Solving Elliptic Partial Differential Equations on Free-form Spline Surfaces with a Polyhedral Control Net

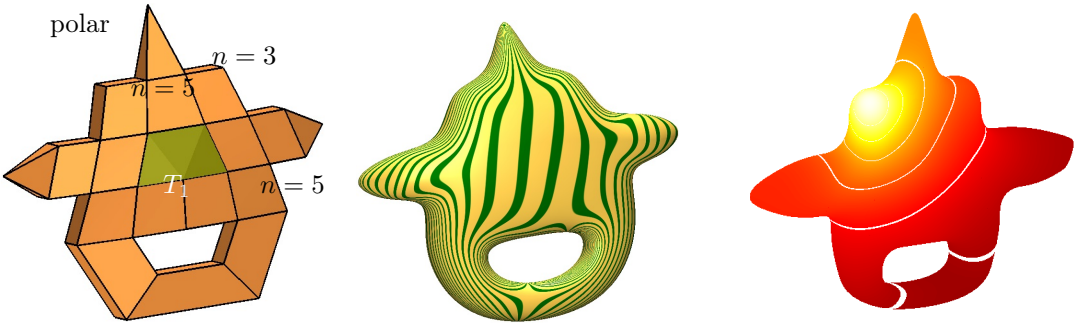
Anonymous as required by submission instructions

[a] *anonymous*

Abstract

Bi-cubic polyhedral-net splines (PnS, ToMS Algorithm 1032) can represent a wide variety of smooth free-form surfaces, but also serve as a differentiable spline space to approximately solve a partial differential equation (PDE) on PnS surfaces. This dual role of PnS avoids re-meshing and yields a parsimonious C^1-smooth Galerkin PDE formulation. Here we apply the approach to solve (time-dependent) elliptic PDEs up to fourth order on spline manifolds leveraging the `dolfinx` environment. The spline manifolds can be open surfaces where the solution matches boundary constraints. The output is the 4-tuple of surface geometry and associated coefficients of the PDE solution, all in polyhedral-net spline form.

Keywords: polyhedral-net spline, free-form surface, n-sided face, C1 spline, T-junction, extraordinary point, polar layout, functions on manifolds, isogeometric analysis



(a) polyhedral control net with irregular configurations

(b) PnS highlight lines indicate differentiability

(c) geodesic isolines from solving the heat equation on the PnS surface

Figure 1: Solving the heat equation on a polyhedral-net spline (PnS) surface. (a) A clearly non-tensor product polyhedral-net with closely-packed irregular patterns (cf. Fig. 2). (b) Verification of geometric smoothness. (c) Geodesic iso-lines representing the progress of heat over the PnS surface.

1. Motivation: Elliptic PDEs on bivariate curved manifolds

Elliptic partial differential equations (PDEs) on free-form surfaces need to take intrinsic surface geometry into account. In the Galerkin formulation of such equations, the surface geometry enters via the first fundamental form, which measures area via inner products of the surface gradients. Whether the free-form

*Corresponding author

[1]This is the first author footnote.

surface arises from human design or as interfaces in volumetric field, a differentiable surface representation is needed for exact computation - and, fourth-order PDEs in the (unmodified) weak form, require a differentiable solution space to compute on the surface. Indeed, for domains with re-entrant, concave corners, the use of differentiable elements is necessary to avoid incorrect outcomes [26]. Similarly, weak penalty enforcement of boundary conditions can yield fictitious solutions.

A sensible approach, that avoids re-meshing and yields a parsimonious C^1-smooth Galerkin formulation, is the higher-order iso-parametric approach of [10, 59, 2, 55]. However, this iso-geometric (IgA) approach [31] has historically been restricted to tensor-product splines with a irregular grid-like layout implying that the domain must be a torus, cylinder or the plane. In 2000, the approach was extended to subdivision surfaces [15] and, in 2015, [44] solved a fourth-order elasticity PDE on geometrically continuous free-form surfaces defining both the geometry and the force field by the same piecewise polynomial space. Since then, higher-order finite element structural analysis has been more widely developed, for example in [45, 13, 8, 47, 63, 29] to list just a few.

Here, we focus on representing geometry and PDE solution space by polyhedral-net splines (PnS, [52]). PnS generalize tensor-product splines by allowing patterns as shown in Fig. 2. Fig. 1 illustrates the combined use of polyhedral-net spline for geometry and PDE analysis space. Fig. 1a shows a control net with closely-packed non-tensor product polyhedral-net patterns. For example, the top 'pole' modeled by 4 triangles is a direct neighbor of four nodes of valence $n = 5$ that also abuts a vertex of valence $n = 3$. Fig. 1b verifies the smooth flow of the PnS surface normal via evenly-distributed highlight lines [6]. Fig. 1c illustrates the geodesic iso-lines of a solution to the heat equation on a polyhedral-net spline using polyhedral-net spline elements. The point source of heat is the white turning to yellow, in the upper left.

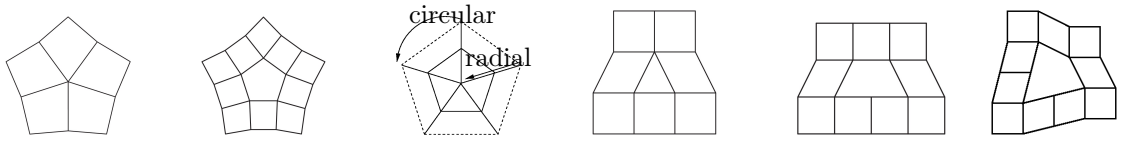
For the gamut of mesh modeling control net configurations, multi-sided, polar and mesh-line shedding as illustrated in Fig. 2, there are currently no, by default differentiable, constructions other than PnS. Replacing mesh shedding and polar configurations by pure quad layouts typically results in shape defects, T-splines require a global parameterization, say of Fig. 1, and Catmull-Clark subdivision would lead to good analysis on bad geometry. The advantage of using PnS directly to solve PDEs is that splines efficiently model both the surface and the PDE solution space while admitting local control net pattern that do not require tensor-structure, see Fig. 1 and Fig. 2. PnS can be re-expressed in the standard industry formats, such as IGES and STEP, and therefore admit many possible applications in design, visualization, animation, moment computation, re-approximation, reconstruction, etc.

Although consistent with the theory and convenient when paired with the same smooth surface representation, C^1 elements are not necessary for second-order equations and fourth-order equations on domains with a convex boundary. Even discontinuous flow lines, arising from a discontinuous Galerkin approach, can be smoothed out in a post-processing step [62]; the operator can be split, requiring only a mixed weak formulation; or applying a penalty method. Moreover, it is possible to model non-grid geometry via trimming, i.e., restricting the tensor-product geometry to a non-rectangular domain, or use subdivision surfaces [53] consisting of an infinite sequence of nested polynomial surface rings. However, these approaches require special treatment of differentiation and integration: trimmed tensor-product splines [18], known under the acronym NURBS, can have slivers of poor aspect ratio, heterogeneity in size, parameter orientation, continuity and polynomial degree; and subdivision surfaces generate singularly-parameterized central limit points. The PnS approach avoids these problems and guarantees no gaps and exactly matching derivatives.

Overview. After reviewing polyhedral-net splines (PnS) and related literature in Section 2, we summarize the isogeometric approach on free-form surfaces in Section 3. Sections 4, 5, 6, 7 respectively formulate and express in Galerkin (weak) form: elliptic PDEs of degree 2, time-dependent PDEs of degree 2, PDEs of degree 4, and time-dependent PDEs of degree 4 – on free-form domains with irregular patterns as in Fig. 2.

2. Polyhedral control nets and polyhedral-net splines

Polyhedral-net splines (PnS), introduced in [52] as Algorithm 1032, are defined, and geometrically outlined, by a topological polyhedron. This type of control net generalizes tensor-product grids and forms a piecewise polynomial, first-order differentiable space with one function associated with each vertex of the



(a) star-net $n = 5$ (b) n-gon-net, $n = 5$ (c) polar-net $m = 5$ (d) T_0-net (e) T_1-net (f) T_2-net

Figure 2: Some admissible non-tensor-product polyhedral control net patterns defining surface pieces (from [52]). The first three patterns isotropically merge n features, the last three reduce or increase the number of control-net lines.

polyhedron. Akin to the well-known tensor-product splines [19], the control net outlines and exaggerates the smooth PnS surface. The polyhedron facets need not be flat, but, for brevity, we drop the qualifier 'topological' from topological polyhedron in the following.

