega} \cap \tilde{\mathbb{X}}$, let $\tilde{\Xi}$ be the lattice containing every M^{th} point $\tilde{\mathbf{x}}_i \in \hat{\Omega} \cap \tilde{\mathbb{X}}$, in each Cartesian direction. When M does not evenly divide these points in any given Cartesian direction, include the M^{th} endpoints $\tilde{\mathbf{x}}_i \in \partial \hat{\Omega} \cap \tilde{\mathbb{X}}$ as well; see, e.g., the black dots in Figure 4.1. The stencil functions $\Phi_\delta(\tilde{\mathbf{x}})$ are evaluated at all points $\tilde{\mathbf{x}}_i^s \in \tilde{\Xi}$ and these values are used as the support points of the ensuing B-spline interpolant $\tilde{\Phi}_\delta(\tilde{\mathbf{x}})$.

An additional benefit of this choice is seen in that $\tilde{\Phi}_\delta(\tilde{\mathbf{x}}_i^s) = \Phi_\delta(\tilde{\mathbf{x}}_i^s)$, at each point $\tilde{\mathbf{x}}_i^s \in \tilde{\Xi}$. This leads to an increased point-wise accuracy and lower potential cost, since each entry $\tilde{A}_{ij} = \tilde{\Phi}_\delta(\tilde{\mathbf{x}}_i^s)$ in the surrogate stiffness matrix is equal to the correct entry, $A_{ij} = \Phi_\delta(\tilde{\mathbf{x}}_i^s)$. Here, it is appropriate to point out that when $M = 1$ every point $\tilde{\mathbf{x}}_i$ is sampled, $\tilde{\Xi} = \hat{\Omega} \cap \tilde{\mathbb{X}}$. In this case, $H = h$ and there is no difference from the surrogate \tilde{A} and the true A .

In order to evaluate $\Phi_\delta(\tilde{\mathbf{x}}_i^s)$, we identify the matrix rows which correspond to the sampling points $\tilde{\mathbf{x}}_i^s \in \tilde{\Xi}$. Additionally, we include the rows which correspond to basis functions near the domain boundary. Each of these rows needs to be assembled using quadrature formulas; see the red and green points in Figure 6.1. The number of interior rows depends on M , whereas the number of rows corresponding to the boundary depends on the order of the basis functions p . After that, we identify all of the active elements which need to be assembled to compute the estimated rows, cf. Figure 6.2. Note that the number of required elements for each sample point depends on the order p . In order to assemble these elements, we employ the `op_gradu_gradv` function, but skip the elements which are not active.

To construct the interpolated stencil functions $\tilde{\Phi}_\delta$, it is possible to use the builtin MATLAB functions `interp2` and `interp3`. However, in 2D, we use the `RectBivariateSpline` function provided by the SciPy Python package [31], which supports spline interpolation up to order 5. These interpolated stencil functions are then evaluated in order to retrieve the remaining values of \tilde{A} ; cf. the blue off-diagonal entries in Figure 6.1. The assembly functions for the mass matrix, the biharmonic equation, and the Stokes problem were modified in a similar way. For symmetric matrices, only the upper-diagonal entries are interpolated and copied to the lower-diagonal entries (cf. (5.1b)). In the Poisson and biharmonic case, we additionally enforce the zero-row sum property by changing the diagonal entries for all rows which include at least one interpolated value (cf. (5.1c)).

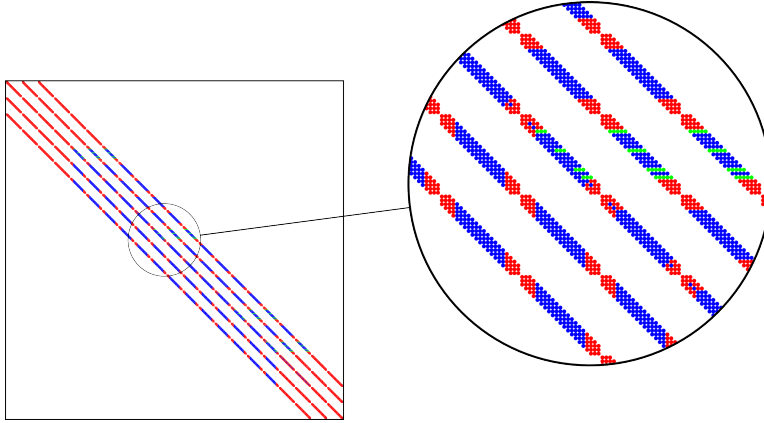


Figure 6.1: Sparsity pattern of the surrogate stiffness matrix $\tilde{\mathbf{A}}$. The red and green points indicate the entries of the stiffness matrix which are evaluated in the standard way. The blue points indicate the entries which are obtained by evaluating the interpolated stencil functions. The red points correspond to the basis functions near the boundaries and the green entries are used as supporting points for the interpolation. Some of the diagonal entries are drawn in blue due to the modification of preserving the kernel.

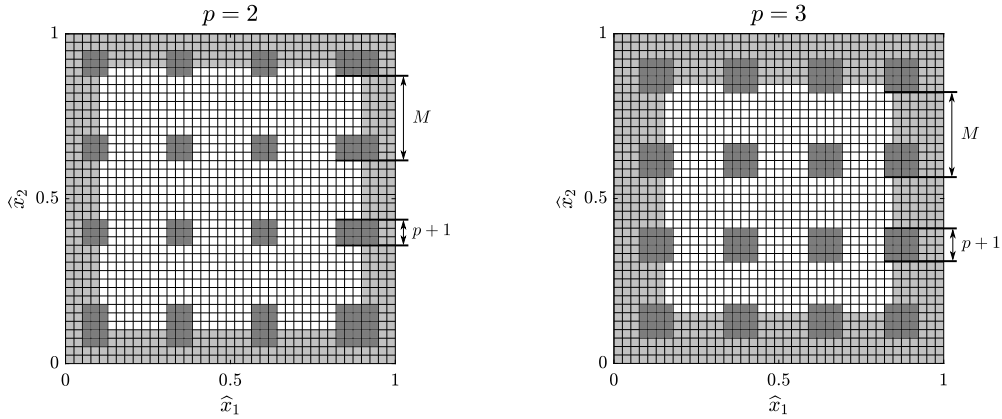


Figure 6.2: The active elements (shown in gray) involved in the surrogate assembly for $M = 10$ with forty knots in each Cartesian direction. The light gray elements correspond to the active boundary elements and the dark gray elements correspond to the inner active elements required for the sampling of the stencil functions.

In order to show that the surrogate approach may be easily applied to other IGA frameworks, we tried to keep the modifications as simple as possible. However, in an IGA implementation tailored to the surrogate approach, even more properties may be exploited to achieve better performance. For example, in the current implementation, the complete local stiffness matrices of the active elements are computed via quadrature, but in practice only a single row of the local matrix is required. Exploiting this fact would save a significant amount of unnecessary computation, especially as p grows, but such an implementation in GeoPDEs would also involve the modification of low-level functions.

REMARK 6.1. *Dirichlet boundary conditions are enforced for surrogate methods in the standard way; that is, by eliminating dofs from the original linear system. As usual, let $\tilde{\mathbf{A}}$ be the full matrix without any consideration for boundary conditions. Without loss of generality, we may assume that all the smallest global indices correspond to the set of interior dofs, denoted by I , and all of the largest correspond to the set of Dirichlet dofs, denoted by D . The unconstrained linear system, $\tilde{\mathbf{A}}\tilde{\mathbf{u}} = \mathbf{f}$, may then be written in block form as follows:*

$$(6.1) \quad \begin{bmatrix} \tilde{\mathbf{A}}_{II} & \tilde{\mathbf{A}}_{ID} \\ \tilde{\mathbf{A}}_{DI} & \tilde{\mathbf{A}}_{DD} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_I \\ \tilde{\mathbf{u}}_D \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_D \end{bmatrix}.$$

Recall that all cardinal basis functions vanish at domain boundary $\partial\Omega$. Therefore, following definition (5.1c), $\tilde{\mathbf{A}}_{ID} = \mathbf{A}_{ID}$ and $\tilde{\mathbf{A}}_{DD} = \mathbf{A}_{DD}$, where \mathbf{A}_{ID} and \mathbf{A}_{DD} are the corresponding submatrices of the standard stiffness matrix \mathbf{A} . Furthermore, because the values of $\tilde{\mathbf{u}}_D$ are prescribed, (6.1) may be reduced to the linear system $\tilde{\mathbf{A}}_{II}\tilde{\mathbf{u}}_I = \mathbf{f}_I - \mathbf{A}_{ID}\tilde{\mathbf{u}}_D$. This reduced system may be solved to determine all the unprescribed solution coefficients, $\tilde{\mathbf{u}}_I$.

7. Poisson's equation. In this section, we analyze a surrogate IGA discretization of Poisson's equation on the domain $\Omega = \varphi(\hat{\Omega})$. Here, as well as in the forthcoming problems, we restrict our attention to Dirichlet boundary conditions. This simplifies the analysis while also retaining all of its interesting features. Given a function $f \in L^2(\Omega)$, the corresponding weak form is the following:

$$(7.1) \quad \text{Find } u \in H_0^1(\Omega) \text{ satisfying } a(u, v) = F(v) \text{ for all } v \in H_0^1(\Omega),$$

where $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx$ and $F(v) = \int_{\Omega} f v \, dx$. At this point, it has been made well-understood that the bilinear form $a(\cdot, \cdot)$ can be rewritten on the parametric domain $\hat{\Omega}$, using the expression (3.5). Recalling this detail, we continue on with the simplifying assumption $K \in [W^{q+1, \infty}(\hat{\Omega})]^{n \times n}$.

7.1. Inconsistency. Recall (2.1) and (3.1) and take $V_h = \text{span}\{N_i\}$, where every $N_i = \hat{N}_i \circ \varphi^{-1}$. Analysis of surrogate methods best proceeds using the surrogate bilinear form $\tilde{a} : V_h \times V_h \rightarrow \mathbb{R}$ inherent to the surrogate matrix $\tilde{\mathbf{A}}$. Explicitly,

$$\tilde{a}(w_h, v_h) = \mathbf{v}^\top \tilde{\mathbf{A}} \mathbf{w},$$

for all $w_h = \sum_i w_i N_i, v_h = \sum_i v_i N_i \in V_h$. Here and throughout, we shall use definition (5.1c) in constructing the surrogate stiffness matrix $\tilde{\mathbf{A}}$ and its associated surrogate bilinear form $\tilde{a}(\cdot, \cdot)$.

In the proceeding analysis, Theorem 7.2 is of fundamental importance. Its proof is a simple consequence of Theorem 4.2 and Lemma 7.1. From now on, for any matrix \mathbf{N} , we use the notation $|\mathbf{N}|_{\max} = \max_{i \neq j} |\mathbf{N}_{ij}|$.

LEMMA 7.1. *For all $v_h, w_h \in V_h$, the following upper bound holds:*

$$(7.2) \quad |a(v_h, w_h) - \tilde{a}(v_h, w_h)| \leq C_3 h^{2-n} |\mathbf{A} - \tilde{\mathbf{A}}|_{\max} \|\nabla v_h\|_0 \|\nabla w_h\|_0,$$

where C_3 is a constant depending only on φ and p .

