Decoupling Simulation Accuracy from Mesh Quality

TESEO SCHNEIDER, New York University, USA
YIXIN HU, New York University, USA
JÉRÉMIE DUMAS, New York University, nTopology, USA
XIFENG GAO, New York University, Florida State University, USA
DANIELE PANOZZO, New York University, USA
DENIS ZORIN, New York University, USA

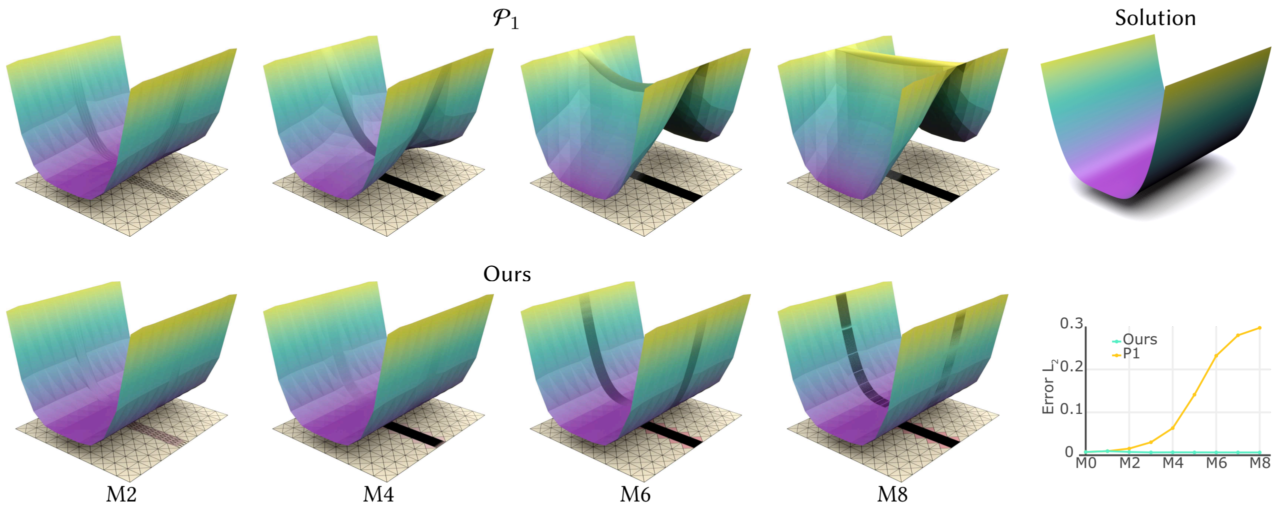

Fig. 1. The simulation accuracy is tied to mesh resolution in traditional FE methods (top, Poisson problem reconstructing a quartic function with Dirichlet boundary conditions, the exact solution is shown in the top right), and dramatically decreases as the quality of the mesh worsens, leading to inaccurate and unpredictable results (bottom right, orange). We introduce a new technique to decouple the simulation accuracy from the mesh quality (bottom), which opens the door to accurate, black box finite element analysis pipelines.

For a given PDE problem, three main factors affect the accuracy of FEM solutions: basis order, mesh resolution, and mesh element quality. The first two factors are easy to control, while controlling element shape quality is a challenge, with fundamental limitations on what can be achieved.

We propose to use p-refinement (increasing element degree) to decouple the approximation error of the finite element method from the domain mesh quality for elliptic PDEs.

Our technique produces an accurate solution even on meshes with badly shaped elements, with a slightly higher running time due to the higher cost of high-order elements. We demonstrate that it is able to automatically adapt the basis to badly shaped elements, ensuring an error consistent with high-quality meshing, without any per-mesh parameter tuning. Our construction reduces to traditional fixed-degree FEM methods on high-quality meshes with identical performance.

Our construction decreases the burden on meshing algorithms, reducing the need for often expensive mesh optimization and automatically compensates for badly shaped elements, which are present due to boundary constraints or limitations of current meshing methods. By tackling mesh generation and finite element simulation jointly, we obtain a pipeline that is both more efficient and more robust than combinations of existing state of the art meshing and FEM algorithms.

CCS Concepts: • **Computing methodologies → Modeling and simulation**;

Additional Key Words and Phrases: Finite Elements, P-Refinement, Error Estimates, Mesh Quality

ACM Reference Format:
Teseo Schneider, Yixin Hu, Jérémie Dumas, Xifeng Gao, Daniele Panozzo, and Denis Zorin. 2018. Decoupling Simulation Accuracy from Mesh Quality. *ACM Trans. Graph.* 37, 6, Article 280 (November 2018), 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

This work was partially supported by the NSF CAREER award 1652515, the NSF grant IIS-1320635, the NSF grant DMS-1436591, the NSF grant 1835712, the SNSF grant P2TIP2_175859, a gift from Adobe Research, and a gift from nTopology.

Authors' addresses: Teseo Schneider, New York University, USA, teseo.schneider@nyu.edu; Yixin Hu, New York University, USA; Jérémie Dumas, New York University, nTopology, USA; Xifeng Gao, New York University, Florida State University, USA; Daniele Panozzo, New York University, USA; Denis Zorin, New York University, USA.

1 INTRODUCTION

The numerical solution of partial differential equations is a basic building block in many algorithms in graphics, geometry processing, digital fabrication, mechanical engineering, and many other scientific disciplines. Depending on the application, different levels of accuracy are required, leading to a plethora of methods that vary from inaccurate but real-time, to very accurate and requiring many hours per integration step. Finite element method (FEM), is the most common approach to solve PDEs.

The accuracy of a FEM solution is determined by three main factors: basis approximation order (usually determined by the degree of the polynomials used), mesh resolution, and mesh quality. Increasing element order (often referred to as *p*-refinement) and increasing mesh resolution (*h*-refinement) are both common and widely used ways to increase simulation accuracy. Both can be achieved in a straightforward manner. In contrast, mesh quality is much harder to control, yet its impact on accuracy can be very substantial. Overwhelmingly, FEM basis constructions *assume* that the input mesh has elements of adequate quality.

Ensuring good element quality is a particular challenge for computer graphics applications, where coarse discretizations on fixed meshes with low-order elements (particularly sensitive to quality) are preferred, and highly complex shapes need to be handled. Even state-of-the art meshing algorithms cannot guarantee element quality without nontrivial assumptions on the input mesh surface. Moreover, fundamentally, high-quality element shape cannot be always obtained: if the boundary of the mesh needs to be preserved, some elements *have* to be of poor shape dictated by the boundary geometry (for example, the CAD model shown in Figure 2).

In this paper, we introduce a simple, yet very effective, technique to decouple the simulation accuracy of elliptic PDEs from mesh element quality. That is, we propose a method that produces a solution whose approximation error is largely independent of the presence of badly shaped elements in the input mesh. This is achieved by using an *a priori* error estimate with the dependence on the mesh quality made explicit. Our technique uses standard Lagrangian elements, and differs from commonly implemented *p*-refinement methods in one critical way: the criterion for determining the degree is based on the input *domain* mesh, and not on the solution (e.g., deformed mesh in elasticity). It is fully compatible with adaptive methods that additionally perform *h*-, *p*-, or *hp*-refinement based on an *a posteriori* error estimation in the solution.

Our approach reduces the burden of achieving high quality for *all* elements in the meshing phase, and allows us to obtain high-accuracy even on models where a high quality mesh is impossible to construct due to the presence of sharp features.

We validate our technique by testing it on two large datasets (each contains 9 809 models), demonstrating that a completely automated, controlled accuracy, analysis pipeline is enabled by combining our technique with robust mesh generators [Shewchuk 1996; Hu et al. 2018]. We run our experiments with both state-of-the-art direct and iterative solvers, demonstrating that our technique benefits both small and large scale simulation scenarios, and that the introduction of higher-order elements does not result in solver problems. Our technique is a step toward designing black-box analysis pipelines for shape optimization, which rely on running millions of simulations to compute shape derivatives, since it provides unprecedented robustness to meshing quality, at the cost of a minor increase of running times. To foster its adoption and further research, we attach a self-contained reference implementation, which we used to generate all the results reported in this paper, as well as the test datasets.

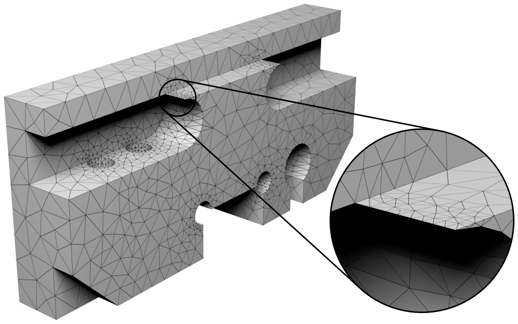

Fig. 2. A model with two adjoining feature lines, forming a zero angle: it is impossible to create a high-quality tetrahedral mesh of its interior, since mesh refinement with samples on the surface results in elements with angles approaching zero at the tip.

2 RELATED WORK

The Finite Elements Method (FEM) is, by far, the most common approach to discretize PDEs in science and engineering applications. FEM has been introduced to computer graphics by [Terzopoulos et al. 1987], and it has been widely used for physically-based simulation, geometric modeling, geometry processing, and computational fabrication.

Graphics Overview. A large fraction of computer graphics techniques rely on solving a PDE as a submodule, the most common examples being diffusion equations, elastic deformations, and their variations (elastoplasticity and viscoelasticity). We briefly review some examples of work in computer graphics that motivated our method. Due to the vast scale of applications of FEM, we do not intend this list to be comprehensive.

In geometric modeling, deformation methods heavily rely on FE, including construction of bases for efficient real-time methods [Sorkine 2005; Jacobson et al. 2011; Wang et al. 2015]. In geometry processing, FEM is used in surface/volume parametrization [Hormann et al. 2008], surface reconstruction [Kazhdan and Hoppe 2013], vector field design [Vaxman et al. 2016], quadrangulation/remeshing [Bommes et al. 2012; Ling et al. 2014], image retargeting [Kaufmann et al. 2013], and surface/volumetric texturing [Orzan et al. 2008; Takayama et al. 2010]. In computational fabrication, elasticity FEM is essential for predicting properties of printed shapes (e.g., checking structural soundness [Zhou et al. 2013]) and for shape optimization [Panetta et al. 2015; Musialski et al. 2016; Zhu et al. 2017].

Several desirable features are common to many of these applications: (1) ability to handle complex shapes robustly and automatically, and (2) maximal performance for an acceptable level of error.