ToMS Algorithm 1032 provides code for PnS of degree bi-3, i.e. degree 3 in each of the two variables, Only the elements on a regular grid are bi-quadratic (bi-2) splines, but are represented in bi-3 form (degree-raised) for uniformity. Bi-3 (bi-cubic) polynomial pieces are universally supported in geometric modeling softwares and computing environments that support curved, higher-order elements. The bi-3 pieces are output in the convenient Bernstein-Bézier (BB-) form [23] or as a collection of pieces in uniform tensor-product B-spline form [19]. Admissible polyhedral control nets consist of quadrilateral faces in a local grid layout, often called 'regular tensor-product' layout, and irregular configurations as illustrated in Fig. 2. Three types of T-junctions shed or birth quad-strips. Polar configurations form a cone of n triangles meeting at a vertex, surrounded by a ribbon of n quadrilaterals. In n-gon configurations, $2n$ quadrilaterals surround an n-gon. Star-configurations allow $n \neq 4$ quadrilateral faces to join around an interior vertex. The polyhedral patterns listed in Fig. 2 can be placed in close proximity, enabling complex layouts such as Fig. 1. Patterns can arise from joining primary surfaces and also as a form of local adaptation.

The polynomial pieces of a polyhedral-net spline depend linearly on the polyhedral control net. Evaluating, differentiating, integrating, or performing another operation on a PnS is as efficient as converting a tensor-product spline from B-spline form to BB-form and performing the corresponding calculation. The pieces of a polyhedral-net spline join with matching derivatives after a known change of variables (reparameterization), and the polyhedral control net expresses the neighbor relations of the polyhedral-net spline generating functions, outlines shape, and provides handles for manipulating shape. Due to these properties, the polyhedral control net vertices can be used as computational degrees of freedom, e.g., for least squares fitting, computing moments or, as is our goal, for solving partial differential equations. Bi-cubic polyhedral-net splines can both model geometry and functions on the geometry, and approximate solutions to PDEs in particular. The input and output of Algorithm 1032 define the PnS space:

Input: A polyhedral (control) mesh.

Output: A smooth polyhedral-net spline with polynomial pieces of degree bi-3.

The algorithm collects any of the itemized sub-nets of control points and applies a linear transformation to the sub-net to generate the polynomial coefficients of the output pieces. By default, the transformation automatically enforces geometric continuity, i.e. smoothness after change of variables, between the pieces, based on [37] (Fig. 2a, Fig. 2b); [39] Fig. 2c; [38] (Fig. 2d, 2e, 2f).

2.1. Alternative surface representations

Commonly-used computational function spaces for polyhedral unstructured layout either sacrifice differentiability near irregularities, as in the (dis)continuous Galerkin approach [16], or restrict the computation to non-conforming, internally smooth spline regions subject to penalty functions [46]. Piecewise polynomial, a.k.a. spline functions on triangulations [40], and radial basis functions [11] provide built-in smoothness but are typically restricted to planar domains as opposed to free-form splines that not restricted to be functions on a fixed joint domain.

A survey of free-form surface constructions controlled by meshes with changing vertex valence can be found in [50] which classifies smooth surface representations into three major classes: (a) singular representations, (b) rational blending, and (c) geometrically continuous constructions. The classes differ by their math-

ematical strategies for mapping the domains of n pieces locally, without overlap and gap, into one common parameter plane. (a) Singular representations include those with singularities at corners [48, 54, 45, 60, 65], singular edges [43], and an infinite sequence of contracting faces, a.k.a. subdivision algorithms [20, 53], that are usually approximated by a finely-faceted model. (b) Rational blending constructions typically have rational singularities outside the domain or at the corners [27, 61, 30]. (c) Geometrically continuous (GC) constructions join smoothly after a change of variables. General GC constructions include, for example, [49, 44, 17, 8], and for data fitting and simulation specific to planar domains [7, 33, 35, 34]. Collocation in [56] is based on T-splines, i.e. on [22] for n-valent vertices. [64] re-introduces the approach of [36] to the IGA community and [24] is a non-linear construction yielding non-uniform angles where patches meet.

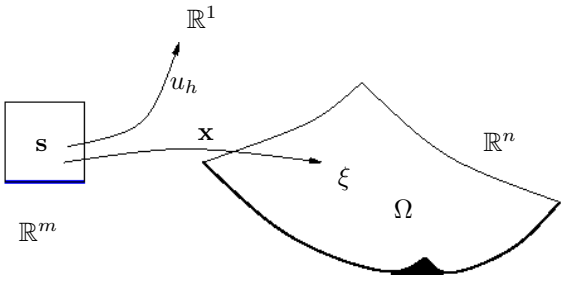
3. The isogeometric approach

Let

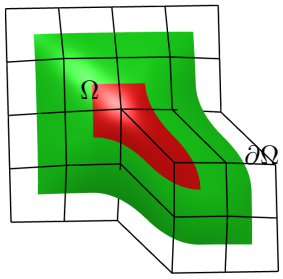
$$\mathbf{x} : \mathbf{s} \in \square \subset \mathbb{R}^m \to \xi = (\xi_1, \ldots, \xi_n) \in \Omega \subset \mathbb{R}^n \quad .$$

be the geometry map that defines the physical domain, see Fig. 3a. The function $u_h : \square \subset \mathbb{R}^m \to \mathbb{R}$ is to be determined so that $q = u_h \circ \mathbf{x}^{-1}$ satisfies constraints, for example as a solution of the Poisson equation or the biharmonic equation. Here, $\mathbf{x}^{-1}$ the pullback from Ω to $\square$. In our context, the 'physical domain' Ω embedded in 3-space is a free-form surface parameterized by polyhedral-net splines:

$$\mathbf{x} := \sum_i \mathbf{x}_i \phi_i, \quad \phi_i : \mathbb{R}^2 \to \mathbb{R} \text{ with control points } \mathbf{x}_i \in \mathbb{R}^3 \text{ and gradient } \nabla\phi_i = \begin{bmatrix} \partial_s \phi_i \\ \partial_t \phi_i \end{bmatrix}. \tag{1}$$



(a) Isogeometric maps



(b) A polyhedral control net and spline.

Figure 3: (a) Scalar-valued function u_h is defined on the same parameter domain mapped by $\mathbf{x}$ to span the physical domain Ω. (b) A polyhedral control net and polyhedral-net spline (green: regular bi-2 splines, red: degree bi-3 PnS) that parameterize Ω.

For the Galerkin (weak) formulation of the elliptic PDEs, we use PnS test functions $\psi_i = \phi_i$. We implemented the approach both in C++, and – by extending the change of basis approach of [9, 32] to PnS – in a Python `dolfinx` framework. This allows us to use existing FEA software by expressing PnS elements as constrained Lagrange elements native to the solver package `dolfinx` [5, 58, 57, 1], leveraging its assembly and matrix functionality to solve the equations in terms of the PnS control points. By default, we compute integrals by the tensored 5-point Gauss Quadrature Rule. Since the number of irregular patterns does not increase under refinement, we do not focus on measuring 2-norm convergence. Convergence in $\mathcal{L}^2$ is anyhow dominated by the increasing regularity of the a refined mesh, see Fig. 6. To properly analyze the error caused by the isolated irregular configurations, we therefore work with the more informative L^∞ norm errors, displayed graphically, for a gallery of equations where the solution is known. This allows us to assess both correctness and gauge the impact of patterns on the error.

