Subdivision Exterior Calculus for Geometry Processing

Fernando de Goes
Pixar Animation Studios

Mathieu Desbrun
Caltech

Mark Meyer
Pixar Animation Studios

Tony DeRose
Pixar Animation Studios

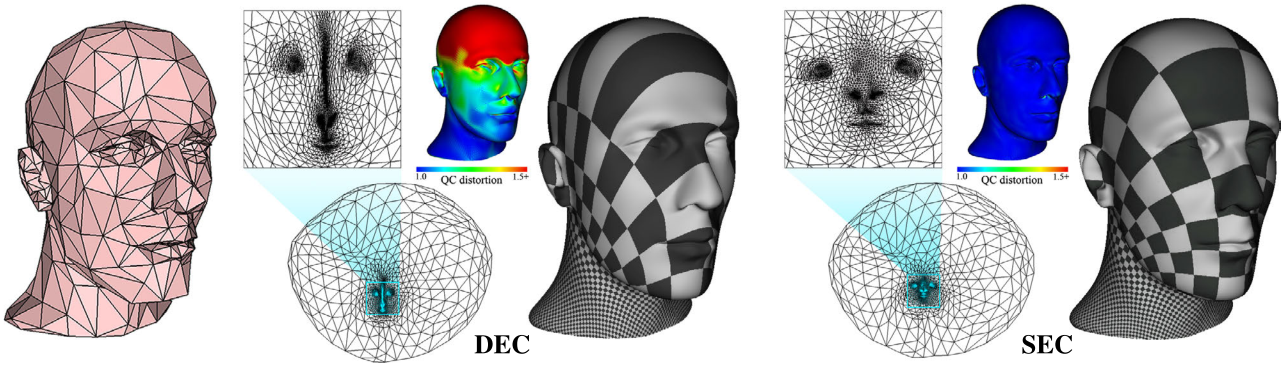

Figure 1: ***Subdivision Exterior Calculus (SEC).*** *We introduce a new technique to perform geometry processing applications on subdivision surfaces by extending Discrete Exterior Calculus (DEC) from the polygonal to the subdivision setting. With the preassemble of a few operators on the control mesh, SEC outperforms DEC in terms of numerics with only minor computational overhead. For instance, while the spectral conformal parameterization [Mullen et al. 2008] of the control mesh of the mannequin head (left) results in large quasi-conformal distortion* ($\mathrm{mean}=1.784$, $\mathrm{max}=9.4$) *after subdivision (middle), simply substituting our SEC operators for the original DEC operators significantly reduces distortion* ($\mathrm{mean}=1.005$, $\mathrm{max}=3.0$) *(right). Parameterizations, shown at level 1 for clarity, exhibit substantial differences.*

Abstract

This paper introduces a new computational method to solve differential equations on subdivision surfaces. Our approach adapts the numerical framework of Discrete Exterior Calculus (DEC) from the polygonal to the subdivision setting by exploiting the refinability of subdivision basis functions. The resulting *Subdivision Exterior Calculus* (SEC) provides significant improvements in accuracy compared to existing polygonal techniques, while offering exact finite-dimensional analogs of continuum structural identities such as Stokes' theorem and Helmholtz-Hodge decomposition. We demonstrate the versatility and efficiency of SEC on common geometry processing tasks including parameterization, geodesic distance computation, and vector field design.

Keywords: Subdivision surfaces, discrete exterior calculus, discrete differential geometry, geometry processing.

Concepts: **•Mathematics of computing → Discretization;** *Computations in finite fields;*

1 Introduction

Subdivision surfaces have become the *de facto* geometry representation in the entertainment industry [Loop et al. 2013]. By recursively refining a control mesh using linear combinations of vertices, subdivision provides an effective tool for modeling, animation, and rendering of smooth surfaces of arbitrary topology [Zorin and Schröder 2000; Warren and Weimer 2001]. In spite of this prominence, little attention has been paid to numerically solving differential equations on subdivision surfaces. This is in sharp contrast to a large body of work in geometry processing that developed discrete differential operators for polygonal meshes [Botsch et al. 2010] serving as the foundations for several applications ranging from parameterization to fluid simulation [Crane et al. 2013a].

Among the various polygonal mesh techniques, Discrete Exterior Calculus (DEC) [Desbrun et al. 2008] is a coordinate-free formalism for solving scalar and vector valued differential equations. In particular, it reproduces, rather than merely approximates, essential properties of the differential setting such as Stokes' theorem. Given that the control mesh of a subdivision surface is a polygonal mesh, applying existing DEC methods directly to the control mesh may seem tempting. However, this approach ignores the geometry of the limit surface, thus introducing a significant loss of accuracy in the discretization process (Fig. 1). A customary workaround is to perform computations on a denser polygonal mesh generated by a finite number of subdivision steps to improve accuracy, but this solution dramatically increases the number of degrees of freedom, computation time, and memory footprint. Furthermore, the mismatch between the geometry representation and the basis functions used for solving the differential equations weakens the convergence rate of the numerics. This issue is broadly recognized in the finite element literature and has led to the development of isoparameteric and isogeometric methods [Hughes et al. 2005]. Yet most of these recent techniques are neither adapted to subdivision surfaces nor do they preserve structural properties of the smooth theory.

To overcome these limitations, we introduce in this paper a systematic derivation of discrete differential operators for subdivision surfaces through a technique we call *Subdivision Exterior Calculus* (SEC). This variant of DEC can be easily retrofitted to existing implementations of a variety of geometry processing algorithms, and yields an effective numerical framework that retains core properties of smooth differential operators defined on the limit subdivision surface, while offering the accuracy expected from the use of higher-order basis functions.

SIGGRAPH '16 Technical Paper,, July 24-28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925880

1.1 Related Work

Our contributions relate to a number of research efforts in computer graphics, computational physics, and discrete differential geometry, which all seek numerical solutions of partial differential equations.

Finite element methods. The finite element methodology has long been regarded as the foundational tool to numerically solve variational problems. Using test and trial function spaces of finite dimensions, a differential equation is discretized by characterizing its solution in a so-called weak, or integral, form. These computational methods are commonly based on a polygonal approximation of the domain geometry and basis functions over each polygonal element. Increased accuracy is obtained either through *h-refinement*, where the geometry approximation (and thus the solution space) is refined using more polygons, or *p-refinement*, where the geometric discretization is kept intact but higher-order basis functions are used [Babuska and Suri 1994]. However, any mismatch between the basis functions describing the geometry and the ones defining the solution space on the geometry hampers the analysis and convergence of these approaches, an issue coined a "variational crime" [Strang and Fix 1973].

Isogeometric analysis. The concept of isogeometric analysis (IGA) was introduced in [Hughes et al. 2005] to mitigate this mismatch in computer-aided design. Typically, a B-spline representation is used both for modeling the geometry of a physical domain and for defining function spaces needed for the weak formulation employed by the finite element methodology (see [Cirak et al. 2002] for an early extension to subdivision surfaces). A primary advantage of IGA methods is that they are geometrically exact no matter how coarse the discretization is: the coarsest level of discretization entirely defines the geometry, while refinements of the function space can be performed to better capture the solution of a differential equation without affecting the shape of the domain. However, exact computation of integrals appearing in weak formulations (such as stiffness or mass matrices) is not possible in general, so quadrature rules are invoked to approximate them. In particular, adaptive quadrature strategies are required to resolve the integrands in regions of high curvature and near extraordinary vertices [Barendrecht 2013; Nguyen et al. 2014; Jüttler et al. 2016].

Structure-preserving computing. Alongside the developments mentioned above, computational approaches involving "compatible" or "mimetic" discretizations have been shown to improve numerics by ensuring exact preservation of vector calculus identities [Bossavit 1998]. The relevance of exterior calculus (Cartan's calculus of differential forms [Frankel 2004]) to computations has become particularly important: having function spaces forming a sequence with gradient, curl and divergence operators that reproduces the de Rham cohomology turns out to be crucial in guaranteeing, e.g., proper null spaces for differential operators with no spurious modes or locking artifacts. The DEC framework, in which many geometry processing algorithms can be formulated [Crane et al. 2013a], offers such a discretization scheme in the case of triangulations through the use of (refinable) low order Whitney basis functions [Desbrun et al. 2008]. Similarly, Finite Element Exterior Calculus (FEEC) allows for the use of higher-order Whitney bases defined over simplicial meshes [Arnold et al. 2006]. More recently, isogeometric extensions of these structure-preserving methods were proposed based on tensor-product splines [Buffa et al. 2011; Back and Sonnendrücker 2014] or T-splines [Buffa et al. 2014]. However, no existing approach currently offers both structure-preserving and isogeometric numerics on subdivision surfaces.

Geometry processing. The prevalence of subdivision surfaces in geometric modeling has motivated the adaptation of polygonal mesh processing algorithms for some specific applications. For instance, Zhou et al. [2007] adapted Laplacian surface editing tools to subdivision surfaces, He et al. [2010] proposed a hierarchical as-rigid-as-possible parameterization technique for subdivision, and Liu et al. [2014] combined skinning weights and subdivision. In [Stam 2003], the differential operators involved in fluid animation were altered from regular grids to Catmull-Clark subdivision surfaces. In [Lounsbery et al. 1997], the authors showed how wavelets could be constructed from subdivision surfaces. Grinspun et al. [2002] extended adaptive finite elements using subdivision basis functions for the simulation of thin shells, while Thomaszewski et al. [2006] computed bending forces on the limit surface of a Loop scheme. The work of Riffnaller-Schiefer et al. [2015] also simulated thin shells via NURBS-compatible subdivision surfaces. These methods have used quadratures based on exact evaluation [Stam 1998] at selected limit surface locations to approximate differential operators, similar to the approach favored in the IGA literature, but none offer a structure-preserving calculus that can be directly used for increasing accuracy of scalar and vector computations.

Smooth Whitney forms. Our work is closely related to the construction of Whitney forms through subdivision of vertices, edges, and faces introduced in [Wang et al. 2006]. However, the authors exploited smooth differential forms for visualization purposes only, reverting to DEC operators on the control mesh for solving differential equations. Instead, we show that the refinability of these higher-order Whitney forms can be leveraged to design accurate and structure-preserving discrete differential operators for subdivision surfaces.

