Geometric Discretization through Primal-Dual Meshes

Thesis by

Fernando de Goes

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2014
(Defended April 29, 2014)

Acknowledgements

A doctorate is not only the end of graduate school, but symbolizes the legacy in my life of many people who inspired, motivated, guided and challenged me over the years. My sincere gratitude!

I would like to thank my committee members—**Peter Schröder**, **Houman Owhadi**, **Pierre Alliez**, and **Mathieu Desbrun**—for their encouraging comments. Special thanks to **Lance Williams**, **Michael Kass**, **Tony DeRose**, and **Mark Meyer** for showing me that academia and industry can walk hand in hand, as it should be. I also thank **Al Barr** for his many advices and long discussions; **Santiago Lombeyda** and **Alain Martin** for wonderfull chats during lunch; and **Maria Lopez** and **Sheila Shull** for keeping me out of trouble. Before I joined Caltech, I had the guidance and friendship of three special professors: **José Portes**, **Siome Goldenstein** e **Luiz Velho**, muito obrigado pelos conselhos e oportunidades.

I have been lucky to collaborate with brilliant people: thanks **Patrick Mullen** and **Pooran Memari** for the HOT team; thanks **Katherine Breeden** for the Cube team and Singapore slingers; thanks **Pierre Alliez** and **David Cohen-Steiner** for many productive collaborations and a wonderful summer in Sophia-Antipolis; thanks **Victor Ostromoukhov** for being *truly* enthusiastic and sharing your expertise with us; thanks **Yiying Tong** and **Beibei Liu** for being patience and helping me derive many and many equations; thanks **Jin Huang** and **Siwang Li** for making the T. Rex alive; thanks **Keenan Crane** for talking "zucchini" and rendering a Möbius banana. Thank you Caltech friends: **Richard Chen**, **Peyman Tavallali**, **Eyal En Gad**, **Manuel Monge**, **Luis Otero**, **Eldar Akhmetgaliyev**, **Matthew Faulkner**. Thank you geometry lab: **Corentin Wallez**, **Dmitry Pavlov**, **Max Budninskiy**, **Tomasz Tyranowski**, **Henry Jacobs**.

Obrigado a toda a minha família, avós, primos, e tios. Obrigado **João**, **Mari** e **Rafa** por nos apoiarem e entenderem nossas rotinas malucas. Obrigado **Vi** por ser sempre firme e forte. Obrigado **mãe** por me fazer acreditar nos meus sonhos. Obrigado **pai** por me mostar que conhecimento pode nos levar além. Obrigado **Ju** por topar esta aventura, por compartilhar desafios e conquistas, por estar ao meu lado, pela sua amizade, pelo nosso amor.

At last, I would like to thank my adivosr **Mathieu Desbrun**. I have learned so much with you. Thanks for being such a special person, examplar, dedicated, tireless, courteous, and kind. Thank you for believing in my ideas, encouraging me, and keeping me focus. *It was great to work for you. It is great to work with you. It is great to work with a friend!*

Abstract

This thesis introduces new tools for geometric discretization in computer graphics and computational physics. Our work builds upon the duality between weighted triangulations and power diagrams to provide concise, yet expressive discretization of manifolds and differential operators. Our exposition begins with a review of the construction of power diagrams, followed by novel optimization procedures to fully control the local volume and spatial distribution of power cells. Based on this power diagram framework, we develop a new family of discrete differential operators, an effective stippling algorithm, as well as a new fluid solver for Lagrangian particles. We then turn our attention to applications in geometry processing. We show that orthogonal primal-dual meshes augment the notion of local metric in non-flat discrete surfaces. In particular, we introduce a reduced set of coordinates for the construction of orthogonal primal-dual structures of arbitrary topology, and provide alternative metric characterizations through convex optimizations. We finally leverage these novel theoretical contributions to generate well-centered primal-dual meshes, sphere packing on surfaces, and self-supporting triangulations.

Contents

Chapter 1

Introduction

Developing reliable numerical methods is a fundamental task at the core of computer science, computational physics, and scientific computing. In essence, a numerical method provides an algorithmic description of mathematical equations. These computational tools can thus be applied to simulate natural phenomena, analyze the stability of engineering structures, and even design and experiment with new theoretical models.

A key challenge in computational modeling is the process of *discretization*: how can one encode smooth mathematical concepts through a finite dimensional representation amenable to computations? Ultimately, the discretization must be accurate, converging to the continuous objects as more computational resources are allocated. Based on this requirement, existing discretization techniques make use of finite set of samples that minimizes an approximation error with respect to the continuous formulation. However, naive sampling strategies may lead to suboptimal approximations, failing to capture structural features of the smooth problem and thus negatively impacting the stability and predictive power of the numerical methods. These limitations are often found in the discretization of physical systems, such as the locking problem in finite-element methods for incompressible elasticity, the appearance of spurious modes in the solution of vector field wave equations, and the numerical dissipation of energy and momenta in computer animations, to cite a few.

This thesis investigates new discretization tools with the goal of offering structure preserving models with bounded approximation error. Intuitively speaking, we aim at designing a finite set of "coordinates" that are *computationally simple* to store and manipulate, while being sufficiently *expressive* to reproduce structural properties of the continuous problem at the discrete realm. Our work is motivated by a series of recent methods, referred to as *geometric discretization*, that leverages the underlying geometry of the continuous problem as a guiding principle to develop reliable discretized formulations. In particular, we introduce new tools for the construction of power diagrams and their dual orthogonal triangulations as a means to derive discrete models of differential operators with improved accuracy, while being computationally efficient.

Related Work

Geometric discretization has led to a rich body of work in applied mathematics, computational physics, and computer science. We review next a few topics related to our work.

Discrete Differential Geometry. Geometry processing applications rely heavily on differential geometry objects, such as curvature, parameterization, and the Laplacian operator. These concepts are particularly suitable for geometric discretization, due to their inherent geometric definition. Meyer et al. [Mey04], for instance, made use of the characterization of mean cuvature vectors as local area gradients and proposed an approximation of curvature operators for triangle meshes. In [Hir03], a discrete version of exterior calculus, including operators such as the divergence and the Laplacian, was introduced for simplicial complexes by leveraging the notion of chain and cochains. This framework led to a discrete notion of Hodge decomposition for piecewise linear vector fields based on the encoding of vectors as scalars per oriented edges interpolated with Whitney basis functions [Whi57]. In a similar fashion, Wardetzky [War06a] proposed a discretization of Hodge decomposition for piecewise constant vector fields based on conforming basis functions at vertices and non-conforming functions at edges. In [Spr03], a conformal equivalence of triangle meshes was presented by discretizing curvature as vertex tip angles and metric as edge lenghts. Our work builds on these previous methods, and proposes an augmented discretization of metric tensors and divergence-free tensor fields through the construction of primal-dual structures on triangulations.

Discrete Mechanics. Geometric discretization is also benefitial for the simulation of mechanical systems. Variational integrators [HLW10], in particular, are based on the discretization of the Hamiltonian of physical systems instead of their subsequent equations of motions. By formulating a discrete Hamiltonian, symmetries and invariants can be reliably identified and preserved throughout the discretization. The work of [Kha10], for example, employed variational integrators for elasticity simulation, enforcing energy and momentum preservation. In [Wes04, Lew03], variational integrators were combined with asynchronous time stepping, and then applied to contact mechanics [Har10, Vou13]. Bergou [Ber10] used geometric discretization to simulate elastic rods and viscous threads. Finally, Mullen [Mul11] proposed an Eulerian geometric discretization through the use of triangulations and their dual diagrams, serving as the foundation for structure preserving simulation methods in computational fluid dynamics [Pav09], electromagnetics [Ste09], and magnetohydrodynamics [Gaw10]. We instead develop a new Lagrangian method for fluid simulation based on power diagrams.

Contributions and Outline

This thesis is organized in two main parts. We first address geometric discretization in Euclidean spaces based on the construction of power diagrams, and then augment the discretization of surfaces through weighted triangulations. The interplay between weighted triangulations and power diagrams is at the core of our approach to geometric discretization, leading to a concise and efficient representation for discrete manifolds and differential operators. Much of the work presented in this thesis appears in the following publications:

- de Goes, Breeden, Ostromoukhov, Desbrun
  Blue Noise through Optimal Transport
  SIGGRAPH Asia 2012 / ACM Transactions on Graphics, 31(6), 2012.
- de Goes, Alliez, Owhadi, Desbrun
  On the Equilibrium of Simplicial Masonry Structures
  SIGGRAPH 2013 / ACM Transactions on Graphics, 32(4), 2013.
- de Goes, Memari, Mullen, Desbrun
  Weighted Triangulations for Geometry Processing
  ACM Transactions on Graphics, 33(3), 2014.
- de Goes, Wallez, Huang, Pavlov, Desbrun
  Power Particles: a new Lagrangian method for fluid simulation
  (in preparation).

We start our presentation by revisiting basics definitions on the construction of power diagrams in Chapter 2. We also introduce new geometric tools to control the shape of power cells, including an optimization procedure that generates volume-controlled partitions and evenly spaced sampling in domains of arbitrary shape and topology. We then employ power diagrams to derive a novel discretization of calculus that verifies structural geometric properties such as the linear accuracy for the gradient and Laplacian operators. In Chapter 3, we make use of power diagrams to design a new blue noise sampling algorithm for image stippling that outperforms existing methods both in efficiency and quality. In Chapter 4, we present a new Lagrangian method for fluid simulation that preserves local volumes and divergence-free velocity fields, while maintaining even-spaced particle distributions for large time-steps. Chapter 5 introduces a generalization of power diagrams as cell complexes on triangulated surfaces. Our representation of surfaces through primal-dual structures provides an augmented discretization of metric tensors, and thus leads to an extension of the discrete exterior calculus framework with improved accuracy. We detail in Chapter 6 the characterization of primal-dual meshes based on convex optimization routines that combine geometric measurements such as angle, curvature, length ratio, and area. Finally, we show that divergence-free tensor fields can be discretized by constructing orthogonal dual diagrams on planar triangulations. Reduced coordinates for discrete divergence-free tensors are then employed for the design of simplicial masonry structures.

Chapter 2

Geometric Discretization through Power Diagrams

In this chapter, we describe the key concepts in geometric discretizations based on power diagrams. We start by spelling out basic definitions of cell complexes and the construction of power diagrams. We then introduce new discrete differential operators derived directly from power diagrams and detail their structure-preserving features. Departing from previous work, we consider power diagrams as our primal discretization, while dual triangulations provide a complementary geometric structure. These concepts will serve as the foundation of our computational tools for meshing and computational fluid dynamics in Chapters 3 and 4.

2.1 Preliminaries

We begin with the definition of cell complexes as a combinatorial structure.

Definition 2.1.1. *A cell complex is a partitioning of $\mathbb{R}^d$ into a collection of n cells $\mathcal{V}=\{\mathcal{V}_i\}_{i=1..n}$ satisfying two conditions:*

1. *Every cell $\mathcal{V}_i$ has trivial topology, i.e., it is homeomorphic to a d-dimensional ball;*
2. *The intersection of any pair of cells $\mathcal{V}_i$ and $\mathcal{V}_j$, denoted as the cell facet $\mathcal{A}_{ij}$, is either empty or a subset of co-dimension one.*

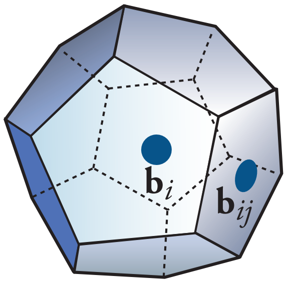
In order to tile Euclidean space entirely, cells may be unbounded and we refer to these cells as *infinite cells*. We also define a *vertex* of a complex in $\mathbb{R}^d$ as the intersection of at least $d+1$ cells. Cell complexes are said to be *combinatorially equivalent* if they share the same combinatorics. This equivalence relation characterizes, for instance, different immersions of a cell complex. Similarly, combinatorially equivalent complexes lead to a notion of Poincaré duality [Mun84].

Definition 2.1.2. *A pair of cell complexes $\mathcal{V}$ and $\widetilde{\mathcal{V}}$ is said to be dual to one another if there exists a duality map $*$ from $\mathcal{V}$ to $\widetilde{\mathcal{V}}$ that associates any element σ of i-dimension $(i = 0, .., d)$ in $\mathcal{V}$ to an element $*\sigma$ of $(d-i)$-dimension in $\widetilde{\mathcal{V}}$, and that verifies the combinatorial condition: $\sigma_i \subset \sigma_j \iff *\sigma_i \supset *\sigma_j$. We call such a pair of complexes a primal-dual complex, where $\mathcal{V}$ is referred to as the primal complex, and $\widetilde{\mathcal{V}}$ as the dual complex.*

We now consider the geometry of cell complexes by enriching every cell with an affine structure.

Definition 2.1.3. *A polytopal complex in $\mathbb{R}^d$ is a cell complex such that every cell corresponds to a d-dimensional polytope.*

As an example, a polytopal cell $\mathcal{V}_i$ corresponds to a polyhedron in 3D (resp., a polygon in 2D), while a polytopal facet $\mathcal{A}_{ij}$ is represented by a planar polygon in 3D (resp., a line segment in 2D). In our notation, we use V_i to indicate the volume of cell $\mathcal{V}_i$, and A_{ij} as the volume of facet $\mathcal{A}_{ij}$. We also denote the centroid (i.e., center of mass) of $\mathcal{V}_i$ as $\mathbf{b}_i$, and the centroid of $\mathcal{A}_{ij}$ as $\mathbf{b}_{ij}$. The affine structure of polytopal complexes can be further used to analyze the geometry of primal-dual complexes.



Definition 2.1.4. *A primal-dual complex is called orthogonal when the supporting hyper-planes of any primal element σ and of its dual element $*\sigma$ meet at an angle of $\pi/2$.*

Finally, we introduce two special categories of polytopal complexes.

Definition 2.1.5. *A convex complex is a polytopal complex with convex cells.*

Definition 2.1.6. *A simple complex in $\mathbb{R}^d$ is a convex complex such that every vertex is a vertex of exactly $d+1$ cells.*

2.2 Power Diagrams

We describe next a special family of cell complexes called *power diagrams*, which are constructed as the intersection of half-spaces. To this end, we first introduce the notion of power distance.

Definition 2.2.1. *For a given point $\mathbf{q} \in \mathbb{R}^d$ and scalar $w \in \mathbb{R}$, the power distance from any point $\mathbf{x} \in \mathbb{R}^d$ to $(\mathbf{q}, w)$ is defined as: $\|\mathbf{x} - \mathbf{q}\|^2 - w$, where $\|\cdot\|$ indicates the Euclidean L_2 norm.*

As an additive modification of the Euclidean metric, the power distance draws the separation of any pair of weighted points $(\mathbf{q}_i, w_i)$ and $(\mathbf{q}_j, w_j)$ as a hyper-plane with points $\mathbf{x}$ verifying $\|\mathbf{x}-\mathbf{q}_i\|^2 - w_i = \|\mathbf{x}-\mathbf{q}_j\|^2 - w_j$. Therefore, the difference of weights acts as a shift of the hyper-planes towards the smallest weight. By associating cells $\mathcal{V} = \{\mathcal{V}_i\}_i$ to weighted points $(\mathbf{q}, w) = \{(\mathbf{q}_i, w_i)\}_i$, we can further employ the power distance to compute a cost function $\mathcal{E}$ on cell complexes:

$$\mathcal{E}(\mathbf{q}, w, \mathcal{V}) := \sum_i \int_{\mathcal{V}_i} \left(\|\mathbf{x} - \mathbf{q}_i\|^2 - w_i\right) d\mathbf{x}. \tag{2.1}$$

For a fixed list of weighted points, the cell complex minimizing energy $\mathcal{E}$ is uniquely determined as the intersection of half-spaces induced by the weighted bisectors, leading to the construction of power diagrams [Aur87b].

Definition 2.2.2. *A power diagram is a cell complex $\mathcal{V}$ defined by a list of n points $\{\mathbf{q}_i\}_{i=1..n}$, called* sites, *and their associated scalar values $\{w_i\}_{i=1..n}$, called* weights. *For each weighted point $(\mathbf{q}_i, w_i)$, its power cell $\mathcal{V}_i$ is defined as:*

$$\mathcal{V}_i = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x} - \mathbf{q}_i\|^2 - w_i \leq \|\mathbf{x} - \mathbf{q}_j\|^2 - w_j \; \forall j\}. \tag{2.2}$$

Notice that the definition of power cells in Equation (2.2) is invariant to an additive translation of the weights. In particular, when weights are all equal, power diagrams reduce to Voronoi diagrams [Vor08, ES85].

The case of Voronoi diagrams also determines a dual triangulation, known as Delaunay triangulation, in which a vertex of the cell complex is placed at the circumcenter of its respective dual simplex. Similarly, a power diagram defines a dual triangulation, called a *weighted Delaunay triangulation*, that connects sites $\mathbf{q}_i$ and $\mathbf{q}_j$ sharing a power facet $\mathcal{A}_{ij}$. We indicate by $\mathcal{N}_i$ the set of neighboring sites sharing a power facet with site i, and use l_{ij} to indicate the distance $\|\mathbf{q}_i - \mathbf{q}_j\|$. The vertices of a power diagram correspond to weighted circumcenters, a generalization of circumcenters constructed as the intersection of weighted bisectors. More concretely, for a simplex σ formed by $d{+}1$ sites $\{\mathbf{q}_i\}_{i=0..d}$ in $\mathbb{R}^d$ with weights $\{w_i\}_{i=0..d}$, the weighted circumcenter $\mathbf{c}_\sigma$ is expressed as:

$$\mathbf{c}_\sigma = \mathbf{q}_0 + \frac{1}{2d!|\sigma|} \sum_{i=1..d} \left(l_{0i}^2 + w_0 - w_i\right) \boldsymbol{\sigma}_i, \tag{2.3}$$

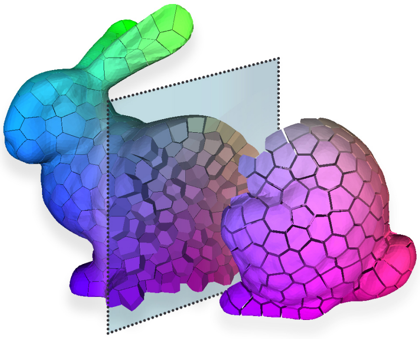
where $|\sigma|$ indicates the volume of σ, and $\boldsymbol{\sigma}_i$ denotes the inward-pointing normal of the face of σ opposite to $\mathbf{q}_i$ scaled by the volume of the face. We thus conclude that any power diagram and its weighted Delaunay triangulation form an orthogonal primal-dual complex.

The geometry of power diagrams is easily manipulated by moving a site's location or adjusting its weight. The variational nature of power diagrams as the minimizer of $\mathcal{E}$ (Equation (2.1)) also unveils that the space of power diagrams is continuous with respect to both $\mathbf{q}$ and w, even through local connectivity changes [MMdGD11]. In fact, an edge flip in a weighted Delaunay triangulation corresponds to a vanishing facet in the power diagram. It is worth noticing, however, that drastic differences between weights may induce empty power cells, and thereby isolate sites with no neighbors. We denote sites associated with zero volume cells as *hidden sites* [Aur87b].

Algorithms to construct power diagrams make use of the same strategies as in the case of Voronoi diagrams, with the caveat that Euclidean distances are replaced by power distances. Therefore, the worst time complexity to generate a power diagram with n weighted points is $\mathcal{O}(n \log n)$ in $\mathbb{R}^2$ and $\mathcal{O}(n^{\lceil d/2 \rceil})$ in $\mathbb{R}^{d>2}$. Efficient implementations of these algorithms, such as in the CGAL library [CGA14], have reported that the complexity bound is typically reached in cases with coplanar sites, while the general case performs close to linear in the number of sites.

The construction and properties of power diagrams can be further extended to any closed d-dimensional subset $\Omega \subset \mathbb{R}^d$, of arbitrary shape and topology, and with a manifold boundary $\partial\Omega$. The only additional requirement is a clipping procedure that intersects power cells with the domain boundary $\partial\Omega$, enforcing bounded power cells.

Definition 2.2.3. *For a domain $\Omega \subset \mathbb{R}^d$, we define a clipped power diagram of Ω as a power diagram generated by weighted points $\{(\mathbf{q}_i, w_i)\}_i$ clipped at the domain boundary $\partial\Omega$.*



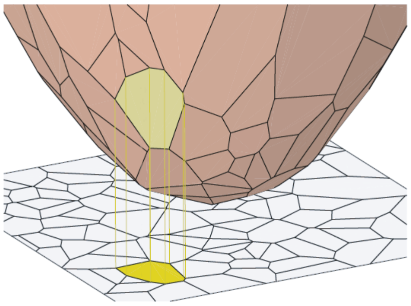
Power cells that intersect the boundary $\partial\Omega$ can be clipped by either using the Sutherland-Hodgman algorithm [YWLL13], or based on mirrored copies of the subset of sites with dual facets straddling $\partial\Omega$. If the domain Ω is periodic, a similar mirroring procedure can be used as well [YWLA11]. The resulting clipped cells form finite (and possibly non-convex) polytopes (inset), from which one can compute all relevant geometric quantities (centroids $\mathbf{b}_i$ and $\mathbf{b}_{ij}$, volumes V_i, areas A_{ij}, and lengths l_{ij}).

2.3 Cell Complexes versus Power Diagrams

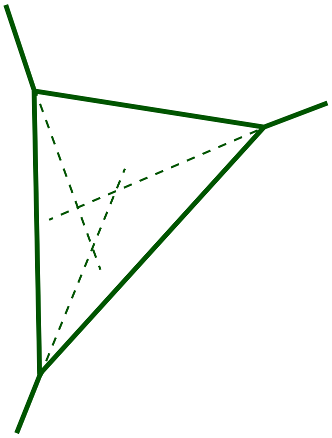
The geometric parameterization offered by site locations and weights makes power diagrams a convenient representation of cell complexes. In the following, we discuss necessary and sufficient conditions to characterize cell complexes as power diagrams in $\mathbb{R}^d$.

Convexity & Orthogonality. We first observe that power diagrams in $\mathbb{R}^d$ are *convex* complexes, a direct implication of Equation (2.1). We thus conclude that *convexity* is a necessary condition for a power diagram. In the previous section, we also showed that power diagrams form an orthogonal primal-dual complex with weighted Delaunay triangulations. Aurenhammer et al. [Aur87a] proved that this property is not only necessary, but it is also a *sufficient* condition for power diagrams. This result implies that there is a one-to-one mapping from weighted Delaunay triangulations to power diagrams. It bears to pointing out, however, that not any simplicial complex is a weighted Delaunay triangulation: even though every triangulation is combinatorially equivalent to a cell complex, such dual complexes may not be convex and therefore may differ from a power diagram.

Lifting. The duality to weighted Delaunay triangulations also provides an alternative interpretation of power diagrams via the *lifting* procedure [Aur87b]. In this approach, every site $\mathbf{q}_i$ in $\mathbb{R}^d$ with weight w_i is lifted to $\mathbb{R}^{d+1}$ by appending an additional Cartesian coordinate of the form $\|\mathbf{q}_i\|^2 - w_i$. The weighted Delaunay triangulation is then constructed as the orthographic projection to $\mathbb{R}^d$ of the convex hull of the lifted points. Similarly, the power diagram is constructed as the lower envelope tangent to the convex hull (see inset). Observe that both complexes rely exclusively on the convex paraboloid in $\mathbb{R}^{d+1}$ induced by the lifted points. We can thus convert any triangulation into a weighted Delaunay triangulation by computing the convex hull of its lifted vertices, which may require edge flips as well as the removal of (hidden) sites lifted above the paraboloid envelope.



Simple Complexes. The valence of a vertex in a power diagram is bounded from below by the number of facets of the associated simplex in the dual triangulation. Therefore, the valence in $\mathbb{R}^d$ numbers at least d+1. If we further assume that sites $\{\mathbf{q}_i\}_i$ are in general position, we then enforce non-vanishing power facets and this counting matches exactly to d+1. We thus conclude that power diagrams with sites in general position correspond to *simple* complexes. The converse statement was investigated in [Dav59, Aur87a], which showed that any simple complex in $\mathbb{R}^d$ (for $d \geq 3$) is also a power diagram. Note, however, that not any 2D simple complex is a power diagram, as illustrated by the counter-example in the inset.



Generalizations. At last, we point out that alternative partitionings have been proposed to encode more general cell complexes. In [MMD11], for instance, a non-orthogonal primal-dual complex was presented by adding one displacement vector per site while retaining the affine structure of cells. Extensions of

Voronoi diagrams based on anisotropic metrics were also introduced to generate curved complexes, at the cost of possibly creating multiple disconnected components per cell (known as orphans) [LS03]. More recent efforts have shown mild conditions on the metric anisotropy in order to guarantee orphan-free curved complexes [CG11] and the existence of a dual triangulation [CG12].

2.4 Discrete Differential Operators

In this section, we detail the geometric discretization of differential operators on scalar and vector fields derived from power diagrams. We discretize scalar and vector fields on a power diagram using one sample per cell. For example, a scalar function f (resp., a vector field $\mathbf{v}$) is approximated as a piecewise constant function with value f_i (resp., vector $\mathbf{v}_i$) at cell $\mathcal{V}_i$. In the following, we consider discrete differential operators for power diagrams tessellating a domain Ω, which can be either periodic (i.e., a unit box with identification of opposite faces) or bounded.

Identities. Reynolds transportation theorem will be a useful tool for our derivations, as it states that the rate of change of the integral of a scalar function f within a region $\mathcal{V}$ is equal to the region integral of the change of f, plus the boundary integral of the rate at which f flows through the boundary $\partial\mathcal{V}$ with outward unit normal $\mathbf{n}$, i.e.:

$$\nabla\left(\int_{\mathcal{V}} f(\mathbf{x})\ d\mathbf{x}\right) = \int_{\mathcal{V}} \nabla f(\mathbf{x}) d\mathbf{x} + \int_{\partial\mathcal{V}} f(\mathbf{x})\ (\nabla\mathbf{x})^t\, \mathbf{n}(\mathbf{x})\, d\mathbf{x}. \tag{2.4}$$

When applied to power cell volumes, this theorem simplifies to boundary integrals (in $\mathbf{q}$ or w):

$$\nabla V_i = \sum_{j\in\mathcal{N}_i} \frac{1}{l_{ij}} \int_{\mathcal{A}_{ij}} (\nabla\mathbf{x})^t\, (\mathbf{q}_j - \mathbf{q}_i)\ d\mathbf{x}. \tag{2.5}$$

We also point out that any point $\mathbf{x}$ on a power facet $\mathcal{A}_{ij}$ can be expressed as:

$$\mathbf{x} = \frac{1}{2}\,(\mathbf{q}_i + \mathbf{q}_j) + \frac{w_i - w_j}{2\, l_{ij}^2}\,(\mathbf{q}_j - \mathbf{q}_i) + s_{\mathbf{x}} \mathbf{R}_{\mathbf{x}}\,(\mathbf{q}_j - \mathbf{q}_i)\,, \tag{2.6}$$

where the three terms indicate, respectively, the midpoint of the edge connecting $\mathbf{q}_i$ and $\mathbf{q}_j$, a displacement along the edge vector $\mathbf{q}_j - \mathbf{q}_i$, and finally a displacement within the facet $\mathcal{A}_{ij}$ (thus orthogonal to $\mathbf{q}_j - \mathbf{q}_i$) of magnitude $s_{\mathbf{x}}$ along the rotated edge $\mathbf{R}_{\mathbf{x}}\,(\mathbf{q}_j - \mathbf{q}_i)$, with $\mathbf{R}_{\mathbf{x}}$ as a d-dimensional rotation matrix. At last, as cell volumes $\{V_i\}_i$ sum to the total volume of the domain, we have:

$$\sum_i V_i = |\Omega| \quad\implies\quad \nabla_i V_i = -\sum_j \nabla_i V_j. \tag{2.7}$$

Divergence. Based on the well-defined notion of volume per power cell (namely, V_i), divergence can be measured as the change of local volume induced by the motion of a site. Consequently, we define a discrete divergence operator $\mathbf{D}$ as the volume gradient induced by displacing sites, i.e., $\mathbf{D} = \nabla_{\mathbf{q}} V$.