4. Second-order elliptic equations

Denote by $\Delta_{\mathbf{x}}$ the Laplace Beltrami operator, i.e. the Laplacian on the surface $\mathbf{x}$, and by $\partial \mathtt{u}_\Omega$ the partial derivative of $\mathtt{u}_\Omega$ transversal to the boundary $\partial\Omega>$ Poisson's equation seeks

$$\mathtt{u}_\Omega : \Omega \to \mathbb{R} : \begin{cases} \Delta_{\mathbf{x}} \mathtt{u}_\Omega = -f & \text{on } \Omega \in \mathbb{R}^3, \\ \mathtt{u}_\Omega = \mathtt{u}_b \text{ or } \partial \mathtt{u}_\Omega = \mathtt{u}'_b & \text{on } \partial\Omega. \end{cases} \tag{2}$$

Here, $\mathtt{u}_\Omega$ has to satisfy on each piece of the boundary either Dirichlet $\mathtt{u}_\Omega = \mathtt{u}_b$ or von Neumann $\partial \mathtt{u}_\Omega = \mathtt{u}'_b$ constraints. The weak Galerkin form of (2), pulled back by $\mathbf{x}^{-1}$ to the domain $\square$ of ϕ_j and projected into the space of polyhedral-net spline functions ϕ_j is

$$\int_\Omega \nabla \mathtt{u}_h \cdot \nabla \psi_i(\mathbf{x}^{-1}) d_\Omega = \int_\Omega f \psi_i(\mathbf{x}^{-1}) d_\Omega, \quad \mathtt{u}_h := \sum_j c_j \phi_j(\mathbf{x}^{-1}). \tag{3}$$

For (3) to be valid, the test functions ψ_j must vanish at the domain boundary $\partial\Omega$. Eq. (3) is solved as a matrix equation in terms of the coefficient vector $\mathbf{c}$ of $\mathtt{u}_h$:

$$K\mathbf{c} = \mathbf{f} \qquad K_{ij} := \int_\Omega \nabla \phi_i(\mathbf{x}^{-1}) \cdot \nabla \psi_j(\mathbf{x}^{-1}) d_\Omega, \quad \mathbf{f}_i := \int_\Omega f \psi_i(\mathbf{x}^{-1}) d_\Omega. \tag{4}$$

Since each polyhedral-net spline ϕ_i can be expressed in piecewise bivariate Bernstein-Bézier form, the domain of each ϕ_i consists of a collection of unit squares $\square := [0..1]^2$. That is, the pre-image of Ω consists of squares that we enumerate with the label α. It is therefore good to re-express Eq. (4) by changing to variables on $\square$. We denote the gradient and its determinant as

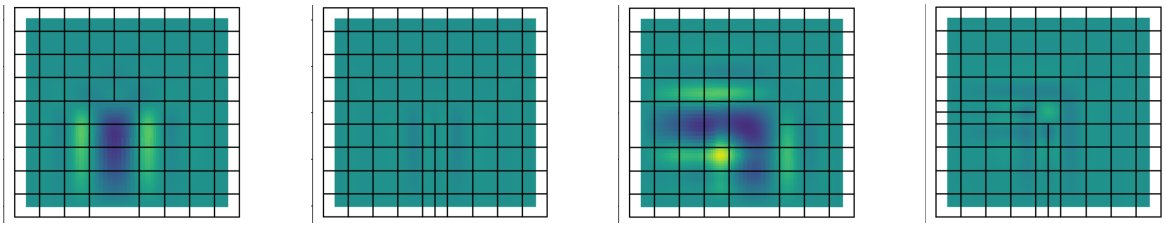
$$J_\alpha := \nabla_{\mathbf{s}} \mathbf{x}_\alpha = \begin{bmatrix} \partial_s x & \partial_t x \\ \partial_s y & \partial_t y \\ \partial_s z & \partial_t z \end{bmatrix} \in \mathbb{R}^{3\times 2} \qquad \mathtt{J} := \sqrt{\det(J_\alpha^{\mathtt{t}} J_\alpha)} \in \mathbb{R}.$$

Then integration over the unit square domain, Eq. (4) becomes as in [44] for row i and column j,

$$K_{ij} = \sum_\alpha \int_\square (\nabla \phi_j)^{\mathtt{t}} (J_\alpha^{\mathtt{t}} J_\alpha)^{-1} (\nabla \psi_i) \ \mathtt{J} \, \mathrm{d}\square, \quad \mathbf{f}_i = \sum_\alpha \int_\square f \psi_i \ \mathtt{J} \, \mathrm{d}\square. \tag{5}$$

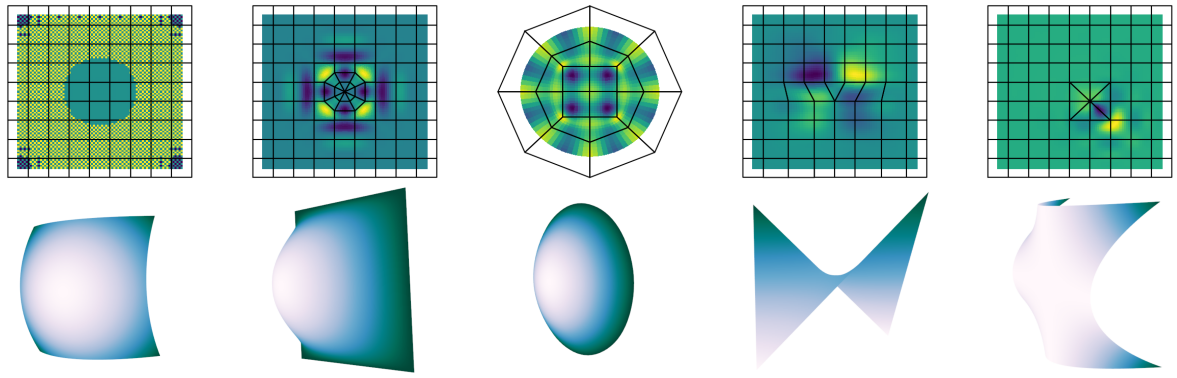
The sum in (5) extends over all pieces α where ϕ_i and ψ_j have support.

Dirichlet and von Neumann (gradient normal to the boundary) *boundary conditions* are enforced by the values associated with the outermost two layers of the control net. Alternatively, von Neumann boundary conditions are introduced by replacing the normal component of the gradient of the solution in the formulation of the weak form with the von Neumann values.



(a) T1 subtractive (b) T1 additive (c) T2 subtractive (d) T2 additive

Figure 4: Error patterns in Poisson's equation with solution $u = -(x^2-1)(z^2-1)$ when subtracting or adding partial mesh lines to form T-junctions. Since $\mathtt{u}$ is bi-quadratic the error on the regular 10×10 grid (and most regular 4×4 regular sub-grids in a,b,c,d) is 10^-8.



(a) $\mathtt{u} = (x^2 + z^2)/2$ (b) $\mathtt{u} = -(x^2 - 1)(z^2 - 1)$ (c) $\mathtt{u} = (x^2 + z^2)/2$ (d) $\mathtt{u} = xz$ (e) $\mathtt{u} = x^2 z^2$

Figure 5: Error distribution ($1e^{-8}$ of size for (a) and $\sim 1e^{-3}$ otherwise) (*top*) and shape (*bottom*) when, except for (a), introducing irregular configurations.

Our first set of tests, illustrated by Fig. 4 looks at Poisson's Equation with homogeneous boundary conditions on the unit square – but using control nets with *artificially introduced irregular patterns*. We solve

$$\Delta \mathtt{u} = 2x^2 + 2z^2 - 4, \quad x, z \in [-1, 1]^2, \tag{6}$$

which has the known solution $\mathtt{u} = -(x^2 - 1)(z^2 - 1)$. Here x, y and z are parametric PnS and we set $y = 0$ to simulate bivariate functions. Fig. 4 shows the error distribution when computing $\mathtt{u}$ in PnS-form on irregular meshes created by removing or adding partial mesh lines. The disappearance of the error in the additive cases shows that the error is not due to the irregularity, but rather to the lower resolution when removing part of a mesh line. Refining the mesh from which we remove a partial mesh line as in Fig. 4(a), we determine $\mathcal{L}^2$ error ratios $r_n := e_n/e_{n+1}$ where e_n is the $\mathcal{L}^2$ error on the grid with spacing 2^{-n}. We find the error ratios to be in line with, and slightly faster than, predicted by the (bi-)quadratic reproduction of PnS:

$$(r_3, r_4, r_5, r_6) = (8.581346846, 11.60928394, 11.51746173, 11.39926402).$$

The same information is displayed as log-scale L^2 errors in Fig. 6. The decay of the error in Fig. 6 is representative for all other PnS configurations since the $\mathcal{L}^2$ error is dominated by the increasing regularity of the refined mesh, i.e. meshes are minimally modified from the regular case. To allow comparison, these initial measurements are not shown for free-form surfaces, but bivariate functions.