1.2 Contributions

In this paper, we adapt DEC, the coordinate-free language for calculus on polygonal meshes, to subdivision surfaces. This is done by extending the work of Wang et al. [2006] that unified subdivision with the exterior derivative operator. Here we complete the picture by unifying subdivision with the inner product of differential forms. The resulting *Subdivision Exterior Calculus* (SEC) allows for computations on the control mesh with significantly improved accuracy compared to DEC as it reduces discretization errors, while retaining core properties of the smooth theory. In particular, we prove that our SEC-based differential operators satisfy the properties recognized in [Auchmann and Kurz 2006; Wardetzky et al. 2007]. Finally, we apply the SEC framework to archetypal geometry processing tasks to demonstrate that graphics algorithms can be easily extended from the polygonal to the subdivision setting.

2 Preliminaries

Since our contribution extends the DEC methodology from polygonal meshes to subdivision surfaces, we begin by reviewing important concepts, definitions, and properties of both exterior calculus and subdivision schemes that we will need in later sections.

2.1 Primer on Exterior Calculus

The differential equations we consider involve operators from calculus on manifolds. A particularly elegant and modern formalization is given by exterior calculus [Frankel 2004]. We give here a brief informal primer on exterior calculus in the smooth setting.

Differential forms are the central objects in exterior calculus. A differential k-form ω defined on a manifold $\mathcal{S}$ is a smooth antisymmetric multilinear function that maps, for each point $x \in \mathcal{S}$, k tangent vectors to a scalar; that is, $\omega : T_x\mathcal{S} \times \ldots \times T_x\mathcal{S} \rightarrow \mathbb{R}$, where $T_x\mathcal{S}$ is the tangent space at x. Differential forms themselves define vector spaces, and we denote the vector space of k-forms as

Ω_k. The antisymmetric tensorial nature of differential forms also makes them the building blocks of integration: k-forms are (the only) objects that can be integrated over k-dimensional submanifolds. Scalar-valued functions are thus 0-forms (one can evaluate them at points), while vector fields are akin to 1-forms (one can compute line integrals of a vector field),[1] and area densities are 2-forms (one can integrate them over a 2D region). Since we are dealing with surfaces, we are only concerned with 0-, 1-, and 2-forms, though most of the formalism throughout the paper holds for smooth manifolds of any dimension.

The exterior derivative d extends the notion of derivative of a function to differential forms in a metric-independent manner. It defines a linear operator mapping k-forms to $(k+1)$-forms via differentiation. More specifically, if f is a 0-form, then df is essentially the gradient of f; the exterior derivative of a 1-form represents the curl of the associated vector field, and if ω is a 2-form on a surface then $d\omega = 0$. The exterior derivative not only unifies familiar differential operators from scalar and vector calculus like gradient and curl, it also plays a fundamental role in Stokes' theorem, which states that

$$\int_{\mathcal{R}} d\omega = \int_{\partial\mathcal{R}} \omega, \tag{1}$$

for any differential k-form ω and $(k+1)$-dimensional region $\mathcal{R}$ with boundary $\partial\mathcal{R}$. This relationship implies that d applied twice to any form must be zero, since $\partial\mathcal{R}$ has no boundary and thus $\partial\partial\mathcal{R}$ is the empty set. The property $dd = 0$ is in fact tantamount to the well-known calculus identity $\nabla\times\nabla f = 0$ for any function f.

We also need a notion of inner product on differential forms to be able to compare them. Two additional concepts from exterior calculus are required for this purpose: the *wedge product* $\wedge$ and the *Hodge star operator* $\star$. The wedge product extends the notion of product to forms so that, given a k-form α and a m-form β, it returns a $(k+m)$-form $\alpha\wedge\beta$. The Hodge star operator, on the other hand, employs the metric of the n-dimensional manifold $\mathcal{S}$ to establish an isomorphism mapping a k-form to a $(n-k)$ form. Notice that the Hodge star operator $\star$ is metric dependent, in contrast to d and $\wedge$ that are defined with no reference to a metric. Armed with wedge and star, we can now define inner products between arbitrary differential k-forms α and β as

$$\langle\alpha,\beta\rangle \overset{\text{def.}}{=} \int_{\mathcal{S}} \alpha\wedge\star\beta. \tag{2}$$

Based on this definition, we say that the forms ω and $\star\omega$ are *duals* of one another in the sense that wedging the two forms defines a n-form that can then be integrated over the entire domain, returning $\langle\omega,\omega\rangle \equiv \|\omega\|^2$, i.e., the squared norm of ω.

2.2 Discrete Exterior Calculus

Discrete exterior calculus (DEC) is a convenient discretization of differential forms and the exterior calculus operators that maintains structural properties of the smooth theory [Desbrun et al. 2008]. Next, we summarize the DEC framework on polygonal meshes, which will inform our formulation on subdivision surfaces.

Discrete surface. In the following, a polygonal mesh $\mathcal{M}$ of arbitrary topology, possibly with boundaries, consists of V vertices, E edges, and F faces. Vertices are given positions indicated by a $V\times 3$ matrix $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^3\}$, edges are assigned an arbitrary but fixed orientation, and faces are oriented so that any two adjacent faces induce opposite orientations on their common edge. We further assume the mesh to be manifold, as it is supposed to be the discretization of a smooth surface.

Discrete forms. In the polygonal setting, a discrete k-form is an assignment of one scalar value to each k-cell of the polygonal mesh. Therefore, discrete 0-forms are scalars representing the evaluation of a function at vertices, discrete 1-forms are scalars representing a tangent vector field integrated along oriented edges, and discrete 2-forms are scalars encoding a density through its area integral over oriented faces. We define the vector of coefficients of a discrete k-form on $\mathcal{M}$ by $\mathbf{w}_k = \{w_{\sigma_k}\}$, where σ_k is a k-cell. Note that the sign of discrete forms changes as the orientation of their respective mesh elements is reversed, e.g., $w_{ij} = -w_{ji}$.

Exterior derivative. For polygonal meshes, Stokes' theorem in Eq. (1) is preserved *exactly* by discretizing the exterior derivatives as the transpose of the signed incidence matrices of $\mathcal{M}$. More concretely, the discrete exterior derivative for 0-forms, denoted by $\mathbf{d}_0$, is an $E\times V$ matrix with rows containing ± 1 for the endpoints of a given edge with the sign being determined by the edge orientation, while the discrete exterior derivative for 1-forms $\mathbf{d}_1$ is an $F\times E$ matrix with ± 1 entries according to the orientation of the edges as one moves around oriented polygonal faces. Observe that this definition is metric-independent (i.e., purely combinatorial) and verifies $\mathbf{d}_1\mathbf{d}_0 = 0$ as in the smooth case, creating a sequence on forms [Desbrun et al. 2008], as depicted in Fig. 2. The existence of such a sequence will have important consequences on the properties of operators like the Laplacian.

Inner product. The concept of inner product is discretized by assigning a matrix $\mathbf{M}_k$, denoted as the *inner product matrix*, to each space Ω_k so that $\langle\mathbf{a}_k,\mathbf{b}_k\rangle = \mathbf{a}_k^t\mathbf{M}_k\mathbf{b}_k$. In contrast to the discrete exterior derivatives which are purely combinatorial and hence exact, the inner product matrices are approximations of the continuous inner product. Therefore, their discretization controls the numerical accuracy of DEC computations. Several approaches have been proposed to improve the accuracy of the inner product matrices on polygonal meshes [Arnold et al. 2006; Mullen et al. 2011]. In the process, key properties of these matrices have been identified to guarantee good numerics [Auchmann and Kurz 2006]: an inner product matrix $\mathbf{M}_k$ should be *symmetric*, *positive-definite*, *convergent*, *scale-aware*, *local*, and *constant accurate in the plane*. The first two properties imply a proper inner product. Convergence to the continuous inner product ensures that the approximation error decreases under appropriate mesh refinement. Scale awareness indicates that, when the surface is uniformly scaled by a constant s, the entries of $\mathbf{M}_k$ change by a factor of $s^{2(1\text{-}k)}$ as the continuous inner product does. Locality guarantees that the matrices $\mathbf{M}_k$ are sparse, approximating the continuous inner product integrated over small neighborhoods of the mesh cells. The last property indicates that constant differential forms are reproduced exactly in the plane.

Primal versus dual forms. Mimicking the smooth setting, each inner product matrix $\mathbf{M}_k$ determines a linear mapping from a discrete k-form $\mathbf{w}_k$ to a discrete version of the *dual* $(2\text{–}k)$-form $\star\,\mathbf{w}_k$ via the approximation $\star\,\mathbf{w}_k \approx \mathbf{M}_k\mathbf{w}_k$. This property hints at why previous works often call the matrix $\mathbf{M}_k$ the discrete Hodge star operator for discrete primal k-forms, while $(-1)^k\mathbf{M}_k^{\text{-}1}$ is the discrete Hodge star operator for discrete dual $(2-k)$-forms [Desbrun et al. 2008] (see Sec. 3.5). This algebraic definition of duality is typically associated with the geometric construction of a dual mesh, e.g., circumcentric dual meshes [Hirani 2003], weighted dual meshes [de Goes et al. 2014], barycentric dual meshes [Auchmann and Kurz 2006] and their generalization to general polygonal meshes [Alexa and Wardetzky 2011]. While the geometric realization of duality is a practical tool for DEC, we will not need this concept in our work. Moreover, note that the notion of dual forms now leads to a pair of sequences involving $\mathbf{d}_0$ and $\mathbf{d}_1$ for primal forms, and $\mathbf{d}_1^t$ and $\mathbf{d}_0^t$ for dual forms (see Fig. 2), while the inner product matrix for dual $(2-k)$-forms corresponds to $\mathbf{M}_k^{\text{-}1}$.