Definition 2.4.1. *For n power cells in a d-dimensional domain Ω, the discrete divergence operator $\mathbf{D}$ is constructed as an $n \times n$ matrix with row-valued entries:*

$$\begin{cases} \mathbf{D}_{ij} = \left(\nabla_{\mathbf{q}_j} V_i\right)^t = \dfrac{A_{ij}}{l_{ij}}\,(\mathbf{q}_j - \mathbf{b}_{ij})^t, \\[2ex] \mathbf{D}_{ii} = \left(\nabla_{\mathbf{q}_i} V_i\right)^t = -\displaystyle\sum_{j\in\mathcal{N}_i} \left(\nabla_{\mathbf{q}_i} V_j\right)^t. \end{cases} \tag{2.8}$$

Lemma 2.4.2. *The divergence* $\mathbf{Dv}$ *of a discrete vector field* $\mathbf{v} = \{\mathbf{v}_i\}_i$ *is expressed as:*

$$[\mathbf{Dv}]_i = \sum_{j\in\mathcal{N}_i} (\mathbf{D}_{ij}\mathbf{v}_j - \mathbf{D}_{ji}\mathbf{v}_i) = \sum_{j\in\mathcal{N}_i} \frac{A_{ij}}{l_{ij}} \big((\mathbf{q}_j - \mathbf{b}_{ij})^t\,\mathbf{v}_j - (\mathbf{q}_i - \mathbf{b}_{ij})^t\,\mathbf{v}_i \big). \tag{2.9}$$

Proof. The diagonal term $\mathbf{D}_{ii}$ is a direct result of Equation (2.7), while the off-diagonal term $\mathbf{D}_{ij}$ is obtained by observing that the directional derivative of $s_\mathbf{x}\mathbf{R}_\mathbf{x}\,(\mathbf{q}_j - \mathbf{q}_i)$ along $\mathbf{q}_j{-}\mathbf{q}_i$ is idempotent (due to the orthogonality between these two vectors), i.e.:

$$\forall \mathbf{x} \in \mathcal{A}_{ij}, \quad (\nabla_{\mathbf{q}_j}\mathbf{x})^t\,(\mathbf{q}_j - \mathbf{q}_i) = \mathbf{q}_j - \mathbf{x}. \tag{2.10}$$

Combining Equation (2.10) with Equation (2.5), the off-diagonal term thus reduces to:

$$\nabla_{\mathbf{q}_j} V_i = \frac{1}{l_{ij}} \int_{\mathcal{A}_{ij}} (\mathbf{q}_j - \mathbf{x})\; d\mathbf{x} = \frac{A_{ij}}{l_{ij}}\,(\mathbf{q}_j - \mathbf{b}_{ij})\,. \tag{2.11}$$

□

Note that, as in the continuous setting, our discrete divergence verifies $\int_\Omega \nabla\cdot\mathbf{v}(\mathbf{x})\;d\mathbf{x} = 0$. This is easily shown by simple algebric manipulation:

$$\sum_i [\mathbf{Dv}]_i = \sum_i\sum_j \mathbf{v}_j^t\,(\nabla_{\mathbf{q}_j}V_i) = \sum_j \mathbf{v}_j^t \underbrace{\left(\nabla_{\mathbf{q}_j}\sum_i V_i\right)}_{=0} = 0. \tag{2.12}$$

Curl. In 2D and 3D, we can use the discrete divergence to define a curl operator $\mathbf{C}$, which is written for a given a vector field $\{\mathbf{v}_i\}_i$ as:

$$[\mathbf{Cv}]_i = \sum_{j\in\mathcal{N}_i} \frac{A_{ij}}{l_{ij}} \big((\mathbf{q}_j - \mathbf{b}_{ij}) \times \mathbf{v}_j - (\mathbf{q}_i - \mathbf{b}_{ij}) \times \mathbf{v}_i \big). \tag{2.13}$$

Gradient. We can further employ the divergence operator to derive a discrete gradient based on Stokes' theorem:

$$\int_\Omega \nabla f(\mathbf{x})^t\,\mathbf{v}(\mathbf{x})d\mathbf{x} + \int_\Omega f(\mathbf{x})\,\nabla{\cdot}\mathbf{v}(\mathbf{x})d\mathbf{x} = \int_{\partial\Omega} f(\mathbf{x})\,\big(\mathbf{v}(\mathbf{x})^t\mathbf{n}(\mathbf{x})\big)\,d\mathbf{x}, \tag{2.14}$$

where f is a scalar function and $\mathbf{v}$ is a vector field. While the right-hand term is zero for periodic domains, it requires proper boundary conditions for arbitrarily bounded domains. By assuming Neumann boundary conditions (i.e., $\mathbf{v}(\mathbf{x})^t\mathbf{n}(\mathbf{x}) = 0\ \forall\mathbf{x} \in \partial\Omega$), we can then define the gradient operator $\mathbf{G}$ as follows.

Definition 2.4.3. *For n power cells in a d-dimensional domain Ω, the discrete gradient operator* $\mathbf{G}$ *is defined as the volume-weighted negated transpose of the divergence operator, i.e.,*

$$\mathbf{G} {=} -\operatorname{diag}(V)^{-1}\mathbf{D}^t,$$

where $\operatorname{diag}(V)$ *is the diagonal matrix containing every cell volume. For a discrete function $\{f_i\}_i$, the gradient then reduces to:*

$$[\mathbf{G}f]_i = \frac{1}{V_i}\sum_{j\in\mathcal{N}_i} (\mathbf{D}_{ji})^t\,(f_i - f_j) = \frac{1}{V_i}\sum_{j\in\mathcal{N}_i} \frac{A_{ij}}{l_{ij}}\,(\mathbf{q}_i - \mathbf{b}_{ij})\,(f_i - f_j). \tag{2.15}$$

Proof. Based on Equation (2.14), we have that:

$$\forall f, \mathbf{v}, \quad \mathbf{v}^t\,\big(\operatorname{diag}(V)\mathbf{G} + \mathbf{D}^t\big)\,f = 0 \quad\Longrightarrow\quad \mathbf{G} {=} -\operatorname{diag}(V)^{-1}\mathbf{D}^t.$$

□

We can also verify the following identities related to the discrete gradient operator $\mathbf{G}$.

Proposition 2.4.4. *Let $f = \{f_i\}_i$ be a discrete function with Dirichlet boundary conditions (i.e., $f_i = 0$ for any cell $\mathcal{V}_i \cap \partial\Omega \neq 0$). The discrete gradient $\mathbf{G}$ satisfies:*

$$\begin{cases} \sum_i V_i \left[\mathbf{G}f\right]_i = 0, & (2.16a) \\ \sum_i V_i \left[\mathbf{G}f\right]_i \times \mathbf{q}_i = 0. & (2.16b) \end{cases}$$

Proof. For the first identity, we have:

$$\sum_i V_i \left[\mathbf{G}f\right]_i = \sum_i f_i \left(\sum_j \nabla_{\mathbf{q}_i} V_j \right) = \sum_i f_i \left(\sum_{j \in \mathcal{N}_i} \frac{A_{ij}}{l_{ij}} \left(\mathbf{q}_i - \mathbf{q}_j\right) \right) = \sum_i f_i \int_{\partial\mathcal{V}_i} (-\mathbf{n}) \, d\mathbf{x} = 0.$$

For the second identity, we have:

$$\sum_i V_i \left[\mathbf{G}f\right]_i \times \mathbf{q}_i = \sum_i f_i \left(\sum_j \nabla_{\mathbf{q}_i} V_j \times \mathbf{q}_j \right) = \sum_i f_i \left(\sum_{j \in \mathcal{N}_i} \frac{A_{ij}}{l_{ij}} \mathbf{b}_{ij} \times \left(\mathbf{q}_i - \mathbf{q}_j\right) \right) = \sum_i f_i \int_{\partial\mathcal{V}_i} \mathbf{n} \times \mathbf{x} \, d\mathbf{x} = 0.$$

□

Laplacian. With these discrete operators, one can now assemble a discrete Laplace operator as $\mathbf{L} = -\mathbf{DG}$. Despite being consistent with the divergence and gradient operators as well as positive semi-definite, this linear operator has a large stencil involving two-away neighbors. Instead we propose a sparser, staggered alternative for the discrete Laplacian that improves computational cost.

Definition 2.4.5. *The discrete Laplace operator Δ is defined as twice the local volume gradient with respect to weights, i.e., $\Delta = 2\nabla_w V$, corresponding to an $n \times n$ symmetric matrix with entries:*

$$\begin{cases} \Delta_{ij} = 2\nabla_{w_j} V_i = -\dfrac{A_{ij}}{l_{ij}}, \\ \Delta_{ii} = 2\nabla_{w_i} V_i = \sum_{j \in \mathcal{N}_i} \dfrac{A_{ij}}{l_{ij}}. \end{cases} \tag{2.17}$$

Proof. The diagonal term Δ_{ii} is a direct consequence of Equation (2.7). For the off-diagonal term Δ_{ij}, we notice that only the second term in Equation (2.6) depends on the weights, which leads to:

$$\forall \mathbf{x} \in \mathcal{A}_{ij}, \quad \left(\nabla_{w_j} \mathbf{x}\right)^t \left(\mathbf{q}_j - \mathbf{q}_i\right) = -\frac{1}{2}. \tag{2.18}$$

Combining Equation (2.18) with Equation (2.4) results in the off-diagonal terms Δ_{ij}. □

Observe that our construction of the discrete Laplacian Δ is equivalent to a finite-volume discretization applied to power cells, with Neumann boundary conditions. We also notice that this discrete operator corresponds to a weighted Laplacian [Gli07, MMdGD11] and, as such, verifies all the properties advocated in [WMKG07] for the construction of a "perfect" Laplacian.

Proposition 2.4.6. *The discrete Laplace operator Δ satisfies (1) locality; (2) symmetry; (3) positive semi-definiteness; (4) linear precision (for interior nodes).*

Proof. The first two properties are trivially verified based on Equation (2.17). Because the coefficients A_{ij}/l_{ij} are non-negative by construction, the third property follows from the quadratic form:

$$\forall f, \quad f^t \Delta f = \sum_{ij} A_{ij}/l_{ij} \left(f_i - f_j\right)^2 \geq 0.$$

To verify the last property, we discretize linear functions as $\{f_i = \mathbf{a}^t \mathbf{q}_i + b\}_i$, where $\mathbf{a}$ is a constant vector and b is a scalar, which leads to $\Delta f = \mathbf{a}^t \Delta \mathbf{q}$. For interior cells, we point out that $[\Delta \mathbf{q}]_i$ is equivalent to the closeness of the boundary loop of the power cell $\mathcal{V}_i$, and thence $[\Delta \mathbf{q}]_i = 0$. For boundary cells, the term $[\Delta \mathbf{q}]_i$ corresponds to the integration of boundary normals over $\mathcal{V}_i \cap \partial\Omega$. □

2.5 Power Diagrams with Constraints

We now show that power diagrams can be constructed by enforcing geometric properties. In particular, we consider constraints on cell volumes and on site locations. To this end, we first assign a target volume $\overline{V}_i$ for each cell $\mathcal{V}_i$ such that $\sum_i \overline{V}_i = |\Omega|$, and then introduce a new energy $\mathcal{F}$ as a slight modification of $\mathcal{E}$ (Equation (2.1)):

$$\mathcal{F}(\mathbf{q}, w) = \sum_i w_i \overline{V}_i + \mathcal{E}(\mathbf{q}, w, \mathcal{V}) = \sum_i \left(\int_{\mathcal{V}_i} \|\mathbf{x} - \mathbf{q}_i\|^2 d\mathbf{x} \right) - \sum_i w_i \left(V_i - \overline{V}_i \right), \tag{2.19}$$

where $\{\mathcal{V}_i\}_i$ are the power cells generated by the weighted points $(\mathbf{q}, w)$.

2.5.1 Volume Control

In contrast to Voronoi diagrams, power diagrams offer additional degrees of freedom (namely, the weights) to fully control the connectivity and volume of power cells. Indeed, Aurenhammer et al. [AHA98] showed that, for a given set of sites $\{\mathbf{q}_i\}_i$, any arrangement of power cell volumes is determined by a unique power diagram. This result is an implication of the following theorem.

Theorem 2.5.1. *For given sites $\mathbf{q} = \{\mathbf{q}_i\}_i$, each associated with a target volume $\overline{V}_i$, the energy $\mathcal{F}$ is concave with respect to weights $w = \{w_i\}_i$ of a power diagram.*

Proof. By applying Reynolds theorem (Equation (2.4)), we compute the gradient of $\mathcal{F}$ as:

$$\begin{aligned}\nabla_{w_i} \mathcal{F} &= \overline{V}_i + \nabla_{w_i} \mathcal{E} \\ &= \overline{V}_i - V_i + \sum_{j \in \mathcal{N}_i} \frac{1}{l_{ij}} \int_{\mathcal{A}_{ij}} \underbrace{\left(\left(\|\mathbf{x} - \mathbf{q}_i\|^2 - w_i \right) - \left(\|\mathbf{x} - \mathbf{q}_j\|^2 - w_j \right) \right)}_{=0 \ \forall \mathbf{x} \in \mathcal{A}_{ij}} \left(\nabla_{w_i} \mathbf{x} \right)^t \left(\mathbf{q}_j - \mathbf{q}_i \right) d\mathbf{x}.\end{aligned}$$

$$\boxed{\therefore \nabla_w \mathcal{F} = \overline{V} - V.} \tag{2.20}$$

Therefore, the critical point of $\mathcal{F}$ enforces the volume constraints. We can also compute the second derivatives of $\mathcal{F}$ based on Equation (2.17):

$$\mathrm{Hess}_w(\mathcal{F}) = \mathrm{Hess}_w(\mathcal{E}) = -\nabla_w V = -\frac{1}{2}\Delta. \tag{2.21}$$

Since Δ is positive semi-definite (Proposition 2.4.6), we conclude that the energy $\mathcal{F}$ is concave with respect to the weights. □

As a concave optimization, the critical point $\nabla_w \mathcal{F} = 0$ determines a connected isolevel maximizing $\mathcal{F}$. This solution space is characterized by a weight configuration *inside* the optimal set, plus the kernel of the Hessian of $\mathcal{F}$. Note, however, that the null space of Δ corresponds to an additive constant to bounded domains (or linear functions in a periodic domain), and thereby any weight assignment in the optimal set determines the same power diagram. Therefore, volume constraints are enforced by at most one power diagram. The optimal weights for a given set of cell volumes can be efficiently computed based on a Newton solver [NW99]. More specifically, the weights are updated iteratively through a series of sparse linear systems of the form:

$$\frac{1}{2}\Delta\delta w = \overline{V} - V. \tag{2.22}$$

This weight-based volume enforcement extends, as is, to domains of arbitrary shape, topology, and dimension. Recently, Gu et al. [XG13] introduced mild conditions on the spacing between sites and on the convexity of the domain in order to turn the space of weights into a convex set, thereby guaranteeing the existence of a solution for the energy $\mathcal{F}$.

2.5.2 Centroidal Power Diagrams

The energy $\mathcal{F}$ also provides control over the placement of sites $\mathbf{q}$, favoring even distributions. This sampling property leads to the notion of the centroidal power diagram.

Definition 2.5.2. *A centroidal power diagram is a power diagram such that the center of mass $\mathbf{b}_i$ of each cell $\mathcal{V}_i$ coincides with its site $\mathbf{q}_i$, i.e., $\mathbf{q}_i = \mathbf{b}_i$.*

Note that, when weights are all equal, centroidal power diagrams reduce to Centroidal Voronoi Tessellations [DFG99]. The construction of a centroidal power diagram also corresponds to a critical point of the energy $\mathcal{F}$, as described next.

Proposition 2.5.3. *A centroidal power diagram corresponds to the extremization of the energy $\mathcal{F}$ with respect to $\mathbf{q}$.*

Proof. By applying Reynolds theorem (Equation (2.4)), we compute the gradient of $\mathcal{F}$ as:

$$\begin{aligned}\nabla_{\mathbf{q}_i}\mathcal{F} &= \nabla_{\mathbf{q}_i}\mathcal{E}\\ &= 2V_i\left(\mathbf{q}_i - \mathbf{b}_i\right) + \sum_{j\in\mathcal{N}_i}\frac{1}{l_{ij}}\int_{\mathcal{A}_{ij}}\underbrace{\left(\left(\|\mathbf{x}-\mathbf{q}_i\|^2 - w_i\right) - \left(\|\mathbf{x}-\mathbf{q}_j\|^2 - w_j\right)\right)}_{=0\ \forall\mathbf{x}\in\mathcal{A}_{ij}}\left(\nabla_{\mathbf{q}_i}\mathbf{x}\right)^t\left(\mathbf{q}_j - \mathbf{q}_i\right)d\mathbf{x}.\end{aligned}$$

$$\boxed{\therefore\ \nabla_{\mathbf{q}}\mathcal{F} = 2V\left(\mathbf{q}-\mathbf{b}\right).} \tag{2.23}$$

Therefore, the critical point of $\mathcal{F}$ generates a centroidal power diagram:

$$0 = \nabla_{\mathbf{q}_i}\mathcal{F} = 2V_i\left(\mathbf{q}_i - \mathbf{b}_i\right) \iff \mathbf{q}_i = \mathbf{b}_i\ \forall i. \tag{2.24}$$

□

It is worth mentioning that the energy $\mathcal{F}$ is non-convex with respect to $\mathbf{q}$ and, consequently, it may present many critical points. Algorithmically, there are many strategies to construct a centroidal power diagram. We can apply, for instance, a Lloyd approach in which each site $\mathbf{q}_i$ is individually optimized to its cell centroid $\mathbf{b}_i$. Instead we can also use quasi-Newton strategies as proposed in [LWL$^+$09], or a fully Newton solver based on second derivatives. We provide next the closed-form second derivatives of $\mathcal{F}$.

Proposition 2.5.4. *The Hessian of the energy $\mathcal{F}$ with respect to n sites $\{\mathbf{q}_i\}_i$ in a d-dimensional domain Ω is a $nd \times nd$ matrix with $d \times d$ blocks of the form:*

$$\begin{cases} \nabla_{\mathbf{q}_i,\mathbf{q}_i}\mathcal{F} = 2V_i\mathbb{I} - \displaystyle\sum_{j\in\mathcal{N}_i} \frac{2}{l_{ij}} \int_{\mathcal{A}_{ij}} \left(\mathbf{q}_i - \mathbf{x}\right)\left(\mathbf{q}_i - \mathbf{x}\right)^t, \\ \\ \nabla_{\mathbf{q}_j,\mathbf{q}_i}E = \displaystyle\sum_{j\in\mathcal{N}_i} \frac{2}{l_{ij}} \int_{\mathcal{A}_{ij}} \left(\mathbf{q}_i - \mathbf{x}\right)\left(\mathbf{q}_j - \mathbf{x}\right)^t, \end{cases} \tag{2.25}$$

where $\mathbb{I}$ is the $d \times d$ identity matrix.

Proof. These results are directly derived by combining Equation (2.4) and Equation (2.10). □

Chapter 3

Blue Noise Sampling through Power Diagrams

In this chapter, we present a fast, scalable algorithm to generate high-quality blue noise point distributions of arbitrary density functions. At its core is a novel formulation of blue noise sampling as an optimal transport problem. We show that this geometric formulation of blue noise is equivalent to the construction of centroidal power diagrams with equally sized cells. We then leverage the geometric tools developed in Chapter 2 to design an efficient optimization technique of point distributions. Our mathematical, algorithmic, and practical contributions lead to high-quality blue noise point sets with improved spectral and spatial properties.

3.1 Introduction

Coined by Ulichney [Uli87], the term *blue noise* refers to an even, isotropic, yet unstructured distribution of points. Blue noise was first recognized as crucial in dithering of images since it captures the intensity of an image through its local point density, without introducing artificial structures of its own. It rapidly became prevalent in various scientific fields, especially in computer graphics, where its isotropic properties lead to high-quality sampling of multidimensional signals, and its absence of structure prevents aliasing. It has even been argued that its visual efficacy (used to some extent in stippling and pointillism) is linked to the presence of a blue-noise arrangement of photoreceptors in the retina [Yel83].

3.1.1 Previous Work

Over the years, a variety of research efforts targeting both the characteristics and the generation of blue noise distributions have been conducted in graphics. Arguably the oldest approach to algorithmically generate point distributions with a good balance between density control and spatial irregularity is through error diffusion [FS76, Uli87], which is particularly well adapted to low-level hardware implementation in printers. Concurrently, a keen interest in uniform, regularity-free distributions appeared in computer rendering in the context of anti-aliasing [Cro77]. Cook [Coo86] proposed the first dart-throwing algorithm to create Poisson disk distributions, for which no two points are closer together than a certain threshold. Considerable efforts followed to modify and improve this original algorithm [Mit87, MF92, Jon06, Bri07, GM09]. Today's best Poisson disc algorithms are very efficient and versatile [DH06, EDP$^+$11], even running on GPUs [Wei08, BWWM10, XXSH11]. Fast generation of irregular low-discrepancy sequences have also been proposed [Nie92, Lem09]; however, these methods based on the radical-inverse function rarely generate high-quality blue noise.

In an effort to allow for fast blue noise generation, the idea of using patterns computed offline was raised in [DW85]. To remove potential aliasing artifacts due to repeated patterns, Cohen et al. [CSHD03] recommended the use of non-periodic Wang tiles, which subsequently led to improved hierarchical sampling [KCODL06] and a series of other tile-based alternatives [ODJ04, LD06, Ost07]. However, all precalculated structures used in these approaches rely on the offline generation of high-quality blue noise.

Consequently, a number of researchers focused on developing methods to compute point sets with high-quality blue noise properties, typically by evenly distributing points over a domain via Lloyd-based iterations [MF92, DHOS00, Sec02, BSD09, XLGG11, CYC$^+$12], electrostatic forces [SGBW10], statistical-mechanics interacting Gaussian particle models [Fat11], or farthest-point optimization [SHD11]. These iterative methods consistently generate much improved point distributions, albeit at sometimes excessive computational complexity.

Finally, recent efforts have provided tools to analyze point sets using spatial/spectral [LD08, SD11] and differential [WW11] methods. Extensions to anisotropic [LWSF10, XHGL12], non-uniform [WW11], multiclass [Wei10], and general spectrum sampling [ZHWW12] have also been recently introduced.

3.1.2 Motivation and Contributions

Despite typically being slower, optimization methods based on iterative displacements of points have consistently been proven superior to other blue noise generation techniques. With the exception of [SGBW10, Fat11], these iterative approaches rely on Voronoi diagrams and Lloyd's relaxations [Llo82]. To our knowledge, the use of Lloyd's algorithm for blue noise sampling was first advocated in [MF92] to distribute points by minimizing the root mean square (RMS) error of the quantization of a probability distribution. However, the authors noticed that a "somewhat suboptimal solution" was desirable to avoid periodic distribution: run to convergence, Lloyd's algorithm tends to generate regular regions with point or curve defects, creating visual artifacts. Hence, a limited number of iterations was used in practice until Balzer et al. [BSD09] proposed the use of a Capacity-Constrained Voronoi Tessellation (CCVT), a rather drastic change in which a constraint of equi-area partitioning is added to algorithmically ensure that each point conveys equal visual importance. However, this approach and its various improvements rely on a discretization of the capacities, and thus suffer from a quadratic complexity, rendering even GPU implementations [LNW$^+$10] unable to gracefully scale up to large point sets. Two variants were recently proposed to improve performance, both providing an approximation of CCVT by penalizing the area variance of either Voronoi cells [CYC$^+$12] or Delaunay triangles [XLGG11].

We show that CCVT can be formulated as a constrained optimal transport problem. This insight leads to a continuous formulation able to enforce the capacity constraints exactly, unlike related work. The variational nature of our formulation is also amenable to a fast, scalable, and reliable numerical treatment. Our algorithm will be shown, through spectral analysis and comparisons, to generate high-grade blue noise distributions.

Key differences from previous methods include:

- a reformulation of CCVT as a continuous constrained minimization based on optimal transport, as opposed to the discretized approximation suggested in [BH08];
- an optimization procedure over the space of *power diagrams* that satisfies the capacity constraints up to numerical precision (Section 2.5.1), as opposed to an approximate capacity enforcement in the space of Delaunay triangulations [XLGG11] or Voronoi diagrams [CYC+12];
- a regularity-breaking procedure to prevent local aliasing artifacts.

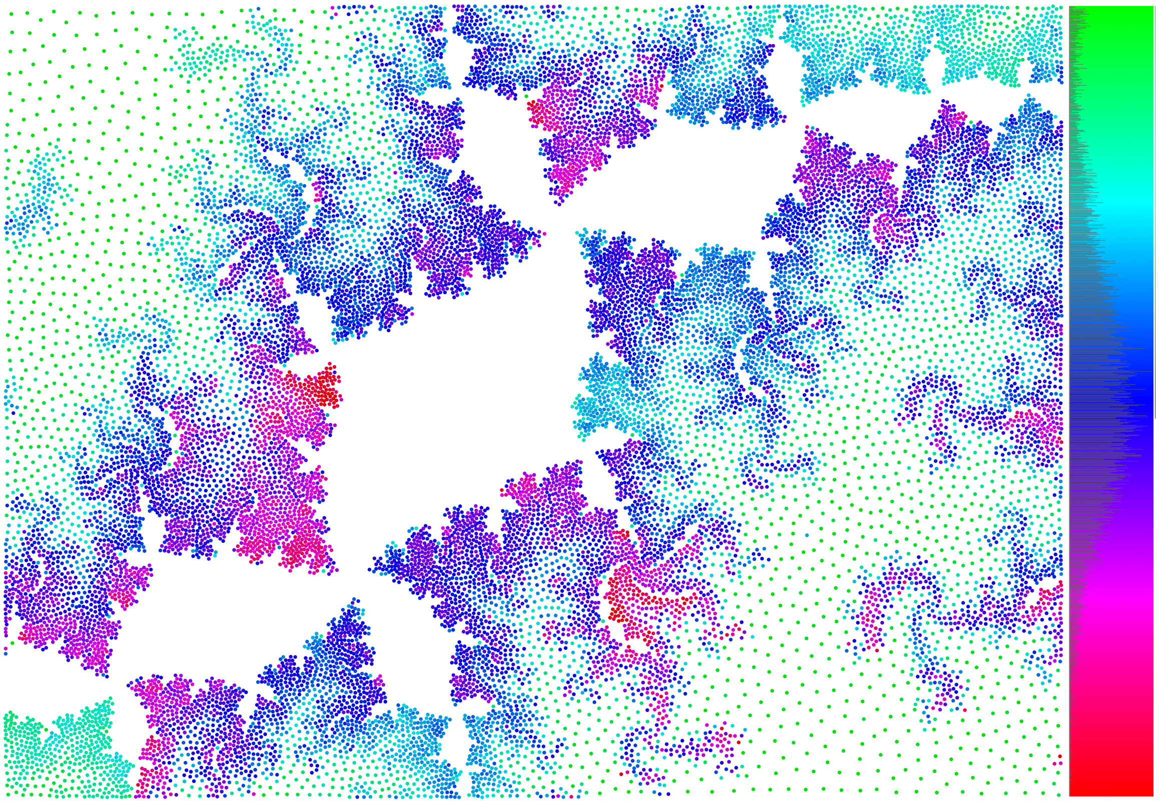


Figure 3.1: **Fractal.** Optimal transport based blue noise sampling of a Julia set image (20K points). Colors indicate (normalized) weight values, ranging from -30% to 188% of the average squared edge length in the triangulation. The histogram of the weights is also shown on top of the color ramp.

3.2 Blue Noise as a Constrained Transport Problem

Before presenting our algorithm for point set generation, we spell out our definition of blue noise as a constrained transport problem. We consider an arbitrary domain Ω over which a piecewise-continuous positive field ρ (e.g., intensity of an image) is defined.

Optimal transport. The optimal transport problem, dating back to Gaspard Monge [Vil09], amounts to determining the optimal way to move a pile of sand to a hole of the same volume—where "optimal" means that the integral of the distances by which the sand is moved (one infinitesimal unit of volume at a time) is minimized. The minimum "cost" of moving the piled-up sand to the hole, i.e., the amount of sand that needs to be moved times the L_p distance it has to be moved, is called the p-Wasserstein metric. The 2-Wasserstein metric, using the L_2 norm, is most common, and is often referred to as the earth mover's distance. Optimal transport has recently been of interest in many scientific fields; see [MMdGD11, BvdPPH11, dGCSAD11] for graphics applications.

Blue noise sampling. Sampling a density function $\rho(\mathbf{x})$ consists of picking a few representative points $\mathbf{q}_i$ that capture ρ well. This is, in essence, the same halftoning process that a black-and-white printer or a monochrome pointillist painter uses to represent an image. In order to formally characterize a blue noise distribution of points, we see sampling as the process of *aggregating* n disjoint regions $\mathcal{V}_i$ (forming a partition $\mathcal{V}$ of the domain Ω) into n points $\mathbf{q}_i$: if ρ is seen as a density of ink over Ω, sampling is the process of coalescing this distribution of ink into n Dirac functions (i.e., ink dots). We can now revisit the definition of blue noise sampling through the following requirements:

A. Uniform Sampling: all point samples should equally contribute to capturing the field ρ. Consequently, their associated regions $\mathcal{V}_i$ must all represent the same amount m of ink:

$$V_i^\rho = \int_{\mathcal{V}_i} \rho(\mathbf{x})\, d\mathbf{x} \equiv m. \tag{3.1}$$

B. Optimal Transport: the total cost of transporting ink from the distribution ρ to the finite point set $\mathbf{q}$ should be minimized, thus representing the most effective aggregation. This ink transport cost for an arbitrary partition $\mathcal{V}$ is given as

$$\mathcal{T}(\mathbf{q}, \mathcal{V}) = \sum_i \int_{\mathcal{V}_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{q}_i\|^2 d\mathbf{x}, \tag{3.2}$$

i.e., as the sum per region of the integral of all displacements of the local ink distribution ρ to its associated ink dot. Note that $\mathcal{T}$ is a modification (of the unweighted part) of the energy $\mathcal{E}$ (Equation (2.1)), now modulated by the density ρ.