As a consequence, triangular or tetrahedral elements of minimal degree (linear) are most commonly used, since they can be robustly generated [Shewchuk 1996; Hu et al. 2018] and they lead to efficient FE algorithms. However, these elements have widely known limitations, and require careful algorithmic choices to lead to stable and efficient simulations.

FEM Fundamentals. Our approach is based on classical techniques from engineering and numerical analysis literature. In particular, we use interpolation error estimates to obtain the error dependence on the shape and size of an element for a general element order. Such estimates in the general case were derived initially in [Ciarlet and Raviart 1972], with additional refinements later, most recently in [Kobayashi and Tsuchiya 2016]. These estimates show that the error is strongly dependent on the element shape. Almost all existing FE methods for tetrahedral meshing, adaptive and non-adaptive, require a bound on mesh quality to guarantee a bound on the error, which is challenging (or impossible, see Figure 2) to obtain in practice. Our approach parsimoniously uses high-order finite elements [Schwab 1998] to eliminate this requirement.

2D and 3D Meshing. Tetrahedral meshing is a central and non-fully solved problem in geometry processing and computational geometry. The inconsistency between what state-of-the art meshing methods can guarantee, and what is required by standard FEM discretizations is a motivation for our work. A number of increasingly robust and efficient meshing algorithms were developed for 2D and 3D [Shewchuk 1996; Dobrzynski 2012; Stuart et al. 2013; Si 2015; Jamin et al. 2015; Hu et al. 2018]. All these methods combine an initial mesh construction with subsequent mesh improvement, that usually dominates the running time. Despite the vast academic and industrial interest, no known 3D meshing methods guarantees a bound on the maximal edge to in-radius ratio required by FEM methods, for a general input surface. While such guarantees exist for triangle meshing [Shewchuk 1996], they still cannot be enforced if the boundary has to be preserved exactly (Figure 2).

Adaptive FEM in Graphics. Adaptive versions of FEM, which are extensively studied in the engineering literature, have a number of important applications in graphics, starting with [Wu et al. 2001] (see [Manteaux et al. 2017] for a recent overview). These methods exploit almost exclusively *h*-refinement (i.e., the size of the elements changes, not their order), and are used for problems involving large deformations, including the emergence of geometric features or discontinuities. Examples include tearing and cracking [Pfaff et al. 2014], resolving features on cloth [Simnett et al. 2009], cutting [Seiler et al. 2011], image retargeting [Kaufmann et al. 2013], elastoplasticity [Wicke et al. 2010; Piovarči et al. 2016], and viscoelasticity [Wojtan and Turk 2008]. At the same time, a sampling of recent works using FEM, [Xu and Barbič 2016; Kim et al. 2017; Liu et al. 2017; Wang et al. 2017] shows that adaptivity is commonly avoided whenever possible, as simulation on a fixed mesh offers substantial benefits, especially for interactive applications, enabling precomputation and avoiding resampling of quantities stored on the mesh (e.g., skinning weights, UV coordinates, colors).

The closest works to ours in the graphics literature are [Bargteil and Cohen 2014], [Edwards and Bridson 2014], and [Kaufmann et al. 2013] which add quadratic elements to increase the accuracy of elasticity simulations, fluid simulations, and image deformations respectively. Similarly to all other adaptive methods, their adaptivity is done *a posteriori*, that is, by using an estimate of the solution error computed after each iteration. *Basis Refinement* [Grinspun et al. 2002] can be viewed as a general form of refinement including both *h* and *p* methods.

While we present our method in the context of static meshes, it is fully compatible with adaptive refinement/coarsening, which is orthogonal to our contribution. An important distinction is that in our case elements are adapted to the *material* mesh, not to its image under deformation: while all existing methods assume that the initial mesh satisfies typical regularity assumptions, our approach lifts this assumption.

Adaptive FEM in Engineering. Different types of adaptive FEM refinement were developed, based on *a posteriori* error estimates; the original fundamentals of *p* and *hp* refinement are summarized in [Babuška and Suri 1994]. A practical parallel implementation is available, e.g., in Deal.II library [Bangerth et al. 2007; Alzetta et al. 2018]. Ten distinct methods for adaptive refinement are summarized and compared in [Mitchell and McClain 2014], using a set of test problems from [Mitchell 2013]. Invariably, these methods assume that the material mesh has sufficient regularity, and adaptation is based on either forces, boundary conditions, or a posteriori estimates of the solution. Pure *p*-refinement was demonstrated to be useful for a number of engineering problems (e.g., [Düster et al. 2003], [Franke et al. 2008]) although is known to be suboptimal for singular and close to singular solutions. Our method increases the basis degree using an *a priori* error estimate based on the geometry of the input mesh, for the purpose of decoupling simulation errors from mesh quality.

3 FINITE ELEMENT NOTATION AND SETUP

The general form of a scalar second-order elliptic PDE on a given domain Ω is

$$\mathcal{F}(x, u, \nabla u, D^2 u) = f(x), \qquad x \in \Omega,$$

subject to

$$u\big|_{\partial\Omega_D} = d(x), \qquad \text{and} \qquad \frac{\partial u}{\partial n}\bigg|_{\partial\Omega_N} = n(x),$$

where $\partial\Omega_D$ is the part of the boundary where the values of the solution u are specified (Dirichlet) and $\partial\Omega_N$ is the part boundary where the normal derivative of the function u is specified (Neumann). We assume that Ω is a polygonal/polyhedral domain, although we do not expect any significant changes for curved boundaries.

The most common PDE in this class is the Poisson's equation $\Delta u = f(x)$. A second-order elliptic PDE with a vector-valued unknown function u (e.g., elasticity) is defined similarly, but boundary conditions on derivatives may take a more complex form compared to Neumann (e.g., conditions on surface force densities).

We use the Poisson, linear elasticity, and non-linear elasticity equations for evaluation, which cover the most common elliptic equations present in graphics applications. We do not consider higher order PDEs (most importantly plates and shells), however, the

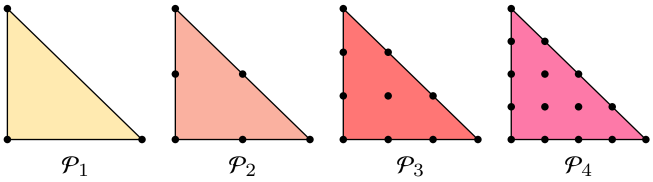

Fig. 3. Node position for the different discretization orders.

approach we describe naturally extends to these cases. We set up the notation using the Poisson equation as an example.

An essential concept in FEM theory is the space $H^m(\Omega)$ of functions on Ω with integrable m-th *weak derivatives* and its associated m-norm $\|\cdot\|_m$, see Appendix A for the definitions.

Weak Form. The FEM is based on discretizing the weak form of the PDE. For the Poisson equation , it has the form:

$$-\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f\, v, \tag{1}$$

which must holds for any *test function* v in $H^1(\Omega)$, vanishing on the boundary. For a twice-differentiable function u, this equation is obtained by integrating by parts $\int \Delta uv = \int fv$, so it is equivalent to the original equation. The weak form requires weaker assumptions on function differentiability, which makes it easier to reason about lower-order approximations (e.g., piecewise-linear) for the solution, and establish their convergence.

Basis Functions. In a finite element discretization the solution u is approximated using the basis functions φ_i, $i = 1, \ldots, n$, as $u_h = \sum_{i=1}^n u_i \varphi_i$ where u_i are the unknown coefficients.

As a basis, we use interpolatory continuous polynomial functions (Lagrange basis) defined on triangulations of Ω in 2D and tetrahedralizations in 3D. In this settings, the parameter h in u_h indicates the largest edge length present in the tessellation. The quality of approximation of a basis depends on the quality of tessellation, which can be quantified, per-element, by the *shape parameter*

$$\sigma = \frac{\rho}{h}, \tag{2}$$

where h is the maximum edge length of the element and ρ is the inscribed sphere radius. Usually, the shape parameter is considered to be bounded from below by a constant K independent of h, for admissible families of tessellations. Such families of tessellations are referred to as *regular*. In contrast to the standard approach, *we do not assume regularity of tessellations.*

For Lagrange basis, each function φ_i is associated with a *node* $n_i \in \Omega$, where it has value one, while being zero at all other nodes. As a consequence, coefficients u_i are just the values of the function u_h at the nodes. The number of nodes (and corresponding basis functions) and their position is directly correlated to the order of the basis, see Figure 3. For instance, for the piecewise linear basis ($\mathcal{P}_1$), the nodes corresponds with the mesh's vertices (Appendix B for higher degrees).

It is useful to view the basis functions as being assembled from local basis functions defined on each element separately; this approach is helpful for defining a basis with polynomial degree depending on the element. From this point of view, each element has a separate set of local nodes, and an interpolatory polynomial function is associated with each local node. Elements sharing a vertex, edge or a face, have coinciding local nodes, which are viewed as a single global node, with an associated single global basis function whose restriction to each element containing the node is equal to the local polynomial basis function. If a node is on a boundary between elements, but is not present in one of them (e.g., if a quadratic element is adjacent to a linear one, this holds for nodes at edge midlpoints) the relationship between global and local basis function coefficients has a more complex form.

Discretized Equations. By using u_h instead of u, and $\varphi_i, i = 1 \ldots n$ as test function v into (1), we obtain a $n \times n$ linear system of the form $\mathbf{Au} = \mathbf{b}$, where $\mathbf{A}$ is referred to as *stiffness matrix*, $\mathbf{b}$ is obtained from f, and $\mathbf{u}$ is the vector of unknown coefficients u_i. Note that if the PDE is not linear this discretization leads to a non-linear system, which is usually solved by using Newton's method. For the Poisson equation, the entries of $\mathbf{A}$ are given by

$$a_{i,j} = -\int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j.$$

Similarly, right-hand side entries are $b_j = \int_\Omega \varphi_j f$.

Dirichlet boundary conditions are treated as fixed values u_i at the boundary nodes. The Neumann boundary conditions are imposed by adding

$$b_j = \int_{\partial\Omega_N} \varphi_j(x) n(x) dx$$

for any node j on the boundary $\partial\Omega_N$.

If we replace φ_i with linear hat functions ($\mathcal{P}_1$ basis, Appendix B), this formula reduces to the commonly used cotangent Laplacian discrete operator [Pinkall and Polthier 1993].