[1] More precisely, one can transform a vector field into a 1-form—and vice versa—for any given metric using the $\sharp$ and $\flat$ operators [Frankel 2004].

$$\begin{tikzcd}
\begin{array}{c}\textit{vertex-based}\\ \text{scalar field}\end{array} \arrow[r, "\mathbf{d}_0"] \arrow[d, shift right, "\mathbf{M}_0"'] & \begin{array}{c}\textit{edge-based}\\ \text{vector field}\end{array} \arrow[r, "\mathbf{d}_1"] \arrow[d, shift right, "\mathbf{M}_1"'] & \begin{array}{c}\textit{face-based}\\ \text{density field}\end{array} \arrow[d, shift right, "\mathbf{M}_2"'] \\
\begin{array}{c}\textit{vertex-based}\\ \text{density field}\end{array} \arrow[u, shift right, "\mathbf{M}_0^{-1}"'] & \begin{array}{c}\textit{edge-based}\\ \text{vector field}\end{array} \arrow[l, "\mathbf{d}_0^t"] \arrow[u, shift right, "-\mathbf{M}_1^{-1}"'] & \begin{array}{c}\textit{face-based}\\ \text{scalar field}\end{array} \arrow[l, "\mathbf{d}_1^t"] \arrow[u, shift right, "\mathbf{M}_2^{-1}"']
\end{tikzcd}$$

Figure 2: ***Discrete de Rham Complex.*** *Discrete exterior calculus relies on a sequence over discrete forms through $\mathbf{d}_0$ and $\mathbf{d}_1$, and on a dual sequence over discrete dual forms through $\mathbf{d}_1^t$ and $\mathbf{d}_0^t$. The same sequences (and their associated structure-preserving properties) carry over to our subdivision setting.*

2.3 Discrete Differential Operators

With discrete exterior derivatives and inner product matrices, one can assemble the discrete differential operators commonly used in graphics. First, the weak (integrated) form of the standard calculus operators are $\nabla \equiv \mathbf{d}_0$, $\nabla\times \equiv \mathbf{d}_1$, and $\nabla\cdot \equiv \mathbf{d}_0^t\mathbf{M}_1$. Since $\mathbf{d}_1\mathbf{d}_0 = \mathbf{0}$, the cohomology structure of differential forms is preserved at the discrete level [Munkres 1984] and, consequently, one can construct a discrete Helmholtz-Hodge decomposition with discrete 1-forms split into curl-free, divergence-free, and harmonic subspaces [Desbrun et al. 2008]. In addition, discrete harmonic 1-forms are guaranteed to span a subspace of Ω_1 of dimension twice the mesh genus, which exactly matches the continuous picture. In practice, this structure preservation translates into computations with no spurious modes and locking artifacts.

Among the several energies commonly used in geometry processing, the Dirichlet energy of a discrete 0-form is defined as:

$$E_D(\mathbf{w}_0) \overset{\text{def.}}{=} \tfrac{1}{2}\|\mathbf{d}_0\mathbf{w}_0\|^2 = \tfrac{1}{2}\mathbf{w}_0^t\mathbf{L}_0\mathbf{w}_0, \tag{3}$$

where $\mathbf{L}_0 \overset{\text{def.}}{=} \mathbf{d}_0^t\mathbf{M}_1\mathbf{d}_0$ is the weak (or integrated) discretization of the Laplace-Beltrami operator that maps discrete 0-forms to dual discrete 2-forms. The strong (or pointwise) Laplacian is given as $\mathbf{M}_0^{-1}\mathbf{L}_0$. We note that the inner product properties mentioned in Sec. 2.2 make the discrete Laplacian $\mathbf{L}_0$ symmetric, positive semi-definite with constant functions in its kernel, convergent, scale invariant, and linear precise in the plane, thus directly preserving the core properties of the smooth Laplacian listed in [Wardetzky et al. 2007]. An additional property of the smooth Laplacian preserved in DEC is the fact that the gradient of the surface area w.r.t. surface positions matches the Laplace-Beltrami operator applied to the position coordinates [Desbrun et al. 1999], i.e., $\nabla_{\mathbf{X}}\mathcal{A} \equiv \mathbf{L}_0\mathbf{X}$ where $\mathcal{A}$ indicates the surface area.

In [Fisher et al. 2007], a smoothness energy for discrete 1-forms (named the anti-holomorphic energy) was discretized via DEC as:

$$E_A(\mathbf{w}_1) \overset{\text{def.}}{=} \tfrac{1}{2}\left(\|\mathbf{d}_1\mathbf{w}_1\|^2 + \|\mathbf{d}_0^t\mathbf{M}_1\mathbf{w}_1\|^2\right) = \tfrac{1}{2}\mathbf{w}_1^t\mathbf{L}_1\mathbf{w}_1, \tag{4}$$

where $\mathbf{L}_1 \overset{\text{def.}}{=} \mathbf{M}_1\mathbf{d}_0\mathbf{M}_0^{-1}\mathbf{d}_0^t\mathbf{M}_1 + \mathbf{d}_1^t\mathbf{M}_2\mathbf{d}_1$ is the weak version of the Hodge-Laplacian operator. Similar to the smooth case, this operator is symmetric and positive semi-definite with harmonic 1-forms in its kernel. To improve locality in $\mathbf{L}_1$, the inverse matrix $\mathbf{M}_0^{-1}$ is often approximated via "mass lumping" [Bossavit 2000].

Finally, we notice that discrete operators for dual forms are deduced by transposing the discrete exterior derivatives and inverting the inner product matrices as sketched in Fig. 2. The properties of DEC are still valid on the space of dual forms, except locality which requires inner product matrices to be diagonal (or lumped).

2.4 Subdivision Surfaces

Subdivision surfaces are a broadly deployed tool in graphics that generates smooth surfaces by recursively refining a polygonal control mesh [Zorin and Schröder 2000; Warren and Weimer 2001]. From a mesh $\mathcal{M}^l$ at subdivision level l, a refined mesh $\mathcal{M}^{l+1}$ is produced by inserting new vertices, edges, and faces to $\mathcal{M}^l$, and updating vertex positions $\mathbf{X}^{l+1}$ to a weighted average of $\mathbf{X}^l$. The weights of a subdivision scheme are determined by the local connectivity of the input mesh, and can be concisely encoded by a sparse matrix. We denote the subdivision matrix at level l by $\mathbf{S}_0^l$, where we use the subscript to indicate that only values at vertices (0-cells) are carried from level l to $l+1$. We also indicate by $\mathbf{A}_0^l$ the accumulation of subdivision matrices from the coarse mesh $\mathcal{M}^0$ to a fine mesh $\mathcal{M}^l$:

$$\mathbf{A}_0^l \overset{\text{def.}}{=} \mathbf{S}_0^{l-1} \cdots \mathbf{S}_0^1\,\mathbf{S}_0^0. \tag{5}$$

As the number of refinement steps increases, the columns of $\mathbf{A}_0^l$ converge to values of the smooth basis functions $\Phi_0^0 = \{\phi_v^0\}$, one for each vertex v in $\mathcal{M}^0$. These basis functions offer a parametric representation for subdivision surfaces, which can then be used to evaluate the limit surface exactly at arbitrary locations [Stam 1998; Niessner et al. 2012]. In particular, we note that the subdivision structure inherited by these bases makes them *refinable*: any basis function ϕ_v^l at level l is a linear combination of the bases at level $l+1$, where the coefficients in the linear combination come from the subdivision matrix $\mathbf{S}_0^l$, i.e., $\Phi_0^l = \Phi_0^{l+1}\mathbf{S}_0^l$. The limit surface can then be written as the manifold $\mathcal{S} = \Phi_0^0\mathbf{X}^0$.

3 Exterior Calculus for Subdivision Surfaces

Existing DEC approaches rely on the geometry of the input polygonal mesh to assemble discrete differential operators. Subdivision surfaces, on the other hand, describe smooth surfaces via a control mesh that only loosely approximates the limit surface. This geometric discrepancy makes the use of DEC on the polygonal control meshes deficient, since measurements on the control mesh can differ significantly from the corresponding measurements on the limit surface. While finer levels of subdivision improve numerics by generating denser meshes that better approximate the limit surface, the exponential growth of degrees of freedom resulting from subdivision makes the cost of computations infeasible. Recent efforts have investigated quadrature strategies that leverage the smooth subdivision bases in order to boost accuracy, even for computations with degrees of freedom restricted to the control mesh [Barendrecht 2013; Nguyen et al. 2014; Jüttler et al. 2016]. In contrast to DEC, these quadrature-based methods are tailored to specific differential equations and do not provide a unified discretization of differential forms and their respective differential operators.

In this section, we address these issues by presenting a new discretization of exterior calculus designed for subdivision surfaces. Our approach borrows heavily from the concepts reviewed in Sec. 2 and the pioneering work of Wang et al. [2006]. As a result, we obtain an isogeometric computational framework for solving differential equations on subdivision surfaces with a higher order of accuracy that maintains the structure-preserving nature of DEC.

3.1 Subdivision Schemes for Discrete Forms

The first task to extend the DEC framework to subdivision surfaces is to define basis functions for discrete k-forms (also known as Whitney bases [Whitney 1957]) that are compatible with the subdivision scheme used to generate the limit surface, while still forming a de Rham sequence. To this end, we adopt the approach proposed in [Wang et al. 2006], which used the subdivision basis functions Φ_0^0 as Whitney 0-form bases, and constructed subdivision schemes

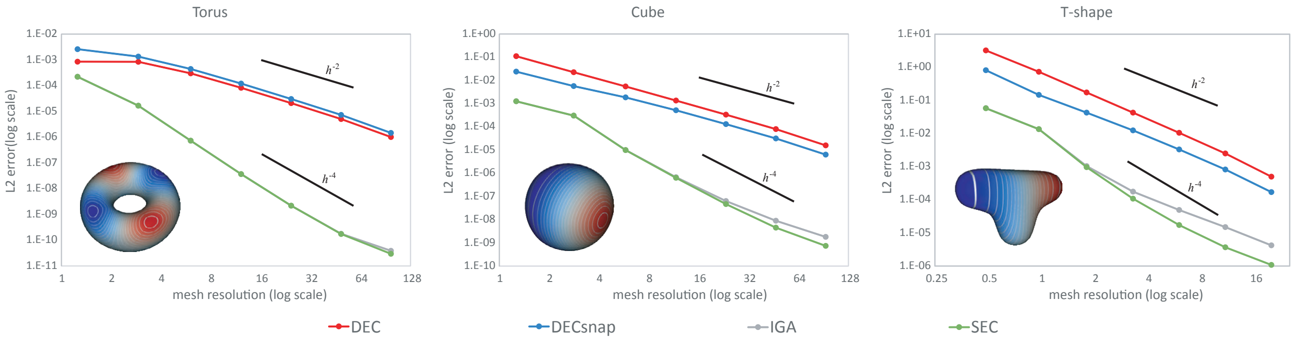

Figure 3: ***Convergence rates.*** *On three different surfaces, we show the log-log plot of the L^2 error of a Poisson solve versus the inverse of the mean edge length h. We observe the expected quadratic convergence rate for DEC and DECsnap (where vertices are first snapped to the limit surface). SEC rivals the higher-order rate of IGA methods, even exceeding it on irregular control meshes such as the cube and T-shape. The right hand sides of the Poisson equations are set to $\sin(\pi x)\sin(\pi y)\sin(\pi z)$, $\sin(\pi x/3)$, and $(2z^2-y^2-x^2)/(x^2+y^2+z^2)$ respectively.*