C. Local Irregularity: the point set should be void of visual artifacts such as Moiré patterns and other aliasing effects; that is, it should be free of local spatial regularity.

Observe that the first requirement implies that the resulting local point density will be *proportional to* ρ as often required in importance sampling. The second requirement favors *isotropic* distribution of points since such partitions minimize the transport cost. The final requirement prevents regular or hexagonal grid patterns from emerging. Together, these three requirements provide a *density-adapted, isotropic, yet unstructured distribution of points*, capturing the essence of a blue noise as a constrained transport problem.

Link to power diagrams. The need for capacity constrained partitions (requirement A) that minimize the cost $\mathcal{T}$ (requirement B) resembles the construction of volume-constrained power diagrams (Section 2.5.1), with the caveat that integrals are now modulated by the density ρ. So, instead of searching through the entire space of possible partitions, we can rather restrict partitions $\mathcal{V}$ to be power diagrams. While methods restricting their search to Delaunay meshes [XLGG11] or Voronoi diagrams [CYC$^+$12] can only approximate the constraints in requirement A, this power diagram formulation has the additional variables (weights) necessary to allow *exact* constraint enforcement, thus capturing sharp feature much more clearly than previous methods (see Section 3.5). In fact, all of our results exhibit uneven weights as demonstrated in Figure 3.1.

3.3 Variational Formulation

Leveraging the fact that requirements A and B can be enforced for power diagrams, we describe next our variational characterization of blue noise distributions of weighted point sets $(\mathbf{q}, w)$. Requirement C will be enforced algorithmically, as discussed in Section 3.4.6, by detecting regularity and locally jittering the point set to guide our optimization towards non-regular distributions.

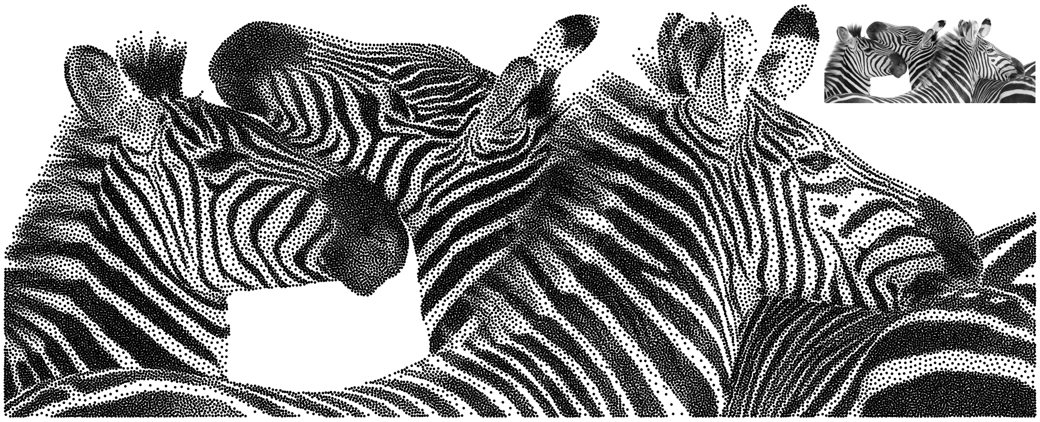


Figure 3.2: **Zebra.** Since our approach accurately captures variations of density, we can blue-noise sample images containing both fuzzy and sharp edges (160K-pixel original image (top right) courtesy of Frédo Durand). 40K points, generated in 159 seconds.

3.3.1 Energy Extremization

We start by properly formulating our constrained minimization to enforce requirements A and B labeled as a function of points and weights:

$$\min_{\mathbf{q},w} \mathcal{T}(\mathbf{q}, w) \quad \text{s.t.} \quad V_i^\rho = m \;\; \forall i. \tag{3.3}$$

We further extend the measurements defined in Section 2.1 by incorporating the density ρ:

$$A_{ij}^\rho = \int_{\mathcal{A}_{ij}} \rho(\mathbf{x}) d\mathbf{x} \quad \text{and} \quad \mathbf{b}_i^\rho = \frac{1}{V_i^\rho} \int_{\mathcal{V}_i} \rho(\mathbf{x}) \mathbf{x} d\mathbf{x}. \tag{3.4}$$

Lagrangian formulation. A common approach to deal with a constrained minimization is to use Lagrange multipliers $\lambda = \{\lambda_i\}_{i=1 \ldots n}$ to enforce the n constraints (one per point) induced by requirement A. The resulting optimization procedure can be stated as:

$$\text{Extremize}_{\;\mathbf{q},w,\lambda}\; \mathcal{L}(\mathbf{q}, w, \lambda) = \mathcal{T}(\mathbf{q}, w) + \sum_i \lambda_i \big(V_i^\rho - m\big). \tag{3.5}$$

Simpler formulation. The Lagrangian multipliers add undue complexity: they contribute an additional n variables to the optimization. Instead, one can show that the extremization above is equivalent to the extremization of a simple extension of the energy $\mathcal{F}$ (Equation (2.19)).

Proposition 3.3.1. *For the sampling of a domain Ω with n points, let the target capacity be $m = |\Omega|/n$ and define the energy:*

$$\mathcal{F}^\rho(\mathbf{q}, w) = \sum_i \left(\int_{\mathcal{V}_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{q}_i\|^2 d\mathbf{x} \right) - \sum_i w_i \left(V_i^\rho - m \right). \tag{3.6}$$

The following optimization problems are equivalent:

$$\textit{Extremize}_{\mathbf{q},w,\lambda}\; \mathcal{L}(\mathbf{q}, w, \lambda) \Longleftrightarrow \textit{Extremize}_{\mathbf{q},w}\; \mathcal{F}^\rho(\mathbf{q}, w). \tag{3.7}$$

Proof. For any critical configuration $(\mathbf{q}^*, w^*, \lambda^*)$ of $\mathcal{L}$, the following conditions hold:

$$\begin{cases} 0 = \nabla_{w_i}\mathcal{L}(\mathbf{q}^*, w^*, \lambda^*) = \nabla_{w_i}\mathcal{T}(\mathbf{q}^*, w^*) + \sum\limits_{j\in\Omega_i} (\lambda_j^* - \lambda_i^*)\nabla_{w_i} V_j^\rho \\ 0 = \nabla_{\mathbf{q}_i}\mathcal{L}(\mathbf{q}^*, w^*, \lambda^*) = \nabla_{\mathbf{q}_i}\mathcal{T}(\mathbf{q}^*, w^*) + \sum\limits_{j\in\mathcal{N}_i} (\lambda_j^* - \lambda_i^*)\nabla_{\mathbf{q}_i} V_j^\rho \\ 0 = \nabla_{\lambda_i}\mathcal{L}(\mathbf{q}^*, w^*, \lambda^*) = V_i^\rho - m. \end{cases}$$

By applying Reynolds theorem (Equation (2.4)), we have:

$$\nabla_{w_i}\mathcal{T} = \sum_{j\in\mathcal{N}_i} \frac{1}{l_{ij}} \int_{\mathcal{A}_{ij}} \rho(\mathbf{x}) \underbrace{\left(\|\mathbf{x}-\mathbf{q}_i\|^2 - \|\mathbf{x}-\mathbf{q}_j\|^2\right)}_{=w_j - w_i \ \forall \mathbf{x}\in\mathcal{A}_{ij}} \underbrace{(\nabla_{w_i}\mathbf{x})^t\,(\mathbf{q}_j - \mathbf{q}_i)}_{\text{Equation (2.18)}} \, d\mathbf{x} = -\frac{1}{2}\left[\Delta^\rho w\right]_i,$$

where $\Delta_{ij}^\rho = -A_{ij}^\rho / l_{ij}$. We can thus expand the first condition as $\Delta^\rho(\lambda^* + w^*) = 0$, which implies: $\lambda^* =$ constant$-w^*$. By replacing the Lagrangian multipliers λ^* with the negated weights w^*, the second condition reduces to $\nabla_{\mathbf{q}_i}\mathcal{L}(\mathbf{q}^*, w^*, \lambda^*) = 2V_i^\rho(\mathbf{q}_i^* - \mathbf{b}_i^\rho) = 0$, which is equivalent to $\nabla_{\mathbf{q}}\mathcal{F}^\rho(\mathbf{q}^*, w^*) = 0$. Similarly, the third condition implies $\nabla_{w_i}\mathcal{F}^\rho(\mathbf{q}^*, w^*) = 0$. Hence, any extremum of $\mathcal{L}$ is also an extremum of $\mathcal{F}^\rho$ and vice-versa. □

Properties. Since the derivatives $\mathcal{F}$ extend trivially to $\mathcal{F}^\rho$, we can not only justify the Lloyd-based algorithmic approaches previously used in [BH08, BSD09, LNW+10], but also derive better numerical methods for finding blue noise point sets. For fixed sites $\mathbf{q}$, the Hessian of $\mathcal{F}^\rho$ with respect to weights is one-half of the negated weighted Laplacian operator Δ^ρ (Section 2.5.1). Consequently, extremizing $\mathcal{F}^\rho$ is a maximization with respect to all weights. This insight thus leads to an efficient numerical approach comparable in speed to recent approximate CCVT methods [XLGG11, CYC+12], but much faster than the quadratic scheme used in [BH08, BSD09, LNW+10]. For fixed weights w, our optimization reduces to the construction of centroidal power diagrams (Section 2.5.2). As previously discussed, several methods can be used to minimize this energy.

3.3.2 Discussion

We now discuss the key differences between our formulation and previous CCVT methods.

Discrete vs. Continuous Formulation. The initial CCVT method and its improvements [BH08, BSD09, LNW+10] adopted a *discrete* formulation in which the density function ρ is represented by a finite set of samples, with the number of samples being orders of magnitude larger than the number n of points. Blue noise point sets are then generated via repeated energy-decreasing swaps between adjacent clusters, without an explicit use of weights. This discrete setup has several numerical drawbacks. First, while samples can be thought of as quadrature points for capacity evaluation, their use causes accuracy issues: in essence, using samples amounts to quantizing capacities; consequently, the transport part of the CCVT formulation is not strictly minimized. Second, the computational cost induced by the number of swaps required to reach convergence is quadratic in the number of samples—and thus impractical beyond a few thousand points. Instead, we provided a continuous energy whose extremization formally encodes the concept behind the original CCVT method [BH08]. The energy $\mathcal{F}^\rho$ in Equation (3.6) was previously introduced in [AHA98] purely as a way to enforce capacity constraints for a fixed point set; here we extend $\mathcal{F}^\rho$ as a function of weights w_i *and* positions $\mathbf{q}_i$, and the closed-form gradient and Hessian we explicitly derived will permit, in the next section, the development of a fast numerical treatment to generate high-quality blue noise distributions in a scalable fashion, independent of the sampling size of the density function.

Approximate vs. Exact Constraints. Attempts at dealing with CCVT through continuous optimization have also been investigated by sacrificing exact enforcement of capacity constraints. In [BDL05, Bal09], for instance, a point-by-point iterative approach is used to minimize the capacity variance of Voronoi cells to best fit the capacity constraints; Chen et al. [CYC$^+$12] recommend adding the capacity variance as a penalty term to the CVT energy instead; Xu et al. [XLGG11] take a dual approach by minimizing capacity variance on Delaunay triangles instead of Voronoi cells. These different variants all mix the requirements of good spatial distribution and capacity constraints into a single minimization, leading to an over-constrained formulation. Minima of their functionals thus always represent a tradeoff between capacity enforcement and isotropic spatial distribution. Instead, our formulation allows exact capacity constraints by controlling the power diagram through the addition of a weight per vertex: we can now optimize distribution quality *while* constraining capacity, resulting in high quality blue noise sampling of arbitrary density field (see the quadratic ramp in Figure 3.5 for a comparison with recent methods).

3.4 Numerical Optimization

We now delve into the numerical methods and algorithmic details we use to efficiently generate blue noise point distributions based on our variational formulation.

3.4.1 Overall Strategy

We proceed with point set generation by computing a critical point of the functional $\mathcal{F}^\rho$ defined in Equation (3.6): we extremize the functional $\mathcal{F}^\rho$ by repeatedly performing a minimization step over positions followed by a projection step over weights to enforce constraints. The power diagram of the weighted point set is updated at each step via the CGAL library [CGA14]. While this alternating procedure is typical for non-linear multivariable problems, we will benefit from several properties of the functional, as already alluded to in Section 3.3:

- enforcing the capacity constraints for a fixed set of point positions is a *concave maximization*;
- minimizing $\mathcal{F}^\rho$ for a fixed set of weights is akin to the minimization of the CVT energy, for which fast methods exist;
- steering clear of regular patterns is enforced algorithmically through a simple local regularity detection and removal.

These three factors conspire to result in a fast and scalable generation of high-quality blue noise point sets as discussed next.

3.4.2 Constraint Enforcement

For given points $\mathbf{q}$, we noted in Section 3.3 that finding the set of weights w_{opt} to enforce the equality of all capacities is a concave maximization. Fast iterative methods can thus be applied to keep computational complexity to a minimum. Since the Hessian of $\mathcal{F}^\rho(\mathbf{q}, w)$ is proportional to the weighted Laplacian, Newton iterations are particularly appropriate to find the optimal set of weights. At each iteration, we thus solve the sparse (Poisson) linear system stated in Equation (2.22). A standard line search with Armijo condition [NW99] is then performed to adapt the step size along the vector δw before updating the vector w of current weights. Given that the Hessian is sparse and symmetric, many linear solvers can be used to efficiently solve the linear system used in each Newton iteration; in our implementation, we use the sparse QR factorization method in [Dav11]. Typically, it only takes 3 to 5 iterations to bring the residual of our constraints to within an accuracy of 10^{-12}.

3.4.3 Transport Minimization

For a fixed set of weights w, we can move the locations of the n points in order to improve the cost of ink transport $\mathcal{F}^\rho(\mathbf{q}, w)$. Previous CCVT-based methods [BSD09, LNW$^+$10] used Lloyd's algorithm as the method of choice for their discrete optimization. In our continuous optimization context, we have more options. A Lloyd update where positions $\mathbf{q}_i$ are moved to the barycenter $\mathbf{b}_i^\rho$ of their associated weighted cell $\mathcal{V}_i$ can also be used to reliably decrease the transport cost: indeed, we prove in the appendix that the gradient of $\mathcal{F}^\rho(\mathbf{q}, w)$ is a natural extension of the gradient of the regular CVT energy. However, Lloyd's algorithm is a special case of a gradient descent that is known to suffer from linear convergence [DFG99]. We improve the convergence rate through line search, again using adaptive timestep gradient descent with Armijo conditions as proposed in [MMdGD11]. Note that the quasi-Newton iterations as proposed in [LWL$^+$09] for the CVT energy are not well suited in our context: alternating weight and position optimizations renders the approximation of the Hessian matrix from previous gradients inaccurate, ruining the expected quadratic convergence.

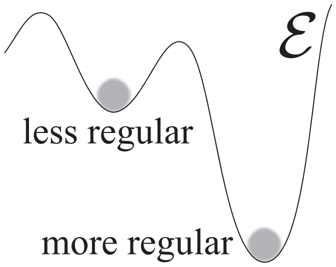
3.4.4 Density Integration

Integrations required by our formulation can be easily handled through quadrature. However, poor quadrature choices may impair the convergence rate of our constraint enforcement. Given that blue noise sampling is most often performed on a rectangular grey-scale image, we design a simple and exact procedure to compute integrals of the density field ρ inside each cell, as it is relatively inexpensive. Assuming that ρ is given as a strictly-positive piecewise constant field, we first compute the value m used in our capacity constraints by simply summing the density values times the area of each constant regions (pixels, typically), divided by n. We then perform integration within each $\mathcal{V}_i$ in order to obtain the capacity V_i^ρ, the barycenter $\mathbf{b}_i^\rho$, and the individual transport cost for each $\mathcal{V}_i$. We proceed in three steps. First, we rasterize the edges of the power diagram and find intersections between the image pixels and each edge. Next we perform a scan-line traversal of the image and construct pixel-cell intersections. Integrated densities, barycenters, and transport costs per cell are then accumulated through simple integration within each pixel-cell intersection where the density is constant. Note that our integration differs from previous similar treatments (e.g., [Sec02, LL06]) as we provide robust and exact computation not only for cell capacities, but also for their barycenters and transport costs—thus avoiding the need for parameter tweaking required in quadrature approximations.

3.4.5 Boundary Treatment

While some of the results we present use a periodic domain (see Section 3.5), most sampling applications involve a bounded domain Ω, often given as a convex polygon (as in the case of a simple image). Dealing with boundaries in our approach is straightforward. First, boundary power cells are clipped by Ω before computing their cell barycenters $\mathbf{b}_i^\rho$ and capacities m_i. Second, the coefficients of the weighted Laplacian are computed through the ratio of (possibly clipped) dual edge lengths and primal edge lengths, as proposed in [MMdGD11]. Thence, the presence of boundaries adds only limited code and computational complexity and it does not affect the convergence rates of any of the steps described above. Note that other boundary treatments could be designed as well, using mirroring or other typical boundary conditions if needed.

3.4.6 Detecting & Breaking Regularities



The numerical procedure described so far solely targets requirements A and B, and as such, nothing preempts regularity. In fact, hexagonal lattices are solutions to our extremization problem in the specific case of constant density and a toroidal domain—and these solutions correspond to “deep” extrema of our functional, as the cost of ink transport $\mathcal{T}$ reaches a *global* minimum on such regular packings of points. Instead, we algorithmically seek “shallow” extrema to prevent regularity (see inset).

For capacity-constrained configurations, local regularities are easily detected by evaluating the individual terms $\mathcal{T}_i$ measuring the transport cost within each region $\mathcal{V}_i$: we assign a regularity score r_i per point as the local absolute deviation of $\mathcal{E}_i^\rho$, i.e.,

$$r_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} |\mathcal{T}_i - \mathcal{T}_j|.$$

We then refer to the region around a point $\mathbf{q}_i$ as *aliased* if $r_i < \tau$, where the threshold $\tau = 0.25\, m^2$ in all our experiments. When aliased, a point and its immediate neighbors are jittered by a Gaussian noise with a spatial variance of $1.0/\rho(\mathbf{q}_i)$ and maximum magnitude $\sqrt{m}$ to break symmetries as recommended in [Luc09]. To prevent a potential return to the same crystalline configuration during subsequent optimization steps, we further relocate 1% of the aliased points to introduce defects. Since our numerical approach relies on a line search with Armijo rule (seeking local extrema), starting the optimization from this stochastically scrambled configuration will fall back to a nearby, shallower extremum—hence removing regularity as demonstrated in Figure 3.3.

It is worth pointing out that all CVT-based methods (including the existing CCVT schemes) may result in point distributions with local regular patterns. While a few approaches avoided regularity by stopping optimization before convergence, we instead prevent regularity by making sure we stop at shallow minima. This τ-based shallowness criterion can be seen as an alternative to the temperature parameter proposed in [Fat11], where the level of excitation of a statistical particle model controls the randomness of the formation of point distributions. Our simple approach is numerically robust and efficient: in practice, we observed that the proposed regularity breaking routine takes place at most once in each example test, independently of τ.

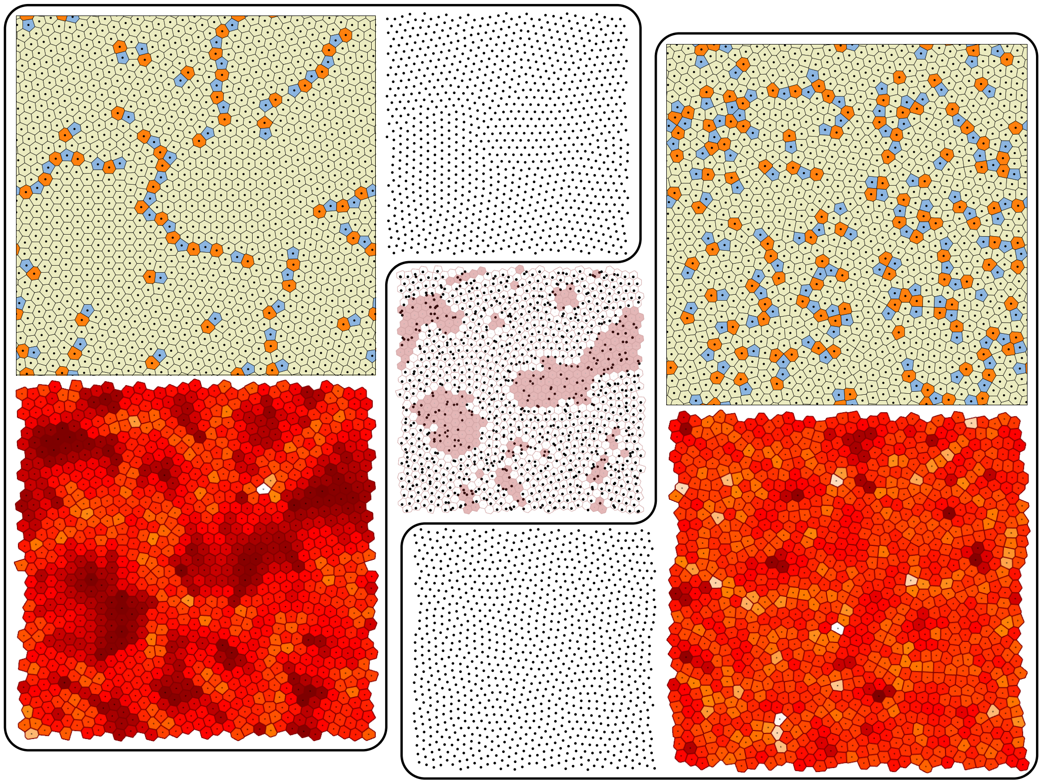


Figure 3.3: **Breaking Regularity.** Optimization of $\mathcal{F}^\rho$ with a strict convergence threshold ($\|\nabla_{\mathbf{q}}\mathcal{F}^\rho\| \leq 10^{-5}$) can produce regularity (left), as revealed by a valence-colored visualization (top) and the distribution of local transport costs $\mathcal{E}_i^\rho$ (bottom). After jittering and relocating aliased regions (middle, colored cells), further optimization brings the point set to a shallower (i.e., less regular) configuration (right) as confirmed by valences and transport costs.

3.4.7 Optimization Schedule

We follow a simple optimization schedule to make the generation process automatic and efficient for arbitrary inputs. We start with a random distribution of points conforming to ρ (better initialization strategies could be used, of course). We then proceed by systematically alternating optimization of weights (to enforce constraints, Section 3.4.2) and positions (to minimize transport cost, Section 3.4.3). Weight optimization is initialized with zero weights, and iterated until $\|\nabla_w \mathcal{F}^\rho\| \leq 0.1\, m$ (the capacity m is used to properly adapt the convergence threshold to the number of points n and the density ρ). For positions, we optimize our functional until $\|\nabla_{\mathbf{q}}\mathcal{F}^\rho\| \leq 0.1\sqrt{n\, m^3}$ (again, scaling is chosen here to account for density and number of points). We found that performing Lloyd steps until the gradient norm is below $0.2\sqrt{n\, m^3}$ reduces computation (it typically requires 5 iterations); only then do we revert to a full-blown adaptive timestep gradient descent until convergence (taking typically 10 iterations). Once an extremum of $\mathcal{F}^\rho$ is found, we apply the regularity detecting-and-breaking procedure presented in Section 3.4.6 and, if an aliased point was found and jittered, we start our optimization again. This simple schedule (Algorithm 1) was used as is on all our results.

3.5 Results

We ran our algorithm on a variety of inputs: from constant density (Figures 3.9) to photographs (Figure 3.8, 3.2, and 3.4) and computer-generated images (Figure 3.1 and 3.5), without any need for parameter tuning. Various

Algorithm 1 Blue noise sampling

```
1: Input: domain $\Omega$, density $\rho$, and number of points $n$.
2: Initialize $\mathbf{q}$ with $n$ random points inside $\Omega$ conforming to $\rho$.
3: repeat
4:     Enforce capacity $m$ through weight optimization (Equation (2.22)).
5:     // Lloyd step
6:     for all points $i$ do
7:         $\mathbf{q}_i \leftarrow \mathbf{b}_i^\rho$
8:     end for
9:     Update power diagram.
10: until $\|\nabla_{\mathbf{q}}\mathcal{F}^\rho\| \leq 0.2\sqrt{n\,m^3}$
11: repeat
12:     Enforce capacity $m$ through weight optimization (Equation (2.22)).
13:     // Gradient descent
14:     $\delta_{\mathbf{q}} = \nabla_{\mathbf{q}}\mathcal{F}^\rho$
15:     Find $\beta$ satisfying Armijo condition.
16:     $\mathbf{q} \leftarrow \mathbf{q} - \beta\delta_{\mathbf{q}}$.
17:     Update power diagram.
18: until $\|\nabla_{\mathbf{q}}\mathcal{F}\| \leq 0.1\sqrt{n\,m^3}$
19: // Detect and break regularities (Section 3.4.6)
20: if Aliased points detected then
21:     Jitter aliased points and immediate neighbors.
22:     Relocate 1% of aliased points.
23:     Restart optimization (line 3).
24: end if
25: Output: $n$ points with blue noise requirements A, B, and C.
```

illustrations based on zoneplates, regularity, and spectral analysis are used throughout the paper to allow easy evaluation of our results and to demonstrate comparisons to previous work.

Spectral Properties. The special case of blue noise point distribution for a constant density in a periodic domain has been the subject of countless studies. It is generally accepted that such a point distribution must have a characteristic blue-noise profile for the radial component of its Fourier spectra, as well as low angular anisotropy [Uli87]. This profile should exhibit no low frequencies (since the density is constant), a high peak around the average distance between adjacent points, followed by a flat tail to guarantee white noise (i.e., no distinguishable features) in the high frequency range. Figure 3.9 demonstrates that we improve upon the results of all previous CCVT-related methods, and fare arguably better than alternative methods such as [Fat11]; in particular, we systematically (i.e., not just on average over several distributions, but for every single run) obtain flat spectrum in low and high frequencies, while maintaining high peaks at the characteristic frequency. Note also that the method of Balzer et al. [BSD09] appears to slowly converge to our results when the ratio m/n (using their notation) goes to infinity with, evidently, much larger timings (Figure 3.6).

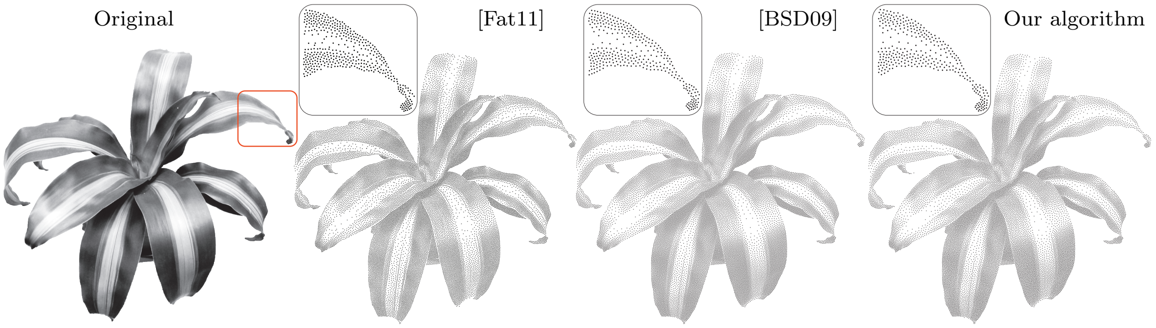


Figure 3.4: **Stippling.** Test from [Sec02] (20K points). While [Fat11] does not clearly capture density gradients (see close-ups), our result is similar to CCVT [BSD09] on this example, at a fraction of the CPU time. Comparative data courtesy of the authors.

Spatial Properties. We provide evaluations of the spatial properties of our results. Figure 3.9 shows two insightful visualizations of the typical spatial arrangement of our point distributions, side by side with results from previous state-of-the-art methods. The second row shows the gaps between white discs centered on sampling points with a diameter equal to the mean distance between two points; notice the uniformity of the gap distribution in our result. The third row compares the number of neighbors for the Voronoi region of each site; as pointed out in [Bal09], the enforcement of the capacity constraints favors heterogenous valences, with fewer noticeable regular regions. Finally, the minimum distance among all points normalized by the radius of a disc in a hexagonal tiling is a measure of distribution quality, known as the normalized Poisson disk radius, and recommended to be in the range $[0.65, 0.85]$ by [LD08]. In all our constant density blue noise examples, the normalized radius is in the range $[0.71, 0.76]$.