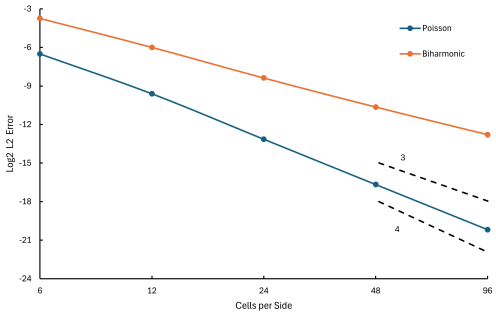


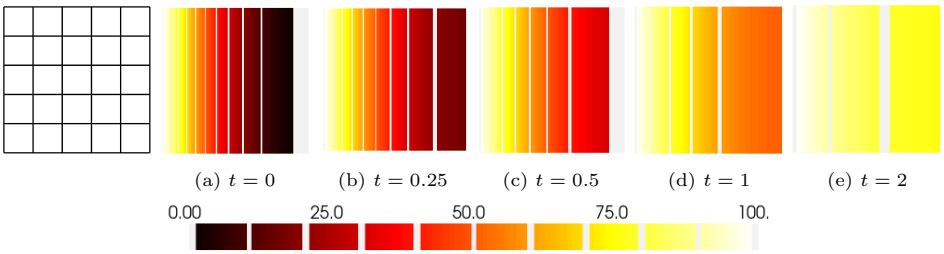
Figure 6: The log L^2 error for refined grids with a central T1 subtractive configuration. Poisson's equation with solution $\mathtt{u} = -(x^2 - 1)(z^2 - 1)$ decays $\sim 2^{-3.5}$. The Biharmonic equation with solution $\mathtt{u} = (x^2 - 1)^2(z^2 - 1)^2$ will be discussed in Section 6.

Fig. 5 shows a gallery of solutions and their errors for Poisson problems with known solutions on meshes with irregularities, and various Dirichlet and Neumann boundary conditions. Besides pure Dirichlet, we

tested pure von Neumann conditions with one anchoring value fixed at a corner. Only Fig. 5a uses a regular mesh and the exact solution $\mathtt{u} = (x^2 + z^2)/2$ for

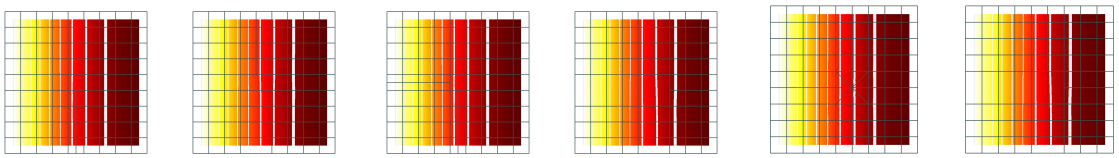
$$\Delta \mathtt{u} = -2, \quad (x, z) \in [-1, 1]^2 \tag{7}$$

with suitable boundary conditions is reproduced to 10^{-7}.



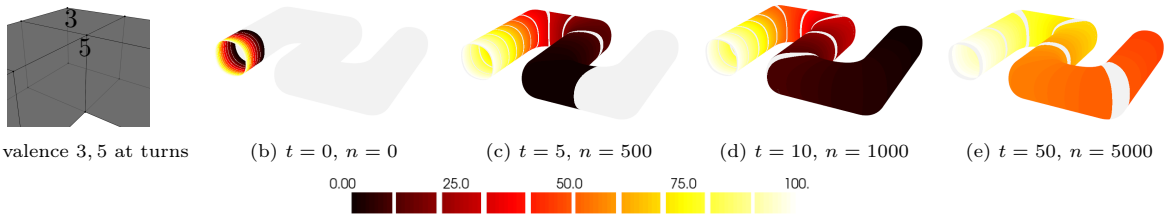
(a) $t = 0$ (b) $t = 0.25$ (c) $t = 0.5$ (d) $t = 1$ (e) $t = 2$

Figure 7: Heat progression from Dirichlet condition of 100 on left, von Neumann condition of 0 on the other sides.



(a) across T1: added, subtracted (b) across T2: added, subtracted (c) across EV (d) across polar

Figure 8: Heat progressing across a plane with irregular PnS configurations at $t = 0.25$. Note in (c) the extraordinary vertex (EV) valences are $n = 5, 8$, and in (d) polar valence is $n = 8$.



(a) valence 3, 5 at turns (b) $t = 0$, $n = 0$ (c) $t = 5$, $n = 500$ (d) $t = 10$, $n = 1000$ (e) $t = 50$, $n = 5000$

Figure 9: Heat progressing through a PnS tube of cube extrusions with EVs of valence 3 and 5 (see (a)) and polar end cap. The Dirichlet value on the only boundary curve (at the front left in b,c,d,e) is 100.

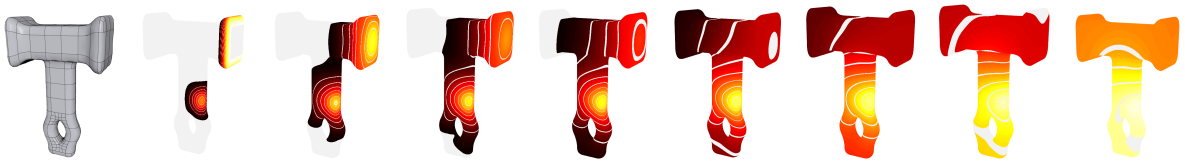
5. Time dependent second-order elliptic equations

The heat equation

$$\frac{\partial \mathtt{u}_\Omega}{\partial t} = \Delta_{\mathbf{x}} \mathtt{u}_\Omega + f, \quad u(t = 0) = u_0 \tag{8}$$

can be formulated as a time-stepping simulation: Given the final time t the equation

$$\left(2M + \frac{n}{t} K\right) \mathtt{u}_h = \left(2M - \frac{n}{t} K\right) \mathtt{u}_h^- \tag{9}$$



(a) PnS (b) $t = 0$, $n = 0$ (c) $t = 0.5$, $n = 50$ (d) $t = 1$, $n = 100$ (e) $t = 2$, $n = 200$ (f) $t = 5$, $n = 500$ (g) $t = 10$, $n = 1000$ (h) $t = 20$, $n = 2000$ (i) $t = 50$, $n = 5000$

Figure 10: Hammer with constant heat source of 100 at handle and a one time initial heat injection at the hammer head.

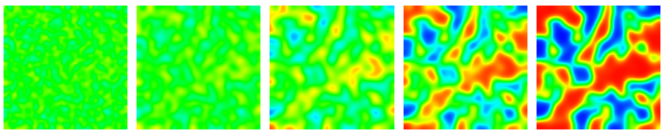


Figure 11: Evolution in time of the Allen-Cahn process on a regular grid initialized by uniform perturbation.

is solved in n steps. Here $\mathtt{u}_h^-$ is the solution at the previous time step, starting with the initial state, and the entries of M are

$$M_{ij} := \begin{cases} \sum_\alpha \int_\square \phi_i \psi_j |J_\alpha| d\square & i \notin O, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The function f enters the computation by setting the value of a collection of heat sources in $\mathtt{u}^-$ before each time step. Boundary conditions are handled as for Poisson's Equation.