for discrete 1- and 2-forms by requiring that their respective subdivision matrices $\mathbf{S}_1^l$ and $\mathbf{S}_2^l$ at any level l commute with the discrete exterior derivatives. More formally, they enforced the following relationship between the subdivision matrix $\mathbf{S}_k^l$ and the discrete exterior derivative $\mathbf{d}_k^l$ of k-form at every level l:

$$\mathbf{d}_k^{l+1}\mathbf{S}_k^l = \mathbf{S}_{k+1}^l\mathbf{d}_k^l. \tag{6}$$

This property guarantees that subdividing a discrete k-form followed by the application of the discrete exterior derivative is equivalent to first applying the discrete exterior derivative followed by a $(k+1)$-form subdivision step. By recursively subdividing the discrete forms, we can define Whitney bases $\Phi_k^0 = \{\phi_{\sigma_k}^0\}$ for discrete k-forms that commute with the continuous exterior derivatives. Note also that, with this construction, the k-form basis functions Φ_k^l induced by subdivision at any level l are *refinable*:

$$\Phi_k^l = \Phi_k^{l+1}\mathbf{S}_k^l. \tag{7}$$

Hence, the spaces of discrete k-forms spanned by subdivision are nested $\Omega_k^0 \subset \Omega_k^1 \subset \cdots \subset \Omega_k^\infty$, $k = 0, 1, 2$. This extends the work of Lounsbery et al. [1997] from functions to differential forms with two important consequences. First, we can now define discrete forms on the control mesh and they automatically define smooth forms on the limit surface. Second, Eq. (6) guarantees that we can directly employ discrete exterior derivatives on the control mesh and they still define a sequence on the corresponding subdivision-based smooth forms, as in Fig. 2. Thus, metric-independent properties of DEC, such as Stokes' theorem, carry over unaltered to the subdivision setting.

In the supplemental material, we provide the complete list of 0-, 1-, and 2-form subdivision rules associated with Loop and Catmull-Clark surfaces, as they are the most used subdivision schemes in graphics. Note that we correct some of the coefficients presented in [Wang et al. 2006] for the Loop scheme to verify the "formule de commutation" for arbitrary valences. We also derive all the 1- and 2-form boundary rules associated with Catmull-Clark surfaces, which were not discussed in previous work [Wang 2008]. These rules were obtained algebraically from existing Catmull-Clark (resp., Loop) rules for 0-forms and Doo-Sabin (resp., half-box splines) rules for 2-forms so that the commutative relations in Eq (6) between $\mathbf{S}_0^l$, $\mathbf{S}_1^l$ and $\mathbf{S}_2^l$ remain valid near boundaries.

3.2 Inner Products for Subdivision Surfaces

While the exterior derivatives are combinatorial and remain unchanged in the subdivision setting, the inner product is metric-dependent, so it is contingent on the geometry of the limit surface generated by the subdivision process. For two discrete k-forms $\mathbf{a}_k^0$ and $\mathbf{b}_k^0$ defined on the control mesh corresponding to smooth k-forms $\alpha_k = \Phi_k^0\mathbf{a}_k^0$ and $\beta_k = \Phi_k^0\mathbf{b}_k^0$, their inner product is:

$$\langle\alpha_k, \beta_k\rangle = \langle\Phi_k^0\mathbf{a}_k^0, \Phi_k^0\mathbf{b}_k^0\rangle = \sum_{\sigma_k,\eta_k} a_{\sigma_k}^0 b_{\eta_k}^0 \langle\phi_{\sigma_k}^0, \phi_{\eta_k}^0\rangle. \tag{8}$$

The coefficients $\langle\phi_{\sigma_k}^0, \phi_{\eta_k}^0\rangle$ induced by the subdivision bases thus define an inner product matrix for discrete k-forms through:

$$\mathrm{M}_k^0 \overset{\text{def.}}{=} \langle\Phi_k^0, \Phi_k^0\rangle. \tag{9}$$

This inner product matrix can be understood either as the mass matrix between subdivision Whitney basis functions, or as an exact evaluation of the dual of a continuous k-form followed by a projection onto each Whitney k-form basis function. Unfortunately, these inner products cannot be evaluated exactly due to the geometry of the limit surface, and thus require some form of approximation.

In IGA methods, each inner product matrix is approximated using quadrature samples [Hughes et al. 2005], with adaptive rules around extraordinary points [Nguyen et al. 2014; Jüttler et al. 2016]. Instead, we propose to leverage the *refinability* of the Whitney basis functions to derive a refinement relation for the subdivision inner product matrices. Eq. (7) implies that M_k^0 can be written in terms of the inner product M_k^1 on the once subdivided control mesh:

$$\begin{aligned}\mathrm{M}_k^0 &\equiv \langle\Phi_k^0, \Phi_k^0\rangle = \langle\Phi_k^1\mathbf{S}_k^0, \Phi_k^1\mathbf{S}_k^0\rangle\\ &= (\mathbf{S}_k^0)^t\langle\Phi_k^1, \Phi_k^1\rangle(\mathbf{S}_k^0) \equiv (\mathbf{S}_k^0)^t\,\mathrm{M}_k^1\,(\mathbf{S}_k^0).\end{aligned} \tag{10}$$

Applying this refinement recursively l times leads to:

$$\mathrm{M}_k^0 \overset{\text{Eq. (5)}}{=} (\mathbf{A}_k^l)^t\,\mathrm{M}_k^l(\mathbf{A}_k^l). \tag{11}$$

Exploiting the rapid convergence of subdivision steps towards the limit surface [Dahmen 1986], our main contribution is to approximate the inner product matrices for the control mesh using:

$$\boxed{\mathbb{M}_k^0 \overset{\text{def.}}{=} (\mathbf{A}_k^l)^t\;\mathbf{M}_k^l\;(\mathbf{A}_k^l).} \tag{12}$$

That is, we approximate the exact inner product matrix M_k^0 at level 0 by first computing the classical DEC inner product matrix $\mathbf{M}_k^l$ at level l, then using the refinement relation in Eq. (11) to assemble $\mathbb{M}_k^0$. This approximation thus maintains the refinability condition up to the subdivision level l. We also point out that, even though we subdivide the control mesh to level l, the final inner product matrices and any subsequent computations are performed on the original control mesh. Therefore, no new degrees of freedom are added in this process.

A first implication of our inner product construction is that the desirable properties of the classical DEC inner product matrices are trivially inherited: symmetry, convergence, and scale-awareness are preserved by construction; since subdivision matrices are full-rank, positive-definiteness is also retained; locality is ensured because the subdivision matrices are sparse (for both Loop and Catmull-Clark, only a two-ring stencil on the control mesh is involved); and finally, our inner products are still constant-accurate on planar control meshes since subdivision reproduces constant fields. As we explain next, this refinement-based definition also has important numerical consequences in terms of structure preservation.

3.3 Discrete Differential Operators & Properties

Equipped with the discrete exterior derivatives $\mathbf{d}_k^0$ and inner product matrices $\mathbb{M}_k^0$ $(k = 0, 1, 2)$, a *Subdivision Exterior Calculus* (SEC) can be employed to perform computations on the control mesh of any subdivision surface: we can consistently discretize all the familiar differential operators from scalar and vector calculus by keeping the exact same expressions as in the DEC case (Sec. 2.2), but now replacing the DEC inner products $\mathbf{M}_k^0$ by our new matrices $\mathbb{M}_k^0$ as indicated in Tab. 1. In practice, this means that only the inner product matrices need to be precomputed through Eq. (12) on a control mesh, and any algorithm involving DEC operators can be trivially modified by incorporating these new subdivision-based discrete operators.

Operator	primal form	dual form
gradient	$\mathbf{d}_0$	$\mathbf{d}_1^t$
curl	$\mathbf{d}_1$	$\mathbf{d}_0^t$
divergence	$\mathbf{d}_0^t \mathbb{M}_1$	$-\,\mathbf{d}_1 \mathbb{M}_1^{-1}$
Laplace-Beltrami	$\mathbf{d}_0^t \mathbb{M}_1 \mathbf{d}_0$	$-\,\mathbf{d}_1 \mathbb{M}_1^{-1} \mathbf{d}_1^t$
Hodge-Laplace	$\mathbb{M}_1 \mathbf{d}_0 \mathbb{M}_0^{-1} \mathbf{d}_0^t \mathbb{M}_1$ $+\mathbf{d}_1^t \mathbb{M}_2 \mathbf{d}_1$	$\mathbb{M}_1^{-1} \mathbf{d}_1^t \mathbb{M}_2 \mathbf{d}_1 \mathbb{M}_1^{-1}$ $+\mathbf{d}_0 \mathbb{M}_0^{-1} \mathbf{d}_0^t$

Table 1: *Discrete differential operators for subdivision surfaces.*

Our discretization of differential operators has a number of structural properties that derive from our enforcement of commutativity and refinability of the discrete operators w.r.t. the subdivision matrices. Since the exterior derivatives $\mathbf{d}_k^0$ commute with subdivision (Eq. (6)), the curl operator also commutes with subdivision. Indeed, if $\mathbf{w}_1^0$ indicates a vector field at subdivision level 0, then the curl of its refinement $\mathbf{w}_1^1$ at level 1 can be computed by evaluating the curl at level 0 and then subdividing the result once:

$$\mathbf{S}_2^0(\nabla \times \mathbf{w}_1^0) \;\equiv\; \mathbf{S}_2^0\,(\mathbf{d}_1^0 \mathbf{w}_1^0) \overset{\text{Eq. (6)}}{=} \mathbf{d}_0^1\,\mathbf{S}_1^0 \mathbf{w}_1^0 \;\equiv\; \nabla \times \mathbf{w}_1^1. \quad (13)$$

As a consequence, if $\mathbf{w}_1^0$ is curl free, then any of its refinements (including its limit smooth form) is also curl-free.