Zoneplates. We also provide zoneplates in Figure 3.9 for the function $\sin(x^2 + y^2)$. Each zoneplate image was created via 32x32 copies of a 1024-point blue noise patch, followed by a Mitchell reconstruction filter to generate a 1024x1024 image with an average of one point per pixel as suggested in [LD06]. Observe the presence of a second noise ring in previous methods, as opposed to our anti-aliased reconstruction.

Quadratic Ramp. Another common evaluation of blue noise sampling is to generate a point set for an intensity ramp, and count the number of points for each quarter of the ramp. Figure 3.5 compares the point sets generated by our technique vs. state-of-the-art methods [Bal09, Fat11, CYC$^+$12]. While all methods recover approximately the right number of points per quarter, our result presents a noticeably less noisy, yet unstructured distribution of points.

Complexity. Previous CCVT methods analyzed the (worst-case) time complexity of a single iteration of their optimization approach. One iteration of our algorithm involves the construction of a 2D power diagram, costing $\mathcal{O}(n \log n)$. It also involves the enforcement of the capacity constraints via a concave maximization w.r.t. the weights via a step-adaptive Newton method; the time complexity of this maximization is of the order of a single Newton step since the convergence rate is quadratic [NW99], and therefore incurs the linear cost of solving a sparse (Poisson) linear system. For N-pixel images and n points, the total complexity of our algorithm thus becomes $\mathcal{O}(n \log n + N)$, with the extra term corresponding to the cost of locating the pixels within each power cell through scan-line traversal. This is significantly better than the discrete versions of CCVT which were either $\mathcal{O}(n^2 + nN \log N/n)$ [Bal09] or $\mathcal{O}(n^2 + nN)$ [LNW$^+$10] and of the same order as the CCVT approximations in [XLGG11, CYC$^+$12]. However, we are unable to match the efficiency of the multi-scale statistical particle model introduced in [Fat11], which scales linearly with the number of points and produces results arguably comparable with the best current methods of blue noise generation.

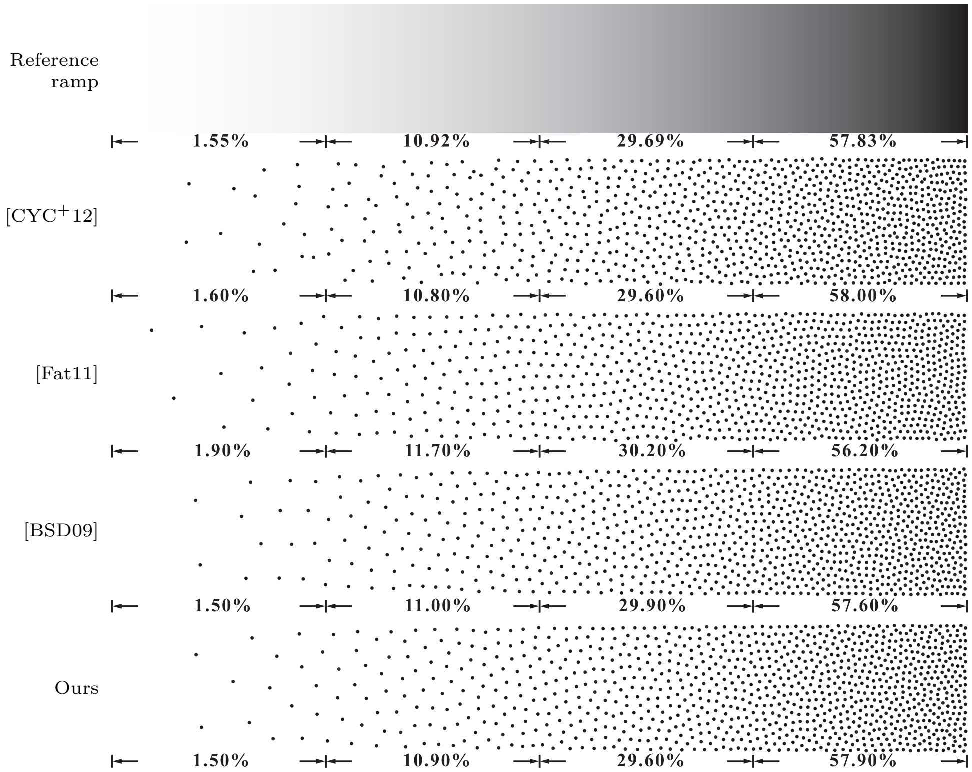


Figure 3.5: **Ramp.** Blue noise sampling of a quadratic density function with 1000 points. The percentages in each quarter indicate ink density in the image, and point density in the examples. Observe that our method returns the best matching of the reference percentages, while still presenting an even and unstructured distribution. Comparative data courtesy of the authors.

Timings. All of our timings in this paper were clocked on an Intel Core i7 2.2 GHz laptop with 4GB RAM. Depending on the input image resolution and the desired number of points, our approach takes from a few seconds (Figures 3.9(right) and 3.5) to 2.1 minutes for Figure 3.4 (20K points, 445x419 image), 2.6 minutes for Figure 3.2 (40K points, 600x267 image), 5.8 minutes for Figure 3.1 (20K points, 600x450 image), and 8.3 minutes for Figure 3.8 (100K points, 512x768 image). The efficiency of the numerics allows us to generate, in 2.8 hours, 1M+ points of the luminance of a high dynamic range 512x768 image, an order of magnitude more complex than the largest results demonstrated by CCVT-based methods. Note that we purposely developed a code robust to any input and any points-to-pixels ratio. However, code profiling revealed that about 40% of computation time was spent on the exact integration described in Section 3.4.4; depending on the targeted application, performance could thus be improved through quadrature [LL06] or image resampling if needed.

Stopping Criteria. As discussed in Section 3.4.7, we terminate optimization when $\|\nabla \mathcal{F}^\rho\| < \varepsilon$, i.e., the first order condition for identifying a locally optimal solution to a critical point search [NW99]. Recent optimization-based blue noise methods [XLGG11, CYC$^+$12], on the other hand, have used the decrease of the objective function per iteration as their stopping criteria. However, a small decrease in the functional does not imply convergence, since a change of functional value depends both on the functional landscape and the step size chosen in each iteration. Favoring guaranteed high quality vs. improved timing, we prefer adopting the first order optimality condition as our termination criteria for robust generation of blue noise distributions. Despite this purposely stringent convergence criteria, the performance of our method is similar to [CYC$^+$12] with their recommended termination based on functional decrease—but twice faster if the method of [CYC$^+$12] is modified to use a stricter termination criterion based on the norm of the functional gradient. Xu et al. [XLGG11] advocate a fixed number of iterations, which, again, does not imply either convergence or high-quality results. Our timings and theirs are, however, similar for the type of examples the authors used in their paper. See Figure 3.7 for a summary of the timings of our algorithm compared to the CCVT-based methods of [XLGG11, CYC$^+$12] for blue noise sampling of a constant density field.

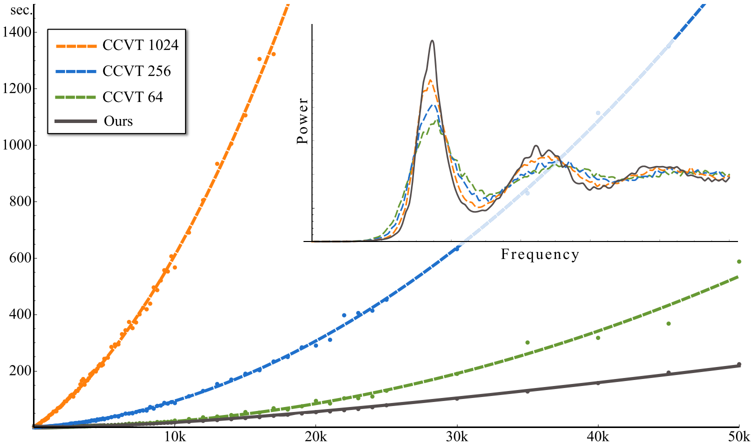


Figure 3.6: **Discrete vs. Continuous CCVT.** Our timings as a function of the number of points exhibit a typical $n \log n$ behavior, systematically better than [BSD09]'s n^2; yet, our radial spectra (inset, showing averages over 10 runs with 1024 points) even outperforms the fine 1024-sample CCVT results. (Here, CCVT-X stands for X "points-per-site" as in [BSD09].)

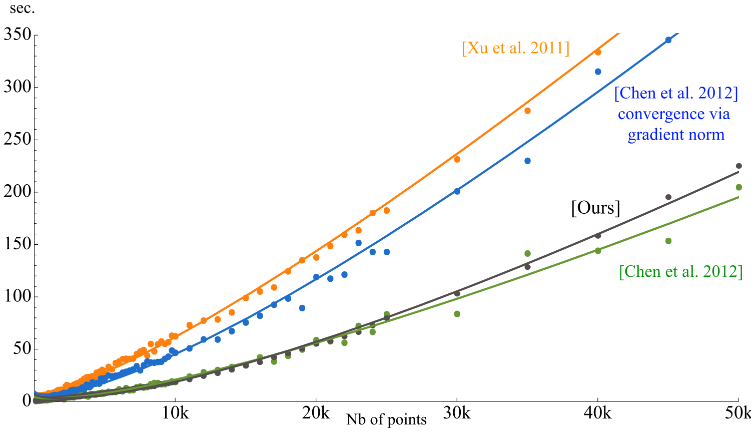


Figure 3.7: **Performance.** Our method (in grey) performs well despite a stringent convergence criteria ($\|\nabla \mathcal{F}^\rho\| < 0.1\sqrt{nm^3}$). The method of [CYC$^+$12] (in green) behaves similarly when using a loose stopping criteria based on the functional decrease per iteration; but becomes twice slower (in blue) if the termination is based on the norm of the functional gradient to guarantee local optimality. The code released by [XLGG11] (in orange) also exhibits comparable performance by terminating the optimization not based on convergence, but after a fixed number of iterations.

3.6 Future Work

We note that our numerical treatment is ripe for GPU implementations as each element is known to be parallelizable. The scalability of our approach should also make blue noise generation over non-flat surfaces practical since our formulation and numerical approach generalizes to these cases without modification. Blue noise meshing is thus an obvious avenue to explore and evaluate for numerical benefits. On the theoretical side it would be interesting to seek a fully variational definition of blue noise that incorporates requirements A, B and C altogether. Generating anisotropic and multiclass sampling would also be desirable, as well as extending our regularity-breaking procedure to other CVT-based methods. Finally, the intriguing connection between HOT meshes [MMdGD11] and our definition of blue noise may deserve further exploration.

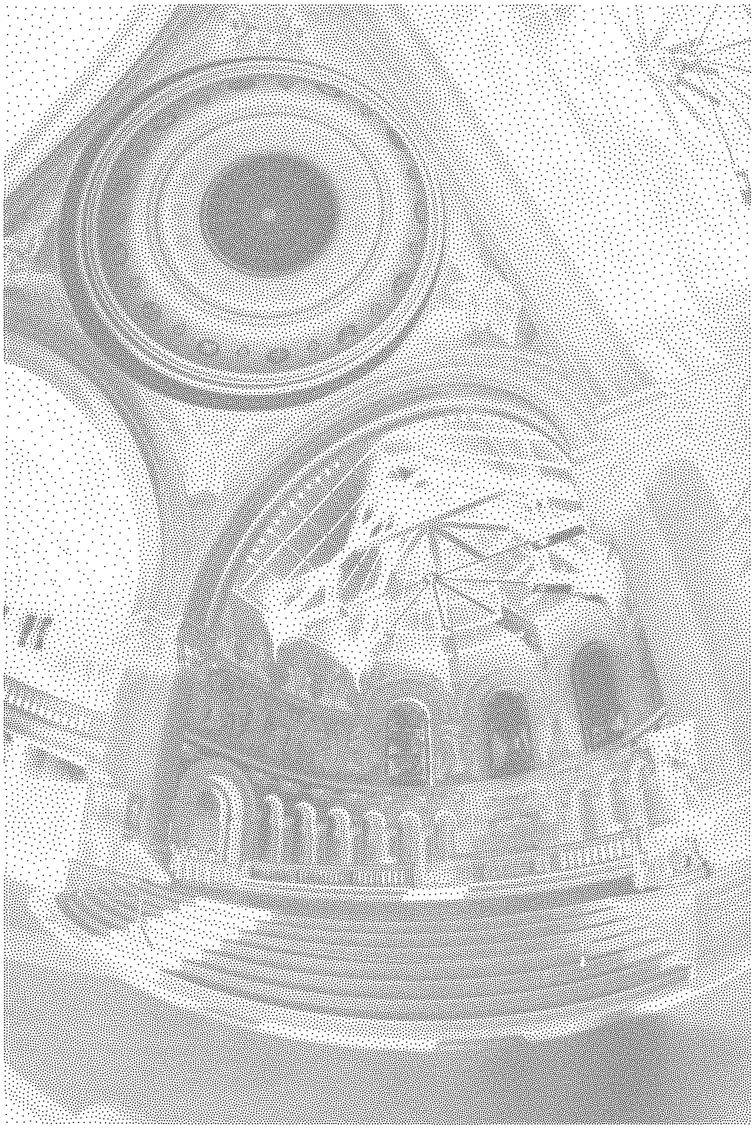


Figure 3.8: **Memorial.** Our variational approach allows sampling of arbitrary functions (e.g., a high-dynamic range image courtesy of P. Debevec), producing high-quality, detail-capturing blue noise point distributions without spurious regular patterns (100K points, 498 s).

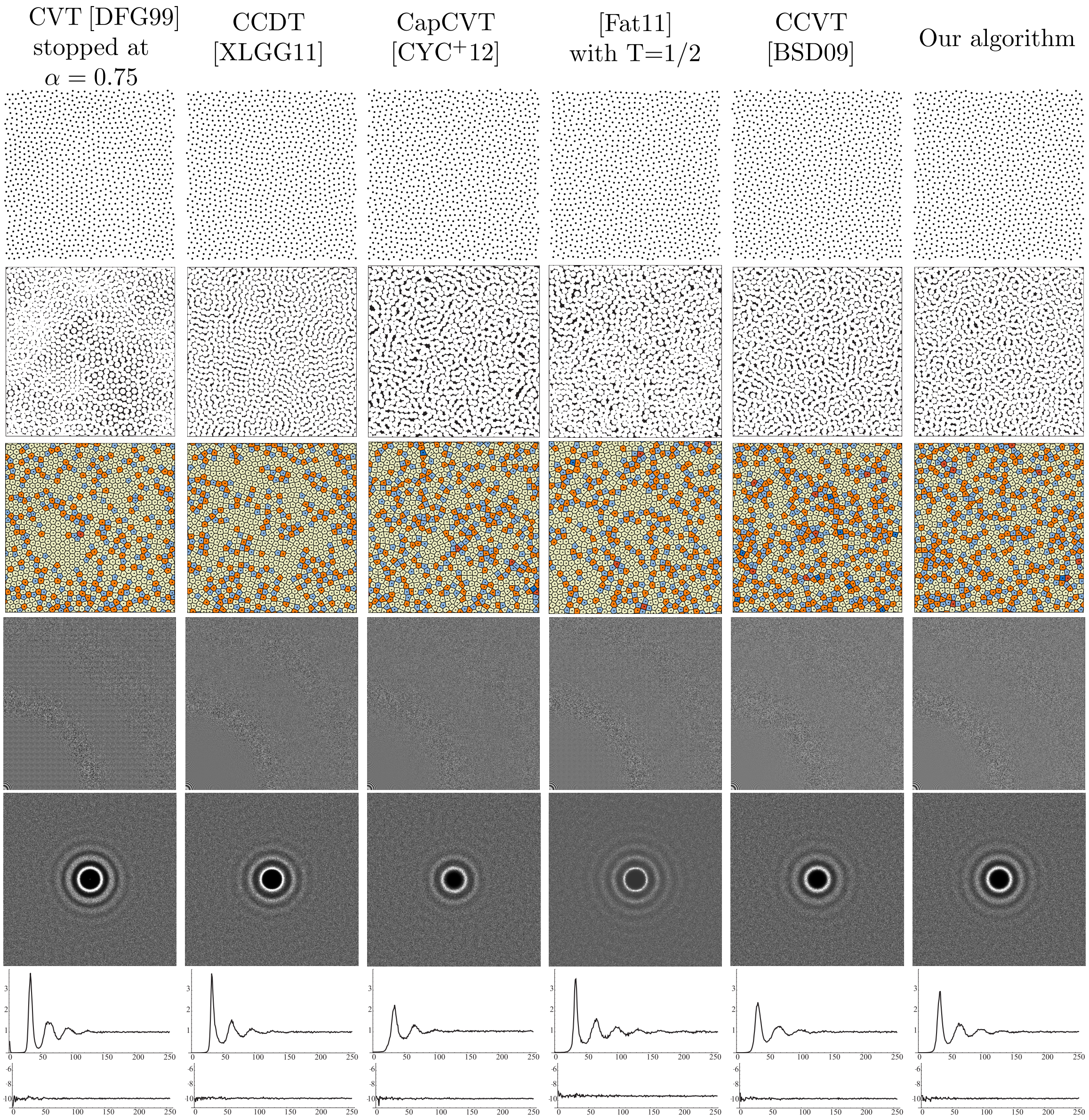


Figure 3.9: **Comparisons.** Different blue noise algorithms are analyzed for the case of constant density over a periodic domain. Top row: distributions of 1024 points. Second row: gaps between white discs centered on sampling points, over black background. Notice the uniformity of gap distribution in two rightmost point sets. Third row: coloring based on number of neighbors for the Voronoi region of each site. Fourth row: $1024x1024$ zoneplates for the function $\sin(x^2 + y^2)$ (see Sec. 3.5 or [LD06] for details). Fifth row: mean periodograms for 10 independent point sets (except for [Fat11], for which only 5 pointsets were available). Sixth row: radial power spectra—note the pronounced peak in our result, without any increase of regularity. Last row: anisotropy in dB ([Uli87], p. 56). Data/code for [Fat11] and [BSD09] courtesy of the authors.

Chapter 4

Fluid Simulation through Power Diagrams

In this chapter, we introduce a novel particle-based approach to fluid simulation. We markedly depart from previous Lagrangian methods by considering particles no longer purely as material points, but also as fluid parcels that partition the fluid domain. Our geometric discretization of fluid flow is based on a time-evolving power diagram (hence the name *power particles*) which provides precise control over local volumes. No kernel-based evaluation of internal forces or density is involved, thus suppressing the need for ghost particles near boundaries. Instead, positions and velocities are updated in time by maintaining the particles well-centered with respect to their associated fluid parcels. We also present a new pressure solver to enforce divergence-free velocity fields through the use of discrete operators derived in Section 2.4 based on power diagrams. We finally demonstrate our method for the simulation of incompressible, compressible, and multi-phase flows.

4.1 Introduction

Fluid particles are a convenient Lagrangian representation of fluid motion that can capture fine scale details while naturally conserving mass. Particle-based methods are popular in animation due to their conceptual simplicity and efficiency, with SPH [Mon92] and FLIP-based [BR86] methods being the common choices. However, enforcing incompressibility with particles remains a serious challenge: many existing approaches suffer from the formation of particle clusters, which cause deficiencies in particle density, and thus result in numerical instabilities. Recent remedies rely on density corrections, and include artificial pressures and viscosity combined with small time steps and loose incompressibility tolerance. Yet, these solutions trade significant artificial damping for improved particle distribution and stability, leaving no satisfactory particle method to handle even basic flows in closed domains, as evidenced by Figure 4.8.

This paper presents a new Lagrangian method for fluid simulation that maintains even distributions of particles over time without significant numerical viscosity. Our work is based on a new representation of fluid particles that associates material points with finite spatial cells obtained from a power diagram partitioning the simulation domain. The flow is then discretized in time through a series of power diagrams, hence the name *power particles*. Our approach is particularly well suited to the case of incompressible fluid simulation, as we can control the construction of power diagrams to enforce local volume preservation up to numerical precision, even for large time steps. We also leverage the new discrete differential operators introduced in Section 2.4 in order to offer a reliable pressure projection to divergence-free velocity fields. Finally, we demonstrate our method in scenarios with complex domains, as well as provide extensions to compressible and multiphase flows.

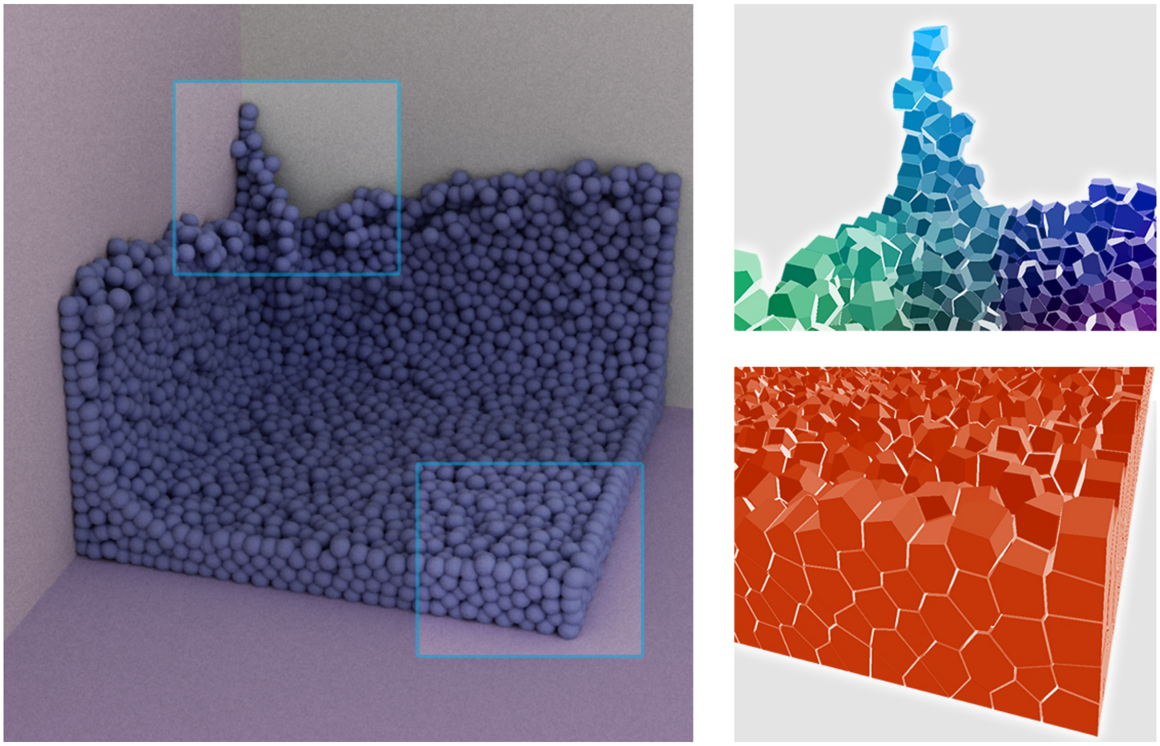


Figure 4.1: **Power Particles.** We present a particle-based approach for fluid simulation. Here, an incompressible liquid splashes in a box; each material point is associated with a well-shaped cell (closeups), whose volume is preserved over time.

4.2 Related Work

A broad variety of methods for fluid simulation have been devised in computational physics and computer graphics—see, e.g., [Bri08] for a review. Here we focus on Lagrangian methods and restrict our discussion to particle-based and hybrid approaches.

Early SPH works. Smoothed Particle Hydrodynamics (SPH) discretizes fluids as material points with interaction forces derived from smooth kernel functions [Mon05]. In computer graphics, SPH was initially used for smoke and fire in [SF95] before being used for liquid simulation in [DG96]. Muller et al. [MCG03] then demonstrated interactive fluid simulation for low stiffness internal forces, while Becker and Teschner [BT07] presented weakly compressible flows via the Tait state equation. These approaches were quickly extended to adaptive sampling [APKG07], multiple fluids [MSKG05, SP08], shallow water simulation [LH10, SBC$^+$11], and parallel computations [GSSP10, IABT11], but remained largely limited to the compressible case.

Incompressibility. Other techniques have addressed incompressibility as a means to improve realism (see a survey in [IOS$^+$14]). The work of [CR99] proposed a projection step to move SPH particles along a divergence-free velocity field; however, they were subjected to inaccuracy and instability due to the projection operator derived from second derivatives of the SPH kernel. Instead, a family of methods was introduced which enforced incompressibility by conserving particle density [PTB$^+$03]. The methods of [ESn07, BLS12], for instance, constrained densities via Lagrangian multipliers. Predictive-corrective incompressible SPH (PCISPH) [SP09] used an iterative Jacobi-style algorithm that accumulates pressure changes and corrects particle location. The convergence of PCISPH was further accelerated in [HLL$^+$12]. Macklin and Muller [MM13] relaxed PCISPH time step requirements by incorporating local pressure estimates into the Position-based Dynamics framework. In [ICS$^+$13], similar position corrections were computed by solving a series of sparse linear systems. Lastly, Hu and Adams [HA07] coupled pressure projection steps and density corrections for the simulation of multiphase fluids.

Lagrangian/Eulerian schemes. Hybrid methods are an alternative to incompressible fluid simulation, combining non-diffusive Lagrangian advection with fast and accurate Eulerian pressure projection. In [RWT11], SPH particles were displaced along a divergence-free velocity field computed on a coarse Eulerian grid. Losasso et al. [LTKF08] proposed a two-way coupled SPH and particle level set method to simultaneously capture small and large phenomena. A finite element discretization for fluids was introduced in [EMB11] alternating Lagrangian advection with mesh optimization. Fluid-implicit particle method (FLIP) [BR86] was adopted in [ZB05] to simulate sand, and later extended to viscous materials [BB08] and two-phase flows [BB12]. More recently, the work of [ATW13] combined FLIP with adaptive sampling of particles and background meshes, while Cornelis et al. [CIPT14] presented a scalable scheme that enriches FLIP velocities with SPH-based density corrections.

Limitations. Both SPH and FLIP-based methods share common limitations. In particular, these methods tend to exhibit particle clumping, which reduces simulation stability and necessitates time step restrictions. Boundary treatment is also sensitive as density fluctuations may happen along contact regions due to particle under- or over-sampling. Existing approaches alleviate these issues through ad-hoc anti-clustering forces and viscosity. A common practice is the use of the XSPH method [Mon92], which averages the local velocity and adds an artificial pressure [Mon00, CBP05]. Discrete element forces [AO11] and cohesion terms [AAT13] were also used to move particles apart. In [SZSY11], stability was improved via non-physical particle smoothing. Boundary handling was addressed through adaptive time steps [IAGT10] and ghost particles [SB12]. However, these strategies introduce severe motion damping, reducing the liveliness of flows. In contrast, our method maintains an even point distribution over time by representing fluid particles as a partition of the domain with well-centered cells, without adding undue numerical viscosity.

Voronoi diagrams. Our work is closely related to Voronoi-based techniques. Voronoi diagrams are commonly used in finite volume methods as a means to construct unstructured meshes for Eulerian simulations. Examples include the work of [ETK$^+$07, MCP$^+$09] that advected vorticity over Voronoi elements, and the method of [BBB10] that placed pressure samples at Voronoi sites. A hybrid Voronoi-based approach for incompressible fluids was proposed in [SBH09] by combining material points with a finite volume pressure solver. A purely Lagrangian scheme using Voronoi diagrams was introduced in [HSS00] for compressible fluids. Later, Serrano et al. [SEZ05] derived discrete divergence and gradient operators based on Voronoi cells that conserve both linear and angular momentum. Voronoi-based methods, however, offer limited control over the volumes of particles, because Voronoi cells are fully determined by the particle location. We suppress this limitation by employing power diagrams, a generalization of Voronoi diagrams, that incorporate extra degrees of freedom for partitioning the domain. Although power diagrams have found many applications in geometry processing [MMdGD11, dGAOD13, LHS$^+$13], their use in simulation has been restricted to the formation of bubbles and foams [BDWR12]. We instead introduce a fluid solver that represents particles as a volume-constrained power diagram, thus allowing the enforcement of particle incompressibility up to numerical precision.

4.3 Power Particles

We now introduce our geometric approach to fluid flow simulation, describing both the concepts and numerical tools behind this computational method.

4.3.1 Rationale and Definition

Traditionally, particles are considered to be material points with no geometric notion of the surrounding volume. SPH methods, for instance, infer the spatial occupancy of a given particle through kernel evaluation based on neighboring positions or densities [Mon92]. This approximation, which effectively amounts to a local Monte Carlo integration over irregular samples, results in notoriously unreliable local volume estimation, and is the primary origin of the multiple inaccuracies inherent to SPH.

Instead, our method proposes to enhance the representation of fluid particles by constructing, at every time frame, a cell complex $\{\mathcal{V}_i\}_i$ that tessellates the domain Ω based on the current location of the material points. We then assign each material point to its corresponding spatial cell, resulting in a precise definition of volume for each fluid particle. Given the properties outlined in Section 2.5, power diagrams are an attractive choice as a way of partitioning a simulation domain in spatial regions. In particular, they involve an extra scalar value per region (the weight), with which we can control the volume of each cell accurately (Section 2.5.1). Moreover, the sampling technique mentioned in Section 2.5.2 can be leveraged to make sure that material points are good proxies of their associated cell by staying near their centroids, enforcing the consistency of these complementary point-based and tessellation-based representations. Finally, power cells are a smooth function of site locations and weights: even if their dual triangulation may experience edge flips during site motion or weight change, the power diagram will smoothly deform.