Proof. In this proof, we will use the symbols " \lesssim " and " \approx " to denote upper bounds and equivalence, respectively, up to constants depending at most on p and φ . For each $i = 1, \dots, N$, let $\mathcal{I}(i)$ be the set of indices j such that $\text{supp}(N_i) \cap \text{supp}(N_j) \neq \emptyset$ and notice that $|\mathcal{I}(i)| \leq |\mathcal{D}| = (2p+1)^n$. We begin with the observation that $\sum_i (\mathbf{A}_{ij} - \tilde{\mathbf{A}}_{ij}) = \sum_j (\mathbf{A}_{ij} - \tilde{\mathbf{A}}_{ij}) = 0$. With

these two identities in hand, we find that

$$\begin{aligned}
(7.3) \quad a(v_h, w_h) - \tilde{a}(v_h, w_h) &= -\frac{1}{2} \sum_{i,j}^{\tilde{\mathcal{I}}} (\mathbf{A}_{ij} - \tilde{\mathbf{A}}_{ij})(\mathbf{v}_i - \mathbf{v}_j)(\mathbf{w}_i - \mathbf{w}_j) \\
&\leq |\mathbf{A} - \tilde{\mathbf{A}}|_{\max} \sum_i \sum_{j \in \mathcal{I}(i)} |\mathbf{v}_i - \mathbf{v}_j| |\mathbf{w}_i - \mathbf{w}_j| \\
&\leq |\mathbf{A} - \tilde{\mathbf{A}}|_{\max} \sum_i \left(\sum_{j \in \mathcal{I}(i)} |\mathbf{v}_i - \mathbf{v}_j|^2 \right)^{1/2} \left(\sum_{j \in \mathcal{I}(i)} |\mathbf{w}_i - \mathbf{w}_j|^2 \right)^{1/2}.
\end{aligned}$$

For the time being, fix the index $i = 1, \dots, N$. For each coefficient vector $\mathbf{v} \in \mathbb{R}^N$, define $|\mathbf{v}|_{\mathcal{I}(i)} = \left(\sum_{j \in \mathcal{I}(i)} |\mathbf{v}_i - \mathbf{v}_j|^2 \right)^{1/2}$. One may easily check that $|\cdot|_{\mathcal{I}(i)}$ is a seminorm on \mathbb{R}^N . In order to identify its kernel, simply observe that $|\mathbf{v}|_{\mathcal{I}(i)} = 0$ iff $\mathbf{v}_j = \mathbf{v}_i$ for each coefficient $j \in \mathcal{I}(i)$. Since $i \in \mathcal{I}(i)$, an equivalent way of stating this condition is that there exists some constant $C \in \mathbb{R}$ such that $\mathbf{v}_j = C$ for each $j \in \mathcal{I}(i)$. Now, recall that $v_h(\mathbf{x}) = \sum_i \mathbf{v}_i N_i(\mathbf{x})$ for all $\mathbf{x} \in \Omega$ and consider the following alternative seminorm:

$$|\mathbf{v}|_{N_i} = \|\nabla v_h\|_{0, \text{supp}(N_i)}.$$

Notice that $|\mathbf{v}|_{N_i} = 0$ iff $v_h(\mathbf{x})$ is equal to a constant on $\text{supp}(N_i)$, say $C \in \mathbb{R}$. From the partition of unity property inherent to all NURBS bases, it must hold that $C = v_h(\mathbf{x}) = \sum_{j \in \mathcal{I}(i)} C N_j(\mathbf{x})$ for all $\mathbf{x} \in \text{supp}(N_i)$. In other words, since $\{N_j|_{\text{supp}(N_i)}\}_{j \in \mathcal{I}(i)}$ is a linearly independent set, $|\mathbf{v}|_{N_i} = 0$ iff $\mathbf{v}_j = C$ for each $j \in \mathcal{I}(i)$.

In the previous paragraph, we showed that the kernels of $|\mathbf{v}|_{N_i}$ and $|\mathbf{v}|_{\mathcal{I}(i)}$ are identical; namely, $|\mathbf{v}|_{N_i} = 0$ iff $|\mathbf{v}|_{\mathcal{I}(i)} = 0$ iff $\mathbf{v} \in Q_i$, where

$$Q_i = \{\mathbf{v} \in \mathbb{R}^N : \mathbf{v}_j = \mathbf{v}_i \text{ for each } j \in \mathcal{I}(i)\}.$$

Clearly, $|\cdot|_{N_i}$ and $|\cdot|_{\mathcal{I}(i)}$ induce norms on the quotient space \mathbb{R}^N/Q_i . The next important observation is that $|Q_i| = N + 1 - |\mathcal{I}(i)|$, which may be witnessed by inspection. Because $\dim(\mathbb{R}^N/Q_i) = |\mathcal{I}(i)| - 1 \leq (2p + 1)^n - 1$ is finite and depends only on p , it follows from the well-known equivalence of norms on finite dimensional vector spaces (e.g., the vector space \mathbb{R}^N/Q_i) that the seminorms $|\cdot|_{N_i}$ and $|\cdot|_{\mathcal{I}(i)}$ are equivalent. Of course, the corresponding equivalence constants will depend on h , φ , and p . Nevertheless, a standard scaling argument is all that is required to see that $|\mathbf{v}|_{\mathcal{I}(i)} \approx h^{2-n} |\mathbf{v}|_{N_i}$. Therefore, employing (7.3), we may simply write

$$|a(v_h, w_h) - \tilde{a}(v_h, w_h)| \lesssim h^{2-n} |\mathbf{A} - \tilde{\mathbf{A}}|_{\max} \sum_i \|\nabla v_h\|_{0, \text{supp}(N_i)} \|\nabla w_h\|_{0, \text{supp}(N_i)}.$$

The proof is completed by applying the discrete Cauchy–Schwarz inequality to the right-hand side of the inequality above and employing the fact that $\left(\sum_i \|f\|_{0, \text{supp}(N_i)}^2 \right)^{1/2} \approx \|f\|_0$, for all $f \in L^2(\Omega)$. \square

THEOREM 7.2. *Let $C_4 = C_2 \cdot C_3$. For all $v_h, w_h \in V_h$, the following upper bound holds:*

$$|a(v_h, w_h) - \tilde{a}(v_h, w_h)| \leq C_4 H^{q+1} \|\nabla v_h\|_0 \|\nabla w_h\|_0.$$

Proof. Obviously, we are done if $a(\cdot, \cdot) = \tilde{a}(\cdot, \cdot)$. Therefore, assume $a(\cdot, \cdot) \neq \tilde{a}(\cdot, \cdot)$ and let $i < j$ be the indices of the maximal value $|\mathbf{A}_{ij} - \tilde{\mathbf{A}}_{ij}| = |\mathbf{A} - \tilde{\mathbf{A}}|_{\max} > 0$. Since $\tilde{\mathbf{A}}$ is defined

by (5.1c), Theorem 4.2 leads us to the inequality

$$|\mathbf{A} - \tilde{\mathbf{A}}|_{\max} = |\Phi_{\delta}(\tilde{\mathbf{x}}_i) - \tilde{\Phi}_{\delta}(\tilde{\mathbf{x}}_i)| \leq C_2 h^{n-2} H^{q+1}.$$

The proof is completed using Lemma 7.1. \square

REMARK 7.1. In Lemma 7.1, it is important that $\tilde{\mathbf{A}}$ be defined using (5.1c). Indeed, in (7.3), it is the symmetry and the zero row sum property preserved in this definition which allows $|a(v_h, w_h) - \tilde{a}(v_h, w_h)|$ to be bounded by products of differences in the coefficients \mathbf{v}_i and \mathbf{w}_j . If we had used definition (5.1a) or (5.1b), one would have to directly work with the upper bound

$$|a(v_h, w_h) - \tilde{a}(v_h, w_h)| \leq \|\mathbf{A} - \tilde{\mathbf{A}}\|_{\max} \sum_i \sum_{j \in \mathcal{I}(i)} |\mathbf{v}_i| |\mathbf{w}_j|.$$

This, in turn, can only be finessed to arrive at an inequality like

$$(7.4) \quad |a(v_h, w_h) - \tilde{a}(v_h, w_h)| \leq C'_3 h^{-n} \|\mathbf{A} - \tilde{\mathbf{A}}\|_{\max} \|v_h\|_0 \|w_h\|_0,$$

for some other constant C'_3 , depending only on φ and p . Notice the loss of an h^2 scaling factor when comparing (7.2) and (7.4). This difference could greatly affect solution accuracy.

7.2. A priori error estimation. Define $V_{h,0} = V_h \cap H_0^1(\Omega)$. The following lemma is identical in spirit to [21, Theorem 7.1]. By the Lax–Milgram theorem, this lemma allows us to conclude that there exists a unique surrogate solution corresponding to (7.1), namely $\tilde{u}_h \in V_{h,0}$, for every sufficiently small $H > 0$.

LEMMA 7.3. Let $\alpha = (1 + C_P)^{-1}$, where C_P is the Poincaré constant for the domain Ω . If $H^{q+1} < \alpha \cdot C_4^{-1}$, then the surrogate bilinear form $\tilde{a} : V_h \times V_h \rightarrow \mathbb{R}$ is coercive on $V_{h,0}$. Letting $\tilde{\alpha} > 0$ be the associated coercivity constant, it also holds that $\tilde{\alpha} \rightarrow \alpha$, as $H \rightarrow 0$.

Proof. Let $S = \{v \in H_0^1(\Omega) : \|v\|_1 = 1\}$ be the surface of the unit ball in $H_0^1(\Omega)$. Notice that $\alpha \leq a(v_h, v_h) \leq \tilde{a}(v_h, v_h) + |a(v_h, v_h) - \tilde{a}(v_h, v_h)|$ for all $v_h \in V_h \cap S$ and, therefore,

$$\alpha - |a(v_h, v_h) - \tilde{a}(v_h, v_h)| \leq \tilde{a}(v_h, v_h) \quad \text{for all } v_h \in V_h \cap S.$$

Invoking Lemma 7.1, the second term on the left may be bounded from above as follows:

$$|a(v_h, v_h) - \tilde{a}(v_h, v_h)| \leq C_4 H^{q+1}.$$

Clearly, if $C_4 H^{q+1} < \alpha$, then $0 < \alpha - |a(v_h, v_h) - \tilde{a}(v_h, v_h)| \leq \tilde{a}(v_h, v_h)$, as necessary. \square

THEOREM 7.4. Let $\theta > 1$. If $u \in H^{p+1}(\Omega)$, then there exists a constant C_5 , depending only on p and φ , such that

$$(7.5a) \quad \|u - \tilde{u}_h\|_1 \leq C_5 h^p \|u\|_{p+1} + \theta \cdot \alpha^{-1} C_4 H^{q+1} \|\nabla u\|_0,$$

for every sufficiently small $H > 0$. Additionally, if $\Omega \subseteq \mathbb{R}^n$ is convex, then there exists a constant C_6 , depending only on p and φ , such that

$$(7.5b) \quad \|u - \tilde{u}_h\|_0 \leq C_6 h^{p+1} \|u\|_{p+1} + \theta \cdot C_P C_4 H^{q+1} \|\nabla u\|_0,$$

for every sufficiently small $H > 0$.

Proof. We first prove (7.5a). Let $u_h \in V_{h,0}$ be the solution of the standard IGA discretization (2.1b) associated to (7.1). Clearly, $\|u - \tilde{u}_h\|_1 \leq \|u - u_h\|_1 + \|u_h - \tilde{u}_h\|_1$. Moreover, by [13, Theorem 6.1], $\|u - u_h\|_1 \leq C_5 h^p \|u\|_{p+1}$. Recalling Lemma 7.3, we find that

$$\tilde{\alpha} \|u_h - \tilde{u}_h\|_1^2 \leq \tilde{a}(u_h - \tilde{u}_h, u_h - \tilde{u}_h) = \tilde{a}(u_h - \tilde{u}_h, u_h) - a(u_h - \tilde{u}_h, u_h).$$

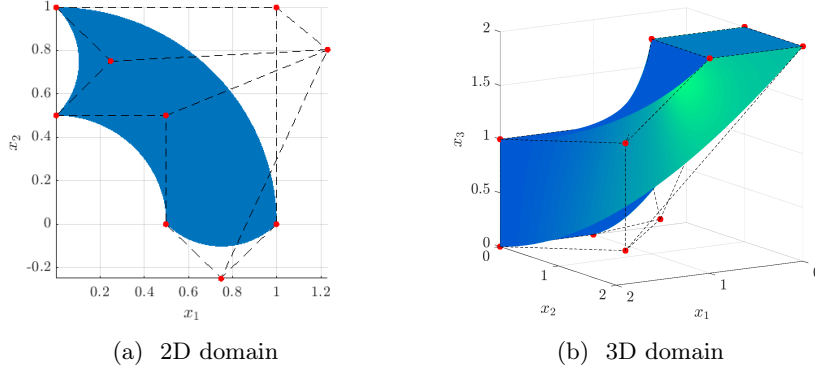


Figure 7.1: Two domains Ω for problem (7.1).