Interestingly, the case for divergence is different due to the presence of the inner product matrix:

$$\begin{aligned} \nabla \cdot \mathbf{w}_1^0 &\equiv (\mathbf{d}_0^0)^t \mathbb{M}_1^0 \mathbf{w}_1^0 \overset{\text{Eq. (12)}}{=} (\mathbf{d}_0^0)^t (\mathbf{S}_1^0)^t \mathbb{M}_1^1 (\mathbf{S}_1^0) \mathbf{w}_1^0 \\ &= (\mathbf{S}_1^0 \mathbf{d}_0^0)^t \mathbb{M}_1^1 \mathbf{w}_1^1 \overset{\text{Eq. (6)}}{=} (\mathbf{d}_0^1 \mathbf{S}_0^0)^t \mathbb{M}_1^1 \mathbf{w}_1^1 \qquad (14) \\ &= (\mathbf{S}_0^0)^t (\mathbf{d}_0^1)^t \mathbb{M}_1^1 \mathbf{w}_1^1 \equiv (\mathbf{S}_0^0)^t \,(\nabla \cdot \mathbf{w}_1^1)\,. \end{aligned}$$

Therefore, if a 1-form $\mathbf{w}_1^0$ on the control mesh satisfies $\nabla \cdot \mathbf{w}_1^0 = 0$, the refined divergence $\nabla \cdot \mathbf{w}_1^1$ is not necessarily zero, and neither is the limit 1-form $\mathbf{w}_1 = \Phi_1^0 \mathbf{w}_1^0$. However, the divergence of the once-refined 1-form $\mathbf{w}_1^1$ is in the kernel of $(\mathbf{S}_0^0)^t$. This means that the divergence of the limit vector field $\mathbf{w}_1$ is in the kernel of $(\Phi_0^0)^t$. In other words, the limit divergence is exactly zero when tested against any of the coarse Whitney 0-form basis functions. This property guarantees that the coarse notion of the divergence of a vector field (i.e., its discrete divergence on the control mesh) is indeed a weak form of the continuous divergence. Note that a quadrature-based approximation of the inner product may not define a divergence that is consistent across levels, thus potentially degrading the numerics. Instead, our approach is compatible throughout the first l subdivision levels by construction due to Eq. (12).

Finally, our formulation ensures that the area gradient of a surface subdivided l times matches the Laplace-Beltrami of its coordinates at level 0. This geometric link between the Laplacian of coordinates and the area gradient has been successfully used for fairing of simplicial meshes [Desbrun et al. 1999], and it remains valid in our subdivision case as well. Indeed, one has for a control mesh $\mathcal{M}^0$ with position $\mathbf{X}^0$ defining a subdivided surface $\mathcal{M}^l$ of area $\mathcal{A}^l$:

$$\nabla_{\mathbf{X}^0} \mathcal{A}^l = \big(\nabla_{\mathbf{X}^0} \mathbf{X}^l\big)^t \big(\nabla_{\mathbf{X}^l} \mathcal{A}^l\big) = \big(\nabla_{\mathbf{X}^0} \mathbf{X}^l\big)^t \big(\mathbf{L}_0^l \mathbf{X}^l\big),$$

where we employed the fact that the DEC inner product matrix at level l guarantees the equality $\nabla_{\mathbf{X}^l} \mathcal{A}^l = \mathbf{L}_0^l \mathbf{X}^l$. Given the DEC Laplacian $\mathbf{L}_0^l = (\mathbf{d}_0^l)^t \mathbf{M}_1^l (\mathbf{d}_0^l)$ and since $\mathbf{X}^l = \mathbf{A}_0^l \mathbf{X}^0$, we have:

$$\begin{aligned} \nabla_{\mathbf{X}^0} \mathcal{A}^l \;&= \big(\mathbf{A}_0^l\big)^t \big(\mathbf{d}_0^l\big)^t \,\mathbf{M}_1^l\, \big(\mathbf{d}_0^l\big) \big(\mathbf{A}_0^l \mathbf{X}^0\big) \\ &\overset{\text{Eq. (6)}}{=} \big(\mathbf{d}_0^0\big)^t \big(\mathbf{A}_1^l\big)^t \,\mathbf{M}_1^l\, \big(\mathbf{A}_1^l\big) \big(\mathbf{d}_0^0\big) \mathbf{X}^0 \qquad (15) \\ &\overset{\text{Eq. (12)}}{=} \big(\mathbf{d}_0^0\big)^t \,\mathbb{M}_1^0\, \big(\mathbf{d}_0^0\big) \mathbf{X}^0 \overset{\text{Tab. (1)}}{=} \mathbb{L}_0^0 \mathbf{X}^0. \end{aligned}$$

Note that a quadrature-based inner product matrix could not enforce, in general, such a direct link between the resulting Laplace-Beltrami evaluation and the area gradient of a subdivided surface. The proof above also confirms that our SEC Laplacian annihilates linear functions, a convenient numerical property in the discrete realm [Wardetzky et al. 2007]. This can be easily shown by noticing that $\nabla_{\mathbf{X}^0} \mathcal{A}^l = \mathbf{0}$ if control points $\mathbf{X}^0$ lie in a plane.

3.4 Numerics & Accuracy

When dealing with numerical approximations of differential equations on surfaces, it is important to distinguish between two sources of numerical error, which we refer to as projection and operator errors. Generally speaking, the exact solution of a differential equation lives in an infinite-dimensional function space defined on the limit smooth surface. When we pick a control mesh, we are projecting this exact solution onto the finite dimensional function space spanned by the subdivision basis functions. The minimum norm residual is what we refer to as *projection error*. This error can only be reduced by enriching our function spaces, i.e., by using a finer control mesh. The *operator error*, on the other hand, occurs because our discretization of the inner product (the only metric-dependent operators in our context) cannot, in general, be computed exactly. Therefore, the discrete version of a differential operator may exhibit limited accuracy for a given function space.

Projection error. By analyzing the rate of convergence of a numerical solution towards a reference solution, we can compare the projection error of our SEC discretization versus DEC and IGA. Fig. 3 shows the L^2 errors generated when solving a Poisson equation over several subdivision levels for three different surfaces. A finite element solution on a high resolution mesh is used as ground truth, and the error norms are all evaluated at level 7 for fairness. We used the approach of [Alexa and Wardetzky 2011] for evaluating inner product matrices $\mathbf{M}_k$ on polygonal meshes for DEC and SEC, and Gauss-Legendre quadrature rules of fourth order for IGA, as recommended in [Jüttler et al. 2016]. Note that the rate of convergence of DEC is roughly quadratic in the mean edge length. Convergence for both SEC and IGA is considerably better, exhibiting a quartic projection error decrease on regular meshes, as expected from the use of higher-order basis functions. We also observe a reduction in the order of convergence for meshes with high curvature regions or in the presence of extraordinary vertices, in agreement with what is observed in recent IGA methods on similar tests [Barendrecht 2013; Nguyen et al. 2014; Jüttler et al. 2016].

Operator error. In order to estimate operator error, we compare the residuals of a Poisson solve for different discretizations of the inner product on the control mesh in Fig. 4 (bottom left). We use as reference a numerical solution computed on the control mesh via the SEC operators in Eq. (12) for $l = 7$. Errors are as follow:

Method	L^2 **error**	L^∞ **error**	H^1 **error**
DEC	1.7343E-2	1.5033E-2	2.2640E-2
DECsnap	5.6270E-3	4.4101E-3	1.5598E-2
SEC $l = 1$	3.1239E-3	2.4947E-3	5.6419E-3
SEC $l = 2$	7.0890E-4	7.5853E-4	1.7026E-3
SEC $l = 3$	1.6839E-4	2.0303E-4	4.5758E-4
SEC $l = 4$	3.9548E-5	4.9458E-5	1.1195E-4
IGA order 4	4.7071E-5	3.9551E-5	8.4435E-5

The conventional DEC operator leads to quite poor results, which can be improved by "snapping" the control vertices to their limit locations; we call this approach DECsnap. The operator error in SEC is reliably controlled by adjusting the subdivision level l used in Eq. (12). By level 3 the L^2 error has been reduced by two orders of magnitude, while level 4 performs similarly to an IGA-based evaluation with Gauss-Legendre quadrature of order 4. Notice that the two-ring support of the basis functions for both Loop and Catmull-Clark schemes implies that the sparsity of $\mathbb{M}_k^0$ matches the sparsity of M_k^0 for $l \geq 2$. Given the fast reduction in operator error with the number of levels l, using $l = 3$ is sufficient to make operator error negligible compared to projection error in typical situations.

While we recommend computing the inner product matrices using SEC (Eq. (12)) with $l = 3$, accuracy at coarser levels can be improved by snapping the refined coefficients to their limit values, and appropriately pulling the result back to level 0. The resulting SECsnap inner product matrices are expressed as

$$\mathbb{M}_k^0 = (\mathbf{P}_k^l \mathbf{A}_k^l)^t \, \mathbf{M}_k^l \, (\mathbf{P}_k^l \mathbf{A}_k^l), \tag{16}$$

where $\mathbf{P}_k^l$ is the matrix that snaps the coefficients at level l to their limit values. This modification thus employs the exact evaluation of the basis functions Φ_k^l for every k-cell in $\mathcal{M}^l$. The same snapping can also be used for larger values of l, but our numerical tests show very limited impact since the limit surface is already quite well approximated. Fig. 4 compares numerical solutions of a Poisson equation using SECsnap for $l = 0$ with DEC and SEC for $l = 3$, for a right hand side set to a pair of opposite-sign Diracs placed on the hood and the trunk.

3.5 Discussion

We conclude this section with a few remarks on the relevance of our subdivision exterior calculus outside of geometry processing.

Discrete Hodge Star. Given that the continuous inner product of differential forms (Eq. (2)) involves both the Hodge star operator and an integral of the wedge product, discrete notions of these operators have also been introduced in previous work (see, e.g., [Hirani 2003]). In particular, the inner product matrix $\mathbf{M}_k$ is seen in DEC as a linear isomorphism from the space of discrete primal k-forms to discrete dual $(2-k)$-forms, thus directly defining a matrix version of the discrete Hodge star operator for discrete primal k-forms [Desbrun et al. 2008]. By analogy to the smooth theory, the discrete Hodge star operator for dual $(2-k)$-forms is then represented by the matrix $(-1)^k \mathbf{M}_k^{-1}$. Therefore, the matrices $\mathbb{M}_k$ defined in SEC offer higher-order approximations of the discrete Hodge stars and can be used, for instance, in computational electromagnetism [Bossavit 1998; Buffa et al. 2014].