Therefore, our approach consists of defining the configuration of a fluid at time t as a set of material points $\{\mathbf{q}_i\}_i$ and associated scalar values $\{w_i\}_i$, acting as sites and weights of a power diagram that constructs fluid parcels $\{\mathcal{V}_i\}_i$ with local volumes $\{\overline{V}_i\}_i$, covering the whole domain. We thus refer to each pair of material point $\mathbf{q}_i$ and associated weight w_i as a *power particle.* Power particles will be called "well-centered" if each site $\mathbf{q}_i$ lies near the centroid $\mathbf{b}_i$ of its power cell $\mathcal{V}_i$, to ensure that material points are good representatives of their associated cells. We also assign to each power particle a fixed mass m_i, from which we calculate the local fluid density as $\rho_i = m_i/\overline{V}_i$. Velocities $\{\mathbf{v}_i\}_i$, pressure samples $\{p_i\}_i$, and even temperatures $\{T_i\}_i$ if needed, are also associated with individual particles. A fluid flow is computed by evolving sites and weights in time according to a given equation of motion, while controlling the well-centeredness of the power cells and their volumes. With this setup at hand, we next describe the basic numerical tools necessary to turn our power particles into a fluid solver.

4.3.2 Controlling Volumes

Being able to control the volume of each fluid parcel throughout an animation is particularly important to enforce incompressibility. Our choice of power diagrams to partition the domain allows us to select a target volume $\overline{V}_i$ for each site $\mathbf{q}_i$, and the weights satisfying these imposed volumes are found by maximizing the concave energy in Equation (2.19). More specifically, we start from the current set of weights and iteratively solve for weight updates based on Equation (2.22) until the deviation from the target volumes is below a tolerance. Since it is a concave optimization, this procedure guarantees the existence of a solution (up to an additive constant since translating weights by a constant keeps the power cells unchanged), and therefore our setup can impose incompressibility up to numerical precision. Notice that this weight solver resembles the iterative Poisson-based scheme used to conserve particle density in incompressible SPH methods [HA07, ICS$^+$13], where the weights play the role of (twice) the pressure samples; however, our weights are purely geometric, and do not directly displace the sites.

4.3.3 Volume-based Operators

Another consequence of using power diagrams to partition the domain is that it naturally leads to discrete operators on scalar and vector fields, as derived in Section 2.4. In particular, the results in Proposition 2.4.4 offer a discretization that preserves physical invariances. We first notice that the discrete gradient (Equation (2.15)) is symmetric, i.e., gradient forces from particle i to particle j cancel forces from j to i, and consequently preserve linear momentum exactly. Angular momentum is also preserved due to the invariance of power cell volumes to global rotations. Furthermore, since this definition is directly derived from a consistent partition of the domain (that provides local volumes exactly summing up to the domain's volume), we do not suffer the usual inaccuracies of kernel-based operators: our gradient operator is in fact linear accurate (Equation (2.16a)), while the symmetrized gradient operator used in SPH methods is not accurate even for constant fields.

4.3.4 Pressure Projection

In addition to local volume preservation, we also enforce incompressibility by flowing particles along a divergence-free velocity field. We correct particle velocities through a pressure projection of the form $\nabla \cdot (\nabla p/\rho) = \nabla \cdot \mathbf{v}$, where p is a pressure scalar field and boundary conditions are properly set. Given the discrete operators in Section 2.4, one can directly assemble the equivalent discrete Poisson equation through:

$$dt\, \mathbf{D}\, \mathrm{diag}(\rho)^{-1} \mathbf{G} p = \mathbf{D}\mathbf{v}, \tag{4.1}$$

where diag(ρ) is the diagonal matrix containing particle density $\rho_i = m_i/\overline{V}_i$. The velocity field $\mathbf{v}$ is then updated by subtracting $dt\ \mathrm{diag}(\rho)^{-1}\mathbf{G}p$, rendering $\mathbf{v}$ divergence-free.

4.3.5 Advection-Correction

The use of *finite* time steps to advect each site $\mathbf{q}_i$ by its velocity $\mathbf{v}_i$ unavoidably introduces drifting both on cell volumes and on the sampling of the sites. While the volume deviation is easily corrected through the weight optimization in Section 4.3.2, we must also make sure that sites remain good proxies of their associated parcels. We leverage the fact that well-centered power particles (as defined in Section 2.5.2) resemble a centroidal power diagram, and thus propose a correction step which directly moves the locations of sites $\{\mathbf{q}_i\}_i$ to their respective cell centroids $\{\mathbf{b}_i\}_i$. This correction step corresponds to a single Lloyd iteration used in blue noise sampling (Section 3.4.7) that decreases the energy $\mathcal{F}$ in Equation (2.19), and thereby maintains a high quality sampling of the flow with well distributed cells. Note that this correction step is a purely geometric resampling and, in essence, matches the partial decoupling of particle velocity from the actual change in particle position advocated in [HA07, SZSY11]. As a result, our approach prevents the bunching up of particles within the domain or at boundaries, without the need for ghost particles or additional viscosity terms.

4.3.6 Boundary handling

Beyond the clipped power diagram partitioning the domain (Section 2.2), our approach does not require any additional treatment at boundaries. This is in sharp contrast with all other particle-based methods which require either ghost particles [SB12] or various remedies to the systematic issue of kernel deficiency near obstacles, creating clumping or voids. Our well-centered power particles clipped at the boundary eliminates these artifacts altogether.

The boundary conditions necessary to solve the pressure projection equation are also quite straightforward. For Neumann condition, we ignore the boundary integral term in Equation (2.14) since it amounts to enforcing $\mathbf{v}\cdot\mathbf{n}=0$; for Dirichlet condition, we set the pressures of boundary particles (i.e., those that are clipped) to the prescribed pressure. For non-slip boundary conditions, we simply remove the tangential velocity component for boundary particles. At coarse resolution, one can scale down the tangential component instead to avoid overly strong adherence to the boundary.

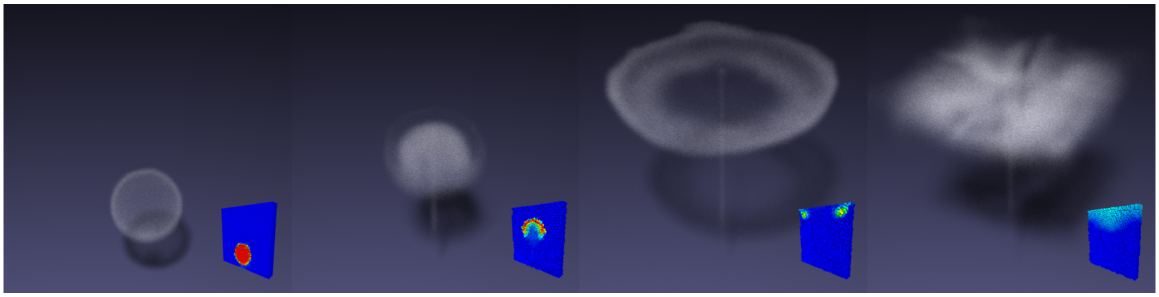


Figure 4.2: **Smoke.** Starting from a ball of hot gas, buoyancy makes the gas rise and curl up inside a cube-shaped box, while temperature diffuses along the way (insets display an internal slice of power cells colored by their temperature).

4.4 Algorithm

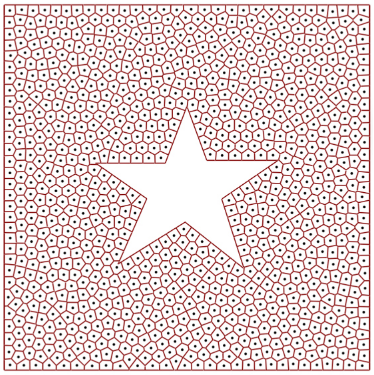
We now delve into the algorithmic steps of our fluid solver based on power particles. We begin by describing how one can compute incompressible fluids with our approach, and show that simple alterations of this incompressible case turns our approach into a compressible and multiphase fluid solver. We also discuss extensions such as surface tension, buoyancy, viscosity, and adaptive sampling. An outline of the simulation loop is given in Algorithm 2.

Algorithm 2 Simulation loop

```
 1: Apply external forces (gravity, buoyancy,...).
 2: Compute pressure $p$.
 3: for all particles $i$ do
 4:     $\mathbf{v}_i \leftarrow \mathbf{v}_i - (dt/\rho_i)\,[\mathbf{G}p]_i$
 5:     $\mathbf{q}_i \leftarrow \mathbf{q}_i + dt\mathbf{v}_i$
 6: end for
 7: Update volume $\overline{V}$.
 8: Enforce volume $\overline{V}$ through weight optimization (Equation (2.22)).
 9: for all particles $i$ do
10:     $\mathbf{q}_i \leftarrow \mathbf{b_i}$
11: end for
12: Enforce volume $\overline{V}$ through weight optimization (Equation (2.22)).
```

4.4.1 Incompressible fluid solver

Our numerical scheme to simulate incompressible fluids makes use of all the tools we presented so far. We now provide detailed comments on their implementations.



Initialization. We begin by instantiating a set of n power particles in the domain Ω that forms a centroidal power diagram of uniform cell volume $\{\overline{V}_i \equiv |\Omega|/n\}_i$. This partitioning is efficiently generated by alternating position and weight optimization of the energy $\mathcal{F}$, as discussed in Section 3.4.7 and originally recommended in [dGBOD12]. Typically, a maximum of 5 to 10 optimization steps of positions and weights were sufficient to generate a uniform sampling of the domain (see inset). The resulting power particles $\{(q_i, w_i)\}_i$ are each given a density ρ_i and a mass m_i such that $m_i = \rho_i \overline{V}_i$. Finally, their velocity is assigned, either to zero or to a user-prescribed initial value. We are now ready to follow the simulation loop described in Algorithm 2.

Divergence-free velocity field. In the case of incompressible fluids, internal forces come directly from the gradient of pressure in order to enforce a divergence-free velocity field. We compute this pressure in Step 2 by solving for the set of scalars $p = \{p_i\}_i$ that satisfy Equation (4.1). Step 4 then subtracts pressure forces from the particles, thus projecting the velocity field onto the space of divergence-free vector fields as desired. Note that no kernel-based approximations of either pressure or internal force (used in SPH methods) are required.

Volume preservation. Displacing particles along a divergence-free velocity field with *finite* time steps would irremediably lead to a numerical drift of the volumes of fluid parcels. Therefore, in addition to controlling the divergence of the velocity field, we also enforce incompressibility directly for the power particles by keeping the volume of each cell constant. This means that Step 7 is rendered unnecessary for incompressible flows as each $\overline{V}_i$ remains fixed throughout the animation, and the weights are computed as described in Section 2.5.1.

The rest of the simulation loop is straightforward: we simply apply the correction defined in Section 4.3.5 to maintain a well-centered geometric sampling of the flow. Note that Steps 8 and 12 involve an update of the weights, and thus, an update of the power diagram connectivity.

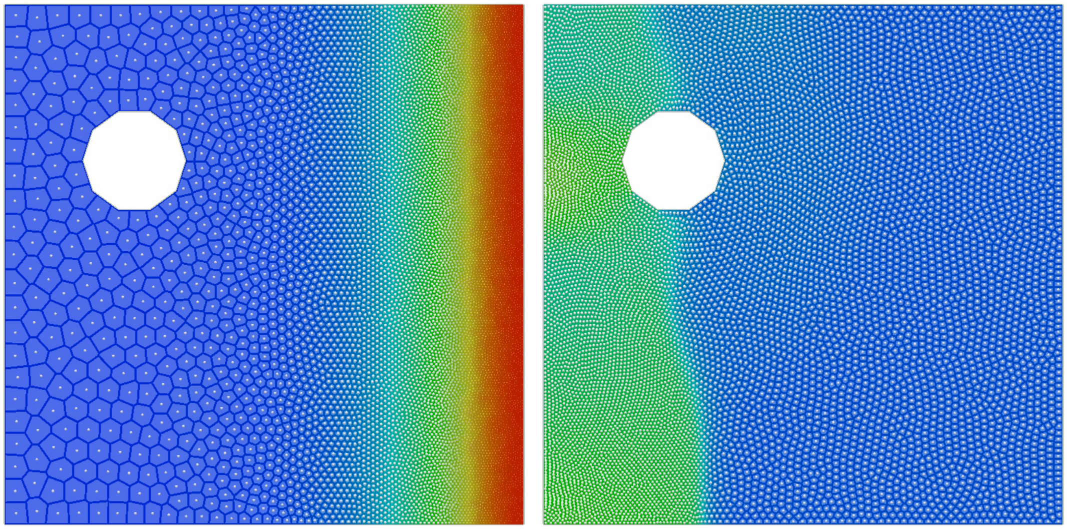


Figure 4.3: **Compressible flow.** Power particles can handle compressible flows as well. In this example, the shallow water equations are simulated in a square domain with a polygonal obstacle. Starting from a step-like height field (represented via color ramp), a wave comes crashing onto the obstacle and the left wall, creating a wide variety of particle sizes indicating compression.

4.4.2 Compressible fluids

The case of compressible fluids requires only minor modifications. First, the scalar pressure p in Step 2 is now computed via an arbitrarily chosen state equation. We use in our implementation the simple constitutive law

$$p_i = \kappa\,(\rho_i - \overline{\rho}_i), \tag{4.2}$$

where κ is tantamount to a stiffness, and ρ_i and $\overline{\rho}_i$ are, respectively, the current and the initial density of particle i. The velocity is then updated with the pressure forces in Step 4, which, unlike in the incompressible case, will no longer result in a divergence-free field. We then apply, in Step 7, the local volume change induced by the velocity divergence as an update of the target volumes $\{\overline{V}_i\}_i$:

$$\overline{V}_i \leftarrow \overline{V}_i + dt\,[\mathbf{Dv}]_i\,. \tag{4.3}$$

Observe that the new volumes always sum to the domain volume, $\sum_i \overline{V}_i = |\Omega|$, as a result of our definition of the divergence operator in Section 2.4 verifying the identity $\sum_i [\mathbf{Dv}]_i = 0$ (Equation 2.12). All other steps remain unchanged.

4.4.3 Multiphase flows

Our power particle framework also accommodates the simulation of multiphase flows, i.e., multiple fluid types interacting within the same domain. For interacting fluids of the same kind but different densities, our method is applicable with no modification other than proper initialization of the densities (and, consequently, of the masses) per particle. For the interaction of incompressible and compressible fluids, Steps 2 and 7 of the algorithm are implemented in such a way that compressible particles are updated through a pressure from a state equation, while incompressible particles are adjusted via a pressure projection. The latter also requires a slight modification for the boundary condition of Equation (4.1): we set Dirichlet boundary conditions along the fluids' interface, and keep Neumann boundary condition for particles at the domain boundary $\partial\Omega$.

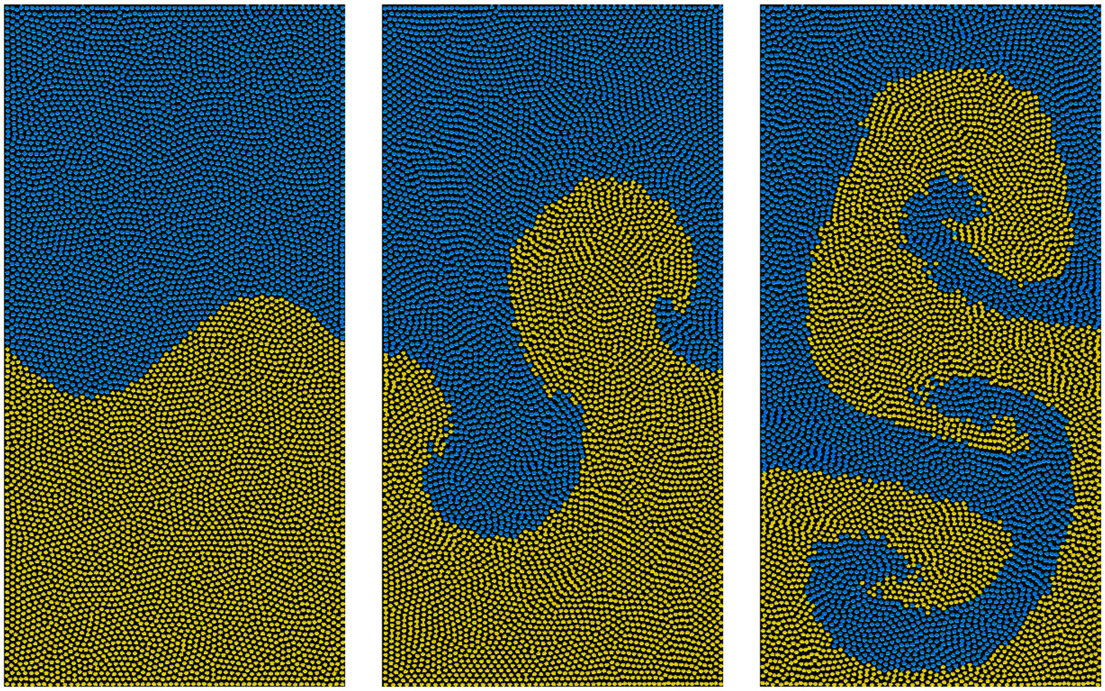


Figure 4.4: **Rayleigh-Taylor instability.** A heavy fluid on top of a lighter fluid develops an instability at the interface as the top fluid is accelerated into the other due to gravity ($t = 1, 3, 5$).

4.4.4 Surface Tension

Surface tension appears at the interface between two fluids due to a difference in cohesion forces on each side. It is often modeled as a stretched elastic membrane, resulting in forces minimizing the area of the interface. Here again, our use of power particles is particularly convenient as it can easily incorporate many of the existing strategies to approximate surface tension forces. Our implementation follows the approach in [LSSF06], which adds the interface curvature as discontinuities of pressure in the projection step. More specifically, we first compute particle normals by normalizing the gradient of the fluid density $\mathbf{n} = \mathbf{G}\rho/\|\mathbf{G}\rho\|$, and then estimate the mean curvature κ at interface particles as $\kappa = -\mathbf{D}\mathbf{n}$. Finally, we add the term $\tau(A_{ij}/l_{ij})\,(d_{ji}\kappa_i + d_{ij}\kappa_j)$ in the right-hand side of Equation (4.1), which accounts for the integrated curvature at the interface facet scaled by the tension coefficient τ.

4.4.5 Viscosity

Viscosity is also easily incorporated into our framework. We compute a viscosity force at particles by applying the Laplacian operator $\mathbf{L}$ for each Cartesian coordinate of the velocity field $\mathbf{v}$, resulting in a viscosity force of the form: $\mathbf{f}_i^{\text{visc}} = \nu \left[\mathbf{L}\mathbf{v}\right]_i$, with ν the viscosity coefficient. These forces are added as external forces in Step 1 of our simulation.

4.4.6 Heat diffusion and buoyancy

Particles can also carry temperatures $\{T_i\}_i$, with heat transfer between material points simulated through the heat equation using the discrete Laplace operator $\mathbf{L}$ as: $T_i \leftarrow T_i + dt \left[\mathbf{L}T\right]_i$. Buoyancy is then added as an upwards external force per particle proportional to the local temperature.

4.4.7 Spatial adaptivity

Power particles can be initialized with an arbitrary density function through importance sampling (see, for instance, Chapter 3). As a result of this spatially-varying density distribution, the target volumes $\{\overline{V}_i\}$ and, consequently the particles masses $\{m_i\}$, are no longer uniform. The simulation loop, however, remains unchanged. See Figures 4.5 and 4.8 for an example of incompressible fluid simulation initialized with different sampling patterns. One can also adapt the number of particles during the animation to provide a fully adaptive method similar to [ATW13]; however, the design of an efficient, yet accurate adaptive strategy is left as future work.

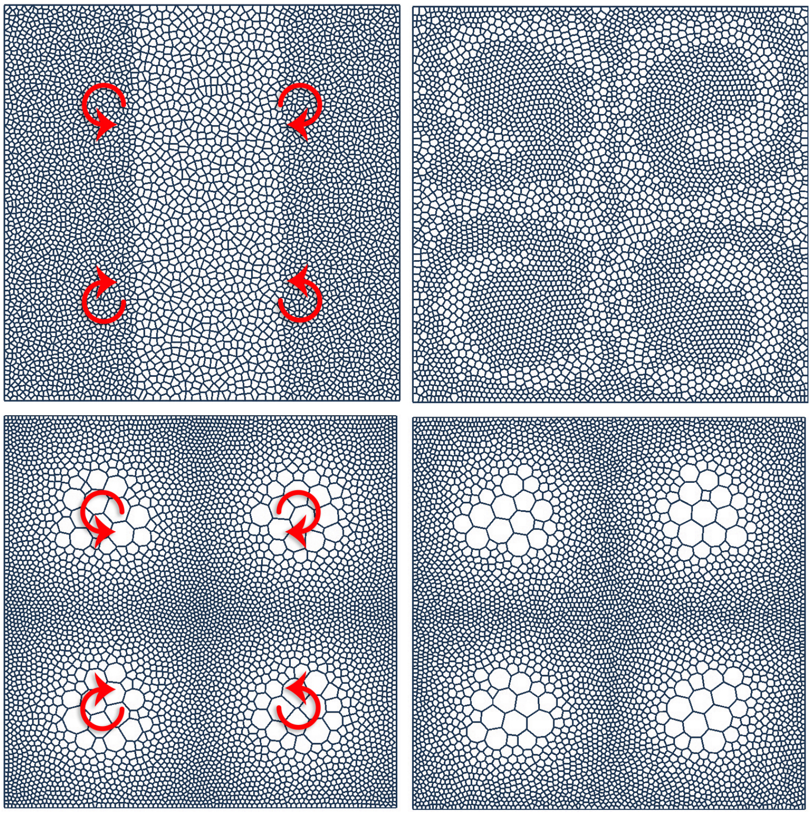


Figure 4.5: **Robustness.** Whether we initialize power particles with variable sampling densities (top) or with an adaptive sampling size (bottom), the four-vortex example of Figure 4.8 is still properly captured (right), with symmetries being preserved over long periods of time.

4.5 Results

We performed a series of tests in 2D and 3D in order to validate our method both in terms of performance and visual quality. A selection of these examples are included in the accompanying video. We describe next the setup of our experiments, and discuss results as compared to previous work.

Implementation. Our implementation makes use of the CGAL library [CGA14] to construct power diagrams in 2D and 3D. For the clipping procedure, we used a mirroring strategy that first detects dual facets intersecting the boundary $\partial\Omega$, then inserts mirrored copies of the incident weighted points. In our experiments, we observed that the mirroring-based clipping gives a speedup of an order of magnitude compared to [YWLL13], specially in the case of simple domains. Note that the mirrored sites are *not* ghost particles, since they are only used temporarily to build dual cells and then discarded. We also used a pre-conditioned conjugate gradient solver to compute the pressure projection in Equation (4.1), as suggested in [Bri08]. We set the tolerance for volume enforcement to 0.1%, which typically requires 1 to 3 Newton steps (Equation (2.22)). Smaller thresholds are easily achieved too, with the computational cost of extra Newton iterations. At last, we notice that Steps (10-12) in Algorithm 2 can be skipped when sites are less than 0.1% of the bounding box size away from their associated centroids, i.e., when the partition remains well-centered after advection.

Comparisons in 2D. Figure 4.8 shows side by side comparisons for the animation of four vortices of an incompressible flow in a 2D unit box, generated by our algorithm versus the FLIP method [ZB05] (code released by the authors) and our own implementation of the Position Based Fluids (PBF) [MM13]. We initialized all cases with the same distribution of 5k particles, and set a time step of $dt = 0.01$. Observe that FLIP exhibits clumping of particles in between vortices, which worsens over time and forms significant voids. The PBF method, on the other hand, maintains an even distribution of particles but at the cost of large numerical damping, even when viscosity is set to zero. Instead, our approach displays an artifact-free motion, keeping particles well-distributed and preserving the vortical structures for uniform and adapted particle sizes alike. Figure 4.8 (top-right) shows the kinetic energy in time for these three methods. Notice that our method has the smallest decay rate, and provides consistent results even with adaptive sampling and with 20x larger time steps (see supplemental video). We also tested the well-known case of a flow in a channel past a disk for a low viscosity fluid. We used 20k particles with non-slip boundary conditions around the disk, and periodic boundary conditions through the vertical walls as a mean to implement inlet and outlet particles in the flow direction. Figure 4.6 shows the expected vortex shedding (von Kármán vortex street) formed by the adherence of the fluid to the boundary, in agreement with physical experiments.

Flows in 3D. We also experimented with our approach for incompressible fluids in 3D (Figure 4.1), including four vortices in a box (see supplemental video) and a smoke flow driven by buoyancy forces (Figure 4.2). For the former, we used 216k particles adaptively initialized with 80% of the particles placed in the top half of the domain (see supplemental video). We also tested our method with non-trivial domains: Fig. 4.7 shows a vortical flow within a twisted torus shape. Lastly, we note that our method keeps a fluid in a container under gravity still, as expected.

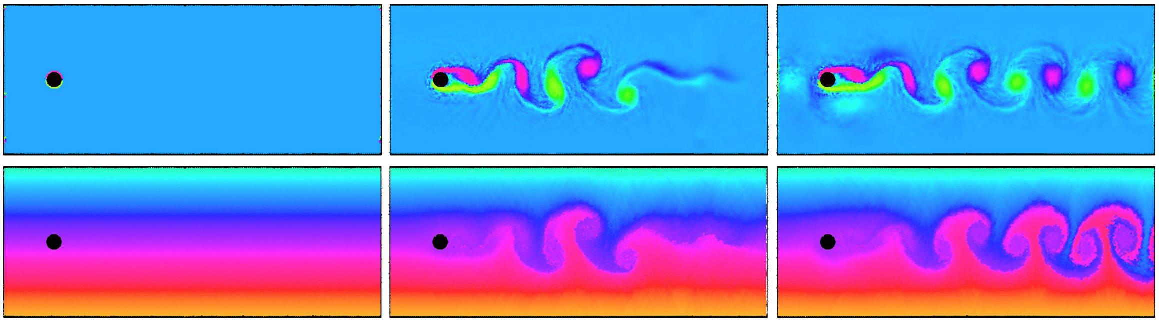


Figure 4.6: **Flow past obstacle.** A slightly viscous flow in a periodic channel with a round obstacle develops the expected von Kármán vortex street: turbulence appears, and the obstacle "sheds" vortices along the channel (top: vorticity plot; bottom: advected dye).

Two-Phase Flows. Our method can directly simulate fluids of variable density. For easy comparison with existing multiphase Lagrangian approaches (see, e.g., [HA07]), we reproduced the 2D Rayleigh-Taylor instability test described in [CR99]. Figure 4.4 shows the fluid configuration at time $t = 1, 3, 5$. Notice the fluid roll-up without smearing the density discontinuities, in contrast to previous results based on SPH methods. We also tested two-phase fluids with larger density ratios in order to mimic water-air interaction, both in 2D and in 3D. Figure 4.1 displays the power particle sites for such flow in 3D, with the addition of gravity and surface tension. We also simulated a liquid-air flow with 140k particles along a U-shaped domain. It is worth mentioning that our method requires the simulation of both air and water particles in order to partition the whole domain. We also addressed air as an incompressible fluid, which is only acceptable for velocities far below the speed of sound.

Compressible Flow. We also demonstrate the flexibility of our approach to simulate compressible fluids. We tested our 2D compressible fluid model to simulate shallow water, in which densities represent particles heights as in [LH10]. Figure 4.3 shows the smooth transition of power cell areas for a shallow water wave in a domain containing an obstacle.

Performance. Our experiments were clocked on an Intel i7 2.2GHz laptop with 4GB RAM. Examples in 2D using between 5k and 20k particles took less than 50 milliseconds per time step (Figures. 4.6, 4.4, 4.3, and 4.8). For profiling purposes, we also tested a 2D example with 250k particles, which took 8 seconds per time step: 30% of the time for the projection step, 50% for two calls of power diagram tessellation, and 20% for two calls of cell clipping. While our 2D clipping procedure is extremely efficient, its 3D counterpart is not, becoming the most time consuming stage of our algorithm. For example, a 3D simulation with 250k particles took 26 seconds per time step, in which 26% of the time was spent on the power diagram tessellation and 67% on clipping. Our implementation relies on the 3D power diagram functionalities of CGAL [CGA14], which are optimized for sequential computing but not thread-safe in the current version. Moderate size examples remain efficient, with 1 second per time step for 5k particles in Figure 4.7, and about 20 seconds per time step for 32k particles in Figures 4.1 and 4.2.