Geometric Mapping. The geometric mapping g is used for computing integrals in a_{ij} and b_i. These integrals need to be computed on every element E (triangle or tetrahedron) which all have different shapes. The integration is performed element by element, with $a_{ij} = \sum_E a^E_{ij}$, where the summation is over all elements where both basis functions i and j do not vanish. Each element term a^E_{ij} is computed on a *reference* element $\bar{E}$ (in our case, a regular right-angle triangle/tetrahedron positioned at the origin, with edges along axes) through a change of variables. Let G be the Jacobian of the mapping g, then

$$a^E_{ij} = -\int_E \nabla\varphi_n \cdot \nabla\varphi_m = -\int_{E^{ref}} (G^{-T}\nabla\bar{\varphi}_n) \cdot (G^{-T}\nabla\bar{\varphi}_m) \det G,$$

where $\bar{\varphi}$ are the bases defined on the reference element $\bar{E}$. We use the simplest piecewise affine geometric map g for each element: $g(\mathbf{x}) = \mathbf{Bx} + \mathbf{c}$, where $x \in \bar{E}$, $\mathbf{B}$ is a matrix and $\mathbf{c}$ a constant, both depending on E.

Quadrature. All integrals are computed numerically by means of quadrature points and weights, which translates the integrals into weighted sums. The quadrature needs to be of sufficiently high order for the error of the method to not be affected. Although it is more expensive, we use quadratures that integrate polynomials of degree k exactly for degree k elements [Xiao and Gimbutas 2010].

4 A PRIORI ESTIMATE

One important limitation of the standard FE construction is that the basis depends on the shape and size of the elements of the discrete mesh: as shown in Figure 1, numerical and approximation errors might lead to inaccurate solutions that are not representative of the actual solutions of the underlying equation.

We propose a modification to the standard discretization to take into account the geometrical quality of the mesh *during* the basis construction, compensating for the potential error by increasing the degree of the basis. Our algorithm is divided into 3 stages: (1) we use an *a priori* error estimate, including explicit dependence on the shape of the element, to decide the polynomial degree of the basis assigned to each triangle, (2) we propagate the polynomial degrees to the adjacent elements of the mesh to ensure the desired polynomial reproduction order on the badly shaped elements, and (3) we construct a basis satisfying a necessary set of constraints on degrees of freedom to ensure C_0 continuity between elements of different orders.

These modifications are easy to add to existing FEM systems, and effectively decouple the mesh quality from the accuracy, as demonstrated in Section 6, while maintaining, for reasonable meshes, the same performance as the lowest degree elements used.

Overview. Let $\hat{k}$ be the user-specified minimal degree, which will likely be 1 (i.e., linear elements) for many graphics applications, and $\hat{h}$ be the average edge length. Our strategy aims at matching the error of every element to the error of a "perfectly shaped" element by increasing the degree k of its basis. That is, by ensuring that the error of every element is lower than the one of a regular triangle or tetrahedron of degree $\hat{k}$ and edge length $\hat{h}$. Therefore, we need an *error predictor relative to perfect shape* for the solution to determine the element's degree.

Let $\Gamma(k, E)\|u\|_{k+1}$ be an error bound for the FE solution, where $\|u\|_{k+1}$ is an H^{k+1}-norm of the exact solution (Appendix A). The error predictor $\Gamma(k, E)$ depends on the degree of the basis k and the geometry of the element E, but not on u. For obtaining k, we solve

$$\Gamma(k, E) = \Gamma(\hat{k}, \hat{E}) \tag{3}$$

where $\hat{E}$ is the regular equilateral triangle or tetrahedron with uniform side $\hat{h}$. We will now explain how we choose the error predictor Γ for linear and non-linear elliptic PDEs.

Standard A priori Error Estimate Summary. The error-estimates for degree k conforming triangular and tetrahedral finite elements for second-order linear and non-linear elliptic problems typically have the form

$$\|u - u_h\|_0 < \max_E \Gamma(k, E)\, \|u\|_{k+1} = Ch^{k+1}\|u\|_{k+1}, \tag{4}$$

where u is the exact solution, u_h is the solution of the finite element system, h is the maximal edge length in the mesh, and C is a constant independent of h, u, and u_h. Note that $\|u - u_h\|_0$ is the L_2-norm of the error. This estimate holds for sufficiently regular solutions and the order of convergence may decrease if the solution has a singularity. C is dependent on a range of other factors, including the PDE coefficients, and, most importantly for our purposes, the *shape of the elements.* A similar estimate can be derived for the error of solution gradient $\|u - u_h\|_1$, with the power of h decreased by one.

These estimates are classically obtained under the regularity assumption, that is, by assuming that the shape quality σ_E (Eq. (2)) is bounded by a constant K independent of h, for all elements E. That is, the minimal quality of the elements is bounded by K, an assumption that rarely holds on real-world meshes. We want to derive estimates avoiding this assumption and explicitly keeping their dependence on element quality.

Shape-Dependent Error Estimate. While this estimate is based on a textbook derivation (e.g., [Ciarlet 1976]) due to ubiquitous mesh regularity assumptions, typically the explicit dependence on the shape parameter σ is lost. We briefly review the steps leading to the appearance of the shape factor in the estimate below, and, for completeness, present a full derivation in a special case in Appendix C. We also note that while the convergence order of an estimate may decrease, as a consequence of solution singularities, we have experimentally observed the shape dependence to remain similar (see the singular solution experiments in Section 6). The error bound of the form (4) is usually obtained in 3 steps (see, e.g., [Ciarlet 1976]):

- *Step 1.* First, by a general estimate (Cea's lemma), the error in the FE solution is bounded by the best possible approximation error in the FE basis, in H^1 norm. The fact that the H^1 norm is used is important, as it is impossible to make a similar estimate in the L_2 norm for the PDEs we consider.
- *Step 2.* The H^1 best possible H^1 approximation error is estimated from above by the H^1 error of interpolation by the FE basis (as it is clearly not better than the best possible).
- *Step 3.* The L_2 error in the solution is bounded by the H^1 error, scaled by the element size h. This is done using the Aubin-Nitsche lemma, which requires using an interpolation error estimate one more time.

The first estimate is quite general, and involves constants dependent on the PDE, but not on the element shapes; the constants arising at the next two steps involve interpolation errors, including the shape factor. At each step, an additional factor $1/\sigma$ appears in the formula (Appendix C), leading to the following L_2 error estimate

$$\|u - u_h\|_0 < \max_E C_1 \frac{h_E^{k+1}}{\sigma_E^2} \|u\|_{k+1}, \tag{5}$$

where σ_E is the shape parameter and h_E is the maximal edge length of an element E. Figure 4 shows the relation between σ and the error for the example in Figure 1, as Eq. (5) confirms that the L_2 error is proportional to σ^2, while the H^1 error is proportional to σ.

Formula for choosing k. Based on Eq. (5) and (4), we define the *per-element* function Γ:

$$\Gamma(k, E) = C_1 \frac{h_E^{k+1}}{\sigma_E^2}. \tag{6}$$

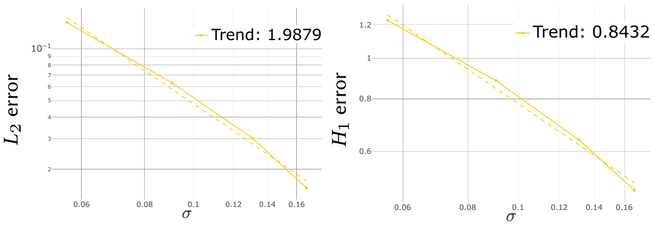

Fig. 4. Relation between σ and the L_2 (left) and H^1 (right) error for $\mathcal{P}_1$ elements.

Plugging (6) into (3) (and adding a user-provided element-independent tolerance B, we obtain:

$$C_1 \frac{h_E^{k+1}}{\sigma_E^2} \|u\|_{k+1} = B\, C_1 \frac{\hat{h}^{\hat{k}+1}}{\hat{\sigma}^2} \|u\|_{k+1},$$

where $\hat{h}$ is the average edge length of the mesh, $\hat{\sigma} = \sqrt{6}/12$ in 3D and $\hat{\sigma} = \sqrt{3}/6$ in 2D (shape parameter for a perfectly regular element). By solving for the degree k of an element E

$$k = \frac{\ln\left(B\hat{h}^{\hat{k}+1} \frac{\sigma_E^2}{\hat{\sigma}^2}\right) - \ln\, h_E}{\ln\, h_E}.$$

Comparison with Interpolation Error. Before proceeding, we compare the *solution error* (5) to the *interpolation error* of the Lagrange basis, which is commonly used to evaluate tesselation quality. For instance, [Shewchuk 2002] describes accurate formulas for both L_2 and H^1 interpolation errors for piecewise-linear elements.

This demonstrates that the gradient (equivalently H^1) error is inversely proportional to the shape parameter σ, while L_2 error depends only on h, *not* on σ.

Specifically, for an element E of degree k, the interpolation errors have the form

$$\|u - u_h\|_{0,E} \leq C_2 h_E^{k+1} \|u\|_{k+1,E}$$
$$\|u - u_h\|_{1,E} \leq C_3 \frac{h_E^k}{\sigma_E} \|u\|_{k+1,E}$$

where $\|\cdot\|_{m,E}$ is the norm restricted to the element E, and the constants are independent from the element size or shape and k. Note the contrast to (5): the solution error grows *faster* with deterioration in shape, compared to the gradient error, consistent with Figure 4. This distinction is confirmed in practice: badly shaped elements lead to large solution errors, not just large errors in solution derivatives and related quantities (e.g., stresses in elasticity). For this reason, the interpolation error is not necessarily the best way to determine the element quantity. In contrast, (5) measures the effect on shape more directly.

5 TECHNICAL AND IMPLEMENTATION DETAILS

Equipped with our a priori estimate, we briefly describe how to design a FE method that uses it to decouple simulation accuracy from element quality, by assigning degrees to elements based on their shape quality.

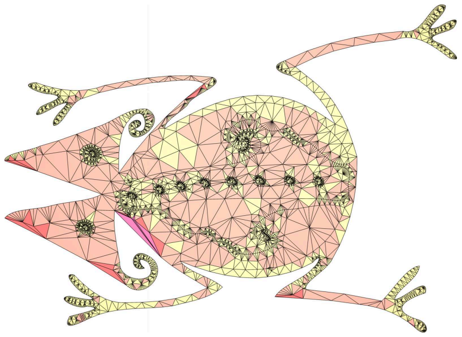

Fig. 5. Element discretization order, $\mathcal{P}_1$ in yellow, $\mathcal{P}_2$, in orange, $\mathcal{P}_3$ in red, and $\mathcal{P}_4$ in pink.