Link to Multiresolution Analysis. Since the matrices $\mathbf{A}_k^l$ are approximations of the bases Φ_k^l, our expression of the inner product matrices can be interpreted as an extension of the work of Lounsbery et al. [1997]: the authors recognized the importance of the refinability of the inner product for functions (0-forms), and used it to compute the inner product matrix exactly in *parameter* space, but left the construction of inner products at the limit surface as future work. Our work offers a practical solution by approximating with arbitrary accuracy the inner product matrix on the limit surface. Using our inner product matrices, one could also use the machinery developed by Lounsbery et. al. [1997] to construct bi-orthogonal wavelets directly on the limit surface.

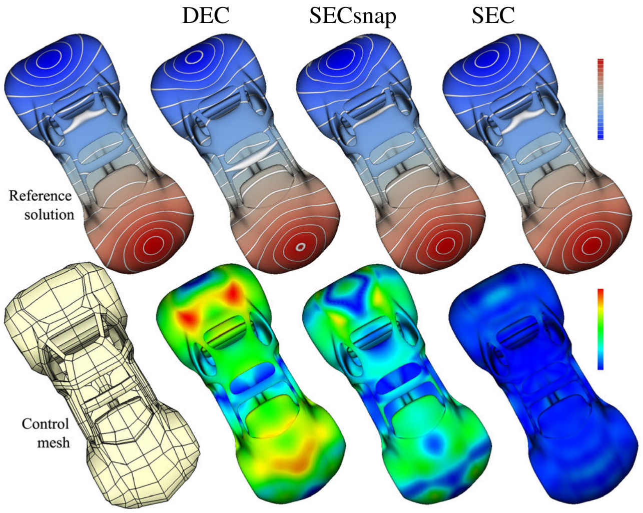

Figure 4: ***Numerical accuracy.*** *Solving a Poisson equation on the control mesh of the car model with DEC, SEC ($l = 0$) with snapping, and SEC ($l = 3$), respectively, demonstrates the increase in accuracy that SEC produces, with the numerical solution nearly matching the reference solution computed on a $16K$ times finer mesh. Top row exhibits color-coded values of the solution on the subdivided mesh once scaled to match the range of the reference solution, while bottom row shows the magnitude of the pointwise difference to the reference solution once normalized by the maximum residual among these three methods.*

Link to Isogeometric Analysis. IGA-based methods using subdivision or NURBS bases employ a Galerkin approach to discretize differential equations. Despite the fact that the subdivision bases are piecewise polynomial, evaluating the integrals involved in weak forms is usually intractable because of the non-linearity introduced by the surface area term [Cottrell et al. 2009]. Gauss-Legendre quadrature rules are thus employed, where the area term is handled by computing the Jacobian matrix that maps the parametric representation to the actual limit surface at each quadrature point. Our SEC approach actually matches (and even outperforms at times) the order of accuracy typically witnessed in IGA formulations, as evidenced in Fig. 3. Moreover, quadrature methods suffer from two main drawbacks. First, it is still unknown if any quadrature rule can retain the properties of smooth operators as we discussed earlier. Second, quadratures are operator-dependent, so composition of, e.g., a weak divergence and a weak gradient operator may not match the direct discretization of the Laplacian operator. Instead, our approach can be seen as a specially tailored quadrature that resolves the problems mentioned above by placing samples at the cells of the subdivided mesh and approximating the area term using the geometry of a fine polygonal mesh. SEC thus provides a consistent and structure-preserving discretization of differential forms and associated operators. It is also worth noting that SEC can leverage existing DEC treatments of various boundary conditions without modification.

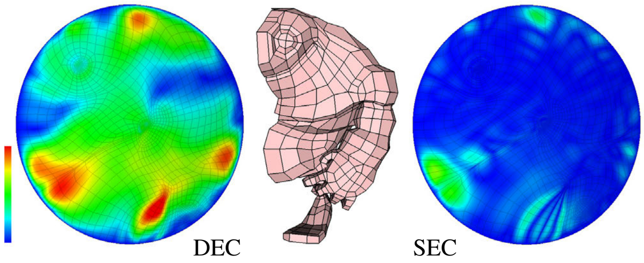

Figure 5: ***Fixed-boundary parameterization.*** *With the boundary of the half big guy model (middle) set to a circle, minimizing the conformal energy on the control mesh using DEC (left, color-coded error map) is significantly worse than using SEC (right, $l = 3$) when compared to a reference solution computed at higher resolution.*

Link to Multigrid. Our approach also bears similarities to multigrid methods. If one thinks of the subdivision matrices as playing the role of upsampling operators, then our inner product construction can be seen as a Galerkin multigrid approach [McCormick 1984]: the coarse discretization of the operator is evaluated by first performing upsampling, then discretizing the upsampled problem, and finally gathering the result back to the coarse level.

Lumping. We notice that our inner product matrices are sparse, but not diagonal due to the two-ring support of Loop and Catmull-Clark basis functions. As generally done in Galerkin methods, one can improve efficiency by computing a "lumped" version of these matrices without significantly hurting accuracy. Since 0- and 2-forms are associated to (scalar and density) functions, their lumping can be computed as: $\text{lumped}(\mathbb{M}_0^0) = \text{diag}(\mathbb{M}_0^0\mathbf{1})$ and $\text{lumped}(\mathbb{M}_2^0) = \text{diag}(\mathbb{M}_2^0\mathbf{1})$, rendering both matrices diagonal (where $\mathbf{1}$ is the column vector of all ones). Lumping the inner product matrix for 1-forms can also be achieved by replacing Φ_1^0 with $d\Phi_0^0$, as suggested in [Bossavit 2000] for the case of the linear Whitney bases. Note that this lumping reduces the sparsity of the resulting inner product by keeping the discrete Laplacian for 0-forms intact, i.e., $\mathbf{d}_0^{0\,t}\text{lumped}(\mathbb{M}_1^0)\mathbf{d}_0^0 = \mathbf{d}_0^{0\,t}\mathbb{M}_1^0\mathbf{d}_0^0$, but it does not make the matrix diagonal due to the two-ring support of Φ_0^0.

4 Applications

Based on the machinery developed in the preceding section, geometry processing algorithms formulated using DEC are now immediately applicable to subdivision surfaces by simply using our SEC operators in place of the polygonal DEC operators. In the following, we demonstrate how our SEC operators can improve the results of common graphics applications such as parameterization, geodesic distance computation, and vector field design. In all our examples, we use the inner product matrices $\mathbf{M}_k$ derived from the work of Alexa and Wardetzky [2011] as they apply to both triangle and quad meshes. This amounts to setting $\mathbf{M}_0$ and $\mathbf{M}_1$ to their mass matrices M_0 and M_1, and $\mathbf{M}_2$ to the diagonal matrix with the inverse area of each polygon's maximal projection. All results were clocked on an Intel Core i7 2.2 GHz laptop with 4GB RAM.

4.1 Parameterization

Parameterization computes a mapping from an input mesh to the plane. Various approaches have been proposed to generate angle-preserving (or conformal) parameterizations. Among them, Mullen et al. [2008] presented a spectral technique for computing a parameterization using a constrained minimization of the quadratic conformal energy:

$$E_C(\mathbf{u}) = E_D(\mathbf{u}) - \tfrac{1}{2}\,\mathbf{u}^t\,\mathbf{B}\,\mathbf{u}, \tag{17}$$

where $\mathbf{u}$ is a vector of (u, v) coordinates of the mesh vertices, E_D is the Dirichlet energy defined in Eq. (3) involving the Laplace-Beltrami operator of the surface, and $\mathbf{B}$ is the quadratic form representing the area of the parameterization.

Using our SEC operators in lieu of the DEC operators, we can directly apply this method to subdivision surfaces. Fig. 1 displays the results of computing on the control mesh a conformal parameterization of a subdivision surface using DEC versus SEC for $l = 3$. Both texture distortion and quasi-conformal error plots show that SEC results in far improved accuracy: while DEC results in an area-weighted mean error of 1.784 and a max error of 9.4, SEC leads to a mean error of 1.005 and a max error of 3.0. For comparison purposes, IGA produces a mean error of 1.006 and a max error of 6.7. Because the sparsity of the inner product matrices are only changing from a one-ring neighborhood in DEC to a two-ring neighborhood in SEC, the eigen computations are only slightly affected: the power method for our SEC approach requires 8ms compared to 3ms for the original DEC approach. Additionally, we compare in Fig. 5 the parameterization of a control mesh to a circular boundary shape using both SEC and DEC. This experiment allows us to evaluate their respective error to a reference solution computed with the same boundary conditions on a thrice subdivided mesh. Here again, the results are quite different, with SEC improving the L^2 (resp., L^∞) error by a factor 7 (resp., 2) compared to DEC. Yet, the linear solve involved in SEC is still of the size of the control mesh, thus ending up being six times faster than the reference solution. Substituting SEC by IGA in Fig. 5 results in negligible differences, with $L^2_{\text{SEC}} = 0.10$ and $L^\infty_{\text{SEC}} = 0.91$ versus $L^2_{\text{IGA}} = 0.12$ and $L^\infty_{\text{IGA}} = 0.89$. We point the reader to the supplemental material for a visual comparison of DEC, SEC and IGA.

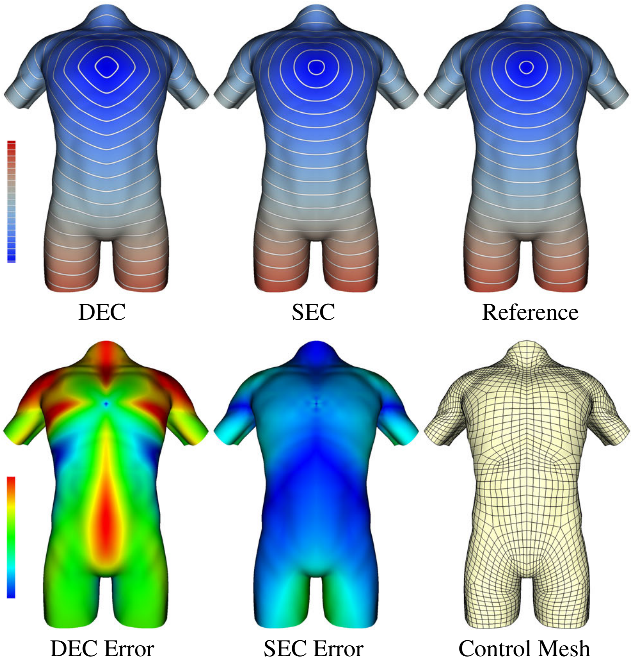

Figure 6: ***Geodesic distances on Torso.*** *Applying the heat method of Crane et al. [2013b] using DEC operators on the control mesh of the torso model results in large distance errors on the subdivision surface (left). Our SEC ($l = 3$) operators (middle) provide much improved distances which are visually comparable to the reference solution (top right) computed on a 64 times finer mesh.*

4.2 Fast Geodesics

Our SEC operators can also be used in the heat method [Crane et al. 2013b] to compute geodesic distances on subdivision surfaces. This approach consists of two steps: first, a short diffusion is solved with Robin boundary conditions and time step set to the maximum edge length of the control mesh; then the distance function is computed such that its gradient best matches the normalized gradient of the diffusion result.