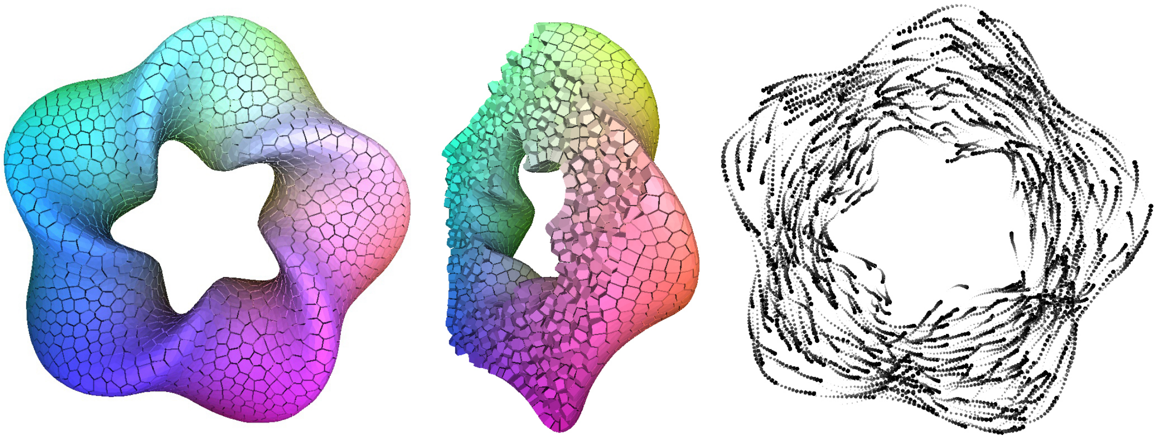


Figure 4.7: **Twisted torus.** An equipartition of a 3D twisted shape (left; cross section displayed in the middle) is used to simulate an incompressible, vortical flow (right).

4.6 Discussion

Power particles offer a new geometric discretization of fluid flows that combines material points and finite volumes tessellating the simulation domain. Both representations are evolved in time without resorting to error-prone kernel evaluations, and result in flows with even particle distributions and full control over local volumes. Our numerical method accommodates the simulation of both compressible and incompressible fluids as well as surface tension and multiphase flows, without the usual clumping of particles near boundaries or excessive numerical damping that is typical of density correction strategies.

Our approach is not without limitations. The performance and scalability of our method are primarily bounded by the construction and clipping of power cells. We are currently investigating extensions to recent parallel implementations of Voronoi diagrams [DS11]. We also conjecture that large improvements can be made by exploring the time coherence and continuity of power particles, which restrict connectivity updates to localized operations. Another limitation relates to our current approach of tessellating the entire domain to generate a water-air animation, instead of directly simulating the liquid with a free surface. Surface tracking with, e.g., level sets could be used to define and advect the interface while power particles could be clipped at the interface. As future work, we are interested in applying our power particle formulation to a broader family of physical simulations, including incompressible elasticity.

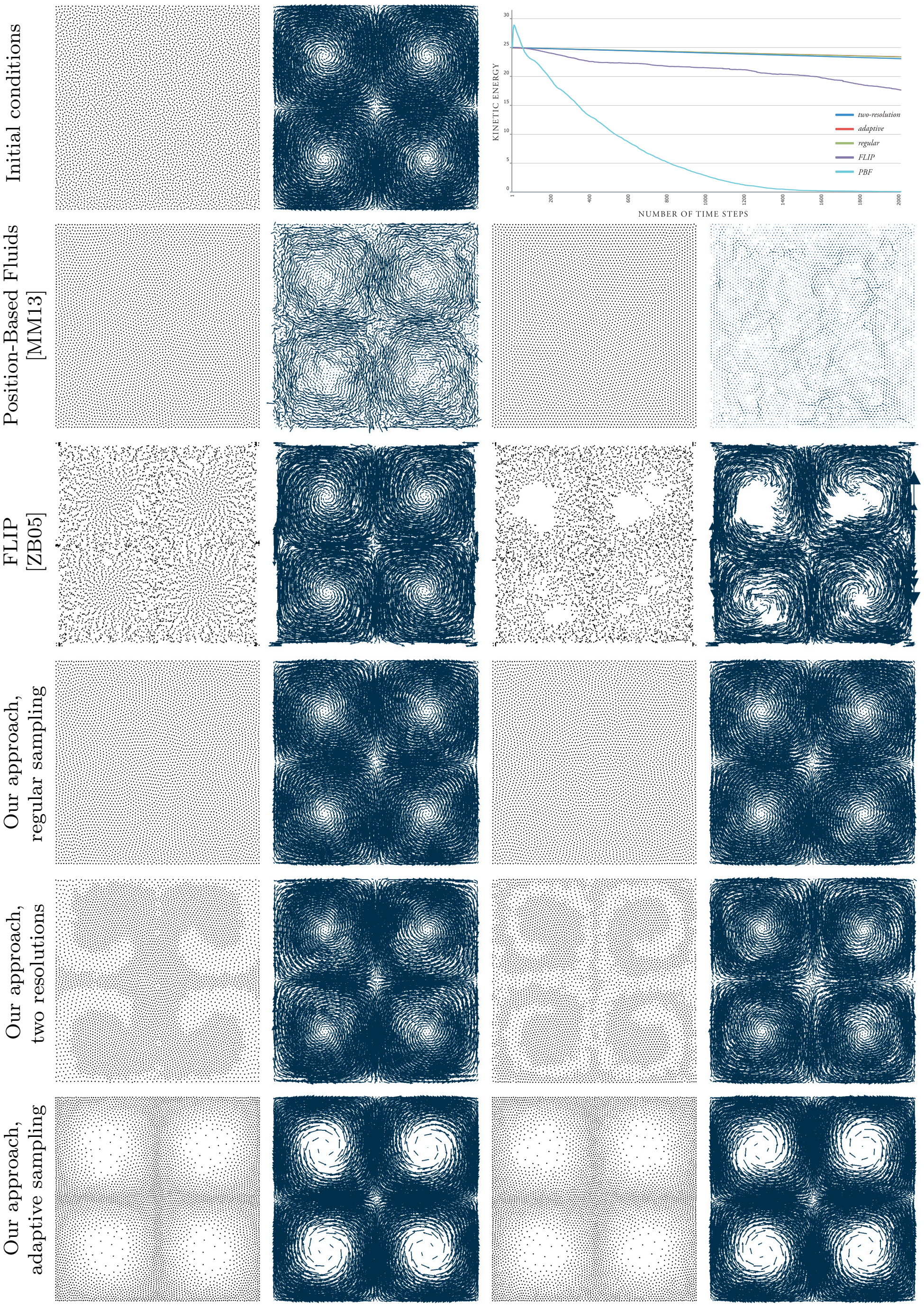


Figure 4.8: **Comparisons.** For an initial sampling of the square domain and an initial divergence-free velocity field forming four vortices (top left), current incompressible SPH methods such as Position Based Fluids [MM13] keep a good particle distribution at the cost of significant motion damping (second row; position and velocity after 1000 and 2000 frames; $dt = 0.01$). FLIP-based methods [ZB05] exhibit less artificial viscosity, but fail to maintain a good distribution (third row, same snapshots). Our approach, instead, preserves the symmetry of the flow even for very different initializations of positions and sizes. Moreover, the flow experiences only limited dissipation as illustrated by the energy plot (top right).

Chapter 5

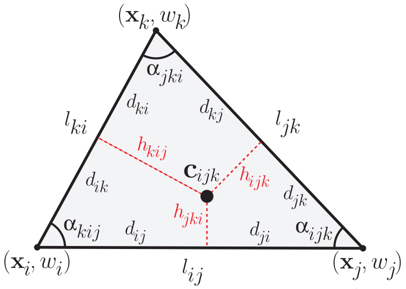
Geometric Discretization through Weighted Triangulations

Thus far we have presented the discretization of Euclidean spaces through power diagrams. However, many computational problems require the discretization of curved domains as well, particularly surfaces. Discrete surfaces are predominantly represented by triangulations, since triangles can conform to and approximate a broad range of shapes and levels of details. Yet, a diversity of geometry processing tasks rely, overtly or covertly, on an orthogonal dual structure to the triangle mesh, inducing the construction of primal-dual complexes. In this chapter, we extend the design of cell complexes introduced in Chapter 2 to surfaces immersed in 3D through the notion of weighted triangulations. We first provide basic definitions of triangulated surfaces, and then derive discrete differential operators that leverage the primal-dual structure of weighted triangulations.

5.1 Preliminaries

We begin with the definition of piecewise linear triangle meshes to approximate smooth surfaces.

Definition 5.1.1. *A primal mesh T is a piecewise linear triangulation with vertices V, edges E, and faces F, defining a discrete orientable manifold surface of any topology in $\mathbb{R}^2$ or $\mathbb{R}^3$, with or without a boundary.*



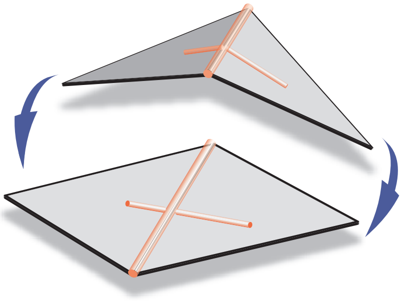
We use single indices (e.g., i) to refer to vertices of T, pairs of indices (e.g., ij) for edges, and triples (e.g., ijk) for triangles. The positions of the vertices are denoted as $\mathbf{p}_i$, and an edge ij can be seen as a 2D/3D vector $\mathbf{e}_{ij} = \mathbf{p}_j - \mathbf{p}_i$. The length of edge ij is denoted by $l_{ij} = |\mathbf{e}_{ij}| \in \mathbb{R}^+$, while $a_{ijk} \in \mathbb{R}^+$ represents the area of triangle ijk. We use ψ_i to refer to the piecewise linear basis function over T for the vertex i (i.e., $\psi_i(\mathbf{p}_j) = \delta_{ij}$, $\forall j \in V$), with which nodal values can be linearly interpolated over T. Angles in T are indexed by the triangle they live in, with the middle index referring to the originating vertex: α_{ijk} thus denotes the tip angle of triangle ijk at vertex j (see inset). Additionally, we define $\theta_i \in \mathbb{R}^+$ as the sum of all tip angles at vertex i, and $\phi_{ij} \in (0, 2\pi)$ as the sum of the (up to two) angles opposite to edge ij. Finally, we call $\mathcal{N}_i$ the set of simplices incident to vertex i.

We enrich the surface discretization by incorporating a dual diagram to the triangulation T, defined via the duality map $*$ that transforms a simplex of T into its dual element [Mun84].

Definition 5.1.2. *The dual diagram of T associates every triangle ijk to a dual vertex $*ijk$, every edge ij to a dual edge $*ij$ connecting the dual vertices of its adjacent triangles, and every vertex i to the dual cell $*i$ formed by the cycle of its neighboring dual edges.*

Note that, similar to Definition 2.1.2, this notion of a dual diagram is purely combinatorial since the actual locations of dual vertices (and thus of the edges) are not yet given. We now focus on the case where dual vertices are placed within the plane of their associated triangles.

Definition 5.1.3. *A dual diagram is denoted intrinsic to T if every dual vertex $*ijk$ is located at a position $\mathbf{c}_{ijk}$ restricted to the plane supporting triangle ijk.*



This definition generalizes dual diagrams to any triangulation, even when triangles are not coplanar. Indeed, any dual edge $*ij$ of T can be constructed by isometrically unfolding the pair of adjacent triangles ijk and imj into the plane, and joining the dual vertices $\mathbf{c}_{ijk}$ and $\mathbf{c}_{imj}$ with a straight path (see inset). We hence refer to such dual edges as *intrinsically straight* segments. We finally introduce an orthogonality condition between primal and dual edges.

Definition 5.1.4. *An orthogonal dual diagram is an intrinsic dual diagram in which the supporting line of each isometrically unfolded dual edge is orthogonal to its respective primal edge.*

A common example of an orthogonal dual diagram consists of associating each dual vertex $*ijk$ to its triangle circumcenter, i.e., the point in the plane of the triangle that is equidistant to the three vertices $\mathbf{p}_i$, $\mathbf{p}_j$, and $\mathbf{p}_k$. In the case of Delaunay triangulations, these circumcentric dual diagrams correspond to Voronoi diagrams. Hereafter we use $\mathbf{c}^{\diamond}_{ijk}$ to denote the circumcenter of triangle ijk.

5.2 Weighted Triangulations

We now focus on *weighted triangulations* [Gli05], a family of triangulations that provides explicit control over the construction of orthogonal dual diagrams.

Definition 5.2.1. *A weighted triangulation T^w is a primal mesh T with an additional set of real numbers $w = \{w_i\}$, called weights, assigned to the vertices of T.*

From these additional weights, we define the position $\mathbf{c}_{ijk}$ of the dual vertex $*ijk$ as the displacement of the circumcenter $\mathbf{c}^{\diamond}_{ijk}$ by (half) the local gradient of the linear function defined by the vertex values w_i, w_j, and w_k.

Definition 5.2.2. *The location $\mathbf{c}_{ijk}$ of the dual vertex $*ijk$ in T^w is defined as:*

$$\begin{aligned} \mathbf{c}_{ijk} &= \mathbf{c}^{\diamond}_{ijk} - \tfrac{1}{2}\left(\nabla w\right)_{ijk} \\ &= \mathbf{c}^{\diamond}_{ijk} - \tfrac{1}{2}\left(w_i \nabla\psi_i + w_j \nabla\psi_j + w_k \nabla\psi_k\right). \end{aligned} \tag{5.1}$$

Note that the displacement per triangle occurs within the triangle's supporting plane, and thus the dual diagram of any T^w is by construction intrinsic to the primal mesh T. Moreover, a gradient vector is

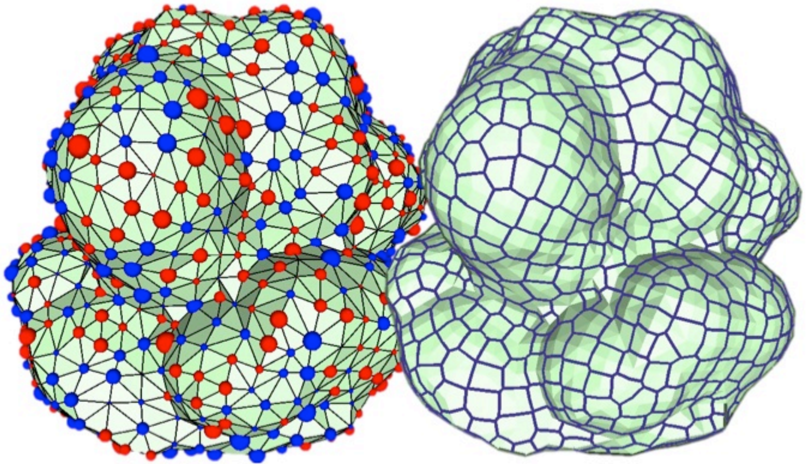


Figure 5.1: **Weighted Triangulation.** Example of a triangle mesh equipped with vertex weights (left) displayed as spheres with squared radii corresponding to the weight magnitudes and colors according to their signs (red+, blue−). The set of weights endows the triangulation with an orthogonal, non-circumcentric dual structure (right).

unchanged if one adds a constant to all weights; thus, weights add $|V|-1$ degrees of freedom to a primal mesh T. It is worth pointing out that this construction does not necessarily result in dual nodes being inside their own triangle. This notion of "off-centered" dual is well documented in the specific case of circumcentric dual for obtuse triangles [MDSB02], and our construction is no different. We can further show that the dual structure of any T^w is an orthogonal diagram.

Proposition 5.2.3. *A weighted triangulation T^w defines an orthogonal dual diagram.*

Proof. Based on Equation (5.1), one can orthogonally project the dual vertex $\mathbf{c}_{ijk}$ to the supporting line of the primal edge ij:

$$(\mathbf{c}_{ijk} - \mathbf{p}_i)^t \, \mathbf{e}_{ij} = \frac{1}{2}\left(l_{ij}^2 + w_i - w_j\right).$$

Since this expression only depends on the primal edge ij, we conclude that both vertices of the dual edge $*ij$ orthogonally project onto the supporting line of the primal edge ij at the same point:

$$\mathbf{c}_{ij} = \mathbf{p}_i + \left(\frac{l_{ij}^2 + w_i - w_j}{2\, l_{ij}^2}\right)\mathbf{e}_{ij}.$$

As a result, the supporting line of $*ij$ passes through $\mathbf{c}_{ij}$ and is orthogonal to the primal edge. □

With the dual diagram in place, we can introduce explicit formulae for the signed measures of the elements in an orthogonal dual diagram. We first define the signed distance between a vertex $\mathbf{p}_i$ and point $\mathbf{c}_{ij}$ as d_{ij}, with a positive sign if $(\mathbf{p}_i - \mathbf{c}_{ij})\cdot(\mathbf{p}_i - \mathbf{p}_j) \geq 0$, and negative otherwise; i.e.,

$$d_{ij} = \frac{l_{ij}^2 + w_i - w_j}{2\, l_{ij}} \quad \text{and} \quad d_{ji} = \frac{l_{ij}^2 + w_j - w_i}{2\, l_{ij}}. \tag{5.2}$$

Note that $d_{ij} + d_{ji} = l_{ij}$. We also define the "height" distance h_{jki} to be the signed distance between $\mathbf{c}_{ijk}$ and $\mathbf{c}_{ij}$, with a positive sign if the triangle $(\mathbf{c}_{ijk}, \mathbf{p}_i, \mathbf{p}_j)$ has the same orientation as the triangle $(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$, and negative otherwise; i.e.,

$$h_{jki} = \frac{l_{ij} \cot \alpha_{jki}}{2} + \frac{\cot \alpha_{kij}}{2\, l_{ij}}(w_j - w_k) + \frac{\cot \alpha_{ijk}}{2\, l_{ij}}(w_i - w_k). \tag{5.3}$$

This height definition allows us to define a signed length A_{ij} of the dual edge $*ij$ associated with a primal edge ij common to ijk and imj as:

$$A_{ij} = h_{jki} + h_{imj}. \tag{5.4}$$

Note that the term A_{ij} extends the facet measure defined in Chapter 2 to signed meshes on surface meshes. This choice of signs matches the traditional sign convention for circumcentric duals (see [HKV13] for a recent exposition). Similarly, one can compute V_i as the signed area of a dual cell $*i$ on a discrete surface through the sum of the signed areas of triangles formed by vertex i and each surrounding dual edge, resulting in:

$$V_i = \frac{1}{2} \sum_{j \in \mathcal{N}_i} d_{ij}\, A_{ij}. \tag{5.5}$$

5.3 Dual Diagrams versus Weighted Triangulations

Weighted triangulations offer parameterized coordinates (namely, the weights) for the construction of orthogonal primal-dual structures on discrete surfaces. We describe next how this scheme relates to the notion of power diagrams introduced Chapter 2, and then discuss further generalizations.

Weighted Circumcenters. Our embedding of dual vertices through displacements of circumcenters (Equation (5.1)) is equivalent to the notion of weighted circumcenter (Equation (2.3)). By rewriting the gradient of basis function ψ_i in triangle ijk as $\mathbf{e}_{jk}^{\perp}/(2\, a_{ijk})$ (where $\perp$ denotes in-plane counterclockwise rotation by $\pi/2$) and using the geometric expression of the circumcenter $\mathbf{c}_{ijk}^{\diamond}$ given in [ACSYD05], we can express the dual position as:

$$\mathbf{c}_{ijk} = \mathbf{p}_i + \frac{1}{4\, a_{ijk}} \left((l_{ij}^2 + w_i - w_j)\, \mathbf{e}_{ij}^{\perp} + (l_{ki}^2 + w_i - w_k)\, \mathbf{e}_{ki}^{\perp} \right). \tag{5.6}$$

Power Diagrams. In Section 2.2, we showed that weighted circumcenters correspond to sites of a power diagram, and hence induce the formation of a dual orthogonal triangulation called weighted Delaunay triangulation. In particular, the weights of a power diagram fully determine the connectivity of the triangulation and enforce non-negative dual lengths ($A_{ij} \geq 0$). Instead, weighted triangulations generalize weighted Delaunay triangulations by decoupling mesh connectivity from weight assignment, and thus allowing orthogonal dual diagrams with negative dual lengths. In fact, any weighted triangulation can be converted to a weighted Delaunay triangulation through a series of edge flips that enforces positive dual lengths [Gli07].

Harmonic Diagrams. Glickenstein [Gli05] showed that the family of weighted triangulations includes all possible pairs of orthogonal primal and dual meshes for *simply connected domains*. In the case of meshes of *arbitrary topology*, additional orthogonal dual diagrams can be constructed based on topological degrees of freedom. As detailed in Chapter 7, these topology-based diagrams correspond to the displacement of dual vertices through *harmonic* vector fields.

Non-Orthogonal Diagrams. We can also use displacement vectors to construct non-orthogonal dual diagrams. In fact, based on the discrete Hodge decomposition of vectors per triangle proposed in [PP03, War06b], one can show that orthogonal dual diagrams correspond to *curl-free* displacement fields, while the non-orthogonal case is determined by displacing circumcenters via the (rotated) gradient of non-conforming functions.

5.4 Discrete Differential Operators

Equipped with an orthogonal primal-dual structure, we can now derive discrete operators useful in geometry processing and graphics applications. Discrete Exterior Calculus (DEC) is a particularly convenient language to easily deduce such operators [Bos98, DKT07, GP10].

Exterior Calculus is a language for differential geometry based on the notion of differential p-forms ($p{=}0,1,2$). In the case of smooth surfaces, 0-forms indicate scalar functions, 1-forms correspond to tangent vector fields, and 2-forms represent anti-symmetric 2×2 matrices. Operations on differential forms are mostly based on the *Hodge star operator* $\star$, which is a metric-dependent map from p-forms to $(2{-}p)$-forms, and the *exterior derivative* d that maps p-forms to $(p+1)$-forms. In particular, exterior derivatives define a chain complex since $d^2=0$. We can also use these operators to characterize a harmonic p-form h as a form that verifies $dh=0$ and $\star^{-1}d\star h=0$. Another key result in the exterior calculus is the Hodge decomposition, with states that any p-form β can be orthogonally split into the sum of a $p-1$-form α, a $p+1$-form γ and a harmonic, i.e., $\beta = d\alpha + \star^{-1}d\star\gamma + h$. We point to the textbook of [AMR88] for a detailed introduction to exterior calculus.

DEC. Differential p-forms are discretized as integrated measurements of the continuous p-form over mesh elements of dimension p, either at the primal or at the dual mesh. For example, a discrete primal 1-form measures the circulation of a vector field along a primal edge, while a discrete dual 0-form evaluates a scalar function at dual vertices. Based on these quantities, discrete exterior derivatives are constructed in order to define a discrete chain complex [Mun84]. The discrete exterior derivative from 0-forms to 1-forms, for instance, corresponds to the incidence matrix $\mathbf{d}_0$ of vertices and edges ($|E|$ rows, $|V|$ columns), in which each row contains a single $+1$ and -1 for the endpoints of a given edge (the sign being determined from the chosen edge orientation), and zero otherwise. Similarly, the discrete exterior derivative from 1-forms to 2-forms corresponds to the incidence matrix $\mathbf{d}_1$ of edges and faces ($|F|$ rows, $|E|$ columns), with $+1$ or -1 entries according to the orientation of edges as one moves counterclockwise around a face. Note that this discretization is purely combinatorial and verifies, by construction, the chain complex condition: $\mathbf{d}_1\mathbf{d}_0=0$. The discrete Hodge star $\star$, on the other hand, takes geometry into account by encoding how integrated measures over primal elements are transferred to their associated dual elements. More concretely, the discrete Hodge star for discrete p-forms is defined as a diagonal matrix with entries per primal element σ of dimension p given as:

$$\star_\sigma := |*\sigma|/|\sigma|, \tag{5.7}$$

where $|\sigma|$ indicates the signed measure of σ. With these operators, we define a discrete Hodge decomposition for a p-form β as $\mathbf{d}\alpha + \bigstar^{-1}\mathbf{d}\bigstar\gamma + h$.

The orthogonality of primal-dual meshes can be directly leveraged in the construction of DEC operators. While the discrete exterior derivatives remain unchanged, we make use of the closed-form expressions for primal and dual volumes to define a weighted version of the discrete Hodge star operator, leading to the following expressions:

- Primal two-forms: $\bigstar_2 = \mathrm{diag}(\{\star_{ijk}\}_{ijk\in F})$, with
$$\star_{ijk} = 1/a_{ijk}. \tag{5.8}$$
- Primal one-forms: $\bigstar_1 = \mathrm{diag}(\{\star_{ij}\}_{ij\in E})$, with
$$\star_{ij} = A_{ij}/l_{ij}. \tag{5.9}$$
- Primal zero-forms: $\bigstar_0 = \mathrm{diag}(\{\star_i\}_{i\in V})$, with
$$\star_i = V_i. \tag{5.10}$$

Observe that the dual structure for constant weights corresponds to the circumcentric dual, and therefore the Hodge star for one-forms reduces to the well known *cotan* formula in this case [Mac49]. We will use $\star^{\diamond}_{ij}$ to denote this "unweighted" star, i.e., for an edge ij between triangles ijk and imj,

$$\star^{\diamond}_{ij} \;\equiv\; \star_{ij}\Big|_{w=\text{const}} = \frac{1}{2}\Big(\cot\alpha_{jki} + \cot\alpha_{imj}\Big). \tag{5.11}$$

We also use $A^{\diamond}_{ij}$ and $V^{\diamond}_i$ to indicate the circumcentric dual edge length and cell volume, respectively. Based on Equations. (5.3) and (5.4), the weighted version of the Hodge star for one-forms is therefore:

$$\star_{ij} = \star^{\diamond}_{ij} + \frac{\cot\alpha_{kji}}{2l^2_{ij}}(w_i - w_k) + \frac{\cot\alpha_{jik}}{2l^2_{ij}}(w_j - w_k) + \frac{\cot\alpha_{ijm}}{2l^2_{ij}}(w_i - w_m) + \frac{\cot\alpha_{mij}}{2l^2_{ij}}(w_j - w_m). \tag{5.12}$$

Laplacian. A particularly common operator, used in countless geometry processing tools, is the Laplace-Beltrami operator. Using the DEC framework, we can define it for weighted triangulations as the matrix $\Delta = \mathbf{d}_0^t \bigstar_1 \mathbf{d}_0$. This extends the familiar cotan-Laplacian matrix $\Delta^{\diamond}$ [PP93], which corresponds to the Laplace-Beltrami operator with zero weights (or, equivalently, constant weights). Note that Δ is a symmetric matrix providing the "weak" (i.e., integrated) version of the operator, while the "strong" (i.e., pointwise) version is simply the generally non-symmetric matrix $\bigstar_0^{-1}\Delta$. It bears pointing out that, due to the inherent primal-dual orthogonality of weighted triangulations, the operator Δ shares structural properties with the smooth Laplace-Beltrami operator such as symmetry, locality, scale invariance, as well as linear precision on flat domains [WMKG07]. For the case of weighted Delaunay triangulations (i.e., $A_{ij} \geq 0$), this matrix is equivalent to the Laplacian operator derived from power diagrams (Proposition 2.4.5), and it is thus positive semi-definite with constant functions in its kernel. Glickenstein [Gli05] (Theorem 36) further showed that $d_{ij} \geq 0$ for every oriented edge ij is a sufficient condition to reproduce positive semi-definite Laplacians for more general surface meshes, even with negative dual edge lengths.

Chapter 6

Discrete Metric through Weighted Triangulations

We now investigate the use of weighted triangulations to enrich the approximations of surfaces for digital geometry processing. By incorporating a scalar weight per mesh vertex, we introduce a new notion of discrete metric that improves the discretization of shapes and differential operators. We also present alternative characterizations of this primal-dual structure (through combinations of angles, areas, and lengths) and, in the process, uncover closed-form expressions of mesh energies that were previously known in implicit form only. Finally, we demonstrate how weighted triangulations improve the robustness of geometry processing applications, such as the generation of well-centered meshes and sphere packing.

6.1 Introduction

The metric of a smooth surface defines an inner product between tangent vectors, thus allowing for the measure of length, area, and Gaussian curvature on the surface. On a triangle mesh, this notion of metric needs to be properly discretized in order to provide the same set of measurements, and one common option is to use a piecewise Euclidean metric per triangle. However, this choice makes the accuracy and robustness of computations heavily dependent on the shape and size of the triangulation elements [She02]. In this chapter, we propose an augmented notion of discrete metric that additively perturbs the piecewise Euclidean metric by decorating mesh vertices with weights. Our approach builds upon the concept of weighted triangulations (as described in Chapter 5), and hence offers more flexibility to construct accurate discrete operators.

We begin our exposition by defining an admissible set of discrete metrics for weighted triangulations. We then present three alternative representations of the metric of a weighted triangulation, tying together a large number of previous works. Finally, we use our metric representations to design new meshing algorithms: we first extend the work of [MMdGD11] to generate well-centered surface meshes, and then provide a new approach to sphere packing on surfaces that improves robustness compared to [SHWP09].