Propagation of Tags. While the degrees of freedom of $\mathcal{P}_1$ elements corresponds to corners of the mesh only, higher order elements have DOFs on edges/faces ($\mathcal{P}_2$), and on their interior ($\mathcal{P}_3$ and higher). To ensure C_0 continuity on edges/faces, it is necessary to impose constraints on some of the degrees of freedom on shared edges/faces of the elements. However, locking degrees of freedoms reduces the representation power of the basis on the higher-order element: an example is shown in Figure 6, where a $\mathcal{P}_3$ element touches a $\mathcal{P}_1$ element. As a result, to ensure C_0 continuity, the functions in the space spanned by the basis have to be linear on the shared edge of the elements, that is, not all cubic functions on $\mathcal{P}_3$ element can be reproduced. To ensure that the basis for an element E has a complete $\mathcal{P}_{k_e}$ basis, which is required for the H^1 interpolation error to work we need to ensure that all its edge/face neighbors have *at least* degree $\mathcal{P}_{k_e}$.

To enforce this condition, we do a pass over all elements, in ascending order by degree, and for each element E we navigate its edge and face neighbors and set their order to k_e if it is strictly smaller than k_e (Figure 5).

C_0 Basis Construction. To achieve the optimal convergence order of $\hat{k}+1$, three conditions needs to be satisfied [Braess 2007]: (a) polynomial reproduction up to degree $\hat{k}$, (b) quadrature accuracy, and (c) consistency. (a) is satisfied by construction in our setting since we only use $\mathcal{P}_k$ elements, with $k \geq \hat{k}$. (b) can be easily satisfied by using a proper quadrature order that integrates polynomials of degree k exactly. Consistency (c) is ensured by using conforming, that is, C_0 elements.

To guarantee continuity of the basis, we mark as *interface element* all elements which have at least one neighbor with a smaller order, and mark as *regular* the remaining ones. For regular elements, the basis and nodes construction follows the standard finite elements construction [Braess 2007].

For the interface between elements of different orders, we need to introduce a set of constraints to ensure continuity. We observe that, for the specific case of $\mathcal{P}$ basis (Appendix B), a useful property holds: any polynomial of order $k_1 < k_2$ can be viewed as a polynomial of order k_2; therefore, any function in the span of the $\mathcal{P}_{k_1}$ basis can be expressed in $\mathcal{P}_{k_2}$ basis. In particular, this holds on

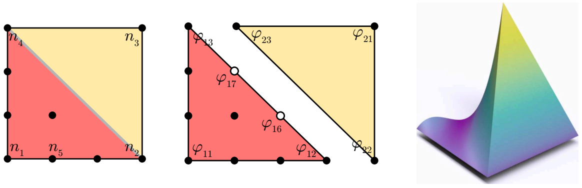

Fig. 6. Illustration of the basis construction between a cubic and linear element. The element nodes, the evaluation of the constraint, and the plot of the basis.

interface faces/edges of elements E_1 and E_2 of different orders k_1 and k_2.

As $k_1 < k_2$, some of the nodes of E_2 at the interface with E_1 are not present in E_1 (but, for the Lagrange basis, all nodes of E_1 are present in E_2) and, as a consequence, if a basis is associated with such a node, it will be discontinuous, as there is no matching polynomial local basis on the other side. For this reason, we eliminate the degrees of freedom at these nodes, and require that the unknowns at these nodes are constrained to the values of the lower-order polynomial interpolant of order k_1, computed from the unknowns at shared nodes of E_1 and E_2. This ensures continuity at the interface.

As the positions of the nodes in barycentric coordinates are independent of the element geometry, these interpolant values are linear combinations of known values with coefficients depending only on k_1 and k_2. The resulting constraints can be either added to the stiffness matrix, or used to eliminate the dependent unknowns from the system (we choose the latter in our implementation).

For the two-dimensional example in Figure 6, the constraint, e.g., for the local DOFs n_{16} at the edge (in gray) that $\mathcal{P}_3$ element E_1 shares with the $\mathcal{P}_1$ element E_2, is obtained as follows: first evaluate the three lower-order $\mathcal{P}_1$ bases

$$\varphi_{21} = x + y - 1, \quad \varphi_{22} = 1 - y, \quad \text{and} \quad \varphi_{23} = 1 - x$$

at the node n_{16} giving the constraint $v_{16} = \frac{1}{3}v_4 + \frac{2}{3}v_2$ for the local node n_{16} of the $\mathcal{P}_3$ element that we eliminate. All local degrees of freedom on E_1 other than those at n_{16} and n_{17} correspond directly to the global degrees of freedom, so the restriction of global interpolant to E_1 has the form

$$\sum_{i \neq 6,7} v_i \varphi_{1i} + \left(\frac{1}{3}v_4 + \frac{2}{3}v_2\right)\varphi_{16} + \left(\frac{1}{3}v_2 + \frac{2}{3}v_4\right)\varphi_{17}$$

Collecting the terms for each remaining unknown, we get the expressions for the restriction of the global basis functions at nodes n_2 and n_4 to E_1:

$$\phi_2^G = \left(\varphi_{12} + \frac{2}{3}\varphi_{16} + \frac{1}{3}\varphi_{17}\right)$$

and a similar expression for φ_4^G. Note that, restricted to the common edge of E_1 and E_2, these functions are linear.

We apply this construction for all interface elements in increasing order of degree: in this way, we are guaranteed to always have all the information that we need to define the continuity constraints. This holds since for an element of degree k we need to know the explicit representation of the boundary only for the neighbours of degree lower than k.

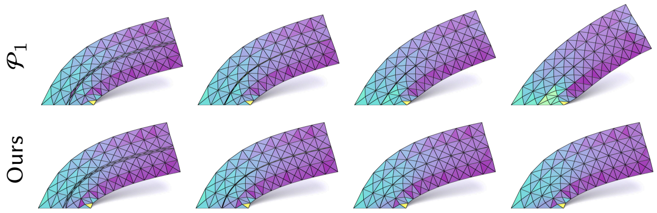

Fig. 7. Neo-Hookean elasticity example on a 2D bar with different discretizations ranging from high-quality (left) to low-quality (right, note that the strip of triangles in the middle of the bar is almost collapsed). The deformation is induced by applying a horizontal force to the top part. Standard $\mathcal{P}_1$ elements (top) exhibit a large displacement error, while our technique (bottom) produces accurate solutions (identical to the leftmost results) for all meshes. The colors represent the von Mises stress.

6 EVALUATION

We implemented our algorithm in C++, using Eigen [Guennebaud et al. 2010] for linear algebra routines and PARDISO [De Coninck et al. 2016; Verbosio et al. 2017; Kourounis et al. 2018] (figures 7, 8, 11, 21 and Table 1) and HYPRE [Falgout and Yang 2002] (figures 9, 10, 16, 17, and 20) for solving sparse linear systems. The source code of our reference implementation is available in the additional material. The experiments were performed on cluster nodes with 2 Xeon E5-2690v4 2.6GHz CPUs and 250GB memory, running 8 processes in parallel on each node, each with 60GB of reserved memory.

h-efficiency. In most cases, we measure the error in units of $\hat{h}^2$ or $\hat{h}$, eliminating the resolution-dependent part, thus allowing us to measure the efficiency of the dicretization for a fixed resolution h. This approach allows us to compare the results in a uniform and consistent way, since our datasets contain meshes of different resolution. To measure h-efficiency for the solution error and gradient error we use

$$E_{L_2}(u_h) = \frac{\|u - u_h\|_0}{\hat{h}^2} \qquad \text{and} \qquad E_{H^1}(u_h) = \frac{\|u - u_h\|_1}{\hat{h}}.$$

Most tests are done for problems with known exact solutions u, that is, the boundary conditions and the right-hand side of the problem are obtained analytically from a known function u.

Mesh Independence. Our technique effectively decouples the accuracy of the solution from the mesh used, as we demonstrate in a series of examples of increasing complexity.

(1) We bend a discretized 2D bar in Figure 7 and a 3D bar in Figure 8, by applying a tangential force to the top segment/plane. Different discretizations produce visibly different results, due to the approximation errors in the anisotropic elements in the center of the bar. When our technique is applied, the solution is similar for all meshes.

(2) We demonstrate that our method produces indistinguishable solutions for a Poisson problem on two sequences of tetrahedral meshes generated with TetWild [Hu et al. 2018] and progressively

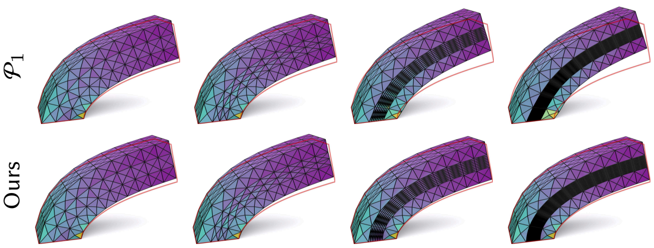

Fig. 8. Neo-Hookean elasticity example on a 3D bar, the layout is identical to Figure 7. Similarly to the 2D case, standard $\mathcal{P}_1$ elements exhibit a major displacement error (highlighted by an outline of the reference solution), while our technique is not affected by the mesh quality.

damaged (Figure 9). The meshes are damaged by adding noise of increasing strength to the vertex positions, moving them one at a time and disallowing operations that flip any tetrahedra. We also show the influence of the parameter B on the accuracy. Lower values of B introduce more high order elements, resulting in a lower error but a larger number of degrees of freedom. On the other hand, by allowing for a looser bound on acceptable error, we get fewer high order elements and higher error. We use the Franke test function [Franke 1979] as the analytic solution. Note that as the quality of the mesh improves, our method introduces fewer high-order elements, since they are not required. The assembly and construction time also decreases accordingly. While the time required by uniform elements stays constant, their errors follow the quality of the discretization (Figure 1), while ours is not affected by the bad elements.

(3) The insensitivity to mesh quality is also demonstrated for the same Poisson problem in Figure 10, by using meshes created with 5 different tetrahedral meshing algorithms to mesh 1 137 surface meshes that could be meshed by all of them: in all cases, the E_{L_2} h-efficiency is better (lower values) and stable when our technique is used, while it heavily varies with standard linear elements (top). The running time slightly increases with our technique (bottom) due to the additional DOFand increase in the quadrature order.

Large Scale Validation. To validate the robustness and effectiveness of our method, we use it to solve a Poisson problem on a large collection of 19 618 3D models (Figure 14). We created two datasets (optimized HQ and semi-optimized LQ) by running TetWild [Hu et al. 2018] on the Thingi10k dataset [Zhou and Jacobson 2016], with two different stopping criteria (maximum conformal AMIPS 3D energy reaches 10 and 300), limiting to one-hour running time and 16GB of memory.