By substituting our SEC operators for the DEC operators used on the original polygonal mesh, we can directly apply the heat method to subdivision surfaces. Fig. 6 shows the geodesic distances computed using polygonal DEC operators on the control mesh (left) versus the SEC operators (l=3, middle), compared to a reference solution using the original heat method computed on a mesh subdivided three times (right). The color-coded distance error maps exhibit significantly better results for our SEC approach, with $L^2_{\text{SEC}} = 0.27$ and L^∞_{SEC}=0.98 versus L^2_{DEC}= 0.67 and L^∞_{DEC}=2.01. Once the assembly of the inner product matrices and prefactorization of the Laplace-Beltrami operator on the control mesh are completed (for which SEC is 8 times slower), computing the geodesic distance field from any source requires 12% more time for SEC than for DEC. Results with IGA are quite similar on this example, with $L^2_{\text{IGA}} = 0.27$ and L^∞_{IGA}=1.01 (images included in the supplemental material). Fig. 9 confirms numerical superiority of SEC and IGA over DEC on another surface: here, SEC and IGA are only 5% slower than DEC at computing a distance field, while their respective error residuals compared to a reference solution are $L^2_{\text{DEC}} = 2.38$ and $L^\infty_{\text{DEC}} = 5.35$, $L^2_{\text{IGA}} = 1.05$ and $L^\infty_{\text{IGA}} = 3.25$, versus $L^2_{\text{SEC}} = 1.03$ and $L^\infty_{\text{SEC}} = 3.36$.

4.3 Vector Field Design

Design of vector fields on surfaces is another example application that is a necessary prelude to a number of geometry processing algorithms such as texturing and grooming. The work of Fisher et al. [2007] offers an interactive method for designing tangent vector fields over triangle meshes by solving for a 1-form, representing the line integral of the vector field along each oriented edge. The user can specify sources and sinks (0-forms), vorticity (2-forms), and draw strokes (used as local 1-form constraints) to drive the field. The algorithm then solves a least-squares problem to minimize the anti-holomorphic energy in Eq. (4) with constraint penalties, thus finding the smoothest vector field subject to the user constraints.

In the context of subdivision surfaces, we can adapt this approach directly using our SEC framework, with no modification to the algorithm. This differs from existing IGA methods, which have not handled coordinate-free representations of vector fields on subdivision surfaces. A simple substitution of the DEC divergence, curl, and Hodge-Laplace operators by our equivalent SEC operators (Tab. 1) is enough to now provide an isogeometric, and still interactive, design tool for tangent vector fields on subdivision surfaces. The only added computation time is the pre-assembly of the inner product matrices, which never exceeded 4s in any of the examples shown in this paper. Note that this application carries over to our subdivision setting seamlessly because SEC offers a discrete Helmholtz-Hodge decomposition that is consistent across levels due to the properties discussed in Sec. 3.3. Fig. 8 shows a vector field designed using our SEC operators (l = 3) on a two-holed torus (top left), and its Helmoltz-Hodge decomposition into curl-free, divergence-free, and harmonic components. Compared to a reference solution computed on a thrice subdivided mesh, the difference of the vector field minimizers is L^2_{DEC} = 1.338 versus $L^2_{\text{SEC}} = 0.263$. Fig. 7 exhibits a grooming application where the fur is directed by an optimized vector field. In this example, the user interactively places constraints on the control mesh of the lion head model and our SEC operators are used to compute the constrained vector field at the limit surface.

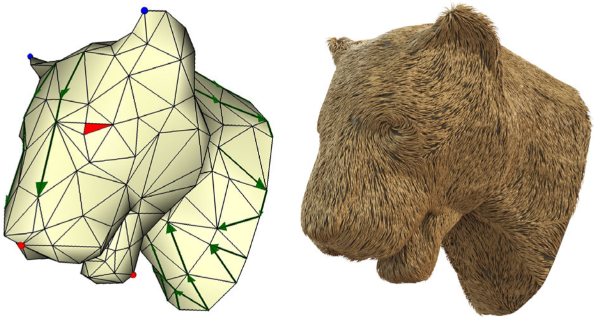

Figure 7: ***Grooming via vector field design.*** *With a small number of constraints on the control mesh of the lion head model, our SEC operators can be used to compute a vector field which drives the fur growth. Red faces indicate curl, red vertices are sinks, blue vertices are sources, and green arrows are direction constraints.*

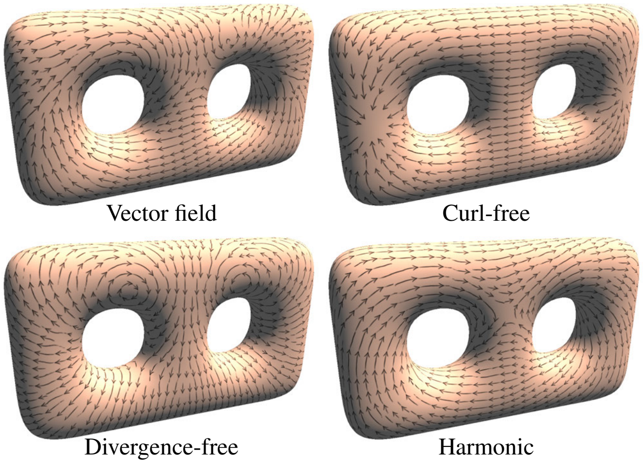

Figure 8: ***Helmholtz-Hodge decomposition.*** *Based on SEC (l = 3), we can perform a Helmholtz-Hodge decomposition of vector fields defined on the control mesh of subdivision surfaces (top left), and extract curl-free (top right), divergence-free (bottom left) and harmonic (bottom right) components. Our method improves the L^2 error of the vector field by 80% over DEC, when compared to the reference solution computed on a 64 times finer mesh.*

4.4 Discussion

While we described SEC assuming that discrete 1-forms $\mathbf{w}$ encode *edge circulations* $\int_e \mathbf{w} \cdot \mathbf{t}$, it bears mentioning that one can also treat them as *edge fluxes* instead, i.e., $\int_e \mathbf{w} \cdot \mathbf{n}$, where $\mathbf{t}$ and $\mathbf{n}$ are the tangent and normal directions of the edge e. This modified convention, called a pseudo-form [Desbrun et al. 2008], only alters the continuous interpretation of the exterior calculus operator: if $\mathbf{w}$ is a pseudo-form, then $\mathbf{d}_1 \mathbf{w}$ now represents the divergence of the vector field, while $(\mathbf{d}_0)^t \mathbb{M}_1 \mathbf{w}$ becomes the curl. The choice of form versus pseudo-form is application-specific, depending on which one makes boundary conditions easier to handle. For instance, fluids on surfaces are often computed using fluxes as main variables [Stam 2003; Elcott et al. 2007] so that no-flux boundary conditions are trivially enforced. Hence, our SEC machinery can also be used for this application: since the advection operator (Lie derivative) can be written purely in terms of the exterior derivative and the Hodge star [Hirani 2003], one can compute the curl of a flux-based vector field, and advect this (integrated) vorticity explicitly in time before converting the advected vorticity back to fluxes through a Poisson equation. Model-reduced fluid simulation can also be done by applying directly the work of Liu et al. [2015], since the eigenstructure of our 2-form and 1-form Laplacians meet the properties that their method requires. Our SEC approach is thus broadly applicable throughout geometry processing and animation.

5 Conclusion & Future Work

Subdivision Exterior Calculus provides a novel extension of Discrete Exterior Calculus for subdivision surfaces. It combines the foundational strengths of refinable finite elements with the simplicity of the DEC methodology through a simple alteration of the inner product matrices. Our results, covering typical geometry processing applications, demonstrate that scalar and vector calculus on subdivision surfaces can be done with only a minor computational overhead, with accuracy rivaling isogeometric techniques. Moreover, a number of geometric identities remains valid for SEC, while they are only approximated in IGA methods.

Our work calls for a number of future investigations. For a given level of accuracy, our inner product (or Hodge star) matrices may be constructed more efficiently in both time and memory by using adaptive refinement, for instance in areas of high curvature or irregular features. It would also be interesting to extend the feature-adaptive method of Niessner et. al. [2012] to the evaluation of 1- and 2-forms. Applying our approach to other refinable schemes, such as NURBS or T-splines, is also of interest. Finally, the use of semi-sharp creases requires the derivation of modified subdivision schemes for 1- and 2-forms such that the exterior derivative commutes with subdivision.

Acknowledgements.

We thank Daniel Garcia for his help in Fig. 7, and David Yu for his help in Fig. 8. We also acknowledge Noam Aigerman, Peter Schröder, Rasmus Tamstorf, and Ke Wang for early discussions, and Yiying Tong for his help with the final version. Meshes are courtesy of Bay Raitt (Car, Big guy, and MonsterFrog), Hugues Hoppe (Mannequin), and Giorgio Marcias (Torso). MD was partially supported through NSF grant CCF-1011944.

References

ALEXA, M., AND WARDETZKY, M. 2011. Discrete Laplacians on general polygonal meshes. *ACM Trans. Graph. 30*, 4, Art. 102.

ARNOLD, D. N., FALK, R. S., AND WINTHER, R. 2006. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica 15*, 1–155.

AUCHMANN, B., AND KURZ, S. 2006. A geometrically defined discrete Hodge operator on simplicial cells. *IEEE Trans. on Magnetics 42*, 4, 643–646.

BABUSKA, I., AND SURI, M. 1994. The *p* and *h-p* versions of the finite element method: Basic principles and properties. *SIAM Review 36*, 4, 578–632.

BACK, A., AND SONNENDRÜCKER, E. 2014. Finite element Hodge for spline discrete differential forms: Application to the Vlasov-Poisson system. *Appl. Numer. Math. 79*, 124–136.