After invoking [Theorem 7.2](#), it now readily follows that $\|u_h - \tilde{u}_h\|_1 \leq \tilde{\alpha}^{-1} C_4 H^{q+1} \|\nabla u\|_0$. Since $\tilde{\alpha} \rightarrow \alpha$, in the limit $H \rightarrow 0$, it also holds that $\|u_h - \tilde{u}_h\|_1 \leq \theta \cdot \alpha^{-1} C_4 H^{q+1} \|\nabla u\|_0$, for all sufficiently small $H > 0$.

Our proof of [\(7.5b\)](#), also involves the triangle inequality: $\|u - \tilde{u}_h\|_0 \leq \|u - u_h\|_0 + \|u_h - \tilde{u}_h\|_0$. If Ω is convex, then $\|u - u_h\|_0 \leq C_6 h^{p+1} \|u\|_{p+1}$. Next, find $w_h \in V_{h,0}$ satisfying $a(w_h, v_h) = (u_h - \tilde{u}_h, v_h)_\Omega$, for all $v_h \in V_{h,0}$. It holds that $\|\nabla w_h\|_0 \leq C_P \|u_h - \tilde{u}_h\|_0$. Now, observe that $\|u_h - \tilde{u}_h\|_0^2 = a(w_h, u_h - \tilde{u}_h) = \tilde{a}(w_h, \tilde{u}_h) - a(w_h, \tilde{u}_h)$. Finally, invoke [Theorem 7.2](#) to arrive at

$$\|u_h - \tilde{u}_h\|_0^2 \leq C_4 H^{q+1} \|\nabla \tilde{u}_h\|_0 \|\nabla w_h\|_0 \leq C_P C_4 H^{q+1} \|\nabla \tilde{u}_h\|_0 \|u_h - \tilde{u}_h\|_0,$$

which works to deliver the stated estimate, since $\|\nabla \tilde{u}_h\|_0 \rightarrow \|\nabla u_h\|_0 \leq \|\nabla u\|_0$, as $H \rightarrow 0$. \square

7.3. Numerical experiments. The two bounds [\(7.5a\)](#) and [\(7.5b\)](#) need to be experimentally verified. Because both estimates depend on the two scales h and H , in order to be overtly thorough, we should provide verification in both scales, independently. That is, first holding the h -scale fixed and varying H and, alternatively, holding the H -scale fixed and varying h . We forgo this verification step and instead refer the reader to similar studies with low-order finite elements in [\[4–6, 21\]](#). In this section, therefore, we verify the bounds above under the assumption $H = H(h)$. We expect that this would be the typical use case.

7.3.1. Overview. In our experiments with Poisson’s equation [\(7.1\)](#), we considered a 2D and a 3D domain Ω ; see [Figure 7.1](#). The 2D domain is given by a second-order NURBS parameterization and the 3D domain is parameterized by second-order B-splines. This particular 2D domain ([Figure 7.1 \(a\)](#)) was chosen instead of the 2D quarter annulus featured later ([Figure 8.1](#)) in order to break some symmetries which would otherwise appear in its stencil functions ([Figure 4.1](#)). (Additional stencil function symmetries can appear if $\varphi: \hat{\Omega} \rightarrow \Omega$ is a conformal mapping.) We also note that, after fixing the NURBS parameterizations for the edges on the boundary of the 2D domain, the parameterization $\varphi: \hat{\Omega} \rightarrow \Omega$ was generated as a Coons patch [\[37\]](#). Because each of the four edges had polynomial weight functions in their parameterizations, the resulting Coons patch NURBS parameterization, $\varphi: \hat{\Omega} \rightarrow \Omega$, contained a global polynomial weight function $W(\hat{\mathbf{x}}) = \sum_j w_j \hat{B}_j(\hat{\mathbf{x}})$. Consequently, the assumptions of [Theorem 4.2](#) were satisfied in all of our experiments.

In constructing $\tilde{\mathbf{A}}$ (cf. [Section 6](#)) and, thus, the surrogate stencil functions $\tilde{\Phi}_\delta = \Pi_H \Phi_\delta$, we tried local B-spline interpolants $\Pi_H: C^0(\hat{\Omega}) \rightarrow S_q(\hat{\Xi})$ of various order $1 \leq q \leq 5$. As

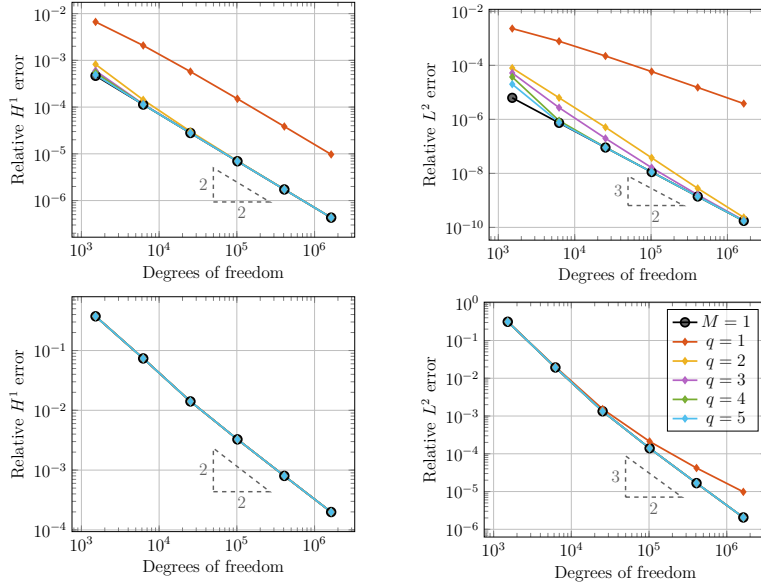


Figure 7.2: 2D domain. Relative errors for $p = 2$, $M = 5$, and two different manufactured solutions. Top row: $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2)$. Bottom row: $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$.

stated previously (see Remark 4.1), we used the convention $\tilde{\mathbf{E}} \subseteq \tilde{\Omega} \cap \tilde{\mathbf{X}}$. In our 2D experiments, we explored both $p = 2$ or $p = 3$ NURBS bases and the full range of possible q . In our 3D experiments, we only explored the case $p = 2$ using $q = 1, 3$. This limitation of our 3D experiments was due to the restraints of the MATLAB B-spline interpolation routine we were using.

In the first set of experiments (Figures 7.2–7.4), the sampling length H was set to $H = Mh$, with the constant $M = 5$. In the second set of experiments (Figures 7.5 and 7.6), we used a mesh-dependent sampling length, defined below.

7.3.2. Constant sampling lengths. For the constant sampling strategy, with $M = 5$, convergence plots of the errors in surrogate solutions are presented in Figures 7.2–7.4. After inspecting these figures, the reader should observe that there is a noticeable difference in the accuracy of the surrogate solutions \tilde{u}_h , dependent on the load f . For instance, with the “low-frequency” manufactured solution, $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2)$, the errors in the surrogate solutions appear to converge to the error generated by the (standard) non-surrogate IGA solution u_h asymptotically, at various rates. However, for the “high-frequency” manufactured solution, $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$, each of the surrogate errors and the corresponding standard IGA error are nearly indistinguishable (except, sometimes, in the case $q = 1$).

This difference can be explained in a simple way. Observe that the h -dependent terms in (7.5a) and (7.5b) are multiplied by the high-order norm $\|u\|_{p+1}$ and, meanwhile, the H -dependent terms are only multiplied by the lower-order seminorm $\|\nabla u\|_0$. Let us call the first term in each bound the *discretization error* and the second term the *consistency error*. Because of the presence of the high-order norm $\|u\|_{p+1}$, the discretization error is much more sensitive to irregularities or oscillations in the solution. Another important detail not to overlook is that C_4 ultimately involves high-order norms of the tensor coefficient $K(\hat{\mathbf{x}})$. Meanwhile, C_5 and C_6 depend (through Ω), at most, on the H^2 -norm of $K(\hat{\mathbf{x}})$.

These observations can lead in many interesting directions, but the conclusion which the

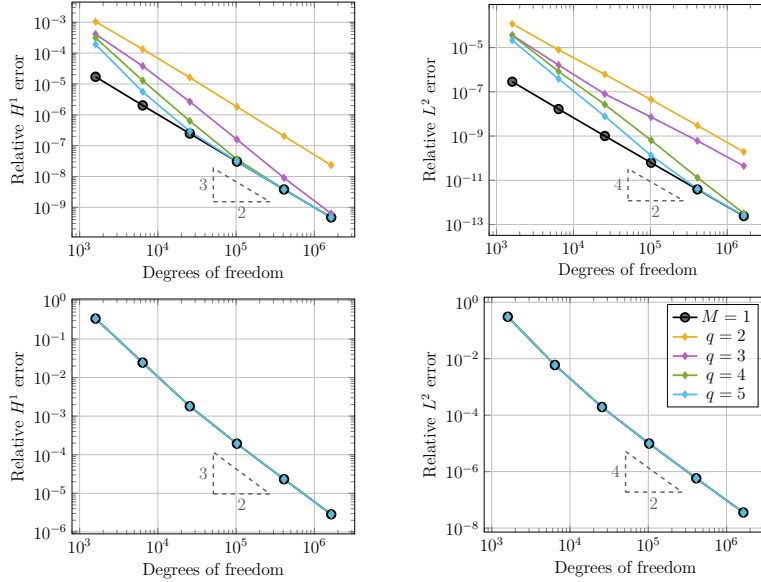


Figure 7.3: 2D domain. Relative errors for $p = 3$, $M = 5$, and two different manufactured solutions. Top row: $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2)$. Bottom row: $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$.

reader should ultimately arrive at is that the assembly of the stiffness matrix should only be carried out to the accuracy required by the problem at hand. Therefore, the correct surrogate assembly strategy must take into account properties of the problem geometry and the given loads, as well as each of the parameters h , p , and q . A simple way of doing this is to use mesh-dependent sampling lengths.

REMARK 7.2. In [21], it was documented that the L^2 error converged like $h^{p+1} + H^{q+2}$ when Π_H resembles an L^2 projection. However, when using an interpolation instead, like it is done in this work, the error converges like $h^{p+1} + H^{q+2}$ whenever the interpolation order q is even and like $h^{p+1} + H^{q+1}$ otherwise. This parity effect can also be observed in results of this work; see, e.g., the top right plot in Figure 7.3. We do not prove this improved convergence rate but refer to proofs of quadrature formulas where the same parity effect can be observed; see, e.g., [17, Chapter 2.5].

7.3.3. Mesh-dependent sampling lengths. In a successful surrogate method, the discretization error should always dominate the consistency error. Of course, however, this domination should not be exercised to such an extent that overall efficiency is compromised. As a rule of thumb, keeping the consistency error at or below 5% of the discretization error is often acceptable. Even within this somewhat conservative threshold, balancing the two errors appropriately can lead to very significant performance advantages.