Ideally, a PDE solver should operate in a “black-box” manner, accepting input geometries and boundary conditions with few restrictions, and producing a solution with guaranteed error. We view the combination of the TetWild mesher and our FEM discretization as a step towards such a solver, and provide an initial evaluation of the complete pipeline. We report the timings and h-efficiencies in Figure 16. The running time of our method is only slightly higher then $\mathcal{P}_1$, and the efficiency is stable for the entire dataset when our technique is used. This effect is particularly noticeable for the dataset with lower quality (right). Some examples where the difference is particularly noticeable are shown in Figure 11.

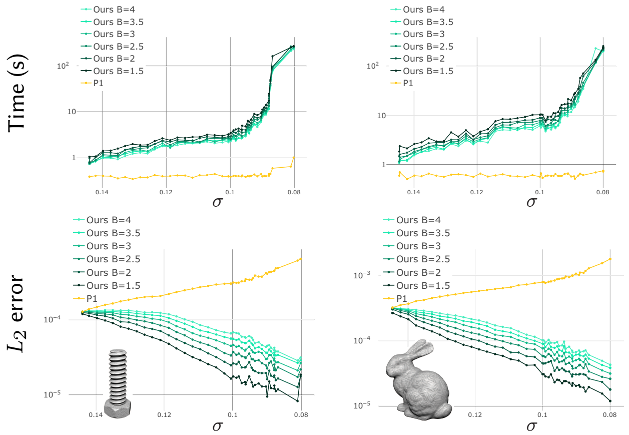

Fig. 9. Poisson solutions on a series of remeshing of the same boundary mesh with varying quality (measured by the shape parameter σ, x-axis). Plot of the running times in seconds (top) and L_2 errors (bottom) for different choices of B.

We also compare the performance of our method compared with full $\mathcal{P}_k$ elements for $k = 1, \ldots, 4$ on 300 randomly chosen high and low quality meshes (Figure 17). As expected, for the user-specified degree $\hat{k} = 1$, our method improves the efficiency for $\mathcal{P}_1$ while having a similar running times as linear elements. Similarly, higher-order elements have a better h-efficiency but the timings are significantly higher.

We show an accumulation plot of the combined timings of the meshing and analysis stages in Figure 15: the pipeline that uses a lower quality threshold for the meshing quality is significantly faster, while obtaining a comparable h-efficiency (Figure 16).

The difference in timings follows from our strategy: our criterion has the tendency of choosing higher degree for low quality meshes, while for high quality meshes almost all elements are $\mathcal{P}_1$ (Figure 12). Our method increases by 6 times the number of DOFs for the low-quality meshes dataset (on average) versus 2 times for the high-quality one (Figure 13).

Singular Solutions. Increasing the mesh density (while keeping quality constant) steadily improves the accuracy of the solution. However, the same does not hold for increasing the degree of the basis: the solution might not be smooth around singularities, and thus not well approximated by the high-order, smooth basis: in fact, it may decrease due to poor behavior of higher-order polynomials in such cases. We experimentally show that our technique is not negatively affected by singularities on the boundary, by testing it on a representative problem from the NIST collection of 2D elliptic problems, which was introduced specifically to compare adaptive algorithms [Mitchell 2013; Mitchell and McClain 2014] (Table 1).

Conditioning of the System. Increasing the degree of the basis usually has a negative effect on the condition number of the stiffness matrix [Babuska et al. 1991]. While this is not particularly relevant for direct solvers, it might affect iterative linear solvers. To assess the effect of the conditioning on our technique, we compute

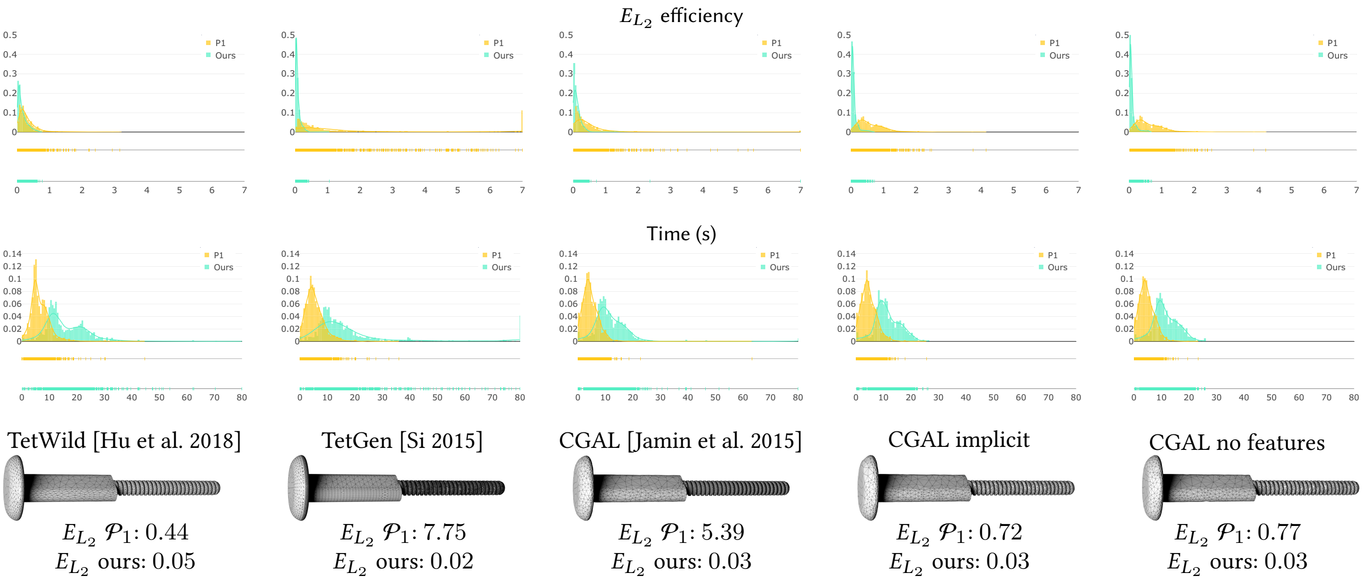

Fig. 10. E_{L_2} efficiency for the Poisson problem (top) and timings (middle) on a dataset of 1313 surfaces meshed with 5 different tetrahedral meshing algorithms.

		$\omega = 3\pi/2$			$\omega = 7\pi/4$		
	Order	L_2	L_∞	H_1	L_2	L_∞	H_1
$\sigma = 0.2$	1	4.58e-03	1.08e-02	1.05e-01	1.27e-02	2.53e-02	1.91e-01
	2	6.07e-04	3.08e-03	3.63e-02	4.26e-03	1.48e-02	1.06e-01
	3	2.12e-04	1.65e-03	2.09e-02	1.88e-03	8.02e-03	7.02e-02
	4	1.05e-04	1.05e-03	1.43e-02	1.26e-03	5.47e-03	5.79e-02
	Ours	9.87e-04	3.06e-03	4.86e-02	3.11e-03	9.37e-03	9.33e-02
$\sigma = 0.1$	1	2.40e-02	3.91e-02	2.71e-01	4.70e-02	7.40e-02	3.98e-01
	2	4.14e-03	1.17e-02	9.77e-02	1.04e-02	2.59e-02	1.67e-01
	3	1.43e-03	5.59e-03	5.67e-02	4.58e-03	1.42e-02	1.09e-01
	4	6.63e-04	3.52e-03	3.84e-02	2.93e-03	1.11e-02	8.67e-02
	Ours	8.32e-04	3.43e-03	4.56e-02	2.97e-03	1.11e-02	8.98e-02

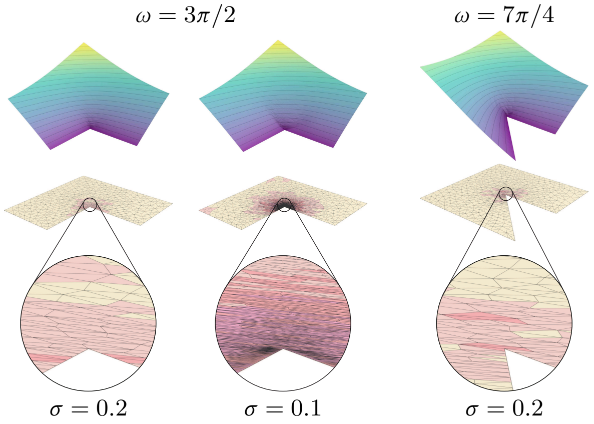

Table 1. Singularities on the boundary have little impact on the accuracy of our method, as demonstrated in the case of the reentrant corner included in the collection of 2D elliptic problems [Mitchell 2013]. The problem corresponds to Laplace's equation with Dirichlet boundary conditions on a domain with a reentrant corner of angle ω. The analytical solution used is $f(x, y) = r^{\frac{\pi}{\omega}} \sin(\frac{\pi}{\omega}\theta)$, where (r, θ) are the polar coordinates of (x, y).

the condition number of the same mesh, with low and high quality, under uniform refinement [Ong 1994] and compare it to standard linear elements (Figure 18). We remark that while our method has an higher conditioning (3.8 for the high quality and 5.4 for low quality), the trend is the same. Moreover, we used the HYPRE iterative solver (with algebraic multigrid preconditioning) for our large scale dataset (Figure 14) and compare the number of interations needed to reach a residual error of 1e-10 (Figure 19), as expected the lower quality meshes requires more iterations than standard $\mathcal{P}_1$ elements because of the large percentage of higher order elements (Figure 12). Note that the significant reduction in the number of iterations for pure linear element is due to the fact that our low quality meshes have less than half DOFs compared to the high quality ones (Figure 12 bottom). Although we noticed an increase in the number of iterations and timings for our method compared to $\mathcal{P}_1$ (likely due to the degradation of the conditioning of the system for high-order elements and the corresponding increase of degrees of freedom), it still gives a considerable timing advantage when a full pipeline including mesh optimization is considered (Figure 15). By comparing the detailed timings on 100 random meshes (Figure 20) we see that most of the overhead of our method comes from the basis construction.