BARENDRECHT, P. J. 2013. *Isogeometric Analysis for Subdivision Surfaces*. Master's thesis, Eindhoven University of Technology.

BOSSAVIT, A., Ed. 1998. *Computational Electromagnetism*. Academic Press.

BOSSAVIT, A. 2000. Computational electromagnetism and geometry. (5): The 'Galerkin Hodge'. *J. Japan Soc. Appl. Electromagn. & Mech. 8*, 2, 203–9.

BOTSCH, M., KOBBELT, L., PAULY, M., ALLIEZ, P., AND LÉVY, B. 2010. *Polygon Mesh Processing*. AK Peters.

BUFFA, A., RIVAS, J., SANGALLI, G., AND VÁZQUEZ, R. 2011. Isogeometric discrete differential forms in three dimensions. *SIAM J. Numer. Anal. 49*, 2, 818–844.

BUFFA, A., SANGALLI, G., AND VÁZQUEZ, R. 2014. Isogeometric methods for computational electromagnetics: B-spline and T-spline discretizations. *J. Comput. Phys. 257*, 1291–1320.

CIRAK, F., SCOTT, M. J., ANTONSSON, E. K., ORTIZ, M., AND SCHRÖDER, P. 2002. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Comput. Aided Des. 34*, 137–148.

COTTRELL, J. A., HUGHES, T. J. R., AND BAZILEVS, Y. 2009. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing.

CRANE, K., DE GOES, F., DESBRUN, M., AND SCHRÖDER, P. 2013. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH Courses*.

CRANE, K., WEISCHEDEL, C., AND WARDETZKY, M. 2013. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5, Art. 152.

DAHMEN, W. 1986. Subdivision algorithms converge quadratically. *J. Comput. Appl. Math. 16*, 2, 145–158.

DE GOES, F., MEMARI, P., MULLEN, P., AND DESBRUN, M. 2014. Weighted triangulations for geometry processing. *ACM Trans. Graph. 33*, 3, Art. 28.

DESBRUN, M., MEYER, M., SCHRÖDER, P., AND BARR, A. H. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. ACM SIGGRAPH, 317–324.

DESBRUN, M., KANSO, E., AND TONG, Y. 2008. Discrete differential forms for computational modeling. In *Discrete Differential Geometry,* A. I. Bobenko *et al.* (Eds), vol. 38 of *Oberwolfach Seminars*. 287–324.

ELCOTT, S., TONG, Y., KANSO, E., SCHRÖDER, P., AND DESBRUN, M. 2007. Stable, circulation-preserving, simplicial fluids. *ACM Trans. Graph. 26*, 1, Art. 4.

FISHER, M., SCHRÖDER, P., DESBRUN, M., AND HOPPE, H. 2007. Design of tangent vector fields. *ACM Trans. Graph. 26*, 3, Art. 56.

FRANKEL, T. 2004. *The Geometry of Physics: An Introduction*. Cambridge University Press.

GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. CHARMS: A simple framework for adaptive simulation. *ACM Trans. Graph. 21*, 3, 281–290.

HE, L., SCHAEFER, S., AND HORMANN, K. 2010. Parameterizing subdivision surfaces. *ACM Trans. Graph. 29*, 4, Art. 120.

HIRANI, A. 2003. *Discrete Exterior Calculus*. PhD thesis, Caltech.

HUGHES, T., COTTRELL, J., AND BAZILEVS, Y. 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng. 194*, 39–41, 4135–4195.

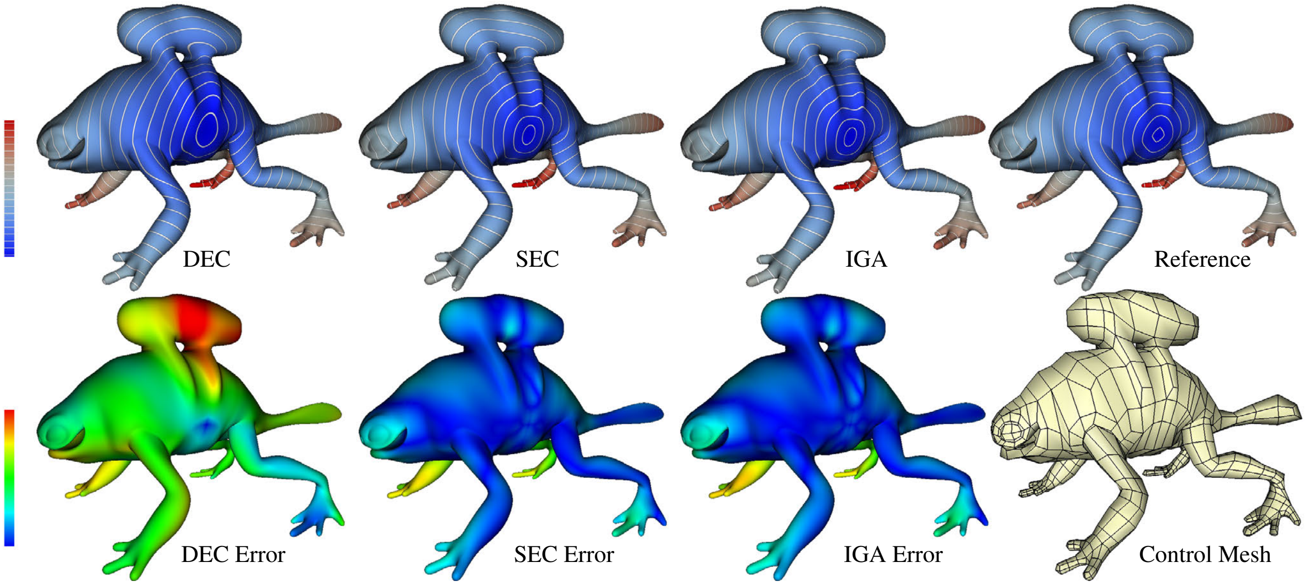

Figure 9: ***Geodesic distances on MonsterFrog.*** *Top row compares DEC versus SEC ($l = 3$) versus IGA for the computation of geodesic distances from a side point on the MonsterFrog model. Bottom row displays color-coded values of the pointwise difference between the numerical solutions and a reference distance field computed on a* 64 *times finer mesh. Our SEC approach is qualitatively similar to IGA and reduces the L^2 (resp., L^∞) error by a factor* 2.3 *(resp.,* 1.6*) compared to DEC.*

JÜTTLER, B., MANTZAFLARIS, A., PERL, R., AND RUMPF, M. 2016. On numerical integration in isogeometric subdivision methods for PDEs on surfaces. *Comput. Methods Appl. Mech. Eng. 302*, 131–146.

LIU, S., JACOBSON, A., AND GINGOLD, Y. 2014. Skinning cubic Bézier splines and Catmull-Clark subdivision surfaces. *ACM Trans. Graph. 33*, 6, Art. 190.

LIU, B., MASON, G., HODGSON, J., TONG, Y., AND DESBRUN, M. 2015. Model-reduced variational fluid simulation. *ACM Trans. Graph. 34*, 6, Art. 244.

LOOP, C., VAN GELDER, D., LITKE, N., EL GUERRAB, R., ELMIEH, B., AND KRAEMER, M. 2013. OpenSubdiv from research to industry adoption. In *ACM SIGGRAPH Courses*.

LOUNSBERY, M., DEROSE, T. D., AND WARREN, J. 1997. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph. 16*, 1, 34–73.

MCCORMICK, S. F. 1984. Multigrid Methods for Variational Problems: Further Results. *SIAM J. Numer. Anal. 21*, 2, 255–263.

MULLEN, P., TONG, Y., ALLIEZ, P., AND DESBRUN, M. 2008. Spectral conformal parameterization. *Comput. Graph. Forum 27*, 5, 1487–1494.

MULLEN, P., MEMARI, P., DE GOES, F., AND DESBRUN, M. 2011. HOT: Hodge-optimized triangulations. *ACM Trans. Graph. 30*, 4, Art. 103.

MUNKRES, J. R. 1984. *Elements of Algebraic Topology*. Addison-Wesley.

NGUYEN, T., KARČIAUSKAS, K., AND PETERS, J. 2014. A comparative study of several classical, discrete differential and isogeometric methods for solving Poissons equation on the disk. *Axioms*, 3, 280–299.

NIESSNER, M., LOOP, C., MEYER, M., AND DEROSE, T. 2012. Feature-adaptive GPU rendering of Catmull-Clark subdivision surfaces. *ACM Trans. Graph. 31*, 1, Art. 6.

RIFFNALLER-SCHIEFER, A., AUGSDÖRFER, U. H., AND FELLNER, D. W. 2015. Isogeometric Analysis for Modelling and Design. In *Eurographics (short papers)*.

STAM, J. 1998. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. ACM SIGGRAPH, 395–404.

STAM, J. 2003. Flows on surfaces of arbitrary topology. *ACM Trans. Graph. 22*, 3, 724–731.

STRANG, G., AND FIX, G. 1973. *An Analysis of the Finite Element Method*. Wellesley-Cambridge.

THOMASZEWSKI, B., WACKER, M., AND STRASSER, W. 2006. A consistent bending model for cloth simulation with corotational subdivision finite elements. In *Symp. Comp. Anim.*, 107–116.

WANG, K., WEIWEI, TONG, Y., DESBRUN, M., AND SCHRÖDER, P. 2006. Edge subdivision schemes and the construction of smooth vector fields. *ACM Trans. Graph. 25*, 3, 1041–1048.

WANG, K. 2008. *A subdivision approach to the construction of smooth differential forms*. PhD thesis, Caltech.

WARDETZKY, M., MATHUR, S., KÄLBERER, F., AND GRINSPUN, E. 2007. Discrete Laplace operators: No free lunch. In *Symp. Geom. Process.*, 33–37.

WARREN, J., AND WEIMER, H. 2001. *Subdivision Methods for Geometric Design: A Constructive Approach*. Morgan Kaufmann Publishers Inc.

WHITNEY, H. 1957. *Geometric Integration Theory*. Princeton University Press.

ZHOU, K., HUANG, X., XU, W., GUO, B., AND SHUM, H.-Y. 2007. Direct manipulation of subdivision surfaces on GPUs. *ACM Trans. Graph. 26*, 3, Art. 91.

ZORIN, D., AND SCHRÖDER, P. 2000. Subdivision for modeling and animation. In *ACM SIGGRAPH Courses*.