Let $q > p$ and consider the mesh-dependent sampling length $M(h) = \max\left\{1, \lfloor c \cdot h^{\frac{p-q+\beta}{q+1}} \rfloor\right\}$, where both $c, \beta \geq 0$ are tuneable parameters. Returning the definition $H = M \cdot h$, we now find that, for any $c > 0$,

$$H^{q+1} \leq (M \cdot h)^{q+1} = c^{q+1} h^{p+1+\beta}, \quad \text{as } h \rightarrow 0.$$

Here, for any $\beta > 0$, both the H^1 and L^2 discretization error will eventually dominate their associated consistency error. Therefore, this parameter exists to maintain the dominance of the

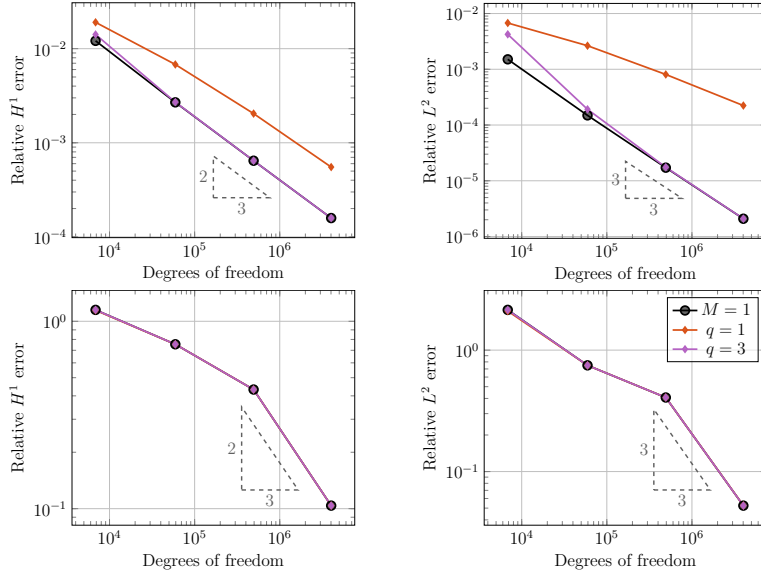


Figure 7.4: 3D domain. Relative errors for $p = 2$, $M = 5$, and two different manufactured solutions. Top row: $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2) \sin(\pi x_3)$. Bottom row: $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2) \sin(20\pi x_3)$.

discretization error, throughout mesh refinements. We found $\beta = 1/2$ to be a suitable choice for our purposes. The parameter c , on the other hand, must be calibrated to the problem at hand.

All run-time measurements in this sections were obtained on a machine equipped with two Intel[®] Xeon[®] Gold 6136 processors with a nominal base frequency of 3.0 GHz. Each processor has 12 physical cores which results in a total of 24 physical cores, but only single core was used to run the following experiments. The total available memory of 251 GB is split into two NUMA domains, one for each socket. For compatibility reasons with using SciPy, we employed the BLAS library in version 3.7.1 provided by the operating system Ubuntu 18.04.2, but using other optimized libraries might improve the performance of the MATLAB solver.

In Figure 7.5 we show the assembly times our implementation accrued for various values of c , when $p = 2$ and $q = 5$. Inspection of Figure 7.6 clearly shows that $c = 0.75$ is suitable for $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2)$ and $c = 3$ is suitable for $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$. Let t_{std} be the time required to assemble the matrix with the standard approach and t_{surr} the time using the surrogate approach. The speed-up from using the surrogate approach is then defined as $\frac{t_{\text{std}}}{t_{\text{surr}}} - 1$. When compared to the non-surrogate assembly strategy at just over one million degrees of freedom, note that the first choice gives an assembly speed-up of around 1500% and the second choice gives a speed-up of more than 5000%. These enormous speed-ups are in fact not that surprising after inspecting the percentage of matrix entries which must be computed using traditional quadrature formulas; see, again, Figure 7.5.

8. Transverse vibrations of an isotropic membrane. One of the great advantages of the IGA paradigm is its superior accuracy in structural vibration problems. Any worthwhile surrogate method should maintain this advantage. Therefore, in this section, we extend the analysis of the previous section to the analysis of transverse vibrations of a two-dimensional

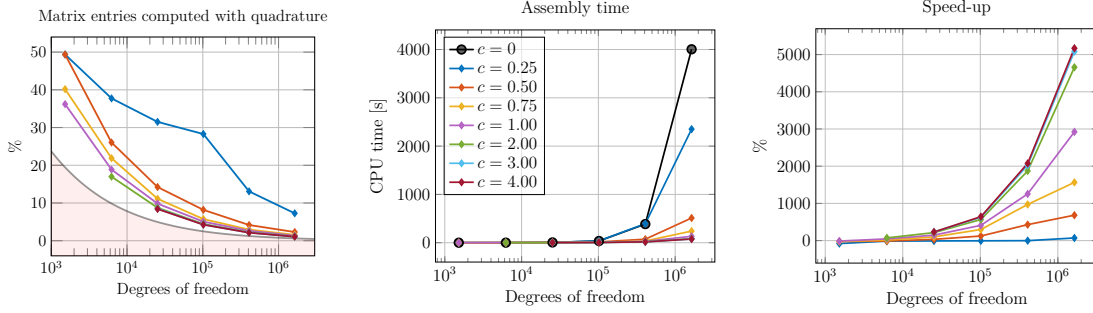


Figure 7.5: 2D domain with $p = 2$ and $q = 5$. Assembly times with the mesh-dependent sampling strategy $M(h) = \max \left\{ 1, \left[c \cdot h^{\frac{p-q+\frac{1}{2}}{q+1}} \right] \right\}$. On the left, note the percentage of matrix entries computed with quadrature. Here, the percentage of entries involving non-cardinal basis functions (roughly $1 - \left(\frac{m-2p}{m}\right)^n$) is shaded out to indicate the theoretical lower bound.

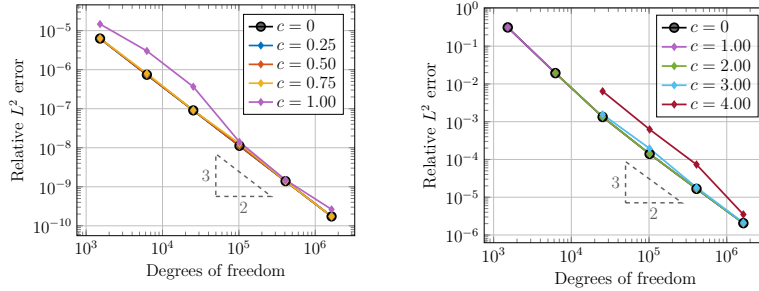


Figure 7.6: 2D domain with $p = 2$ and $q = 5$. Relative L^2 errors for $M(h) = \max \left\{ 1, \left[c \cdot h^{\frac{p-q+\frac{1}{2}}{q+1}} \right] \right\}$, with two different manufactured solutions. Left: $u(\mathbf{x}) = \sin(\pi x_1) \sin(\pi x_2)$. Right: $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$.

elastic membrane. The corresponding weak form is the following:

$$(8.1) \quad \begin{cases} \text{Find all pairs } u \in H_0^1(\Omega) \setminus \{0\} \text{ and } \lambda \in \mathbb{R} \text{ satisfying} \\ a(u, v) = \lambda m(u, v) \quad \text{for all } v \in H_0^1(\Omega), \end{cases}$$

where $a(w, v) = \int_{\Omega} \nabla w \cdot \nabla v dx$ and $m(w, v) = \int_{\Omega} wv dx$. It is well-known that there are countably many solutions to this problem, $\{(u^{(k)}, \lambda^{(k)})\}_{j=1}^{\infty}$, where each $\lambda^{(k)} > 0$. From here on, we make the ordering assumption $\lambda^{(j)} \leq \lambda^{(k)}$, for every $j < k$.

8.1. Surrogate mass matrices. So far, we have only rigorously analyzed surrogates of the elliptic bilinear form $a(\cdot, \cdot)$, written above. Using the techniques developed thus far, results similar to [Theorem 7.2](#) can be proven for other bilinear forms, such as $m(\cdot, \cdot)$. Let the corresponding mass matrix and surrogate mass matrix be denoted \mathbf{M} and $\tilde{\mathbf{M}}$, respectively. Employing definition [\(5.1b\)](#), we take for granted that the accompanying surrogate bilinear form $\tilde{m}(\cdot, \cdot)$ satisfies

$$(8.2) \quad |m(v_h, w_h) - \tilde{m}(v_h, w_h)| \leq C_7 H^{q+1} \|v_h\|_0 \|w_h\|_0,$$

for some C_7 depending only on φ , p , q , and $\|\Pi_H\|$.

In the special case that the geometry mapping $\varphi(\hat{\mathbf{x}})$ is a polynomial, we have a surrogate reproduction property of the mass form $m(u, v)$. This property is formalized in the following corollary to [Proposition 5.1](#).

COROLLARY 8.1. *Assume that the domain mapping $\varphi: \hat{\Omega} \rightarrow \Omega$ is defined through a polynomial of order p , i.e., $\varphi \in [\mathcal{Q}_p(\hat{\Omega})]^n$. Let \mathbf{M} be the coefficient matrix arising from the discretization of $m(\cdot, \cdot)$ and $\tilde{\mathbf{M}}$ the corresponding surrogate matrix. If $q \geq n \cdot p - 1$, it holds that $\mathbf{M} = \tilde{\mathbf{M}}$.*

Proof. Let $\mathbf{J}(\hat{\mathbf{x}})$ be the Jacobian (tensor) of $\varphi(\hat{\mathbf{x}})$. The transformation of the integral from the physical to the reference domain is given by

$$m(u, v) = \int_{\Omega} uv \, dx = \int_{\hat{\Omega}} \hat{u}\hat{v} \det(\mathbf{J}) \, d\hat{\mathbf{x}}.$$

It remains to show that $G(\hat{\mathbf{x}}, \hat{u}(\hat{\mathbf{y}}), \hat{v}(\hat{\mathbf{y}})) = \hat{u}(\hat{\mathbf{y}})\hat{v}(\hat{\mathbf{y}}) \det(\mathbf{J}(\hat{\mathbf{x}}))$ is a polynomial of degree $n \cdot p - 1$ in the $\hat{\mathbf{x}}$ -variable. Since $\varphi \in [\mathcal{Q}_p(\hat{\Omega})]^n$, it holds that $\det(\mathbf{J}) \in \mathcal{Q}_{n \cdot p - 1}(\hat{\Omega})$. Applying [Proposition 5.1](#) yields the desired reproduction property. \square

8.2. A priori analysis of the eigenvalue error. Adopt the obvious notation, $(u_h^{(k)}, \lambda_h^{(k)})$ and $(\tilde{u}_h^{(k)}, \tilde{\lambda}_h^{(k)})$, for the standard IGA and surrogate IGA solutions corresponding to [\(8.1\)](#), respectively. Due to space limitations, we only analyze the convergence of the eigenvalues $\tilde{\lambda}_h^{(k)} \rightarrow \lambda^{(k)}$.

THEOREM 8.2. *Let $\theta > 1$. If $u^{(k)} \in H^{p+1}(\Omega)$, then there exists a constant C_8 , depending only on p and φ , such that*

$$(8.3) \quad \frac{|\lambda^{(k)} - \tilde{\lambda}_h^{(k)}|}{\lambda^{(k)}} \leq C_8 h^{2p} + \theta \cdot (C_4 + C_7) H^{q+1},$$

for every sufficiently small $H > 0$.

Proof. Clearly, $|\lambda^{(k)} - \tilde{\lambda}_h^{(k)}| \leq |\lambda^{(k)} - \lambda_h^{(k)}| + |\lambda_h^{(k)} - \tilde{\lambda}_h^{(k)}|$. It is known that if $u^{(k)} \in H^{p+1}(\Omega)$, then $|\lambda^{(k)} - \lambda_h^{(k)}| \leq C_8 \lambda^{(k)} h^{2p}$. We now focus on the term $|\lambda_h^{(k)} - \tilde{\lambda}_h^{(k)}|$.