This observation shows that our technique is a good fit for problems solved with algebraic multigrid techniques, and it would be

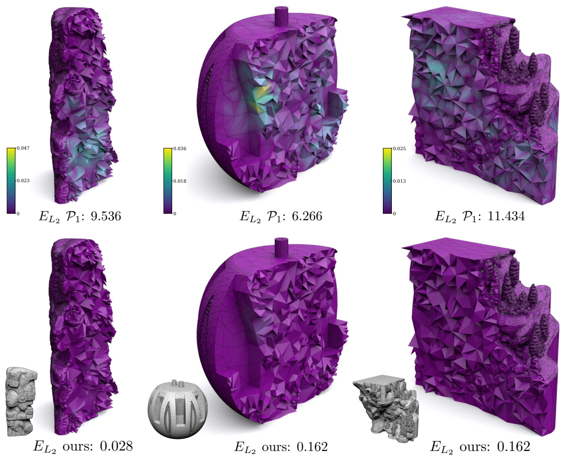
E_{L_2} $\mathcal{P}_1$: 9.536
E_{L_2} $\mathcal{P}_1$: 6.266
E_{L_2} $\mathcal{P}_1$: 11.434
E_{L_2} ours: 0.028
E_{L_2} ours: 0.162
E_{L_2} ours: 0.162

Fig. 11. Example of models where $\mathcal{P}_1$ discretization has a low h-efficiency (large values) for the Poisson equation, the color represents the actual pointwise errors.

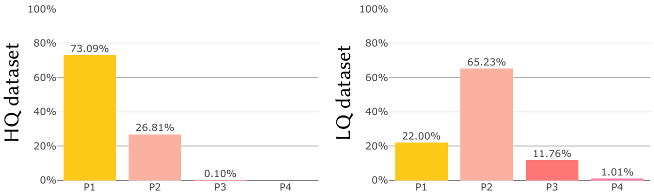

Fig. 12. Distribution of element's degrees for our 3D dataset with high quality (left) and low quality (right).

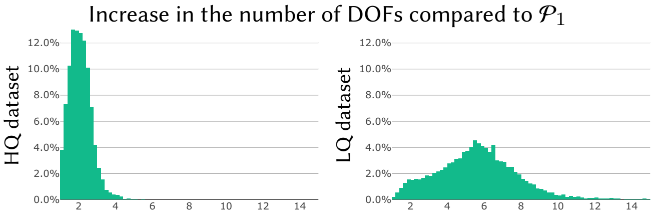

Fig. 13. Histograms of the ratio of the number of degree of freedom of our method over linear $\mathcal{P}_1$ elements for our 3D dataset with high quality (left) and low quality (right).

interesting to conduct similar tests with geometric multigrid techniques which are used in graphics for both unstructured [Aksoylu et al. 2005; Shi et al. 2006; Botsch et al. 2006; Jakob et al. 2015] and structured meshes [Zhu et al. 2010; McAdams et al. 2011].

Linear Elasticity. In addition to the non-linear elasticity simulations shown in figures 7 and 8, we show a sampler of *linear* elasticity simulations with Dirichlet boundary conditions in Figure 21. Note that, the h-accuracy drops to negligible levels with our technique, whereas a standard $\mathcal{P}_1$ discretization would not be usable for these models.

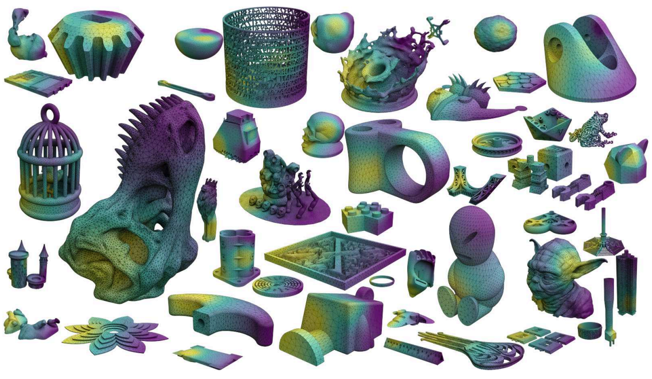

Fig. 14. Sample models from our optimized 3D dataset composed of 9 809 tetrahedral meshes generated with TetWild [Hu et al. 2018] using default parameters, the colors represent the Dirichlet boundary conditions used to solve the Poisson problem on the interior.

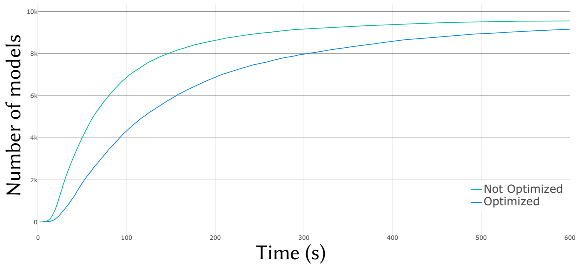

Fig. 15. Accumulation plot of the timings for the combined meshing and analysis pipeline on two datasets. The pipeline is considerably faster if we stop the mesh optimization early (semi-optimized), compared to a complete quality optimization (optimized). With our technique, the h-efficiency of the solution is not affected significantly, providing a simple and effective way of speeding up the computation of the FE solution.

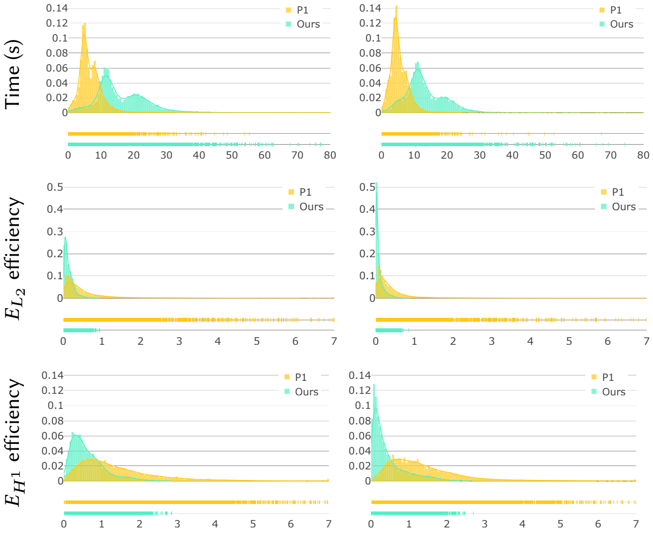

Fig. 16. Histograms of timings (top) and h-efficiency (middle, bottom) for solving a Poisson problem on our 3D dataset with high quality (left) and low quality (right).

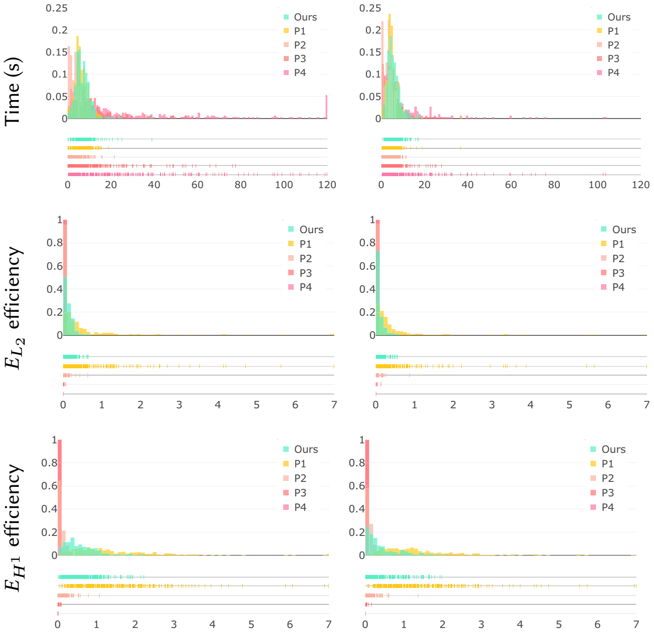

Fig. 17. Histograms of timings (top) and h-efficiency (middle, bottom) for solving a Poisson problem on 300 randomly picked meshes with high quality (left) and low quality (right).

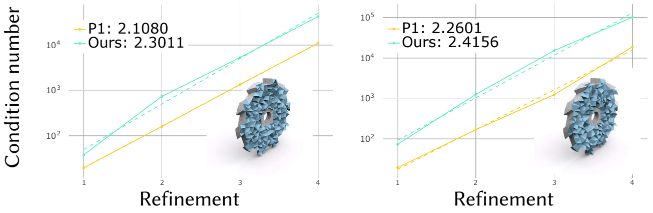

Fig. 18. Evolution of the condition number for different level of refinement of our method compared with linear elements for one model (inset) with high quality min $\sigma = 5.6$e-2 (left) and low quality min $\sigma = 2.4$e-2 (right).

7 CONCLUDING REMARKS

We introduced a simple and effective technique to decouple the simulation error from mesh quality with *a priori* analysis of the mesh geometry. The technique has two downsides: (1) it increases the computational cost, and (2) it requires a somewhat more complex basis construction to set up the stiffness matrix and rhs. We argue that the minor increase in runtime is well worth the benefit of not having to consider or optimize mesh quality, especially when only a few solves need to be done for a specific mesh. To ameliorate the second downside, we provide a self-contained, open-source, C++ reference implementation of our technique, which will allow practitioners and researchers to easily integrate our basis into existing FEM systems (https://github.com/polyfem/polyfem).

An interesting venue for future work is the integration of our technique with existing (and complementary) *hp*-refinement methods, to concentrate DOFs based on a posteriori error estimate and on the right-hand side. Similarly, it would be interesting to explore

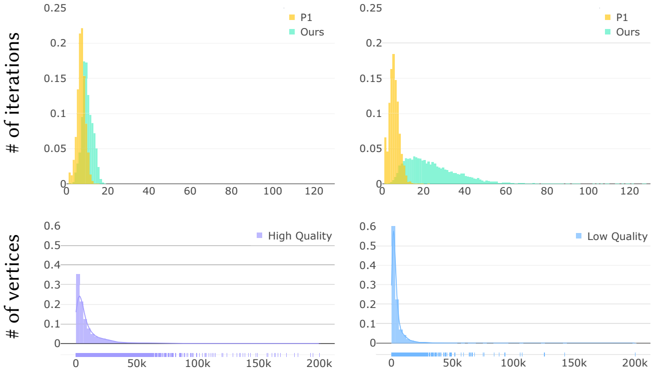

Fig. 19. Histogram of the number of Hypre iterations to reach a residual error of 1e-10 (top) of our method compared with $\mathcal{P}_1$ elements for our large data set with high quality (left) and low quality (right). The bottom row shows the number of vertices of the two data set.

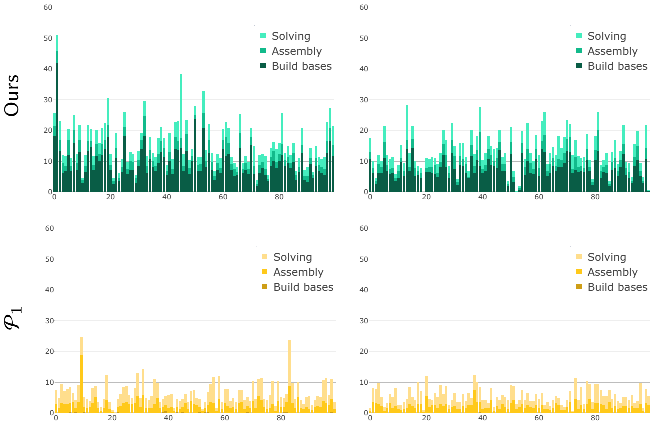

Fig. 20. Timings of our method compared with linear $\mathcal{P}_1$ elements for 100 randomly chosen model with high quality (left) and low quality (right).