5.1. Solving the Heat Equation on manifolds

We confirmed the correct solution of the heat equation (8) initially on the square plate $-1 \le x \le 1, -1 \le z \le 1$ with a Dirichlet heat source $\mathtt{u}_\Omega(-1, 0, 0) = 100$ at the left boundary and perfect von Neumann heat transfer along the other three edges: $0 = \partial \mathtt{u}_\Omega(1, 0, 0) = \partial \mathtt{u}_\Omega(0, 0, -1) = \partial \mathtt{u}_\Omega(0, 0, 1)$. We choose $f = 0$, apply 50 iterations to final time $t = 1$. The result matches linear `dolfinx` FEM using 81 vertices and 128 triangles. As before, x, y and z coordinates are parametric PnS surfaces, where we set $y = 0$ when working with bivariate functions.

Fig. 7 demonstrates the progression of heat when increasing t. Remarkably, see Fig. 8, we can substitute grids with irregular PnS configurations and match the result.

Fig. 9 shows the progression of heat, from a Dirichlet boundary condition of $\mathtt{u}_b = 100$, through a tube. Fig. 10 depicts a hammer with the contact face having nonzero *initial* conditions only but continuing heat injection at the grip so that the contact face starts out warm and cools down over time, while the point on the handle serves as a constant heat source.

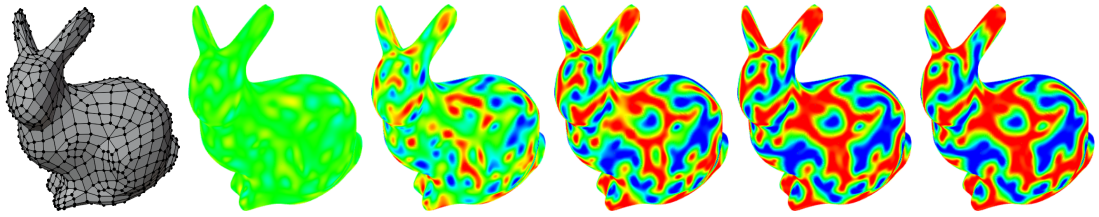


Figure 12: Evolution in uniform time intervals of the Allen-Cahn process on a complex closed surface.

5.2. Allen-Cahn progression on manifolds

The Allen-Cahn second-order time-dependent partial differential equation can be written as [14, 3]:

$$\frac{\partial \mathtt{u}}{\partial t} = \epsilon^2 \Delta \mathtt{u} - f'(\mathtt{u})$$

Fig. 11 shows the PnS solution for $f'(\mathtt{u}) = 1000(\mathtt{u} - \mathtt{u}^3)$ with random initial conditions. Allen-Cahn on a more complex PnS with $f'(\mathtt{u}) = 10000(\mathtt{u} - \mathtt{u}^3)$ is shown in Fig. 12. After initialization, for the rabbit model with 720 PnS degrees of freedom, each iteration takes about 100 milliseconds on a 16 GB RAM Intel i7-8850H 2.60 GHz CPU.

6. Fourth-order elliptic problems

The biharmonic equation

$$\text{find } q : \Omega \to \mathbb{R} : \begin{cases} \Delta^2 q & = -f & \text{in } \Omega = \mathbf{x}(\square), \\ \nabla q \cdot \mathbf{n} & = \mathtt{u}_b' & \text{on } \partial\Omega \\ q & = \mathtt{u}_b & \text{on } \partial\Omega, \end{cases},$$

has the weak form with $\mathtt{u}_h := \sum_j c_j \phi_j(\mathbf{x}^{-1})$

$$\int_\Omega \Delta_\xi u_h(\mathbf{x}^{-1}(\xi)) \Delta_\xi \psi_i(\mathbf{x}^{-1}(\xi)) d_\Omega = \int_\Omega f(\xi) \psi_i(\mathbf{x}^{-1}(\xi)) d_\Omega. \tag{11}$$

For (11) to be valid, we must choose the test functions ψ_j and their normal derivatives $\mathbf{n} \cdot \nabla \psi_j$ to vanish at the domain boundary $\partial\Omega$ so that

$$\begin{aligned} \int_\Omega \nabla \cdot \nabla(\Delta \mathtt{u}_h) \psi_i d_\Omega &= -\int_\Omega \nabla(\Delta \mathtt{u}_h) \cdot \nabla \psi_i d_\Omega + \cancelto{\psi_i(\Gamma) = 0}{\int_\Gamma \nabla(\Delta \mathtt{u}_h) \cdot \mathbf{n}\ \psi_i d_\Gamma} \\ &= \int_\Omega \Delta \mathtt{u}_h \Delta \psi_i d_\Omega - \cancelto{\nabla \psi_i \cdot \mathbf{n}(\Gamma) = 0}{\int_\Gamma \Delta \mathtt{u}_h\ \nabla \psi_i \cdot \mathbf{n}) d_\Gamma} \\ &= \int_\Omega f \psi_i d_\Omega. \end{aligned} \tag{12}$$

To change variables, we define $I_2 := J^\mathtt{t} J \in \mathbb{R}^{2\times 2}$, the first fundamental form of $\mathbf{x}$. I_2 is invertible and symmetric. With $J_{k,:} \in \mathbb{R}^{1\times 2}$ the kth row of J, we abbreviate $\mathtt{j}_k := J_{k,:} I_2^{-1} \in \mathbb{R}^{1\times 2}$. Then for $g \in \{\psi, \phi\}$,

$$H_\omega(g_i) := \Delta_\omega g_i(\mathbf{x}^{-1}) = \sum_{k=1}^{n} \mathtt{j}_k \underbrace{g_{i,\mathbf{ss}}}_{2\times 2} \mathtt{j}_k^\mathtt{t} - \underbrace{g_{i,\mathbf{s}}}_{2\times 1} \cdot \mathtt{j}_k^\mathtt{t}\ (\mathtt{j}_k \underbrace{\mathbf{x}^k_{\mathbf{ss}}}_{2\times 2} \mathtt{j}_k^\mathtt{t}) \quad \in \mathbb{R}. \tag{13}$$

Equation (11) for the ith test function is a row of the following matrix equation in terms of c_j

$$\begin{aligned} K\mathbf{c} = \mathbf{f}, \qquad \mathbf{f}_i(\mathbf{s}) &:= \sum_\alpha \int_\square (f(\mathbf{x}(\mathbf{s}))\ \psi_i(\mathbf{s}), \\ \text{with } i\text{th row } \sum_j \mathbf{c}_j K_{ij} = \mathbf{f_i}, \qquad K_{ij} &:= \sum_\alpha \int_\square H_\omega(\phi_j) H_\omega(\psi_i) |\det J_\alpha|\, \mathrm{d}\square. \end{aligned} \tag{14}$$

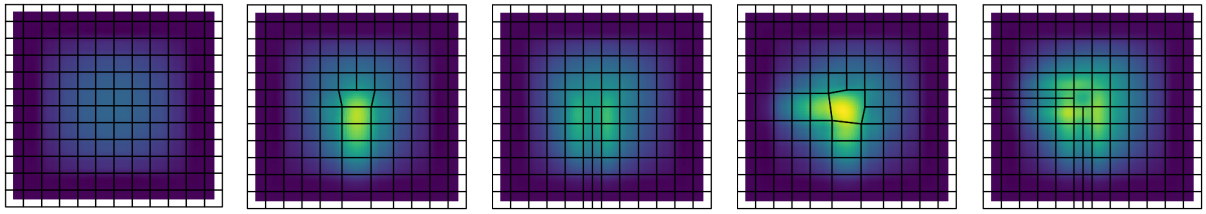
This uses the linearity of $H_\omega(g_i) := \Delta_\omega g_i(\mathbf{x}^{-1})$ in g_i.

We confirm correctness of the implementation by solving the equation

$$\Delta^2 \mathtt{u} = f, \quad f = 24(x^4 + z^4) + 288x^2z^2 - 144(x^2 + z^2) + 80 \tag{15}$$

with homogenous Dirichlet and von Neumann boundary constraints on the square $\{(x, z) : (x, z) \in [-1, 1] \times [-1, 1]\}$ and artificially introduce irregular patterns. The $\mathcal{L}^2$ error ratios for refining the subtractive T1 grid, i.e. the type shown in Fig. 13(b), are

$$(r_3, r_4, r_5, r_6) = (4.775677365, 5.2313035, 4.790458795, 4.441263758).$$



(a) Regular (b) T1 subtractive (c) T1 additive (d) T2 subtractive (e) T2 additive

Figure 13: Error in Biharmonic equation with solution $u = (x^2 - 1)^2(z^2 - 1)^2$ when subtracting, or adding partial mesh lines to form T-junctions. The maximum absolute error is 7% for the coarse $\sim 12 \times 12$ control nets. The error of the regular grid is $\sim 2\%$.