6.2 Related Work

Our contributions relate to a number of research efforts and applications in computer graphics, computational geometry, and discrete mathematics. We briefly discuss the most relevant topics.

Primal-Dual Meshes. While a discrete surface is often described using only a triangle mesh, an increasing number of methods make use of an orthogonal dual structure as well: this form of Poincaré duality [Mun84] is valuable in modeling (e.g., for mesh parameterization [Mer01]) and physical simulation (e.g., for fluid flows [BXH10]) alike. Orthogonal dual meshes are most commonly constructed by connecting neighboring triangle circumcenters [MDSB02]. However, this choice of dual mesh is only appropriate for so-called pairwise-Delaunay triangulations (see, e.g., [DS09, HKV13]), while most triangulations require combinatorial alterations for this dual to be well formed [FSSB07]. This construction is thus often too restrictive for the demands of many graphics applications such as the generation of well-centered meshes [VHGR10, MMdGD11] or the construction of discrete Laplacian operators with only positive coefficients [WMKG07, VHWP12]. To overcome these issues, weighted triangulations [Gli05] were introduced as an alternative with much greater flexibility in the location of dual vertices while maintaining primal-dual orthogonality. In this paper, we present further theoretical foundations and computational methods required for the construction of weighted triangulations and their use in geometry processing.

Discrete Metric. Edge lengths are commonly used to define the discrete metric on triangle meshes: once edge lengths are given, measurements such as angles, areas, and Gaussian curvatures are easily computed [Reg61]. However, alternative representations of the discrete metric have also been shown effective in many geometry processing applications. Angles, for instance, are known to characterize the intrinsic geometry of a triangle mesh up to a global scaling. In fact, Rivin [Riv94] even studied the use of angle sums (per vertex, or per edge) as a more concise encoding of the metric. His ideas were further explored and applied to circle patterns [BS03, KSS06] and conformal equivalence [SSP08]: while the former derives the metric of a triangle mesh by assigning an angle-based curvature per edge, the latter determines the metric as a length cross ratio per edge and a curvature per vertex. More recently, Zeng et al. [ZGLG12] proved that the ubiquitous cotan formula [Mac49, PP93], used in the discrete Laplace-Beltrami operator, also determines all the edge lengths of a triangle mesh up to a global scaling and thus encodes its discrete metric. The notion of discrete metric for weighted triangulations has, comparatively, received far less attention. Our work presents several characterizations of the augmented metric resulting from the orthogonal primal-dual structure of weighted triangulations.

Circle packing. Weighted triangulations are closely related to the circle packing problem, as each vertex weight can be interpreted as the squared radius of a vertex-centered circle. This problem goes back to an unpublished work by W. Thurston that presented circle packing as a discrete notion of conformal mapping [Thu76]. Packing circles on triangle meshes has been extensively investigated in graphics. The assignment of circles per triangle, for instance, corresponds to the circle pattern problem for which convex functionals were introduced in [Riv94, BS03, BPS10] and applied to mesh parameterization [KSS06]. Another example, used in graphics to compute geometric flows [JKLG08], is the Andreev-Thurston circle packing [Ste03, CL03] which defines a family of vertex-centered circles such that the circles incident to any edge intersect. This idea was further extended to non-intersecting circles through inversive distance circle packing [Guo09, YGL$^+$09, Luo10], while tangency of neighboring circles corresponds to sphere packing [Col91]. Schiftner et al. [SHWP09] showed that sphere packing only exists for triangle meshes in which the incircles of neighboring triangles are also tangent. Our definition of the augmented metric of weighted triangulations encompasses (and helps put in perspective) all these variants of the circle packing problem on arbitrary triangulated surfaces. In particular, our methodology to construct orthogonal dual meshes provides a novel and reliable computational framework to generate sphere and incircle packing.

6.3 Metric on Triangulated Surfaces

We now revisit the definition of metric on discrete surfaces and present validity conditions for the case of weighted triangulations. We use the same notation introduced in Section 5.2.

Primal Metric. A triangulation represents a piecewise flat surface, hence the metric of the primal mesh alone is piecewise Euclidian. Arguably the most common representation of this metric is through $|E|$ edge lengths, used in many scientific contexts, including early work in numerical relativity [Reg61]. In order to define valid triangles, the lengths of a mesh must be positive and satisfy triangle inequalities.

Definition 6.3.1. *A primal metric in T^w is a set of edge lengths $\{l_{ij}\}$ such that*

$$\begin{aligned} \forall ij \in E: \quad & l_{ij} > 0, \\ \forall ijk \in F: \quad & l_{ij} + l_{jk} - l_{ki} > 0, \\ & l_{jk} + l_{ki} - l_{ij} > 0, \\ & l_{ki} + l_{ij} - l_{jk} > 0. \end{aligned} \tag{6.1}$$

We hereafter denote by $\mathbb{L}$ the set of primal metrics.

Dual Metric. Our weight-based construction of a dual brings about the need to measure dual elements as well, for which the primal lengths alone do not suffice. With lengths and weights, one can express the measure of each dual edge and cell as described in Sections 5.2 and 5.4. The dual structure generated by the weights is, however, unique to the discrete case since the distinction between primal and dual vanishes in the smooth limit. We thus define a dual metric as a *restricted* set of $|V|-1$ weights whose range vanishes as the mesh is appropriately refined (see, e.g., [HPW06]) towards a smooth surface.

Definition 6.3.2. *A dual metric in T^w is a set of vertex weight $\{w_i\}$ such that*

$$\forall ij \in E: \quad |w_i - w_j| \leq l_{ij}^2. \tag{6.2}$$

Using Equation (5.2), we can reexpress this condition as $d_{ij} \geq 0$. Geometrically, this means that the supporting line of each (isometrically unfolded) dual edge of T^w intersects the *interior* of its associated primal edge. The dual structure is therefore constrained to "follow" the primal structure as it converges to a smooth surface under refinement. Note, however, that this condition does not enforce positivity of the dual edge lengths.

Augmented metric. Once primal and dual metrics are known, one can measure any mesh element of a weighted triangulation T^w, be it primal or dual. We thus refer to the pair (l, w) of edge lengths $l = \{l_{ij}\}$ (determining the primal mesh) and weights $w = \{w_i\}$ (defining the orthogonal dual mesh) as the *augmented metric* of T^w.

Definition 6.3.3. *We define $\mathbb{L}^{\mathbb{W}}$ as the set of augmented metrics (i.e., with lengths and weights satisfying Equations* (6.1) *and* (6.2)*), with a total of $|E|+|V|-1$ degrees of freedom.*

Accuracy. Regarding numerical analysis, tight error bounds of DEC operators on weighted triangulations were previously addressed in [MMdGD11]. This work demonstrated that the inclusion of weights not only extends the typical finite element operators on meshes, but can also improve the accuracy of the discrete operators (without affecting the order of convergence). The authors proposed to compute weights as the minimizer of a family of energies that measures the discretization error of the diagonal Hodge star operators in Equations (5.8), (5.9), and (5.10). We finally point out that higher order accuracy can be achieved, if necessary, through the use of high-order approximations of the Hodge star [BSV10, Arn13].

Convergence. Based on our definition of the dual metric, we can also analyze the convergence of discrete operators derived from weighted triangulations to their smooth counterparts. Under the assumption that a sequence of triangle meshes converges to a smooth surface in the Hausdorff distance [HPW06], the validity of dual metrics (Equation (6.2)) implies that the weights become constant in the limit. As a consequence, Δ tends to the same operator in the limit as $\Delta^{\diamond}$, and thus converges to the smooth Laplace-Beltrami operator. We also notice that Equation (6.2) corresponds to a *sufficient* condition introduced in [Gli05] (Theorem 36) to enforce that the weighted Laplace-Beltrami operator is positive semi-definite with only constant functions in its kernel, and therefore satisfies the Lax-Milgram theorem. We thus conclude that our augmented metric shares the same properties as the usual piecewise Euclidean representation, while offering additional control over the construction and accuracy of discrete operators.

6.4 Alternative Metric Characterization

While edge lengths and vertex weights are a convenient way to represent the metric of a weighted triangulation of a given connectivity, they are far from being the only set of geometrically relevant coefficients carrying the same information. In fact, depending on the targeted applications, more convenient descriptions of the weighted metric are available. In this section, we present three alternative representations of the metric, each being a set of measurements on T^w from which the lengths and weights of the weighted triangulation can be recovered (Figure 6.1). We show their equivalence by describing an optimization over the set of augmented metrics $\mathbb{L}^{\mathbb{W}}$ to convert each of these metric representations back to the lengths-weights representation.

6.4.1 Edge-Angle / Vertex-Area

Our first metric characterization assumes that edge angle sums $\{\overline{\phi}_{ij}\}$ and positive dual cell areas $\{\overline{V}_i\}$ of T^w are given, along with its connectivity (Figure 6.1, top). We show that one can recover the discrete metric (l, w) of T^w by solving two convex optimizations *sequentially*: first we recover the edge lengths l from $\{\overline{\phi}_{ij}\}$, after which we deduce the weights w based on $\{\overline{V}_i\}$.

The prescribed values $\{\overline{\phi}_{ij}\}$ and $\{\overline{V}_i\}$ must satisfy necessary conditions in order to be associated to a discrete metric (l, w). We first notice that $\overline{\phi}_{ij}$ is the sum of two triangle angles, which implies that $0 < \overline{\phi}_{ij} < 2\pi$. Also, since our discrete surface is made out of Euclidean triangles for which interior angles sum to π, one has $\sum_{ij} \overline{\phi}_{ij} = \pi|F|$. Accounting for this linear constraint, the set of edge angles and vertex areas provide $|E|+|V|-1$ values, like $\mathbb{L}^{\mathbb{W}}$.

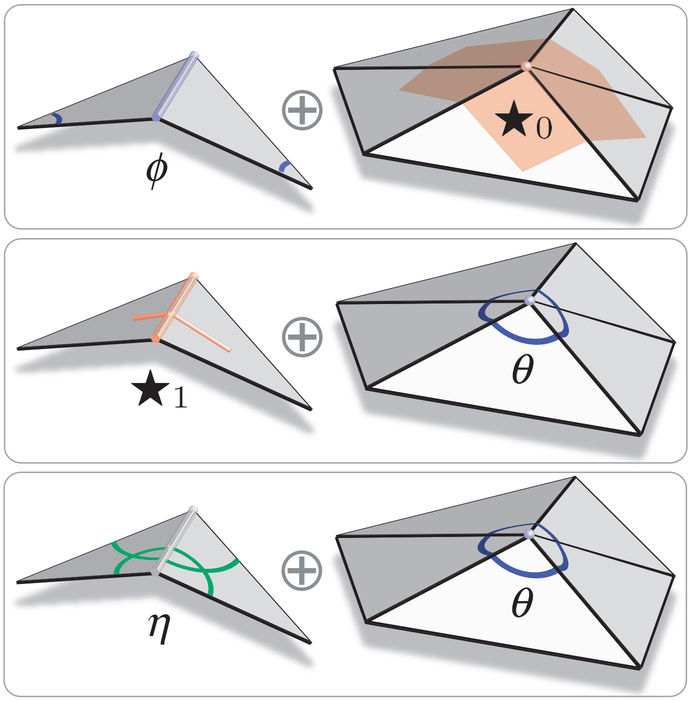


Figure 6.1: **Metric Characterizations.**: We offer three approaches to recover the augmented metric (edge lengths and vertex weights) of a weighted triangulation based on familiar geometric measurements: (top) angle sums ϕ per edge and discrete Hodge star values $\star_0$ per vertex corresponding to dual areas; (center) angle sums θ per vertex and discrete Hodge star values $\star_1$ per edge corresponding to primal-dual length ratios; (bottom) angle sums θ per vertex and inversive distances $\boldsymbol{\eta}$ per edge corresponding to the cosine of intersecting angles of vertex-centered circles.

Circle Pattern Energy. Finding edge lengths $\{l_{ij}\}$ associated to the prescribed edge angle sums $\{\overline{\phi}_{ij}\}$ in T^w is akin to the *circle pattern problem* [Riv94, BS03]. We can solve for lengths by minimizing a convex energy $\mathcal{P}$ as introduced in [BPS10]. We start by adopting the change of variables $u_{ij} = \log l_{ij}$. Note, however, that this logarithmic map transforms the triangle inequalities from linear functions of $\{l_{ij}\}$ (Equation (6.1)) into non-linear functions of length scalings $\{u_{ij}\}$, and thus the convex set $\mathbb{L}$ becomes a non-convex set $\log \mathbb{L}$. We overcome this issue by extending $\log \mathbb{L}$ to a convex set as in [SSP08], at the cost of violating the validity of primal metrics (Section 3.1). Within this space of variables $u = \{u_{ij}\} \in \log \mathbb{L}$, we define $\mathcal{P}$ as follows.

Definition 6.4.1. *The circle pattern energy $\mathcal{P}$ is defined as*

$$\mathcal{P}(u) = \sum_{ijk} \Big(\text{Л}(\alpha_{ijk}) + \text{Л}(\alpha_{jki}) + \text{Л}(\alpha_{kij}) \Big) + \sum_{ij} \Big(u_{ij}(\phi_{ij} - \overline{\phi}_{ij}) \Big), \tag{6.3}$$

where Л$(\cdot)$ *denotes the Lobachevsky function [Mil82], and the angles $\{\alpha_{ijk}\}$ and $\{\phi_{ij}\}$ depend on u.*

We can now deduce the first and second derivatives of $\mathcal{P}$, and present the following theorem.

Theorem 6.4.2. *Assume that a connectivity and angle sums $\{\overline{\phi}_{ij}\}$ are given for T^w. Also assume that $0 < \overline{\phi}_{ij} < 2\pi$ and $\sum_{ij} \overline{\phi}_{ij} = \pi|F|$. There exists at most one primal metric $\{l_{ij}\}$ (up to a global scaling) that recovers the prescribed edge angle sums.*

Proof. We first notice that the Lobachevsky function Л$(\cdot)$ [Mil82] and its derivative are given as:

$$\text{Л}(x) = -\int_0^x \log(2\sin t)dt, \qquad \text{Л}'(x) = -\log(2\sin x).$$

Note that we disregarded the absolute value in Л since all the tip angles are assumed to be in the interval $(0, \pi)$. Partial derivatives of $\mathcal{P}$ are then:

$$\begin{aligned}\nabla_{u_{ij}}\mathcal{P} = & \left(\phi_{ij} - \overline{\phi}_{ij}\right) \\ & + \big(u_{ij} - \log(2\sin\alpha_{jki})\big)\nabla_{u_{ij}}\alpha_{jki} + \big(u_{jk} - \log(2\sin\alpha_{kij})\big)\nabla_{u_{ij}}\alpha_{kij} + \big(u_{ki} - \log(2\sin\alpha_{ijk})\big)\nabla_{u_{ij}}\alpha_{ijk} \\ & + \big(u_{ij} - \log(2\sin\alpha_{ilj})\big)\nabla_{u_{ij}}\alpha_{ilj} + \big(u_{il} - \log(2\sin\alpha_{lji})\big)\nabla_{u_{ij}}\alpha_{lji} + \big(u_{lj} - \log(2\sin\alpha_{jil})\big)\nabla_{u_{ij}}\alpha_{jil}.\end{aligned}$$

By applying the sine law, we have for any edge ij in triangle ijk:

$$u_{ij} - \log(2\sin\alpha_{jki}) = \log(l_{ij}/(2\sin\alpha_{jki})) = \log(R),$$

where R is the triangle circumradius. Since $\nabla_{u_{ij}}\left(\alpha_{ijk} + \alpha_{jki} + \alpha_{jki}\right) = 0$, we thus conclude:

$$\boxed{\nabla_{u_{ij}}\mathcal{P} = \phi_{ij} - \overline{\phi}_{ij},} \tag{6.4}$$

so that $\nabla_u \mathcal{P} = 0$ *iff* the prescribed angles are recovered. The Hessian of $\mathcal{P}$ is now expressed as:

$$\nabla_{u_{jk},u_{ij}}\mathcal{P} = -\cot\alpha_{ijk}, \quad \text{and} \quad \nabla_{u_{ij},u_{ij}}\mathcal{P} = \cot\alpha_{ijk} + \cot\alpha_{kij} + \cot\alpha_{lji} + \cot\alpha_{jil}.$$

This matrix corresponds to half of the Laplacian derived with non-conforming finite elements (see, e.g, [PP03]), and it is therefore positive semi-definite with only constant functions in its null-space. Consequently, the energy $\mathcal{P}$ is convex within $\mathbb{L}^{\mathbb{W}}$. Since the extended domain of $\mathcal{P}$ is defined as a convex set, we conclude that the minimization of $\mathcal{P}$ is a convex optimization and, if a solution $\{l_{ij} = e^{u_{ij}}\}$ exists, it is unique up to a global scaling. □

We can further determine the global scaling of $\{l_{ij}\}$ from the prescribed dual cell areas $\{\overline{V}_i\}$, since their sum is equal to the total area of the mesh. We thus pick the properly scaled $\{l_{ij}\}$ so that:

$$\sum_{ijk} a_{ijk} = \sum_i \overline{V}_i. \tag{6.5}$$

Capacity Constrained Energy. Once the edge lengths are recovered, we compute the weights $w = \{w_i\}$ from the prescribed dual cell areas $\{\overline{V}_i\}$ by observing that these values simply impose a "capacity" constraint for each vertex of T^w. Such a *capacity constrained problem* turns out to be an extension to surface meshes of the variational principle introduced in [AHA98], and a closed form expression of the energy's Hessian was recently given in [dGBOD12] (see Section 2.5.1). Note that our context is different from [AHA98, dGBOD12] since both of them were restricted to weighted Delaunay triangulations. Yet, we can modify their formulation to weighted triangulations with metrics in $\mathbb{L}^{\mathbb{W}}$ by introducing the energy $\mathcal{C}$.

Definition 6.4.3. *The capacity constraint energy $\mathcal{C}$ is defined as:*

$$\mathcal{C}(w) = \sum_i w_i \left(V_i - \overline{V}_i\right) - \sum_{ijk\in F}\left(\frac{d_{ij}^3 h_{jki}}{4} + \frac{d_{ij} h_{jki}^3}{12}\right). \tag{6.6}$$

Note that the domain of $\mathcal{C}$ is the set of dual metrics $w = \{w_i\}$ for a given primal metric $\{l_{ij}\}$ in $\mathbb{L}$, and it is thus a convex polytope (Equation (6.2)). We also point out that the second term of $\mathcal{C}$ is a weighted version of the Centroidal Voronoi Tessellation energy [DFG99] introduced in [MMdGD11]. We find weights $\{w_i\}$ satisfying the prescribed dual cell areas $\{\overline{V}_i\}$ based on the following theorem.

Theorem 6.4.4. *Assume that a primal metric $\{l_{ij}\}$ and positive dual cell areas $\{\overline{V}_i\}$ are given for T^w so that Equation* (6.5) *holds. There exists at most one dual metric $w = \{w_i\}$ (up to an additive constant) that recovers the prescribed dual cell areas.*

Proof. Based on Equations (5.2) and (5.3) and a few steps of algebra, one can show that

$$\boxed{\nabla_{w_i}\mathcal{C} = V_i - \overline{V}_i,} \tag{6.7}$$

showing that $\nabla_w \mathcal{C} = 0$ *iff* the prescribed dual cell areas are matched. The Hessian of $\mathcal{C}$ is one half of the Laplace-Beltrami operator Δ introduced in Section 5.4. This matrix is thus positive semi-definite for any metric in $\mathbb{L}^{\mathbb{W}}$, and its null-space only contains constant vectors, corresponding to the fact that the weights are determined up to an additive constant. Since the domain of $\mathcal{C}$ is convex, we finally conclude that the minimization of $\mathcal{C}$ is a convex optimization, and its solution is unique. □

6.4.2 Edge-Star / Vertex-Angle

Our second metric representation is complimentary to the first one, as we now assume a primal-dual length ratio $\overline{\star}_{ij}$ given per edge, and an angle sum $\overline{\theta}_i$ given per vertex (Figure 6.1, center). However, we will use a *single* variational principle to recover, up to scale, the discrete metric (l, w) of a weighted triangulation T^w from which the length ratios and angle sums are derived.

Note that prescribing the sum of tip angles $\overline{\theta}_i$ around vertex i is equivalent to constraining the discrete Gaussian curvature (or excess angle) κ_i of the surface since $\kappa_i = 2\pi - \theta_i$ for non-boundary vertices. Moreover, since the tip angles of each triangle sum to π, a necessary condition for the existence of a metric (l, w) for a given connectivity is that $\sum_i \overline{\theta}_i = \pi|F|$. On the other hand, prescribing the primal-dual length ratio $\overline{\star}_{ij}$ is equivalent to specifying the diagonal Hodge-star for one-forms and consequently determines the discrete Laplace-Beltrami operator Δ. Although the edge ratios and vertex angles involve $|E|+|V|-1$ values in total, they are invariant to scaling and thus this characterization only recovers the metric up to a global scaling.

Conformal Energy. Our formulation is based on the conformal energy, a functional that commonly arises in geometry processing [PP93, Mer01, DMA02, LPRM02].

Definition 6.4.5. *The conformal energy of a piecewise-linear map* $\mathbf{f}$ *between a triangle mesh* $\overline{T}$ *and another mesh* T *of identical connectivity is defined as:*

$$\mathcal{E}_c(\mathbf{f}) = \frac{1}{2}\sum_{ij} \overline{\star}_{ij} \|\mathbf{f}_i - \mathbf{f}_j\|^2 - \text{Area}(\mathbf{f}), \tag{6.8}$$

where the first term is the Dirichlet energy of $\mathbf{f}$ *(measuring the smoothness of* $\mathbf{f}$ *in the metric of* $\overline{T}$*), and the second term is the area of the image of* $\mathbf{f}$ *(i.e., the total area of* T*).*

Given that $\|\mathbf{f}_i - \mathbf{f}_j\|^2 = l_{ij}^2$, where $\{l_{ij}\}$ are the edge lengths of T, and $\text{Area}(\mathbf{f}) = \frac{1}{2}\left(\sum_{ij} \star_{ij} l_{ij}^2\right) = \frac{1}{2}\left(\sum_{ij} \star^{\diamond}_{ij} l_{ij}^2\right)$ as the total surface area is equal to the sum of all disjoint partition areas, no matter how the surface is partitioned, we can rewrite $\mathcal{E}_c$ as:

$$\mathcal{E}_c(l^2) = \frac{1}{2}\sum_{ij}\left(\overline{\star}_{ij} - \star_{ij}\right) l_{ij}^2 = \frac{1}{2}\sum_{ij}\left(\overline{\star}_{ij} - \star^{\diamond}_{ij}\right) l_{ij}^2. \tag{6.9}$$

Observe that the domain of $\mathcal{E}_c$ is the space $\mathbb{L}^2$ of squared edge lengths satisfying Equation (6.1), and thus corresponds to a convex set homeomorphic to $\mathbb{L}$ [ZGLG12].

Based on the cosine law, one can show that:

$$\nabla_{l_{ij}^2} \alpha_{kij} = -\frac{\cot \alpha_{ijk}}{2 l_{ij}^2}, \quad \nabla_{l_{ij}^2} \alpha_{jki} = \frac{\cot \alpha_{ijk} + \cot \alpha_{kij}}{2 l_{ij}^2} = \frac{1}{4 a_{ijk}}.$$

With these identities, the partial derivatives of $\mathcal{E}_c$ become surprisingly simple despite the non-linear dependence of $\star^{\diamond}_{ij}$ on lengths:

$$\boxed{\nabla_{l^2_{ij}}\mathcal{E}_c = \tfrac{1}{2}\Big(\overline{\star}_{ij} - \star^{\diamond}_{ij}\Big).} \tag{6.10}$$

We thus conclude that the energy $\mathcal{E}_c$ is, in fact, a closed-form expression of the variational formulation introduced in [ZGLG12]: their energy, given in implicit form, admits the same gradient with respect to squared lengths. We can consequently employ their results and present the following theorem.

Theorem 6.4.6. *Assume that the connectivity and length ratios $\{\overline{\star}_{ij}\}$ for an unweighted mesh T are given. There exists at most one primal metric $\{l_{ij}\}$ (up to a global scaling) that recovers the prescribed length ratios.*

Proof. From Equation (6.10), we have that $\{\overline{\star}_{ij}\}$ are recovered *iff* $\nabla_{l^2}\mathcal{E}_c\!=\!0$. Zeng et al. [ZGLG12] showed that the derivatives corresponding to the Hessian of $\mathcal{E}_c$ define a positive semi-definite matrix, with global scaling in its kernel, thus $\mathcal{E}_c$ is a convex energy. Since the domain of $\mathcal{E}_c$ is a convex set, we conclude that the minimization of $\mathcal{E}_c$ in l^2 is a convex optimization and thus the solution $\{l^2_{ij}\}$ is unique up to a global scaling. For completeness, we provide the entries of Hessian of $\mathcal{E}_c$ restricted to triangle ijk with circumradius R:

$$\begin{cases} \nabla_{l^2_{jk},l^2_{ij}}\mathcal{E}_c = -\cot\alpha_{ijk}\dfrac{R^2}{2l^2_{jk}l^2_{ij}}, \\[2ex] \nabla_{l^2_{ij},l^2_{ij}}\mathcal{E}_c = (\cot\alpha_{ijk} + \cot\alpha_{kij})\,\dfrac{R^2}{2l^4_{ij}}. \end{cases}$$

□

Notice that $\mathcal{E}_c$ is zero at its minimum since $\nabla_{l^2}\mathcal{E}_c\!=\!0$ implies the equality of (unweighted) stars, and hence a null conformal energy. However, in our context of weighted triangulations, we want to match the *weighted* Hodge star instead.

Curvature Constrained Conformal Energy. In order to extend the energy $\mathcal{E}_c$ to the augmented metric (l, w), we consider a modified energy $\mathcal{L}_c$ that adds target vertex angles $\{\overline{\theta}_i\}$.

Definition 6.4.7. *The curvature constrained conformal energy $\mathcal{L}_c$ is defined as*

$$\mathcal{L}_c(l^2, w) = \mathcal{E}_c(l^2) - \frac{1}{2}\sum_i w_i\Big(\overline{\theta}_i - \theta_i\Big). \tag{6.11}$$

We provide an intuitive interpretation of our energy: $\mathcal{L}_c$ corresponds to the Lagrangian function of the minimization of $\mathcal{E}_c$ subject to the constraints $\{\overline{\theta}_i\!=\!\theta_i\}$, where the weights play the role of (twice the) Lagrangian multipliers. Remarkably, the gradient of $\sum_i w_i(\overline{\theta}_i - \theta_i)$ with respect to l^2 contains exactly the weighted terms of the Hodge star for one-forms, and hence summing this with Equation (6.10) turns the unweighted star $\star^{\diamond}_{ij}$ into a full Hodge star $\star_{ij}$.

Theorem 6.4.8. *The stationary points of $\mathcal{L}_c$ correspond to discrete metrics (l, w) of T^w that recover the prescribed measures $\{\overline{\theta}_i\}$ and $\{\overline{\star}_{ij}\}$.*

Proof. Since $\mathcal{E}_c$ does not depend on w, the w-derivatives are:

$$\boxed{\nabla_{w_i}\mathcal{L}_c = \frac{1}{2}(\theta_i - \overline{\theta}_i).} \tag{6.12}$$

For the partial derivatives w.r.t squared length, one has instead:

$$\begin{aligned}\nabla_{l_{ij}^2}\mathcal{L}_c = \frac{1}{2}\Bigg(\overline{\star}_{ij} - \frac{\cot\alpha_{jki}}{2} - (w_i - w_k)\frac{\cot\alpha_{ijk}}{2l_{ij}^2} - (w_j - w_k)\frac{\cot\alpha_{kij}}{2l_{ij}^2} \\ -\frac{\cot\alpha_{ilj}}{2} - (w_i - w_l)\frac{\cot\alpha_{lji}}{2l_{ij}^2} - (w_j - w_l)\frac{\cot\alpha_{jil}}{2l_{ij}^2}\Bigg).\end{aligned}$$

$$\boxed{\nabla_{l_{ij}^2}\mathcal{L}_c = \frac{1}{2}\left(\overline{\star}_{ij} - \star_{ij}\right).} \tag{6.13}$$

Therefore, the prescribed edge ratios and vertex angles are recovered *iff* $\nabla_{l^2,w}\mathcal{L}_c = 0$. □

Note that our theorem does not address the uniqueness of the critical point of $\mathcal{L}_c$. In fact, the Hessian of $\mathcal{L}_c$ is in general indefinite, and $\mathcal{L}_c$ is thus non-convex. While the uniqueness of a solution of $\mathcal{L}_c$ remains an open question, we present instead additional analysis of its solution space.