Let $V_h^{(k)}$ be the set of all k -dimensional subsets of V_h and fix $\theta > 1$. Two important members of this set are $E_h^{(k)} = \text{span}\{u_h^{(l)}\}_{l=1}^k$ and $\tilde{E}_h^{(k)} = \text{span}\{\tilde{u}_h^{(l)}\}_{l=1}^k$. Observe that

$$(8.4) \quad \begin{aligned} \tilde{\lambda}_h^{(k)} &= \min_{E_h \in V_h^{(k)}} \max_{v \in E_h} \frac{\tilde{a}(v, v)}{\tilde{m}(v, v)} \leq \max_{v \in E_h^{(k)}} \frac{\tilde{a}(v, v)}{\tilde{m}(v, v)} = \max_{v \in E_h^{(k)}} \frac{a(v, v)}{m(v, v)} \frac{\tilde{a}(v, v)}{a(v, v)} \frac{m(v, v)}{\tilde{m}(v, v)} \\ &\leq \lambda_h^{(k)} \max_{v \in E_h^{(k)}} \frac{\tilde{a}(v, v)}{a(v, v)} \max_{w \in E_h^{(k)}} \frac{m(w, w)}{\tilde{m}(w, w)}. \end{aligned}$$

Note that $\frac{\tilde{a}(v, v)}{a(v, v)} = 1 + \frac{\tilde{a}(v, v) - a(v, v)}{a(v, v)} \leq 1 + C_4 H^{q+1}$, by [Theorem 7.2](#). Similarly, by [\(8.2\)](#), $\frac{m(w, w)}{\tilde{m}(w, w)} = 1 + \frac{\tilde{m}(w, w) - m(w, w)}{\tilde{m}(w, w)} \leq 1 + \theta \cdot C_7 H^{q+1}$, in the limit $H \rightarrow 0$. These observations, together with [\(8.4\)](#), imply

$$(8.5a) \quad \tilde{\lambda}_h^{(k)} \leq \lambda_h^{(k)} + \lambda_h^{(k)} (C_4 + \theta \cdot C_7) H^{q+1},$$

for every sufficiently small $H > 0$. Following a similar argument,

$$(8.5b) \quad \lambda_h^{(k)} \leq \tilde{\lambda}_h^{(k)} \frac{\tilde{a}(v, v)}{a(v, v)} \max_{w \in E_h^{(k)}} \frac{\tilde{m}(w, w)}{m(w, w)} \leq \tilde{\lambda}_h^{(k)} + \tilde{\lambda}_h^{(k)} (\theta \cdot C_4 + C_7) H^{q+1},$$

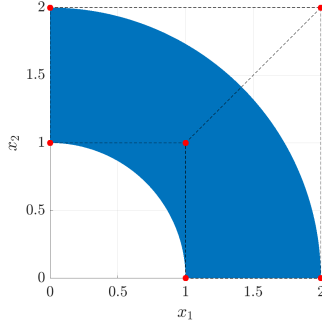


Figure 8.1: The physical domain Ω for problems (8.1) and (9.1).

for every sufficiently small $H > 0$. Note that (8.5a) implies that $\tilde{\lambda}_h^{(k)} \leq \theta \lambda_h^{(k)}$, as $H \rightarrow 0$. After introducing this inequality into (8.5b), we arrive at the upper bound $|\lambda_h^{(k)} - \tilde{\lambda}_h^{(k)}| \leq \lambda_h^{(k)} \theta \cdot (C_4 + C_7)H^{q+1}$, in the limit $H \rightarrow 0$. Because $\lambda_h^{(k)} \rightarrow \lambda^{(k)}$, we also have $\lambda_h^{(k)} \leq \theta \lambda^{(k)}$, as $H \rightarrow 0$. Only elementary algebra remains in order to arrive at (8.3). \square

REMARK 8.1. *One upshot of Theorem 8.2 is that, if $H = \mathcal{O}(h)$, then one may wish to choose $q + 1 > 2p$, in order to recover an optimal spectral convergence rate. Of course, for irregular geometries, it is difficult to maintain the assumption $\lambda^{(j)} \in H^{p+1}(\Omega)$, and a lower q may still provide a useful approximation.*

8.3. Numerical experiments. Our numerical experiments for problem (8.1) involve only the two-dimensional quarter annulus domain Ω , depicted in Figure 8.1. This domain was chosen instead of Figure 7.1 (a) so that $u^{(k)} \in H^3(\Omega)$, for each k . Thus, when $p = 2$, Theorem 8.2 concludes that $|\lambda_h^{(k)} - \tilde{\lambda}_h^{(k)}| \leq \mathcal{O}(h^4 + H^{q+1})$. Figure 8.2, which shows the convergence of the first nine eigenvalues, for $p = 2$, verifies this result. Recall Remark 7.2. One again witnesses the parity present in the L^2 error of the Poisson problems above. That is, we actually observe the stronger conclusion $|\lambda_h^{(k)} - \tilde{\lambda}_h^{(k)}| = \mathcal{O}(h^4 + H^{q+2})$ when q is even.

A close inspection of Figure 8.2 appears to indicate that the accuracy of the surrogate solution improves as the eigenvalues grow. This observation is in line with the previous numerical results for Poisson’s equation, which showed nearly indistinguishable solutions for the “high-frequency” manufactured solution $u(\mathbf{x}) = \sin(20\pi x_1) \sin(20\pi x_2)$ (cf. Figures 7.2–7.4). Naturally, we should compare the accuracy of all eigenvalues computed with the standard IGA method to those coming from the surrogate IGA method. This is done, in part, in Figure 8.3 for both $p = 2, 3$. Here, it is more meaningful to use the natural frequencies $(\omega^{(j)})^2 = \lambda^{(j)}$. Notice that the differences are extremely small across the entire range of computed frequencies.

9. Plate bending under a transverse load. Another clear advantage of IGA is the simplicity of discretizing high-order PDEs. In this section, we briefly demonstrate that the same features hold true for surrogate IGA methods. As a proof-of-concept, consider the simple Poisson–Kirchoff isotropic plate bending model. Given a function $f \in L^2(\Omega)$, the corresponding weak form is the following:

$$(9.1) \quad \text{Find } u \in H^2(\Omega) \cap H_0^1(\Omega) \text{ satisfying } a(u, v) = F(v) \quad \text{for all } v \in H^2(\Omega) \cap H_0^1(\Omega),$$

where $a(u, v) = \int_{\Omega} \Delta u \Delta v \, dx$ and $F(v) = \int_{\Omega} f v \, dx$.

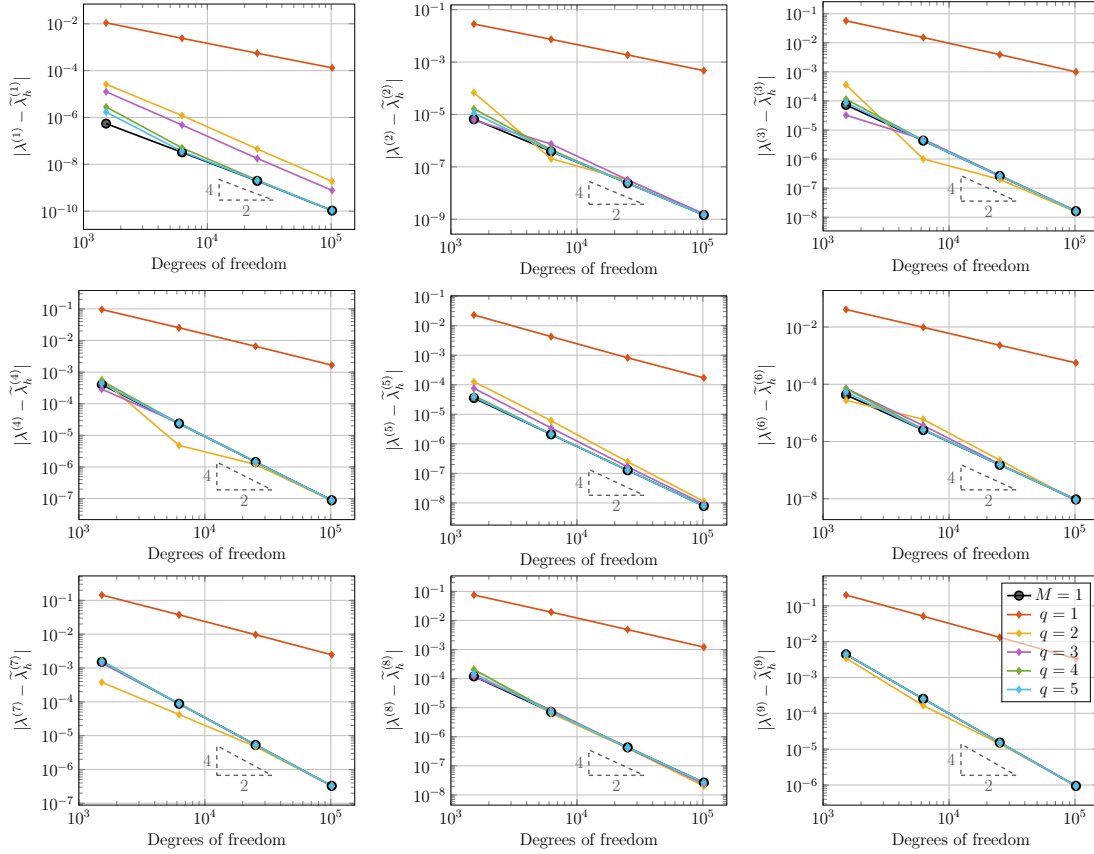


Figure 8.2: Eigenvalue convergence plots. Quarter annulus geometry with $p = 2$ and the constant sampling length $M = 5$. Note that the “exact” solutions, $\lambda^{(k)}$, were taken from precomputed values from a high-order discretization on a much finer mesh.

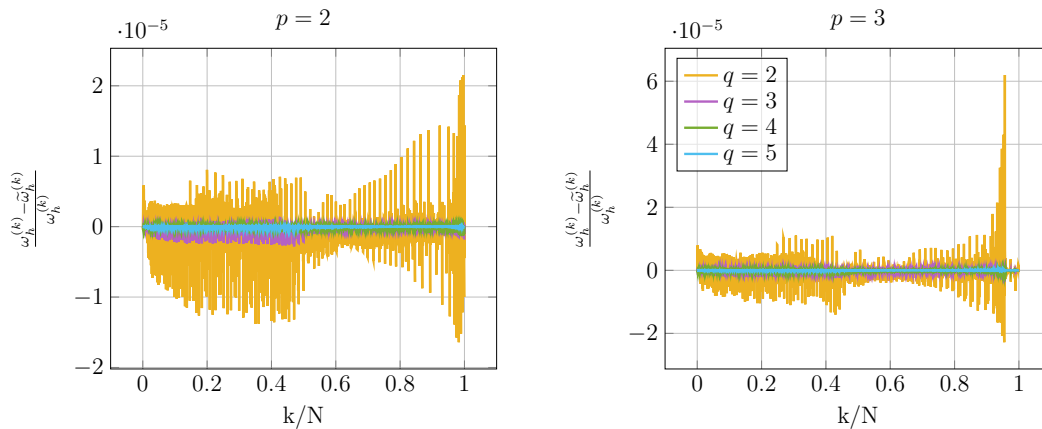


Figure 8.3: Relative differences in the computed natural frequencies from the IGA and the surrogate IGA method for 50×50 control points and $M = 5$.