Fig. 14, shows PnS solving a bi-harmonic equation on the disk using using 1/3 as many G^1 PnS degrees of freedom (*left*) as C^0 Lagrange elements (*right*), yet yielding 1/2 the error.

(a) PnS with 89 elements (b) PnS surface (c) Lagrange: 259 elements

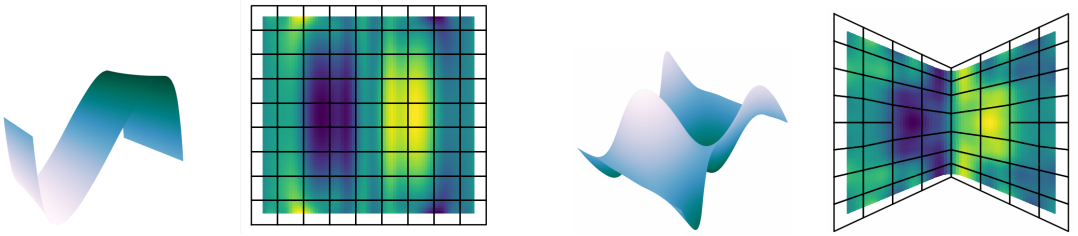
Figure 14: Biharmonic equation with $f = -2$, $u = (x^4 + y^4)/6 - (x^2 + y^2)/2$. Error is in the range of 10^{-3}, with the maximal error (*right*) twice as large as (*left*).

Next, we test the bi-harmonic equation (with known solution $u = \cos \pi x$) for *non-polynomial Dirichlet* and *non-polynomial von Neumann* boundary conditions:

$$f(x, z) = \pi^4 \cos(\pi x), \quad u_{\partial\Omega} = \cos(\pi x), \ (\nabla u)_{\partial\Omega} = \begin{pmatrix} -\pi \sin(\pi x) \\ 0 \end{pmatrix} \quad x, z \in [-1, 1].$$

Specifying the non-polynomial boundary conditions as best-approximating splines results in, as hoped for, an oscillating error Fig. 15a. The 'bow-tie' domain in Fig. 15c,d features two T1 configurations and the known solution for $\mathtt{u} = \sin(\pi x) \cos(\pi z)$.

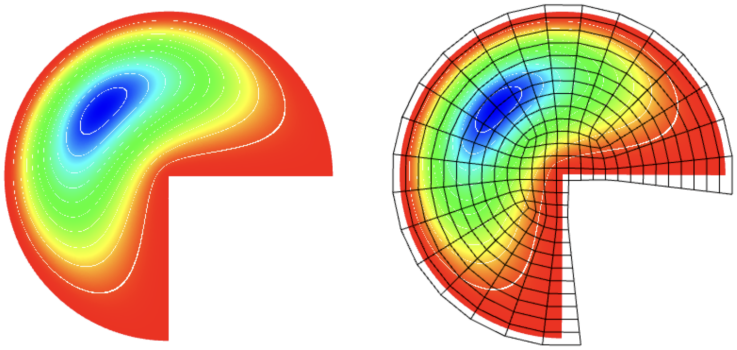
A common approach to solving fourth-order PDEs is to *split the bi-harmonic operator* and solve a system of two coupled second-order equations. If the the boundary constraints are of Dirichlet type for each second order equation, i.e. Dirichlet data and Laplace data are prescribed, this approach does not require C^1 smooth elements [26]. Also, if the constraints are Dirichlet and von Neumann, as above, the splitting approach is correct for smooth boundaries or boundaries with concave corners. However, the splitting approach can lead to incorrect output if one or more domain corners are concave, a.k.a. *re-entrant*. This is important,



(a) $u = \sin(\pi x), \Omega = [-1, 1]^2$ (b) error oscillating in the range $\pm 1.5\%$ (c) $u = \sin(\pi x)\cos(\pi z)$ (d) error oscillating in the range $\pm 5.5\%$

Figure 15: Approximating non-polynomial boundary conditions with low resolution polyhedral control net. The bow-tie (c,d) features two central T1 configurations.

because, for example, `dolfinx` demonstration projects exclusively use the operator split form of the bi-harmonic equation since the "weak form would result in the presence of second-order spatial derivatives, and the problem could not be solved using a standard Lagrange finite element basis" [25].



(a) Exact solution (b) PnS solution

Figure 16: Solution to the bi-harmonic equation [26, Fig. 4.4] whose Laplacian has a singularity at the re-entrant corner. The PnS control net consists of 344 vertices. The function range is $[-1.6, 0]$, and and the maximum error is 7.4×10^{-5}.

Besides proofs of the stated facts, [26] presents a concrete example where $u \in W_0^{2,2}(\Omega)$ and $\Delta u \notin W^{1,2}(\Omega)$ so that the unique minimizer of the fourth-order problem is not a stationary point of the decoupled system. Therefore the split operator approach can fail to reach the correct solution. The example of [26, Fig 4] is a bi-harmonic equation with clamped boundary conditions $u = 0$ and $\frac{\partial u}{\partial \nu} = 0$ on $\partial\Omega$, where Ω is the "Pac-man" domain, a disk with one quadrant removed, see Fig. 16 and the known solution is

$$u = r^{\lambda+1} + (\lambda - 5)r^5 - (\lambda - 4)r^6\Psi_\lambda(\theta), \quad \text{on the "Pac-man" domain} \tag{16}$$

where λ and Ψ_λ are the eigenvalue and eigenfunction of the eigenvalue problem:

$$\begin{cases} \Delta^2(r^{\lambda+1}\Psi(\theta)) = 0 & \text{for } r > 0 \text{ and } 0 < \theta < \omega, \\ \Psi(\theta) = \frac{\partial}{\partial \nu}\Psi(\theta) = 0 & \text{for } \theta \in \{0, \omega\}. \end{cases}$$

The PnS approach solves the equation with 344 elements for a maximal error of 7.4×10^{-5} for $u \in [-1.6, 0]$.

7. Time-dependent fourth-order elliptic problems

The Cahn-Hilliard equation

$$\frac{\partial u}{\partial t} = \Delta\left(\frac{df}{du}\right) - \lambda\Delta^2 u, \qquad f := 100u^2(1-u)^2$$

with homogeneous boundary conditions models the de-mixing of two liquids. The de-mixing has a rapid initialization and then merges clusters with slower progression. Fig. 17 illustrates the PnS solution to the

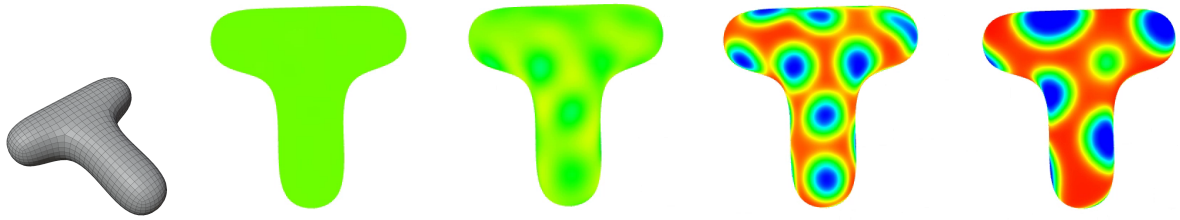


Figure 17: Evolution over time of the Cahn-Hilliard process with randomly perturbed perfect mix initial conditions on PnS control net (a). Green indicates the even mixture and transition between the distinct red and blue liquid phases.