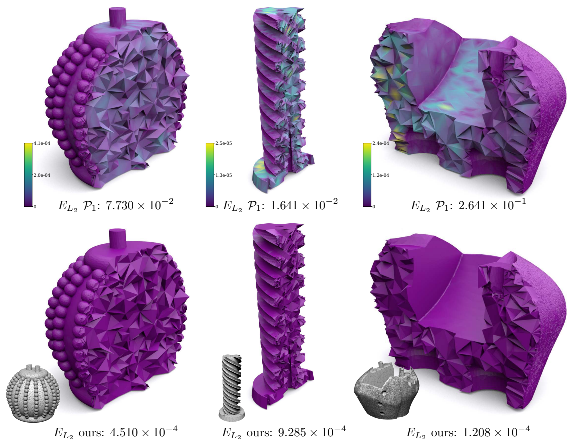
E_{L_2} $\mathcal{P}_1$: 7.730×10^{-2}
E_{L_2} $\mathcal{P}_1$: 1.641×10^{-2}
E_{L_2} $\mathcal{P}_1$: 2.641×10^{-1}
E_{L_2} ours: 4.510×10^{-4}
E_{L_2} ours: 9.285×10^{-4}
E_{L_2} ours: 1.208×10^{-4}

Fig. 21. Examples of linear elasticity simulations with an analytic solution with Dirichlet boundary conditions, the color represents the actual pointwise errors.

the use of our technique in time-dependent simulations where the mesh is deformed at each iteration, dynamically changing the elements' quality, and whether it can be helpful to increase the time step. Despite the fact that most graphics (and engineering) applications uses elliptic PDEs, we plan extend our estimate to a larger class of non-elliptic PDEs, such as wave equations.

We believe that the benefits of our technique heavily outweigh the downsides: we expect our contribution to have a large impact not only on the design of black-box simulation pipelines, but also on improving the quality and reliability of every graphics algorithm relying on solving a discrete PDE.

ACKNOWLEDGEMENTS

We are grateful to Wenzel Jakob for the Mitsuba renderer, and to NYU HPC staff for providing computing cluster service.

REFERENCES

B. Aksoylu, A. Khodakovsky, and P. Schröder. 2005. Multilevel Solvers for Unstructured Surface Meshes. *SIAM J. Sci. Comput.* 26, 4 (April 2005), 1146–1165.

G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. 2018. The deal.II Library, Version 9.0. *Journal of Numerical Mathematics* (2018).

I. Babuska, A. Craig, J. Mandel, and J. Pitkaranta. 1991. Efficient Preconditioning for the p-Version Finite Element Method in Two Dimensions. *SIAM J. Numer. Anal.* 28, 3 (1991), 624–661.

I. Babuška and M. Suri. 1994. The p and h-p versions of the finite element method, basic principles and properties. *SIAM review* 36, 4 (1994), 578–632.

W. Bangerth, R. Hartmann, and G. Kanschat. 2007. deal.II – a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)* 33, 4 (2007), 24.

A. W. Bargteil and E. Cohen. 2014. Animation of deformable bodies with quadratic Bézier finite elements. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 27.

D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. 2012. State of the Art in Quad Meshing. In *Eurographics STARS*. Cagliari, Italy.

M. Botsch, M. Pauly, M. Gross, and L. Kobbelt. 2006. PriMo: Coupled Prisms for Intuitive Surface Modeling. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP '06)*. Eurographics Association, 11–20.

D. Braess. 2007. *Finite elements: Theory, fast solvers, and applications in solid mechanics.* Cambridge University Press.

P. G. Ciarlet. 1976. *Numerical analysis of the finite element method.* Vol. 59. Presses de l'Université de Montréal.

P. G. Ciarlet and P. Raviart. 1972. General Lagrange and Hermite interpolation in Rn with applications to finite element methods. *Archive for Rational Mechanics and Analysis* 46, 3 (1972), 177–199.

A. De Coninck, B. De Baets, D. Kourounis, F. Verbosio, O. Schenk, S. Maenhout, and J. Fostier. 2016. Needles: Toward Large-Scale Genomic Prediction with Marker-by-Environment Interaction. 203, 1 (2016), 543–555.

C. Dobrzynski. 2012. MMG3D: user guide.

A. Düster, S. Hartmann, and E. Rank. 2003. p-FEM applied to finite isotropic hyperelastic bodies. *Computer Methods in Applied Mechanics and Engineering* 192, 47–48 (2003), 5147–5166.

E. Edwards and R. Bridson. 2014. Detailed water with coarse grids: combining surface meshes and adaptive discontinuous Galerkin. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 136.

R. D. Falgout and U. M. Yang. 2002. hypre: A Library of High Performance Preconditioners. In *Computational Science — ICCS 2002*, Peter M. A. Sloot, Alfons G. Hoekstra, C. J. Kenneth Tan, and Jack J. Dongarra (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 632–641.

D. Franke, A. Düster, and E. Rank. 2008. The p-version of the FEM for computational contact mechanics. *PAMM* 8, 1 (2008), 10271–10272.

R. Franke. 1979. A Critical Comparison of Some Methods for Interpolation of Scattered Data.

E. Grinspun, P. Krysl, and P. Schröder. 2002. CHARMS: a simple framework for adaptive simulation. *ACM transactions on graphics (TOG)* 21, 3 (2002), 281–290.

G. Guennebaud, B. Jacob, et al. 2010. Eigen v3.

K. Hormann, K. Polthier, and A. Sheffer. 2008. Mesh Parameterization: Theory and Practice. In *ACM SIGGRAPH ASIA 2008 Courses (SIGGRAPH Asia '08)*. ACM, New York, NY, USA, Article 12, 87 pages.

Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, and D. Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages.

A. Jacobson, I. Baran, J. Popovic, and O. Sorkine. 2011. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.* 30, 4 (2011), 78–1.

W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung. 2015. Instant Field-Aligned Meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA)* 34, 6 (Nov. 2015).

C. Jamin, P. Alliez, M. Yvinec, and J.-D. Boissonnat. 2015. CGALmesh: a generic framework for delaunay mesh generation. *ACM Transactions on Mathematical Software (TOMS)* 41, 4 (2015), 23.

P. Kaufmann, O. Wang, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, and M. Gross. 2013. Finite Element Image Warping. *Computer Graphics Forum (proceedings of EUROGRAPHICS)* 32, 2 (2013), 31–39.

M. Kazhdan and H. Hoppe. 2013. Screened Poisson Surface Reconstruction. *ACM Trans. Graph.* 32, 3, Article 29 (July 2013), 13 pages.

M. Kim, G. Pons-Moll, S. Pujades, S. Bang, J. Kim, M. Black, and S.-H. Lee. 2017. Data-Driven Physics for Human Soft Tissue Animation. *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 36, 4 (2017).

K. Kobayashi and T. Tsuchiya. 2016. Error analysis of Lagrange interpolation on tetrahedrons. *arXiv preprint arXiv:1606.03918* (2016).

D. Kourounis, A. Fuchs, and O. Schenk. 2018. Towards the Next Generation of Multiperiod Optimal Power Flow Solvers. *IEEE Transactions on Power Systems* PP, 99 (2018), 1–10.

R. Ling, J. Huang, B. Jüttler, F. Sun, H. Bao, and W. Wang. 2014. Spectral quadrangulation with feature curve alignment and element size control. *ACM Transactions on Graphics (TOG)* 34, 1 (2014), 11.

T. Liu, S. Bouaziz, and L. Kavan. 2017. Quasi-Newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 23.

P.-L. Manteaux, C. Wojtan, R. Narain, S. Redon, F. Faure, and M.-P. Cani. 2017. Adaptive physically based models in computer graphics. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 312–337.

A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. In *ACM SIGGRAPH 2011 Papers (SIGGRAPH '11)*. ACM, New York, NY, USA, Article 37, 12 pages.

W. F. Mitchell. 2013. A Collection of 2D Elliptic Problems for Testing Adaptive Grid Refinement Algorithms. *Appl. Math. Comput.* 220 (Sept. 2013), 350–364.

W. F. Mitchell and M. A. McClain. 2014. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software (TOMS)* 41, 1 (2014), 2.

P. Musialski, C. Hafner, F. Rist, M. Birsak, M. Wimmer, and L. Kobbelt. 2016. Non-linear shape optimization using local subspace projections. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 87.

M. Ong. 1994. Uniform Refinement of a Tetrahedron. *SIAM Journal on Scientific Computing* 15, 5 (1994), 1134–1144.

A. Orzan, A. Bousseau, H. Winnemöller, P. Barla, J. Thollot, and D. Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-Shaded Images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, Vol. 27.

J. Panetta, Q. Zhou, L. Malomo, N. Pietroni, P. Cignoni, and D. Zorin. 2015. Elastic textures for additive fabrication. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 135.

T. Pfaff, R. Narain, J. M. De Joya, and J. F. O'Brien. 2014. Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 110.

U. Pinkall and K. Polthier. 1993. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics* 2 (1993), 15–36.

M. Piovarči, D. I. Levin, J. Rebello, D. Chen, R. Ďurikovič, H. Pfister, W. Matusik, and P. Didyk. 2016. An interaction-aware, perceptual model for non-linear elastic objects. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 55.

C. Schwab. 1998. *p-and hp-finite element methods: Theory and applications in solid and fluid mechanics.* Oxford University Press.

M. Seiler, D. Steinemann, J. Spillmann, and M. Harders. 2011. Robust interactive cutting based on an adaptive octree simulation mesh. *The Visual Computer* 27, 6–8 (2011), 519–529.

J. Shewchuk. 2002. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley* 73 (2002), 137.

J. R. Shewchuk. 1996. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied computational geometry towards geometric engineering*. Springer, 203–222.

L. Shi, Y. Yu, N. Bell, and W.-W. Feng. 2006. A Fast Multigrid Algorithm for Mesh Deformation. *ACM Trans. Graph.* 25, 3 (July 2006), 1108–1117.