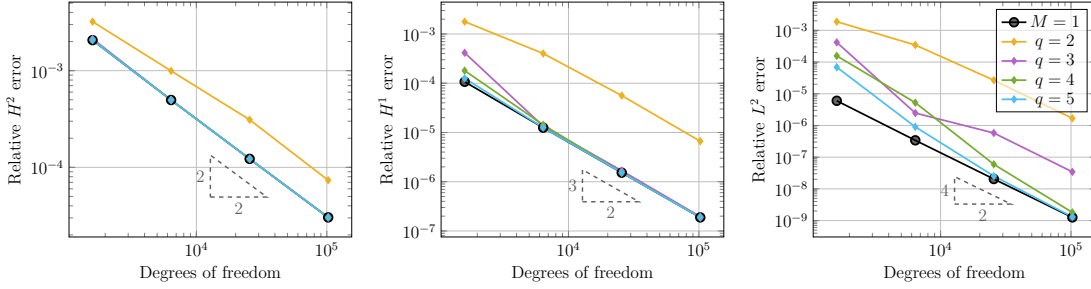


Figure 9.1: Plate bending problem (9.1). Relative errors for $p = 3$, $M = 5$, and the manufactured solution $u(\mathbf{x}) = \sin(\pi x_1) \sinh(\pi x_2)$

9.1. A higher-dimensional kernel. With the same principles as used for Poisson’s equation, one may easily design a surrogate IGA method for (9.1). In our approach, the corresponding surrogate stiffness matrix $\tilde{\mathbf{A}}$ was also defined using (5.1c). Notice that this definition does not preserve the entire kernel found in the true IGA stiffness matrix \mathbf{A} . For instance, one may easily verify that all linear functions lie in the kernel of $a(\cdot, \cdot)$. Therefore, $\mathbf{A}\mathbf{c}^{(1)} = \mathbf{A}\mathbf{c}^{(2)} = \mathbf{0}$, where $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}$ are the x_1 - and x_2 -coefficients of the control points, respectively. In our experiments, this property was only recovered in the limits $h \rightarrow 0$ or $H \rightarrow h$.

9.2. Numerical experiments. Let Ω be the quarter annulus domain depicted in Figure 8.1. Figure 9.1 shows the convergence of the errors, in the H^2 , H^1 , and L^2 norms, corresponding to this geometry Ω and the manufactured solution $u(\mathbf{x}) = \sin(\pi x_1) \sinh(\pi x_2)$. Even though the kernel is not preserved, the numerical results we witnessed are similar to those documented for the $p = 3$ experiments in the Poisson setting (see top row of Figure 7.3). For instance, notice that the surrogate error with $q = 2$ is parallel to the reference IGA error ($M = 1$), in both the $H^1(\Omega)$ and $L^2(\Omega)$ norms.

REMARK 9.1. *Although we will not provide any rigorous analysis, if we recall Remark 7.1, the similarity between our Poisson results and those above may still appear somewhat surprising. Indeed, since only the zero row sum property is inherited in the surrogate $\tilde{\mathbf{A}}$ when using (5.1c), we can not improve on the upper bound in Lemma 7.1. Had the entire kernel of \mathbf{A} been preserved in $\tilde{\mathbf{A}}$, we conjecture that an optimal form of this bound would involve an h^{4-n} scaling factor. Such a factor should permit a surrogate solution \tilde{u}_h of two h -orders higher accuracy.*

10. Stokes’ flow. In this section, we consider a surrogate IGA discretization of Stokes’ flow in a domain $\Omega \subseteq \mathbb{R}^n$. Given a viscosity $\mu \in \mathbb{R}_{>0}$, a function $\mathbf{f} \in [L^2(\Omega)]^n$, and a velocity field on the boundary $\mathbf{g} \in [H^{1/2}(\partial\Omega)]^n$, $\int_{\partial\Omega} \mathbf{g} \cdot \mathbf{n} \, ds = 0$, the corresponding weak form is the following:

$$(10.1) \quad \begin{cases} \text{Find } \mathbf{u} \in [H^1(\Omega)]^n \text{ and } p \in L^2(\Omega)/\mathbb{R} \text{ satisfying } \text{tr } \mathbf{u} = \mathbf{g} \text{ and} \\ a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + b(q, \mathbf{u}) = F(\mathbf{v}) \quad \text{for all } \mathbf{v} \in [H_0^1(\Omega)]^n \text{ and } q \in L^2(\Omega), \end{cases}$$

where $a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mu \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, dx$, $b(p, \mathbf{v}) = \int_{\Omega} p \nabla \cdot \mathbf{v} \, dx$, and $F(\mathbf{v}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx$. In this scenario, the pressure is not unique up to a constant, therefore we enforce the pressure to have zero mean value, i.e., $\int_{\Omega} p \, dx = 0$.

10.1. Surrogate divergence matrices. Since no symmetry can be exploited, the surrogate divergence matrices \mathbf{B} are constructed by employing definition (5.1a). Similarly, as in the mass term arising in Section 8, we have a surrogate reproduction property for the divergence

form $b(q, \mathbf{u})$, when the geometry map is described by a polynomial. This property is formalized in the following corollary of [Proposition 5.1](#).

COROLLARY 10.1. *Assume that the domain mapping $\varphi: \widehat{\Omega} \rightarrow \Omega$ is defined through a polynomial of order p , i.e., $\varphi \in [\mathcal{Q}_p(\widehat{\Omega})]^n$. Let \mathbf{B} be the coefficient matrix arising from the discretization of $b(\cdot, \cdot)$ and $\widetilde{\mathbf{B}}$ the corresponding surrogate matrix. If $q \geq (n-1)p$, it holds that $\mathbf{B} = \widetilde{\mathbf{B}}$.*

Proof. In the following, we only consider the cases $n = 2$ and $n = 3$. Let $\mathbf{J}(\widehat{\mathbf{x}})$ be the Jacobian of $\varphi(\widehat{\mathbf{x}})$. Assuming a gradient preserving transformation to the reference domain, the divergence $\nabla \cdot \mathbf{u}$ in the physical domain is transformed to $\text{tr}(\mathbf{J}^{-1} \widehat{\nabla} \widehat{\mathbf{u}})$, where tr is the trace. Using the property $\mathbf{J}^{-1} = \det(\mathbf{J})^{-1} \text{adj}(\mathbf{J})$, where $\text{adj}(\mathbf{J})$ is the adjugate of \mathbf{J} , the bilinear form $b(\cdot, \cdot)$ may be written as

$$b(q, \mathbf{u}) = \int_{\Omega} p \nabla \cdot \mathbf{v} \, d\mathbf{x} = \int_{\widehat{\Omega}} \widehat{p} \text{tr} \left(\mathbf{J}^{-1} \widehat{\nabla} \widehat{\mathbf{u}} \right) \det(\mathbf{J}) \, d\widehat{\mathbf{x}} = \int_{\widehat{\Omega}} \widehat{p} \text{tr} \left(\text{adj}(\mathbf{J}) \widehat{\nabla} \widehat{\mathbf{u}} \right) \, d\widehat{\mathbf{x}}.$$

It remains to show that $G(\widehat{\mathbf{x}}, \widehat{p}(\widehat{\mathbf{y}}), \widehat{\mathbf{u}}(\widehat{\mathbf{y}})) = \widehat{p}(\widehat{\mathbf{y}}) \text{tr} \left(\text{adj}(\mathbf{J}(\widehat{\mathbf{x}})) \widehat{\nabla} \widehat{\mathbf{u}}(\widehat{\mathbf{y}}) \right)$ is a polynomial of degree $(n-1)p$ in the $\widehat{\mathbf{x}}$ -variable. Applying [Proposition 5.1](#) yields the desired reproduction property. The trace operator tr is linear, thus it suffices to analyze the entries of $\text{adj}(\mathbf{J})$. In 2D, the components of $\text{adj}(\mathbf{J})$ and \mathbf{J} only differ by their position and sign. Since each component of \mathbf{J} is an element of $\mathcal{Q}_p(\widehat{\Omega})$, we conclude that each component of $\text{adj}(\mathbf{J})$ is also in $\mathcal{Q}_p(\widehat{\Omega})$. In 3D, the components of $\text{adj}(\mathbf{J})$ are made up of determinants of 2×2 sub-matrices of \mathbf{J} . Taking the trace yields

$$\text{tr}(\text{adj}(\mathbf{J})) = \det \begin{pmatrix} \mathbf{J}_{22} & \mathbf{J}_{23} \\ \mathbf{J}_{32} & \mathbf{J}_{33} \end{pmatrix} + \det \begin{pmatrix} \mathbf{J}_{11} & \mathbf{J}_{13} \\ \mathbf{J}_{31} & \mathbf{J}_{33} \end{pmatrix} + \det \begin{pmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{pmatrix}.$$

For the first summand, we have $\mathbf{J}_{22} \in \mathcal{P}_p \otimes \mathcal{P}_{p-1} \otimes \mathcal{P}_p$, $\mathbf{J}_{23} \in \mathcal{P}_p \otimes \mathcal{P}_p \otimes \mathcal{P}_{p-1}$, $\mathbf{J}_{32} \in \mathcal{P}_p \otimes \mathcal{P}_{p-1} \otimes \mathcal{P}_p$, and $\mathbf{J}_{33} \in \mathcal{P}_p \otimes \mathcal{P}_p \otimes \mathcal{P}_{p-1}$. From this it follows that $\mathbf{J}_{22} \cdot \mathbf{J}_{33} \in \mathcal{P}_{2p} \otimes \mathcal{P}_{2p-1} \otimes \mathcal{P}_{2p-1}$ and $\mathbf{J}_{23} \cdot \mathbf{J}_{32} \in \mathcal{P}_{2p} \otimes \mathcal{P}_{2p-1} \otimes \mathcal{P}_{2p-1}$. This means that $\mathbf{J}_{22} \cdot \mathbf{J}_{33} - \mathbf{J}_{23} \cdot \mathbf{J}_{32} \in \mathcal{Q}_{2p}(\widehat{\Omega})$. Applying the same arguments to the other summands finally yields that $\text{tr}(\text{adj}(\mathbf{J})) \in \mathcal{Q}_{2p}(\widehat{\Omega})$. \square

In order to discretize [\(10.1\)](#), an inf-sup stable space pair is required. For this purpose, we choose the isogeometric *subgrid* element as described in [\[7\]](#). In this discretization, the velocity field is defined on a subgrid of the pressure where each pressure element is subdivided into 2^n elements. This allows for using a velocity space of order $p+1$ with $C^p(\widehat{\Omega})$ regularity and a pressure space of order p with $C^{p-1}(\widehat{\Omega})$ regularity.

We do not provide *a priori* error estimates for the Stokes problem, but in the case that the divergence matrix is reproduced one would follow similar arguments as presented for the Poisson problem. In the scenario where the divergence matrix is not reproduced, further work is necessary. However, the results in the next subsection suggest that the reproduction is not required in order to obtain an optimal order of convergence.

10.2. Numerical experiments. Our computational study of Stokes' flow is comprised of two separate experiments. In the first experiment, we provide a smooth manufactured solution in order to investigate convergence rates. In the second example, we consider a lid-driven cavity benchmark problem. Due to the discontinuous boundary conditions, the solution of this problem has singularities at two corners of the domain. Both examples are computed on the domain shown in [Figure 10.1](#) which was constructed by a Coons patch with a cubic boundary parameterization. We discretize the problem using the aforementioned subgrid element with a third order velocity and second order pressure. The viscosity is set to $\mu = 1$ for all scenarios.

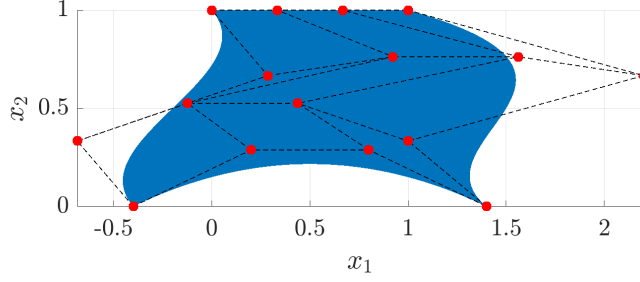


Figure 10.1: The computational domain Ω used in the Stokes flow problem (10.1).