Cahn-Hilliard problem on a closed free-form PnS surface refined to 1410 vertices. Since an explicit solution for such free-form geometry is not available, [28], we only confirmed visual similarity to the `dolfinx` solution based on splitting the operator [25]. To give an impression of the efficiency of the approach, we measured on Ubuntu 24.04.2 with Python 3.12.3 code on an Intel i7-10700 2.9 GHz CPU with 32GB RAM. Since the Cahn-Hilliard equation is non-linear, a main time factor is the number of Newton iterations required for convergence to a tolerance that is deemed sufficient for the solution to be trustworthy. The 'tee' model of Fig. 17 uses on average 4 Newton iterations and 0.3752 seconds per time step to reduce the L^2 norm of residual to $5.42e^{-15}$ (average) and $7.28e^{-15}$ (max) for a total runtime of 1000 time steps of 153 seconds. As a second instance, simulating 100 time steps on a model with 2^{10} quads and boundary constraints takes about 2 minutes (ca. 1.186743 seconds per time step when each step uses an average of 5.12 Newton iterations to obtain an average residual of $8.0e^{-16}$ and a maximal residual of $1.03e^{-15}$.)

8. Conclusion

The broad range of PDEs on manifolds – second-order, second-order time dependent, fourth-order and fourth-order time dependent – provides ample evidence of the feasibility of using geometrically smooth free-form surface spline spaces as engineering analysis spaces without additional meshing. Specifically, by allowing control net configurations for multi-sided configurations, or shedding/merging mesh lines, polyhedral-net spline can not only model smooth geometric free-form shapes, but also a corresponding PnS space for solving PDEs on these surfaces. Moreover the space is sufficiently smooth to solve hard problems with re-entrant corners [26] that demand differentiability to avoid problems due to operator-splitting and C^0 Lagrange elements. By switching to everywhere C^2 PnS, the isogeometric setup for irregular meshes can be extended to sixth order equations, such as the problems in [41, Sect 4.2]. Volumetric fourth order problems on regular partitions have been addressed, e.g. in [28], but C^1 splines over non-regular volume partitions are, at present, only available via singular parameterization, e.g. [42, 4, 21, 51], with applications to second order equations demonstrated in [12] and to fourth order equations in [66].

References

[1] Martin S. Alnaes, Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. 2014. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans. Math. Software* 40 (2014). `https://doi.org/10.1145/2566630`

[2] F.T.K. Au and Y.K. Cheung. 1993. Isoparametric Spline Finite Strip for Plane Structures. *Computers & Structures* 48 (1993), 22–32.

[3] AymaneGr and Jørgen S. Dokken. 2024. Correct implementation of AllenCahn. https://fenicsproject.discourse.group/t/correct-implementation-of-allencahn/13487 [Online; accessed 24-March-2026.

[4] Chandrajit Bajaj, Scott Schaefer, Joe Warren, Guoliang Xu, et al. 2002. A subdivision scheme for hexahedral meshes. *The visual computer* 18, 5-6 (2002), 343–356.

[5] Igor A. Baratta, Joseph P. Dean, Jørgen S. Dokken, Michal Habera, Jack S. Hale, Chris N. Richardson, Marie E. Rognes, Matthew W. Scroggs, Nathan Sime, and Garth N. Wells. 2023. DOLFINx: the next generation FEniCS problem solving environment. preprint. https://doi.org/10.5281/zenodo.10447666

[6] Klaus-Peter Beier and Yifan Chen. 1994. Highlight-line algorithm for realtime surface-quality assessment. *Computer-Aided Design* 26, 4 (1994), 268–277.

[7] Michel Bercovier and Tanya Matskewich. 2017. Smooth Bézier Surfaces over Unstructured Quadrilateral Meshes. *Lecture Notes of the Unione Matematica Italiana* (2017).

[8] Ahmed Blidia, Bernard Mourrain, and Gang Xu. 2020. Geometrically smooth spline bases for data fitting and simulation. *Computer Aided Geometric Design* 78 (March 2020), 101814.

[9] Michael J. Borden, Michael A. Scott, John A. Evans, and Thomas J. R. Hughes. 2011. Isogeometric finite element data structures based on Bézier extraction of NURBS. *Internat. J. Numer. Methods Engrg.* 87, 1-5 (2011), 15–47. https://doi.org/10.1002/nme.2968 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2968

[10] V. Braibant and C. Fleury. 1984. Shape Optimal Design using B-splines. *Computer Methods in Applied Mechanics and Engineering* 44 (1984), 247–267.

[11] Martin D. Buhmann. 2009. *Radial Basis Functions - Theory and Implementations*. Cambridge monographs on applied and computational mathematics, Vol. 12. Cambridge University Press. I–X, 1–259 pages. http://www.cambridge.org/de/academic/subjects/mathematics/numerical-analysis/radial-basis-functions-theory-and-implementations

[12] Daniel Burkhart, Bernd Hamann, and Georg Umlauf. 2010. Iso-geometric Finite Element Analysis Based on Catmull-Clark subdivision Solids. In *Computer graphics forum*, Vol. 29. Wiley Online Library, 1575–1584.

[13] Hugo Casquero, Xiaodong Wei, Deepesh Toshniwal, Angran Li, Thomas JR Hughes, Josef Kiendl, and Yongjie Jessica Zhang. 2020. Seamless integration of design and Kirchhoff–Love shell analysis using analysis-suitable unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 360 (2020), 112765.

[14] Jon Matteo Church, Zhenlin Guo, Peter K. Jimack, Anotida Madzvamuse, Keith Promislow, Brian T. R. Wetton, Steven M. Wise, and Feng Wei Yang. 2019. High Accuracy Benchmark Problems for Allen-Cahn and Cahn-Hilliard Dynamics. *Communications in Computational Physics* (2019). https://api.semanticscholar.org/CorpusID:197678568

[15] F. Cirak, M. Ortiz, and P. Schröder. 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Internat. J. Numer. Methods Engrg.* 47 (April 2000).

[16] B. (Bernardo) Cockburn, George Karniadakis, and Chi-Wang Shu. 2000. *Discontinuous Galerkin methods: theory, computation, and applications*. Vol. 11. Springer-Verlag Inc., pub-SV:adr. xi + 470 pages.

[17] Annabelle Collin, Giancarlo Sangalli, and Thomas Takacs. 2016. Analysis-suitable G1 multi-patch parametrizations for C1 isogeometric spaces. *Computer Aided Geometric Design* 47 (2016), 93–113.

[18] C. de Boor. 1978. *A Practical Guide to Splines*. Springer.

[19] C. de Boor. 1987. B-form basics. In *Geometric Modeling: Algorithms and New Trends*, G. Farin (Ed.). SIAM, 131–148.

[20] Tony DeRose, Michael Kass, and Tien Truong. 1998. Subdivision Surfaces in Character Animation. ACM Press, New York, 85–94.

[21] Alexander Dietz. 2025. *A New Approach to the Construction of Subdivision Algorithms. English Translation of the Dissertation: Ein neuer Ansatz zur Konstruktion von Subdivisionsalgorithmen*. Universitäts- und Landesbibliothek Darmstadt. `https://doi.org/10.26083/TUPRINTS-00030195`

[22] Jianhua Fan and Jörg Peters. 2011. Smooth bi-3 spline surfaces with fewest knots. *Computer-Aided Design* 43, 2 (2011), 180–187.

[23] Gerald Farin. 1988. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press.

[24] Yi-Fei Feng, Li-Yong Shen, Xin Li, Chun-Ming Yuan, and Xing Jiang. 2023. Patching non-uniform extraordinary points. *IEEE Transactions on Visualization and Computer Graphics* 30, 8 (2023), 4683–4693.

[25] FEniCS Project. 2025. Cahn-Hilliard equation – FEniCSx tutorial. `https://docs.fenicsproject.org/dolfinx/main/python/demos/demo_cahn-hilliard.html` [Online; accessed 29-March-2026.

[26] Tymofiy Gerasimov, Athanasios Stylianou, and Guido Sweers. 2012. Corners Give Problems When Decoupling Fourth Order Equations Into Second Order Systems. *SIAM J. Numer. Anal.* 50, 3 (2012), 1604–1623. `https://doi.org/10.1137/100806151` arXiv:https://doi.org/10.1137/100806151

[27] John A Gregory. 1974. Smooth interpolation without twist constraints. In *Computer aided geometric design*. Elsevier, 71–87.