H. Si. 2015. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)* 41, 2 (2015), 11.

T. J. Simnett, S. D. Laycock, and A. M. Day. 2009. An Edge-based Approach to Adaptively Refining a Mesh for Cloth Deformation.. In *TPCG*. 77–84.

O. Sorkine. 2005. Laplacian Mesh Processing. In *Eurographics 2005 - State of the Art Reports*, Yiorgos Chrysanthou and Marcus Magnor (Eds.). The Eurographics Association.

A. Stuart, J. Levine, B. Jones, and A. Bargteil. 2013. Automatic Construction of Coarse, High-Quality Tetrahedralizations that Enclose and Approximate Surfaces for Animation. In *Proceedings of the ACM SIGGRAPH Conference on Motion in Games*.

K. Takayama, O. Sorkine, A. Nealen, and T. Igarashi. 2010. Volumetric modeling with diffusion surfaces. In *ACM Transactions on Graphics (TOG)*, Vol. 29. ACM, 180.

D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. 1987. Elastically deformable models. *ACM Siggraph Computer Graphics* 21, 4 (1987), 205–214.

A. Vaxman, M. Campen, O. Diamanti, D. Bommes, K. Hildebrandt, M. Ben-Chen, and D. Panozzo. 2016. Directional Field Synthesis, Design, and Processing. In *SIGGRAPH ASIA 2016 Courses (SA '16)*. ACM, New York, NY, USA, Article 15, 30 pages.

F. Verbosio, A. D. Coninck, D. Kourounis, and O. Schenk. 2017. Enhancing the scalability of selected inversion factorization algorithms in genomic prediction. *Journal of Computational Science* 22, Supplement C (2017), 99–108.

J.-H. Wang, R. Setaluri, D. L. James, and D. K. Pai. 2017. Bounce maps: an improved restitution model for real-time rigid-body impact. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 150.

Y. Wang, A. Jacobson, J. Barbič, and L. Kavan. 2015. Linear subspace design for real-time shape deformation. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 57.

M. Wicke, D. Ritchie, B. M. Klingner, S. Burke, J. R. Shewchuk, and J. F. O'Brien. 2010. Dynamic local remeshing for elastoplastic simulation. In *ACM Transactions on graphics (TOG)*, Vol. 29. ACM, 49.

C. Wojtan and G. Turk. 2008. Fast viscoelastic behavior with thin features. *ACM transactions on graphics (TOG)* 27, 3 (2008), 47.

X. Wu, M. S. Downes, T. Goktekin, and F. Tendick. 2001. Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. In *Computer Graphics Forum*, Vol. 20. Wiley Online Library, 349–358.

H. Xiao and Z. Gimbutas. 2010. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications* 59, 2 (2010), 663–676.

H. Xu and J. Barbič. 2016. Pose-Space Subspace Dynamics. *ACM Trans. on Graphics (SIGGRAPH 2016)* 35, 4 (2016).

Q. Zhou and A. Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

Q. Zhou, J. Panetta, and D. Zorin. 2013. Worst-case structural analysis. *ACM Trans. Graph.* 32, 4 (2013), 137–1.

B. Zhu, M. Skouras, D. Chen, and W. Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 4, Article 120b (July 2017).

Y. Zhu, E. Sifakis, J. Teran, and A. Brandt. 2010. An Efficient Multigrid Method for the Simulation of High-resolution Elastic Solids. *ACM Trans. Graph.* 29, 2, Article 16 (April 2010), 18 pages.

A WEAK DERIVATIVE AND H^m NORM

Let $f(x_1, \dots x_d) \in H^m$ be a function defined on Ω, g is the *weak derivative* of f with respect to the variables $x_1, \dots x_m$, $m \leq d$ if

$$\int_\Omega f \frac{\partial^m v}{\partial x_1 \dots x_m} = (-1)^m \int_\Omega g v$$

for any function $v \in C_\infty$ vanishing on the boundary of Ω.

Let $\alpha = (\alpha_1, \dots \alpha_m)$ and $|\alpha| = \sum_{i=1}^m \alpha_i$, then for a function $f \in H^m$, the norm $\|f\|_m$ is defined as

$$\|f\|_m = \sum_{|\alpha| \leq m} \|\partial_x^\alpha f\|_{L_2},$$

where $\| \cdot \|_{L_2} = (\int_\Omega | \cdot |^2)^{1/2}$ is the standard L_2-norm and $\partial_x^\alpha f = \partial_{x_1}^{\alpha^1} \dots \partial_{x_d}^{\alpha^d} f$, with $d = 2$ in 2D and $d = 3$ in 3D. For instance, H^0 is the space of integrable functions, with the usual L_2-norm $\|f\|_{L_2} = \|f\|_0$.

B $\mathcal{P}_k$ BASIS

We use Lagrange polynomials to interpolate between the nodes in triangular and tetrahedral elements. The coefficients of those polynomials can be pre-computed beforehand for a fixed degree by solving a linear system formed by the Vandermond matrix of each polynomial basis evaluated at the nodes of the reference element. For convenience, we provide the explicit formulation for $\mathcal{P}_1$ and $\mathcal{P}_2$ both in 2D and 3D, while higher-order bases are provided in the source code.

$\mathcal{P}_1$ in 2D:

$$\varphi_1 = 1 - x - y$$
$$\varphi_2 = x$$
$$\varphi_3 = y$$

$\mathcal{P}_2$ in 2D:

$$\varphi_1 = (x + y - 1)(2x + 2y - 1)$$
$$\varphi_2 = x(2x - 1)$$
$$\varphi_3 = y(2y - 1)$$
$$\varphi_4 = -4x(x + y - 1)$$
$$\varphi_5 = 4xy$$
$$\varphi_6 = -4y(x + y - 1)$$

$\mathcal{P}_1$ in 3D:

$$\varphi_1 = 1 - x - y - z$$
$$\varphi_2 = x$$
$$\varphi_3 = y$$
$$\varphi_4 = z$$

$\mathcal{P}_2$ in 3D:

$$\varphi_1 = (x + y + z - 1)(2x + 2y + 2z - 1)$$
$$\varphi_2 = x(2x - 1)$$
$$\varphi_3 = y(2y - 1)$$
$$\varphi_4 = z(2z - 1)$$
$$\varphi_5 = -4x(x + y + z - 1)$$
$$\varphi_6 = 4xy$$
$$\varphi_7 = -4y(x + y + z - 1)$$
$$\varphi_8 = -4z(x + y + z - 1)$$
$$\varphi_9 = 4xz$$
$$\varphi_{10} = 4yz$$

C L_2-ERROR ESTIMATE FOR POISSON EQUATION

To make the origin of the shape parameter dependence in Γ, we include a short derivation of the estimate in the special case of the Poisson equation with homogeneous Dirichlet boundary conditions, $u|_{\partial\Omega} = 0$. We emphasize that this derivation is not PDE-specific, and effectively the same steps, in a generalized and hence longer form are followed in the general case.

An important assumption required for this derivation is H^2-regularity of the domain, that is $f \in L_2$, $u \in H^2 \cap H_0^1$ and $\|u\|_2 \leq C_1(\Omega)\|f\|_0$.

We define the inner products

$$\langle v, w \rangle|_0 = \int_\Omega v\, w \qquad \text{and} \qquad \langle v, w \rangle|_1 = \int_\Omega \nabla v \cdot \nabla w,$$

and denote by $V^h = \operatorname{span} \varphi_i$ the finite dimensional space spanned by the basis on the triangulation with size parameter h and $|f|_1 =$

$\|\nabla f\|_0$. Finally, $I_h u$ is the finite element interpolant of u, obtained by sampling u at the nodes.

With this notation, the PDE and its finite element discretization can be written as, find u such that

$$\langle u, v\rangle|_1 = \langle f, v\rangle|_0, \text{ for any } v \in H^1;$$
$$\langle u_h, v_h\rangle|_1 = \langle f, v_h\rangle|_0 \text{ for any } v_h \in V_h.$$

Let $e = u - u_h$ be the approximation error, by using $v = v_h$ and subtracting two equations we obtain $\langle e, v_h\rangle|_1 = 0$ for any $v_h \in V_h$ which is the *error orthogonality property.*

(1) Estimate for $|e|_1$. For an arbitrary function in $v_h \in V^h$,

$$|e|_1^2 = \langle e, u - u_h\rangle|_1 = \langle e, u - v_h\rangle|_1 + \langle e, v_h - u_h\rangle|_1.$$

The last term vanishes, by error orthogonality, as $v_h - u_h \in V_h$. By Cauchy-Schwarz inequality,

$$|e|_1^2 = \langle e, u - v_h\rangle|_1 \leq |e|_1 |u - v_h|_1,$$

for any $v_h \in V_h$. Therefore,

$$|e|_1 \leq \inf_{v_h \in V_h} |u - v_h|_1 \leq |u - I_h u|_1,$$

for the particular choice $v_h = I_h u \in V_h$.

(2) Apply interpolation gradient error. On an element E,

$$|u - I_h u|_{1,E} \leq C' \frac{h_E^2}{\rho_E} \|u\|_{2,E} = C' \frac{h_E}{\sigma_E} \|u\|_{2,E},$$

where h_E is the maximal edge length and ρ_E is inscribed circle or sphere radius ([Ciarlet and Raviart 1972]). Therefore, the total interpolation error is bounded by $C' \max_E \frac{h_E}{\sigma_E} \|u\|_2$. This yields to the H_1 error estimate

$$|e|_1 \leq C' \max_E \frac{h_E}{\sigma_E} \|u\|_2.$$

(3) L_2 estimate via auxiliary Poisson problem. Let us consider the auxiliary system $\langle w, v\rangle|_1 = \langle e, v\rangle|_0$ for any $v \in H^1$, that is, the Poisson equation with the right-hand side equal to the error.

Applying the definition of w with $v = e$ followed by error orthogonality, we get

$$\|e\|_0^2 = \langle e, e\rangle|_0 = \langle w, e\rangle|_1 = \langle w - I_h w, e\rangle|_1 \leq |w - I_h w|_1 |e|_1.$$

Using interpolation error the second time, we get

$$\|e\|_0^2 \leq C' \max_E \frac{h_E}{\sigma_E} \|w\|_{2,E}\, |e|_1. \tag{7}$$

Finally, by H^2 regularity, $\|w\|_2 < C_1(\Omega)\|e\|_0$, as e is the right-hand side. Substituting this estimate and the estimate for $|e|_1$ into (7), and cancelling the factor $|e|_0$ we obtain

$$\|e\|_0 \leq C_1 \left(C' \max_E \frac{h_E}{\sigma_E}\right)^2 \|u\|_2,$$

which is the expected estimate.