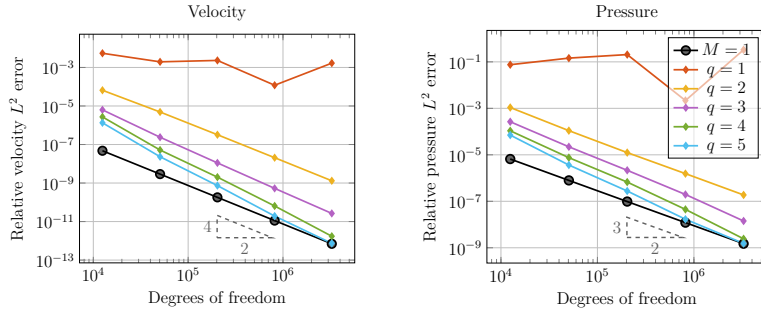


Figure 10.2: Stokes' flow problem (10.1) with manufactured solution (10.2). The velocity is discretized with $p = 3$ and the pressure with $p = 2$. Relative L^2 velocity and pressure errors for $M = 5$.

In the first example, the manufactured solution is chosen to be

$$(10.2) \quad \mathbf{u}(x, y) = \begin{bmatrix} \frac{\sin(x) \cos(y)}{x+1} \\ \frac{((x+1) \cos(x) - \sin(x)) \sin(y)}{(x+1)^2} \end{bmatrix}, \quad p(x, y) = y \sinh(x) + C_p.$$

Note that this solution satisfies $\nabla \cdot \mathbf{u} = 0$ and the constant $C_p \in \mathbb{R}$ is chosen such that the pressure mean is zero. The Dirichlet boundary condition \mathbf{g} and the right hand side \mathbf{f} are set accordingly to match the manufactured solution. In Figure 10.2, we present convergence plots for the velocity and pressure separately for different surrogate orders q and fixed $M = 5$. For reference, we also include the standard discretization with $M = 1$ in these plots. We observe the expected convergence orders for all $q \geq 2$. In agreement with Corollary 10.1, the divergence matrices were perfectly reproduced for every $q \geq 3$.

In the second example, we consider a lid-driven cavity benchmark on the domain Figure 10.1 where the fluid is driven on the top edge by constant velocity $\mathbf{g} = (1, 0)^\top$ and we assume no-slip boundary conditions $\mathbf{g} = \mathbf{0}$ on the remaining parts of the boundary. The degrees of freedom corresponding to the nodal basis functions in the top left and top right corner are set to zero. Furthermore, the volume forces are neglected, i.e., $\mathbf{f} = \mathbf{0}$. In Figure 10.3 (a), we show the velocity streamlines which were computed using a standard IGA approach on a mesh with 320×320 control points. The effect of different surrogate approaches on the velocity streamlines may be observed in Figures 10.3 (b)–10.3 (j). In the case $q = 3$, where the divergence matrices are in fact reproduced, the streamlines show the same behavior as in the standard approach even for $M = 100$. For other values of q and M , the streamline behavior is different, but the streamlines are getting closer to the reference solution the larger q and the smaller M becomes.

We note that actually using an interpolation order of $q = 1, 2, 3$ in computation is still probably not recommended for standard practice. For instance, assembly using $q = 5$ took roughly the same time as either of these lower-order choices and, in this case, the surrogate solution $(\tilde{\mathbf{u}}_h, \tilde{p}_h)$ should be expected to be even more accurate.

Appendix A. Marsden's identity.

The purpose of this appendix is to substantiate some of the claims made in [Subsection 3.3](#), as well as provide a complete proof of [Theorem 4.2](#). In turn, we adopt all of the notation and assumptions introduced in [Subsection 2.2](#). We begin by stating [Theorem A.1](#), the proof of which can be found in [\[18\]](#); see also [\[15, 16, 19\]](#).

THEOREM A.1 (Marsden). *Let $\psi_k(\hat{y}) = (\xi_{k+1} - \hat{y}) \cdots (\xi_{k+p} - \hat{y})$. For any $\hat{x}, \hat{y} \in [0, 1]$,*

$$(\hat{x} - \hat{y})^p = \sum_{k=1}^m b_k(\hat{x}) \psi_k(\hat{y}).$$

This theorem allows us to conclude that the expression [\(3.9\)](#) is valid and, moreover, $w \in \mathcal{Q}_p(\tilde{\Omega}_0)$. This is shown in two steps.

COROLLARY A.2. *Let $\Psi(\tilde{\mathbf{y}}) = \prod_{i=1}^n \prod_{j=0}^{p-1} \left(\frac{j}{m-p} - \frac{p-1}{2(m-p)} - \tilde{y}_i \right)$ and $\mathbf{p} = (p, \dots, p) \in \mathbb{N}^n$. Then, for any $\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \in \tilde{\Omega}_0$,*

$$(A.1) \quad (\tilde{\mathbf{x}} - \tilde{\mathbf{y}})^{\mathbf{p}} = \sum_{\tilde{\mathbf{x}}_i \in \tilde{\mathbb{X}}} \hat{B}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i) \Psi(\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_i).$$

Proof. Observe that, for each $k = p+1, \dots, m-p$, it holds that $\tilde{x}^{(k)} + h = \tilde{x}^{(k+1)} = \xi_{k+1} + (p+1) \cdot h/2$. Recall that $k = p+1, \dots, m-p$ are exactly the indices of the cardinal B-splines $b_k(\hat{x}) = b(\hat{x} - \tilde{x}^{(k)})$. Therefore, by [Theorem A.1](#), we see that

$$(\tilde{x}_i - \tilde{y}_i)^p = \sum_{k=p+1}^{m-p} b(\tilde{x}_i - \tilde{x}^{(k)}) \prod_{j=0}^{p-1} \left(\tilde{x}^{(k)} + \left(j - \frac{p-1}{2} \right) \cdot h - \tilde{y}_i \right),$$

for each $i = 1, \dots, n$. The result now follows from the definitions of \hat{B} and Ψ . \square

[Corollary A.2](#) can be used to write out an elegant expression for any polynomial in $\mathcal{Q}_p(\tilde{\Omega}_0)$, in terms of cardinal B-splines. Indeed, let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)$ be a multi-index, let f be an arbitrary polynomial in $\mathcal{Q}_p(\tilde{\Omega}_0)$, and let $D^{\boldsymbol{\alpha}}$ be the $\boldsymbol{\alpha}$ -derivative operator, in the variable $\tilde{\mathbf{y}}$. By Taylor's Theorem, it holds that

$$(A.2) \quad f(\tilde{\mathbf{x}}) = \sum_{i=1}^n \sum_{\boldsymbol{\alpha}_i \leq \mathbf{p}} \frac{D^{\boldsymbol{\alpha}_i} f(\tilde{\mathbf{y}})}{\boldsymbol{\alpha}_i!} (\tilde{\mathbf{x}} - \tilde{\mathbf{y}})^{\boldsymbol{\alpha}_i},$$

for every $\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \in \tilde{\Omega}_0$. Next, applying $D^{\mathbf{p}-\boldsymbol{\alpha}}$ to both sides of [\(A.1\)](#), we find that

$$(A.3) \quad \frac{(-1)^{|\mathbf{p}-\boldsymbol{\alpha}|} \mathbf{p}!}{\boldsymbol{\alpha}!} (\tilde{\mathbf{x}} - \tilde{\mathbf{y}})^{\boldsymbol{\alpha}} = \sum_{\tilde{\mathbf{x}}_i \in \tilde{\mathbb{X}}} \hat{B}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i) D^{\mathbf{p}-\boldsymbol{\alpha}} \Psi(\tilde{\mathbf{y}} - \tilde{\mathbf{x}}_i).$$

Together, [\(A.2\)](#) and [\(A.3\)](#) imply $f(\tilde{\mathbf{x}}) = \sum_{\tilde{\mathbf{x}}_i \in \tilde{\mathbb{X}}} \hat{B}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_i) (Lf)(\tilde{\mathbf{x}}_i)$, where

$$(Lf)(\tilde{\mathbf{x}}) = \sum_{i=1}^n \sum_{\boldsymbol{\alpha}_i \leq \mathbf{p}} \frac{(-D)^{\mathbf{p}-\boldsymbol{\alpha}_i} \Psi(\tilde{\mathbf{y}} - \tilde{\mathbf{x}})}{\boldsymbol{\alpha}_i!} D^{\boldsymbol{\alpha}_i} f(\tilde{\mathbf{y}}).$$

We may now state our second corollary of [Theorem A.1](#).

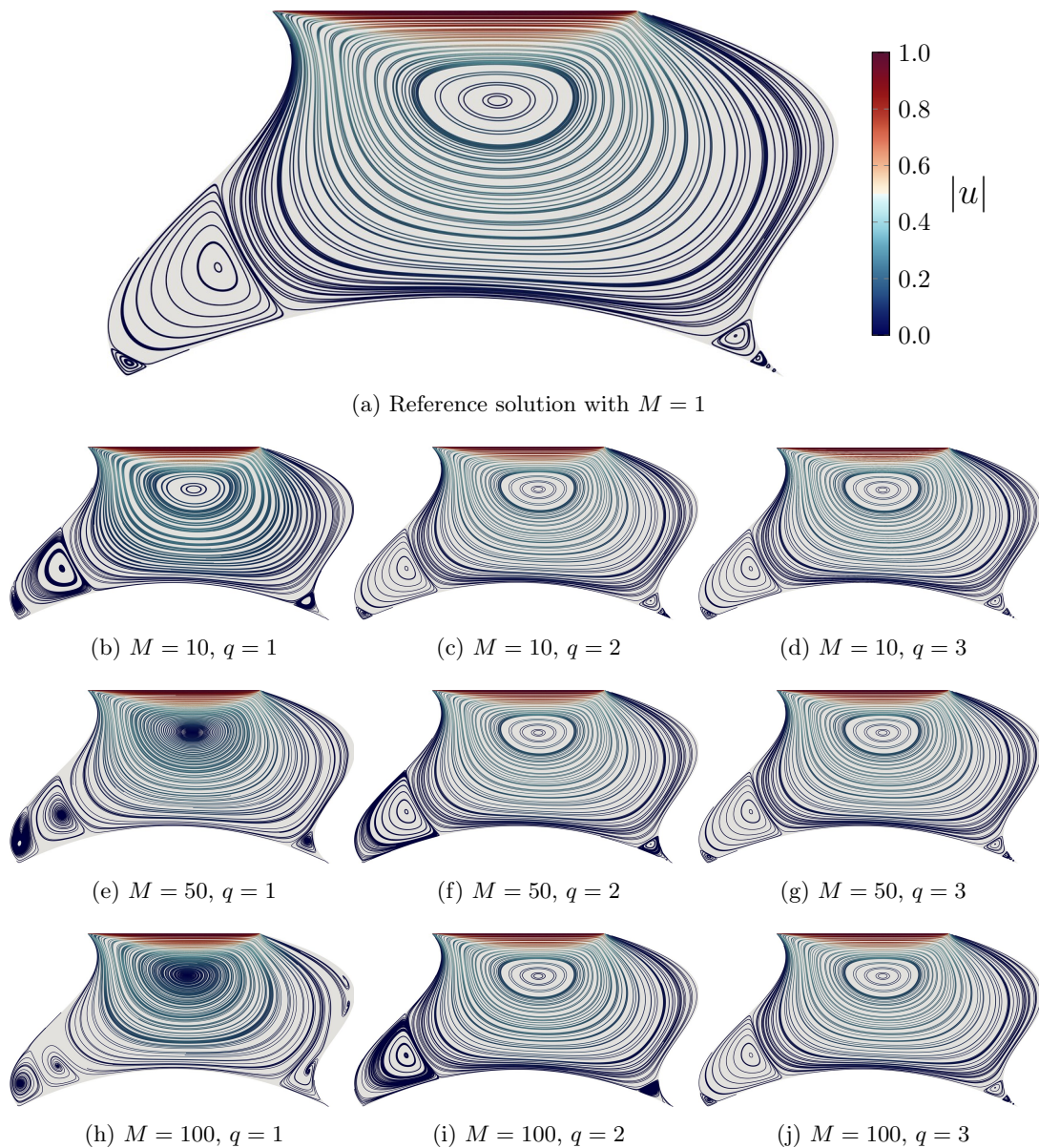


Figure 10.3: Lid-driven cavity benchmark velocity streamlines. The velocity is discretized with $p = 3$ and the pressure with $p = 2$ using surrogate approaches with varying M and q . The mesh is discretized by 320×320 control points. Note that actually using an interpolation order of $q = 1, 2, 3$ is still probably not recommended for standard practice since more accurate results should be expected with $q = 5$ while taking roughly the same time.