[28] Héctor Gómez, Victor M. Calo, Yuri Bazilevs, and Thomas J.R. Hughes. 2008. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering* 197, 49 (2008), 4333–4352. `https://doi.org/10.1016/j.cma.2008.05.003`

[29] Fatima Hasanova, Stefan Takacs, and Thomas Takacs. 2026. Robust approximation error estimates for analysis-suitable G^1 isogeometric multi-patch discretizations. *arXiv preprint arXiv:2605.13270* (2026).

[30] Gerben J Hettinga and Jiří Kosinka. 2018. Multisided generalisations of Gregory patches. *Computer Aided Geometric Design* 62 (2018), 166–180.

[31] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. 2005. Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement. *Computer Methods in Applied Mechanics and Engineering* 194 (2005), 4135–4195.

[32] David Kamensky and Yuri Bazilevs. 2019. tIGAr: Automating isogeometric analysis with FEniCS. *Computer Methods in Applied Mechanics and Engineering* 344 (2019), 477–498.

[33] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. 2018. Construction of analysis-suitable G1 planar multi-patch parameterizations. *Computer-Aided Design* 97 (2018), 41–55.

[34] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. 2019. Isogeometric analysis with C^1 functions on planar, unstructured quadrilateral meshes. *The SMAI journal of computational mathematics* (2019), 67–86.

[35] Mario Kapl, Giancarlo Sangalli, and Thomas Takacs. 2019. An isogeometric C ^{1} subspace on unstructured multi-patch planar domains. *Computer Aided Geometric Design* 69 (2019), 55–75.

[36] Kęstutis Karčiauskas, Thien Nguyen, and Jörg Peters. 2016. Generalizing bicubic splines for modeling and IGA with irregular layout. *Computer-Aided Design* 70 (2016), 23–35.

[37] Kęstutis Karčiauskas and Jörg Peters. 2015. Smooth multi-sided blending of biquadratic splines. *Computers & Graphics* 46 (2015), 172–185.

[38] Kęstutis Karčiauskas and Jörg Peters. 2020. Low degree splines for locally quad-dominant meshes. *Computer Aided Geometric Design* 83 (2020), 1–12.

[39] Kęstutis Karčiauskas and Jörg Peters. 2020. Smooth polar caps for locally quad-dominant meshes. *Computer Aided Geometric Design* 81 (06 2020), 1–12. `https://doi.org/10.1016/j.cagd.2020.101908` PMC7343232.

[40] Ming-Jun Lai and Larry L. Schumaker. 2007. *Spline Functions on Triangulations*. Cambridge University Press. `https://doi.org/10.1017/CBO9780511721588`

[41] Yu Leng, Tianyi Hu, Sthavishtha R Bhopalam, and Hector Gomez. 2022. Numerical solutions of a gradient-elastic Kirchhoff plate model on convex and concave geometries using isogeometric analysis. *Journal of Mechanics* 38 (2022), 238–249.

[42] Ron MacCracken and Kenneth I Joy. 1996. Free-form deformations with lattices of arbitrary topology. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 181–188.

[43] Ashish Myles and Jörg Peters. 2011. C^2 splines covering polar configurations. *Computer-Aided Design* 43, 11 (2011).

[44] Thien Nguyen, Kęstutis Karčiauskas, and Jörg Peters. 2016. C^1 finite elements on non-tensor-product 2d and 3d manifolds. *Appl. Math. Comput.* 272, 1 (2016), 148–158.

[45] Thien Nguyen and Jörg Peters. 2016. Refinable C^1 spline elements for irregular quad layout. *Computer Aided Geometric Design* 43 (March 29 2016), 123–130.

[46] Joachim Nitsche. 1971. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, Vol. 36. Springer, 9–15.

[47] Qing Pan, Chong Chen, Yongjie Jessica Zhang, and Xiaofeng Yang. 2023. A novel hybrid IGA-EIEQ numerical method for the Allen–Cahn/Cahn–Hilliard equations on complex curved surfaces. *Computer Methods in Applied Mechanics and Engineering* 404 (2023), 115767.

[48] J. Peters. 1991. Smooth interpolation of a mesh of curves. *Constructive Approximation* 7 (1991), 221–247.

[49] J. Peters. 1995. C^1-surface splines. *SIAM J. Numer. Anal.* 32, 2 (1995), 645–666.

[50] J. Peters. 2019. Splines for Meshes with Irregularities. *The SMAI journal of computational mathematics* S5 (2019), 161–183.

[51] Jörg Peters. 2020. Refinable tri-variate C^1 splines for box-complexes including irregular points and irregular edges. *Computer aided geometric design* 80 (2020), 101877.

[52] Jörg Peters, Kyle Lo, and Kęstutis Karčiauskas. 2023. Algorithm 1032: Bi-cubic splines for polyhedral control nets. *ACM Tr on Math Software* 49 (March 2023). Issue 1.

[53] J. Peters and U. Reif. 2008. *Subdivision Surfaces*. Geometry and Computing, Vol. 3. Springer-Verlag, New York. i–204 pages.

[54] Ulrich Reif. 1998. TURBS—Topologically Unrestricted Rational B-Splines. *Constructive Approximation* 14 (1998), 57–77.

[55] Uwe Schramm and Walter D. Pilkey. 1993. The coupling of geometric descriptions and finite elements using NURBS - A study in shape optimization. *Finite elements in Analysis and Design* 340 (1993), 11–34.

[56] Michael A Scott, Robert N Simpson, John A Evans, Scott Lipton, Stephane PA Bordas, Thomas JR Hughes, and Thomas W Sederberg. 2013. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering* 254 (2013), 197–221.

[57] Matthew W. Scroggs, Igor A. Baratta, Chris N. Richardson, and Garth N. Wells. 2022. Basix: a runtime finite element basis evaluation library. *Journal of Open Source Software* 7, 73 (2022), 3982. `https://doi.org/10.21105/joss.03982`

[58] Matthew W. Scroggs, Jørgen S. Dokken, Chris N. Richardson, and Garth N. Wells. 2022. Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes. *ACM Trans. Math. Software* 48, 2 (2022), 18:1–18:23. `https://doi.org/10.1145/3524456`

[59] Y.K. Shyy, C. Fleury, and K. Izadpanah. 1988. Shape Optimal Design using higher-order elements. *Computer Methods in Applied Mechanics and Engineering* 71 (1988), 99–116.

[60] D Toshniwal, H Speleers, and TJR Hughes. 2017. Smooth cubic spline spaces on unstructured quadrilateral meshes with particular emphasis on extraordinary points: Geometric design and isogeometric analysis considerations. *Computer Methods in Applied Mechanics and Engineering* 327 (2017), 411–458.

[61] Tamás Várady, Péter Salvi, and Márton Vaitkus. 2024. Genuine multi-sided parametric surface patches–a survey. *Computer Aided Geometric Design* 110 (2024), 102286.

[62] David Walfisch. 2007. *Visualization for high-order discontinuous Galerkin CFD results.* Ph.D. Dissertation. Massachusetts Institute of Technology.

[63] Xiaodong Wei, Yongjie Jessica Zhang, Deepesh Toshniwal, Hendrik Speleers, Xin Li, Carla Manni, John A Evans, and Thomas JR Hughes. 2018. Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 341 (2018), 609–639.

[64] Zuowei Wen, Md Sadman Faruque, Xin Li, Xiaodong Wei, and Hugo Casquero. 2023. Isogeometric analysis using G-spline surfaces with arbitrary unstructured quadrilateral layout. *Computer Methods in Applied Mechanics and Engineering* 408 (2023), 115965.

[65] Meng Wu, Bernard Mourrain, André Galligo, and Boniface Nkonga. 2017. Hermite Type Spline Spaces over Rectangular Meshes with Complex Topological Structures. *Communications in Computational Physics* 21, 3 (2017), 835–866.

[66] Jeremy Youngquist and Jörg Peters. 2022. Solving biharmonic equations with tri-cubic C 1 splines on unstructured hex meshes. *Axioms* 11, 11 (2022), 633.