COROLLARY A.3. Let $W(\hat{\mathbf{x}}) = \sum_{j=1}^N w_j \hat{B}_j(\hat{\mathbf{x}})$. If $W \in \mathcal{Q}_p(\hat{\Omega})$, then there exists a polynomial $w \in \mathcal{Q}_p(\tilde{\Omega}_0)$ such that $w_i = w(\tilde{\mathbf{x}}_i)$, for each $\tilde{\mathbf{x}}_i \in \tilde{\mathbb{X}}$. Moreover, there exists a constant C , depending only on p , such that

$$(A.4) \quad \|w\|_{W^{r,\infty}(\tilde{\Omega}_0)} \leq C \|W\|_{W^{r,\infty}(\hat{\Omega})} \quad \text{for all } r \geq p.$$

Proof. Clearly, $w = LW \in \mathcal{Q}_p(\tilde{\Omega}_0)$. Therefore, one readily determines that $\|w\|_{L^\infty(\tilde{\Omega}_0)} \leq C \|W\|_{W^{p,\infty}(\tilde{\Omega}_0)}$, for some constant C , depending only on p . Due to the equivalence of norms on finite dimensional vector spaces (note that the dimension of $\mathcal{Q}_p(\tilde{\Omega}_0)$ depends only on p) and the fact $\tilde{\Omega}_0 \subseteq \tilde{\Omega}$, we immediately arrive at (A.4). \square

We may now complete the proof of [Theorem 4.2](#).

Proof of Theorem 4.2. Let $\tilde{\mathbf{x}} \in \tilde{\Omega}$ be arbitrary. Then, for any $\delta \in \mathcal{D}$, we have that $\tilde{\mathbf{x}} + \delta \in \tilde{\Omega}_0$. Moreover, for every multi-index $|\alpha| = r$, the product rule can be used to show that

$$\begin{aligned} D^\alpha \Phi_\delta(\tilde{\mathbf{x}}) &= D^\alpha \left[w(\tilde{\mathbf{x}}) w(\tilde{\mathbf{x}} + \delta) \int_{\tilde{\omega}_\delta} \hat{\nabla} \left(\frac{\hat{B}(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}} + \hat{\mathbf{y}})} \right)^\top K(\tilde{\mathbf{x}} + \hat{\mathbf{y}}) \hat{\nabla} \left(\frac{\hat{B}_\delta(\hat{\mathbf{y}})}{W(\tilde{\mathbf{x}} + \hat{\mathbf{y}})} \right) d\hat{\mathbf{y}} \right] \\ &\leq C \cdot \|w\|_{W^{r,\infty}(\tilde{\Omega}_0)}^2 \|K\|_{W^{r,\infty}(\hat{\Omega})} \|W\|_{W^{r,\infty}(\hat{\Omega})}^2 \|\hat{\nabla} W\|_{W^{r,\infty}(\hat{\Omega})}^2 \\ &\quad \cdot (\|\hat{\nabla} \hat{B} \cdot \hat{\nabla} \hat{B}_\delta\|_{L^1(\tilde{\omega}_\delta)} + \|\hat{\nabla} \hat{B} \cdot \hat{B}_\delta\|_{L^1(\tilde{\omega}_\delta)} + \|\hat{\nabla} \hat{B}_\delta \cdot \hat{B}\|_{L^1(\tilde{\omega}_\delta)} + \|\hat{B} \cdot \hat{B}_\delta\|_{L^1(\tilde{\omega}_\delta)}), \end{aligned}$$

for some C depending only on α . Since both functions K , and W are determined by the choice of Ω , [Corollary A.3](#) and a scaling argument show that $\|D^\alpha \Phi_\delta\|_{L^\infty(\tilde{\Omega})} \leq Ch^{n-2}$, where C now depends on p , α , and φ . [Lemma 4.1](#) now completes the proof. \square

Acknowledgments. This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 800898. This work was also partly supported by the German Research Foundation through the Priority Programme 1648 “Software for Exascale Computing” (SPPEXA) and by grant WO671/11-1.

REFERENCES

- [1] Fork of the GeopDEs git repository including the surrogate branch. <https://github.com/drzisga/geopdes>, 2019. [Online; accessed March 13th 2019].
- [2] P. Antolin, A. Buffa, F. Calabr a, M. Martinelli, and G. Sangalli. Efficient matrix computation for tensor-product isogeometric analysis: The use of sum factorization. *Computer Methods in Applied Mechanics and Engineering*, 285:817–828, 2015.
- [3] F. Auricchio, F. Calabro, T. J. Hughes, A. Reali, and G. Sangalli. A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249:15–27, 2012.
- [4] S. Bauer, M. Huber, S. Ghelichkhan, M. Mohr, U. R ude, and B. Wohlmuth. Large-scale simulation of mantle convection based on a new matrix-free approach. *Journal of Computational Science*, 31:60–76, 2019.
- [5] S. Bauer, M. Huber, M. Mohr, U. R ude, and B. Wohlmuth. A new matrix-free approach for large-scale geodynamic simulations and its performance. In *International Conference on Computational Science*, pages 17–30. Springer, 2018.
- [6] S. Bauer, M. Mohr, U. R ude, J. Weism uller, M. Wittmann, and B. Wohlmuth. A two-scale approach for efficient on-the-fly operator assembly in massively parallel high performance multigrid codes. *Applied Numerical Mathematics*, 122:14–38, 2017.
- [7] A. Bressan and G. Sangalli. Isogeometric discretizations of the stokes problem: stability analysis by the macroelement technique. *IMA Journal of Numerical Analysis*, 33(2):629–651, 2013.
- [8] A. Bressan and S. Takacs. Sum-factorization techniques in isogeometric analysis. *arXiv preprint arXiv:1809.05471*, 2018.
- [9] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta numerica*, 13:147–269, 2004.

- [10] T. Butler, P. Constantine, and T. Wildey. A posteriori error analysis of parameterized linear systems using spectral methods. *SIAM Journal on Matrix Analysis and Applications*, 33(1):195–209, 2012.
- [11] F. Calabr a, G. Sangalli, and M. Tani. Fast formation of isogeometric Galerkin matrices by weighted quadrature. *Computer Methods in Applied Mechanics and Engineering*, 316:606–622, 2017. Special Issue on Isogeometric Analysis: Progress and Challenges.
- [12] J. A. Cottrell, T. J. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.
- [13] L. B. Da Veiga, A. Buffa, G. Sangalli, and R. V azquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 2014.
- [14] W. Dahmen, R. De Vore, and K. Scherer. Multidimensional spline approximation. *SIAM Journal on Numerical Analysis*, 17(3):380–402, 1980.
- [15] W. Dahmen, N. Dyn, and D. Levin. On the convergence rates of subdivision algorithms for box spline surfaces. *Constructive Approximation*, 1(1):305–322, 1985.
- [16] W. Dahmen and C. A. Micchelli. Translates of multivariate splines. *Linear Algebra and its Applications*, 52:217–234, 1983.
- [17] P. J. Davis and P. Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.
- [18] C. De Boor. B(asic)-spline basics. Technical report, UW–Madison Mathematics Research Center, 1986.
- [19] C. De Boor, K. H ollig, and S. Riemenschneider. *Box splines*, volume 98. Springer Science & Business Media, 1993.
- [20] C. de Falco, A. Reali, and R. V azquez. GeOPDEs: a research tool for isogeometric analysis of PDEs. *Advances in Engineering Software*, 42(12):1020–1034, 2011.
- [21] D. Drzisga, B. Keith, and B. Wohlmuth. The surrogate matrix methodology: a priori error estimation. *arXiv preprint arXiv:1902.07333*, 2019.
- [22] D. Drzisga, B. Keith, and B. Wohlmuth. The surrogate matrix methodology: A reference implementation for low-cost assembly in isogeometric analysis. *Submitted*, 2019.
- [23] F. Fahrenndorf, L. D. Lorenzis, and H. Gomez. Reduced integration at superconvergent points in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 328:390–410, 2018.
- [24] R. R. Hiemstra, F. Calabro, D. Schillinger, and T. J. Hughes. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:966–1004, 2017.
- [25] R. R. Hiemstra, G. Sangalli, M. Tani, F. Calabro, and T. J. R. Hughes. Fast formation and assembly of finite element matrices with application to isogeometric linear elasticity. ICES Report 19-03, The University of Texas at Austin, 2019.
- [26] C. Hofreither. A black-box low-rank approximation algorithm for fast matrix assembly in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 333:311–330, 2018.
- [27] K. Hollig. *Finite element methods with B-splines*, volume 26. Siam, 2003.
- [28] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41):4135–4195, 2005.
- [29] T. J. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer methods in applied mechanics and engineering*, 199(5-8):301–313, 2010.
- [30] T. J. Hughes and G. Sangalli. Mathematics of isogeometric analysis: a conspectus. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–40, 2018.
- [31] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python. <http://www.scipy.org>, 2001–. [Online; accessed February 11th 2019].
- [32] A. Karatarakis, P. Karakitsios, and M. Papadrakakis. GPU accelerated computation of the isogeometric analysis stiffness matrix. *Computer Methods in Applied Mechanics and Engineering*, 269:334–355, 2014.
- [33] A. Mantzaflaris and B. J uttler. Exploring matrix generation strategies in isogeometric analysis. In M. Floater, T. Lyche, M.-L. Mazure, K. M orken, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces*, pages 364–382, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [34] A. Mantzaflaris and B. J uttler. Integration by interpolation and look-up for Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:373–400, 2015.
- [35] A. Mantzaflaris, B. J uttler, B. N. Khoromskij, and U. Langer. Low rank tensor methods in Galerkin-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:1062–1085, 2017.
- [36] S. A. Mattis and B. Wohlmuth. Goal-oriented adaptive surrogate construction for stochastic inversion. *Computer Methods in Applied Mechanics and Engineering*, 339:36–60, 2018.
- [37] L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 2012.
- [38] G. Sangalli and M. Tani. Matrix-free weighted quadrature for a computationally efficient isogeometric k -method. *Computer Methods in Applied Mechanics and Engineering*, 338:117–133, 2018.
- [39] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part A. on the problem of smoothing or graduation. A first class of analytic approximation formulae.

- Quarterly of Applied Mathematics*, 4:45–99, 1946.
- [40] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part B. On the problem of osculatory interpolation. A second class of analytic approximation formulae. *Quarterly of Applied Mathematics*, 4:112–141, 1946.
- [41] I. J. Schoenberg. *Cardinal spline interpolation*, volume 12. Siam, 1973.
- [42] R. Vázquez. A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0. *Computers & Mathematics with Applications*, 72(3):523–554, 2016.