Corollary 6.4.9. *If a critical point of $\mathcal{L}_c$ exists with primal metric $\{l_{ij}\}$, then the dual metric is unique up to an additive constant.*

Proof. We deduce from Equation (6.13) that:

$$\nabla_{l^2}\mathcal{L}_c = 0 \quad\Longrightarrow\quad \star_{ij} l_{ij}^2 = \overline{\star}_{ij} l_{ij}^2 \;\; \forall ij \quad\Longrightarrow\quad \sum_{j\in\mathcal{N}_i} \star_{ij} l_{ij}^2 = \sum_{j\in\mathcal{N}_i} \overline{\star}_{ij} l_{ij}^2 \;\; \forall i.$$

Since $\star_{ij}$ is linear in weights (Equation (5.12)), we can express the deduced equalities as a Poisson linear system:

$$\Delta^{\diamond}\, w = b, \tag{6.14}$$

where $b_i = \sum_{j\in\mathcal{N}_i} l_{ij}^2\left(\overline{\star}_{ij} - \star_{ij}^{\diamond}\right)$. Note that the kernel of $\Delta^{\diamond}$ only contains constant vectors. Thus, for a critical point of $\mathcal{L}_c$, b is in the range of $\Delta^{\diamond}$, and there is a unique set of weights (up to an additive constant) satisfying Equation (6.14). □

Finally, we show that the space of *weighted* Hodge stars for one-forms is strictly larger than the space of *unweighted* Hodge stars, confirming the flexibility afforded by weighted triangulations in the discretization of differential operators.

Corollary 6.4.10. *Assume a weighted mesh $\widehat{T}^w$ of metric $(\widehat{l}, \widehat{w})$ is given, with non-constant weights. Then, for any* unweighted *mesh T with the same connectivity as $\widehat{T}^w$: $\star_{ij} \neq \widehat{\star}_{ij}\, \forall ij$.*

Proof. Our proof is by contradiction. Suppose there exists a triangle mesh T of edge length $l = \{l_{ij}\}$ and constant weights $w = \{w_i\}$ such that the equality $\star_{ij} = \widehat{\star}_{ij}$ holds. The length assignment l is then a global minimum of the convex energy $\mathcal{E}_c$ with prescribed discrete Hodge star $\overline{\star}_{ij} \equiv \widehat{\star}_{ij}$. Note, however, that $\widehat{l} = \{\widehat{l_{ij}}\}$ is also a global minimizer of $\mathcal{E}_c$ since $\mathcal{E}_c(\widehat{l}\,) = \mathcal{E}_c(l\,) = 0$ (see Equation (6.9)). Due to the convexity of $\mathcal{E}_c$, we conclude that $\widehat{l} = sl$, where the scalar s indicates a global scaling factor. Now, observe that the linear system in Equation (6.14) for these two meshes differs just in $\widehat{b} = s^2 b$. Thus $\widehat{w} = s^2 w$ and is constant, contradicting the initial assumption that $\widehat{w}$ is non-constant. □

6.4.3 Inversive Distance Circle Packing

Our third and last metric characterization involves the concept of circle packing, a topic that has attracted considerable effort since Thurston [Thu76] linked it to conformal mapping (see [Ste03] for a survey). We show that the discrete metric (l, w) of T^w can be determined by solving an *inversive distance circle packing* problem, from prescribed angle sums $\{\overline{\theta}_i\}$ per vertex, and additional positive values $\{\eta_{ij}\}$ per edge known as the inversive distances (Figure 6.1, bottom).

Overview. One can think of the weight w_i as the square of the radius of an intrinsic circle centered at vertex i. These circles define, within each triangle's supporting plane, a unique circle orthogonal to them, and the center of this circle per triangle is our dual vertex $\mathbf{c}_{ijk}$ as shown, e.g., in [Gli05]. It is therefore no surprise that weighted triangulations are relevant in the context of circle packing. The circle packing problem defines a relation between edge lengths and vertex weights of the form:

$$l_{ij}^2 = w_i + w_j + 2\sqrt{w_i w_j}\,\eta_{ij}. \tag{6.15}$$

This resembles the law of cosines where η_{ij} indicates how neighboring circles intersect. By scaling vertex circles, one can then deform edge lengths while keeping the values $\{\eta_{ij}\}$ invariant. The original circle packing problem (used, e.g., in [JKLG08]) assumes positive weights and intersecting circles with acute angles (i.e., $0 \leq \eta \leq 1$), while inversive distance circle packing [YGL$^+$09] extends this concept to non-intersecting circles (i.e., $\eta \geq 0$). Arbitrarily signed weights can further be treated as discussed in [Aur87b, Ped88].

Circle Packing Energy. Recent results [Guo09, Luo10] have shown the existence of an energy (with no known explicit form) that recovers, at its critical point, an augmented metric (l, w) from vertex angles $\{\overline{\theta}_i\}$ and inversive distances $\{\eta_{ij}\}$. They also showed that this energy relies on the change of variables $u_i = \frac{1}{2}\log w_i$, and Equation (6.15) is then rewritten as:

$$l_{ij}^2 = e^{2u_i} + e^{2u_j} + 2e^{u_i+u_j}\eta_{ij}. \tag{6.16}$$

Yang et al. [YGL$^+$09] later derived that the Hessian of this energy as a function of vertex scalings $\{u_i\}$ is the weighted Laplacian Δ defined in Section 5.4. We can then use the augmented metric set $\mathbb{L}^{\mathbb{W}}$ in order to guarantee that the operator Δ is positive semi-definite and thence that the circle packing energy is convex. Lastly, Guo [Guo09] proved that the space of vertex scalings is convex once $\eta \geq 0$, and the minimization of the circle packing energy is thus reduced to a convex optimization. In the following theorem, we summarize the metric characterization of T^w based on the inversive distance circle packing.

Theorem 6.4.11. *Assume that connectivity, vertex angles $\{\overline{\theta}_i\}$ and positive inversive distances $\{\eta_{ij}\}$ are given for T^w. There exists a convex energy such that its unique minimum (if any) is reached at vertex scalings $\{u_i\}$ defining the augmented metric (l, w) that recovers the prescribed $\{\overline{\theta}_i\}$ and $\{\eta_{ij}\}$.*

6.5 Meshing Applications

Weighted triangulations are amenable to most geometry processing tasks due to their added flexibility for the construction of discrete operators through the design of an extended set of orthogonal dual meshes. In this section, we leverage the insights developed in our work to provide concrete examples of weighted triangulations applied to meshing. In particular, we show how to improve upon recent methods in mesh generation and architecture modeling.

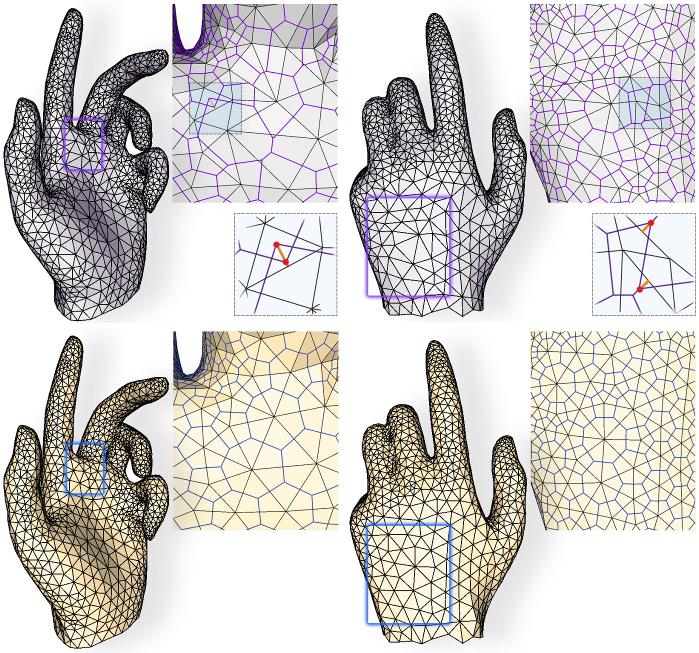


Figure 6.2: **Well-centered Meshes.** The hand model contains many triangles with off-centered circumcenters. Even after optimizing the dual metric w based on Equation (6.18a), most dual vertices get closer to their respective triangle barycenter (top), but a few still remain outside their triangles (see red lines). We can generate a well-centered mesh by optimizing the same energy for both vertex positions and weights, while retaining the original surface shape (bottom).

6.5.1 Well-centered Meshes

In various meshing contexts, it is desirable to place dual vertices inside their associated triangle while retaining orthogonality to the primal mesh [ETK$^+$07, BXH10]. However, generating such "well-centered" meshes with circumcentric duals is challenging [VHGR10]. Alternatively, Mullen et al. [MMdGD11] proposed the use of weighted circumcenters to construct well-centered triangulations in the plane. Their approach is based on the minimization of the functional $\mathcal{M}$ that evaluates the cost of coalescing mesh triangles ijk into their respective dual points $\mathbf{c}_{ijk}$:

$$\mathcal{M}(\mathbf{p}, w) = \sum_{ijk \in F} \int_{ijk} \|\mathbf{y} - \mathbf{c}_{ijk}\|^2 d\mathbf{y}. \tag{6.17}$$

In the case of surface meshes, the authors restricted the minimization of $\mathcal{M}$ only with respect to weights, and showed that the weighted circumcenters move towards their respective triangle barycenters as expected. In order to generate well-centered meshes over non-flat domains, we first point out that the energy $\mathcal{M}$ can be expressed as a sum of two terms:

$$\begin{cases} \mathcal{S}_{\mathbf{b}}(\mathbf{p}, w) = \sum_{ijk} a_{ijk} \|\mathbf{b}_{ijk} - \mathbf{c}_{ijk}\|^2, & (6.18a) \\ \mathcal{I}(\mathbf{x}) = \sum_{ijk} \int_{ijk} \|\mathbf{y} - \mathbf{b}_{ijk}\|^2 d\mathbf{y}. & (6.18b) \end{cases}$$

Note that the first energy $\mathcal{S}_{\mathbf{b}}$ measures how close the dual metric w places the dual vertices $\{\mathbf{c}_{ijk}\}$ to triangle barycenters $\{\mathbf{b}_{ijk}\}$, capturing the key property promoted by [MMdGD11]. The second energy $\mathcal{I}$, on the other hand, corresponds to the sum of second moments of the mesh triangles, and favors the formation of evenly sized triangles. While Mullen et al. [MMdGD11] recommend the sum of these energies in the context of Hodge star optimization, we propose to use only $\mathcal{S}_{\mathbf{b}}$ to construct well-centered surface meshes. We minimize the energy $\mathcal{S}_{\mathbf{b}}$ by alternating optimization of mesh positions $\{\mathbf{p}_i\}$ and weights $\{w_i\}$. We use a L-BFGS method [NW99] to optimize vertex positions. For weight optimization, we exploit our construction of weighted orthogonal duals as displacements of the circumcentric dual (Equation (5.1)), and point out that this minimization amounts precisely to finding a gradient field ∇w on T that best matches (in the L_2 sense) a given vector per triangle. This particular problem is well studied (e.g., in [TLHD03]) and can be efficiently solved with a single linear system:

$$\Delta^{\diamond} w = \nabla \cdot (\mathbf{c}^{\diamond} - \mathbf{b}), \tag{6.19}$$

where $[\nabla \cdot (\mathbf{c}^{\diamond} - \mathbf{b})]_i = \sum_{ijk \in \mathcal{N}_i} \left(\mathbf{c}^{\diamond}_{ijk} - \mathbf{b}_{ijk}\right)^t \mathbf{e}^{\perp}_{jk}$. Figure 6.2 compares the results of weight-only versus position and weight optimization. Note that the weights alone still generate off-centered elements, while our full optimization places dual vertices at the barycenters, and also improves the aspect ratio of the triangles. In our experiments, we used our own implementation of the L-BFGS method with limited memory of size 5, and the library [Dav11] as our sparse linear solver. As the stopping criteria, we set a convergence tolerance of 10^{-10} which was achieved in 10-20 iterations.

6.5.2 Incircle and Sphere Packing

We discussed in Section 6.4.3 the inversive distance circle packing problem and its link to weighted triangulations. A singular case of circle packing occurs when vertex-centered circles are tangent, i.e., when $\eta = 1$. This case, called *sphere packing*, has been used in surface modeling [SHWP09] to design torsion-free structures relevant for architecture. In this same work, the authors noted that a sphere packing corresponds to a triangle mesh with pairwise tangent incircles, a property that induces an orthogonal dual structure in which incenters are dual vertices. We can thus modify Equation (6.18a) to generate incircle and sphere packing through minimization of the energy:

$$\mathcal{S}_{\mathbf{i}}(\mathbf{p}, w) = \sum_{ijk} a_{ijk} \|\mathbf{i}_{ijk} - \mathbf{c}_{ijk}\|^2, \tag{6.20}$$

where $\mathbf{i}_{ijk}$ denotes the incenter of triangle ijk. One can then alternate the minimization of this energy with respect to weights w_i using the Poisson solve in Equation (6.19), and with respect to points $\mathbf{x_i}$ using a

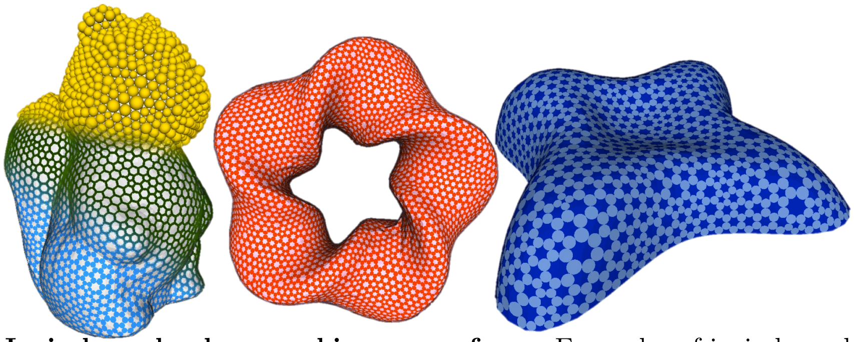


Figure 6.3: **Incircle and sphere packing on surfaces.** Examples of incircle packing on triangulated surfaces with boundary, non-trivial topology, and curved shapes. Incircle packing directly induces sphere packing and an orthogonal dual structure (left).

L-BFGS method [NW99] (with the same parameters as in Section 6.5.1). This approach turns out to be much more robust than the two energies suggested in [SHWP09]: while theirs require an initial guess that is nearly an incircle packing in order to return useful results, our approach is robust to local minima and handles faulty initial guesses quite well, as Figure 6.4 demonstrates. Our approach successfully generates incircle packings on highly curved shapes, with or without boundaries, and with arbitrary topology as shown in Figure 6.3.

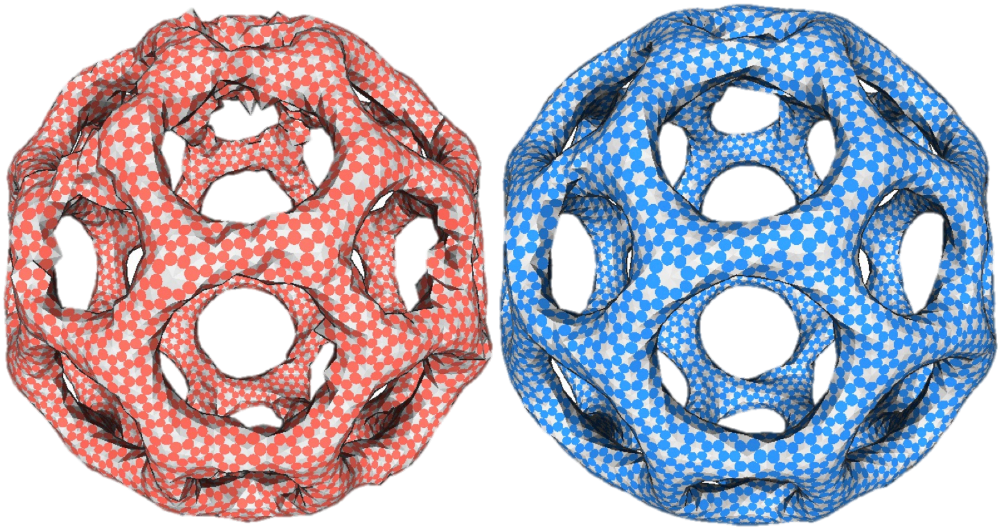


Figure 6.4: **Incircle Packing Comparison.** While the optimization method of [SHWP09] (left) is sensitive to bad initial meshes, our approach (right) reliably generates incircle packings on surfaces of arbitrary shape and topology.

6.6 Conclusion and Future Work

In this chapter, we advocate the use of weighted triangulations in geometry processing as a means to enrich the metric discretization. While our work focused on defining foundations for metric representation on weighted triangulations, we expect that additional benefits will be derived from further studies. On the theoretical side, we are now investigating necessary conditions for the existence of critical points of our metric characterization in Section 6.4. In particular, we conjecture that any weighted Hodge star for one-forms requires at most one set of Gaussian curvatures to be fully determined. This conjecture, if proven correct, would extend the global rigidity theorem presented in [ZGLG12] from triangle meshes to weighted triangulations. Discrete metrics of non-orthogonal primal-dual meshes and polyhedral meshes are also of interest, as well as the extension of our construction to non-intrinsic dual vertices. Further insights may be gathered by linking our characterizations to the use of weighted circle patterns [Spr08, Dim12]. Finally, from a practical standpoint, we believe that geometric dual meshes can bring added robustness to numerous computational tools, and lead to new approaches for geometry processing as demonstrated in Section 6.5; in particular, their use may help extend recent methods to process shapes not solely based on vertex positions, but also on discrete differential geometric quantities [LSLCO05, WBLT12].

Chapter 7

Self-Supporting Structures through Weighted Triangulations

This chapter presents a novel approach for the analysis and design of self-supporting simplicial masonry structures. A finite-dimensional formulation of their compressive stress field is derived, offering a new interpretation of thrust networks through numerical homogenization theory. We further leverage geometric properties of the resulting forces to identify a set of reduced coordinates characterizing the equilibrium of simplicial masonry as the construction of an orthogonal dual diagram. We finally provide computational tools that improve over previous work in efficiency, accuracy, and scalability.

7.1 Introduction

Masonry structures are arrangements of material blocks, such as bricks or stones, that support their own weight. Constructing curved vaults or domes with compression-only structures of blocks, further prevented from slipping through friction and/or mortar, has been practiced since antiquity. It is therefore no surprise that form finding and stability analysis of masonry structures have been an active area of research for years.

The equilibrium of a masonry structure is ensured if there exists an inner *thrust* surface which forms a compressive membrane resisting the external loads [Hey66]. Balance conditions relating the stress field on the thrust surface to the loads are well understood in the continuous setting [GG85, FS03, ABF12]. Discretizing these equations has been done through conforming and non-conforming finite elements, with formulations involving stress, displacements, or both [FAF02, Fra10, Fra11]. The discrete block-based nature of masonry has also led to the analysis of the network of compressive forces keeping masonry blocks together while resisting external loads [Blo09]. The rich geometry of this force diagram has received attention from the geometry processing community as well, with recent work addressing the structural soundness of masonry buildings [WOD09, WSW^{+}12] and on the design of self-supporting polygonal meshes [VHWP12, PBSH13].

In spite of the variety of computational techniques currently available, existing tools still impose stringent limitations on the process of form finding for masonry structures: while previous finite element methods are known to restrict the topology of masonry structures to the case of simply connected domains, thrust network approaches may lead to overconstrained balance equations, depending on the choice of boundary conditions. To overcome these issues, we introduce in this paper a discrete theory of *simplicial masonry structures*. We show that the self-supporting properties of discrete simplicial structures can be derived from a numerical homogenization of the underlying continuous differential equations. By leveraging previous methods, we offer a unified computational framework that enforces the compressive nature and the equilibrium of masonry structures exactly, for surfaces of arbitrary topology. In the process, we introduce reduced coordinates to generate all possible reciprocal force diagrams from simplicial meshes, and reveal geometric connections to well-known continuous notions such as the Airy stress function. Finally, we turn our theoretical contributions into an effective computational technique for the design of simplicial masonry structures that offers improved performance over previous work.

7.2 Discrete Self-Supporting Surfaces

The analysis of masonry structures follows two common assumptions [Hey66, GG85]: no tensile strength or material failure is at play; and there exists a thrust surface, contained within the masonry structure, in static equilibrium with the load applied to the structure (including its own weight). A masonry structure based on these conditions is named a *self-supporting surface*, and we will further concentrate on pure vertical loading. We review next the continuum formulation of self-supporting surfaces, before describing how one can upscale (or homogenize) these equations and turn them into a finite-dimensional problem—which will provide rigorous foundations and extensions to the common use of reciprocal force networks [Blo09, VHWP12].

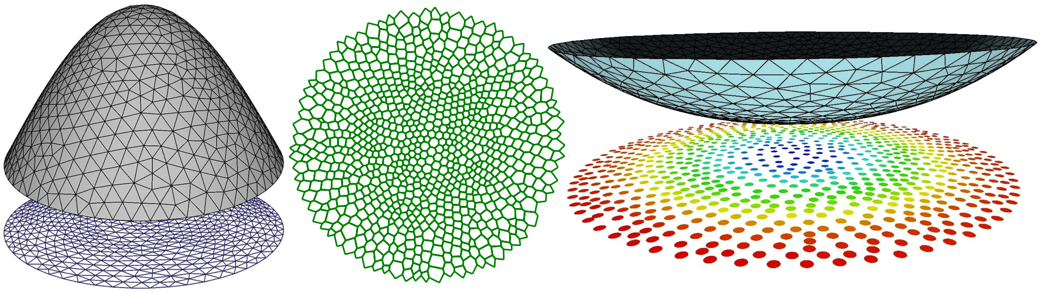


Figure 7.1: **Simplicial Masonry.** A self-supporting simplicial masonry structure is a triangle mesh, defined as a height field over the plane (left). We show through numerical homogenization of the stress tensor that they support their own weights if there exists an orthogonal dual diagram (middle), representing a finite-dimensional approximation of the purely compressive forces at play. We provide discrete counterparts to a number of continuous fields (such as the convex Airy stress function, right) and differential equations traditionally used in describing the equilibrium of these masonry buildings; in particular, we offer a set of reduced coordinates of the space of statically admissible shapes, linking our approach to regular triangulations.

7.2.1 Setup and definitions

Before explaining our approach, we first establish a few notations.

Continuum Setup. A self-supporting surface $\mathcal{S}$ is henceforth represented as a height function z over a two-dimensional domain $\mathcal{U}$ with a Cartesian coordinate system $\mathbf{u} = [u, v]$:

$$[u, v] \to S(u, v) = \big[u, v, z(u, v)\big], \tag{7.1}$$

where we use brackets to concisely denote vectors. The Jacobian matrix $\nabla S = \big[\mathbb{I}, \nabla z^t\big]$ of this map lifts (i.e., pushforwards) vectors from $\mathcal{U} \subset \mathbb{R}^2$ to tangent vectors on $\mathcal{S}$ expressed in $\mathbb{R}^3$, and thence defines an induced metric tensor on $\mathcal{S}$ as $M = (\nabla S)^t \nabla S$. Consequently, quantities on the surface $\mathcal{S}$ (that we will denote for clarity by an overline $\bar{\cdot}$) can be expressed in the plane as a function of u and v. For instance, the mass density $\bar{\rho}$ at a surface point $S(u,v)$ defines a density function ρ in the plane as:

$$\rho(u,v) = \bar{\rho}(S(u,v))\sqrt{\det M}, \tag{7.2}$$

where $\sqrt{\det M}$ accounts for the area change between $\mathcal{S}$ and $\mathcal{U}$. Similarly, a second-order tensor $\bar{\sigma}$ that acts on tangent vectors of the surface expressed in $\mathbb{R}^3$ defines a tensor σ in the plane through:

$$\sigma(u,v) = \nabla S(u,v)^t \, \bar{\sigma}(S(u,v)) \, \nabla S(u,v). \tag{7.3}$$

This setup will allow us to reformulate the 3D equilibrium conditions of masonry surface in 2D. We also refer to [ABF12] for a rigorous derivation of this setup obtained by considering 3D equations in the asymptotic limit of a singular statically admissible stress field concentrated on a surface lying inside the masonry structure.

Discrete Setup. For computational purposes, we will use a planar triangulation T discretizing the domain $\mathcal{U}$, with V vertices, E edges, and F triangles. The 2D positions of vertices are denoted as $\mathbf{u}_i = [u_i, v_i]$, and their associated heights are $z_i = z(u_i, v_i)$, while l_{ij} indicates the length of edge ij and a_{ijk} indicates the area of triangle ijk in 2D. As in Section 5.1, we call $\mathcal{N}_i$ the set of vertices that share an edge with i, and we use ψ_i to denote the piecewise linear basis function over T for vertex i with $\psi_i(\mathbf{u}_j) = \delta_{ij}$. Thus, the height field $z(\mathbf{u}) = \sum_{i \in V} z_i \, \psi_i(\mathbf{u})$ represents a simplicial, piecewise-linear masonry structure. Finally, triangles are oriented counterclockwise, edges are given an arbitrary but fixed orientation, and we then construct the discrete exterior derivative operators $\mathbf{d}_0$ and $\mathbf{d}_1$.

7.2.2 Continuum mechanics description of equilibrium

A self-supporting surface $\mathcal{S}$ is in equilibrium if the compressive forces at play compensate for the dead load created by the surface mass density $\bar{\rho}$ in a gravitational field $\mathbf{g} = [0, 0, -g]$. This amounts to a surface stress tensor $\bar{\sigma}$ defined by a symmetric, negative semi-definite matrix that satisfies:

$$\nabla \cdot \bar{\sigma} = [0, 0, \bar{\rho} g]. \tag{7.4}$$

One can conveniently rewrite this equilibrium condition as two separate equations using the 2D scalar field ρ and tensor σ defined via Equations (7.2) and (7.3):

$$\boxed{\nabla \cdot \sigma = 0 \quad \text{and} \quad \nabla \cdot (\sigma \nabla z) = \rho g.} \tag{7.5}$$

The first equation indicates that the stress tensor σ on $\mathcal{U}$ is *divergence-free* (i.e., the divergence of each of the columns of σ is zero); the second relation equates an elliptic operator on the height to the local dead load in order to enforce balance. This last equation can be further reduced to $\Delta^{\sigma} z = \rho g$, where Δ^{σ} is the Laplacian operator in the metric induced by the stress σ. Moreover, two-dimensional stress tensors over *simply connected* domains can be encoded simply as the Hessian of a real-valued function ζ negated and rotated on both sides by $\pi/2$ [GZ02]. Called the *Airy stress function*, this function must be convex to enforce the compressible nature of stress tensor (Figure 7.1).

7.2.3 Existing discrete approximations of equilibrium

The aforementioned differential description of equilibrium can naturally be leveraged to develop numerical schemes for the generation of self-supporting masonry structures. Next we summarize previous approaches before introducing our rationale for simplicial masonry.

Finite/Discrete Elements. Discretized balance equations can be derived through finite element analysis for a finite-dimensional approximation of the surface—typically, a triangle or quad mesh. However, traditional displacement-based conforming and low-order finite element methods applied to this problem can lead to indeterminacy or divergence under refinement of the solution [GG85]. Stress-based discrete element methods were introduced to remediate these issues, but they often required mesh elements to be aligned with the principal directions of stress (see, for instance, [O'D99]). This unsavory constraint was further removed through the use of non-conforming finite elements, where stress is expressed via a discretized Airy stress function [FAF02, Fra10, Fra11, ABF12]. Since the existence of a stress field deriving from an Airy function is only guaranteed for simply-connected domains, this tight and efficient discrete formulation restricts the topology of the surface.

Force Diagrams. Another well-studied approach is to approximate the continuum medium with an equivalent truss structure: the edges of a mesh are seen as the support of a force diagram acting on the nodes, thereby approximating the stress field through uniaxial singular stresses. Such a *thrust network analysis* (TNA) [BO07, Blo09] only requires a negative value σ_{ij} per edge e_{ij}, interpreted as a force density, to effectively encode the stress tensor. The equilibrium equations for each interior vertex $\mathbf{u}_i$ with height z_i are then reduced to:

$$\begin{cases} \sum_{j\in\mathcal{N}_i} \sigma_{ij}(u_j - u_i) = 0 \\ \sum_{j\in\mathcal{N}_i} \sigma_{ij}(v_j - v_i) = 0 \\ \sum_{j\in\mathcal{N}_i} \sigma_{ij}(z_j - z_i) = m_i g \end{cases} \tag{7.6}$$

where $m_i g$ is the local (lumped) gravitational load on vertex i. Despite great practical relevance stemming from its elegant simplicity, there is no rigorous analysis on the accuracy and convergence of TNA. The recent work of [VHWP12] proposed to help architects design self-supporting structures by applying the TNA model on polygonal meshes; they solved for edge values σ_{ij} by iteratively minimizing the residual of the TNA equations in the least-squares sense while best matching a user-given height field. Unfortunately, this least-squares based methodology is not without shortcomings: for mixed boundary conditions on the height, the resulting system of equations is in general overconstrained, and it may thus fail to find a self-supporting solution without resorting to user interaction.

Towards a Hybrid Approach. It is worth noting that the modeling of a continuous membrane through a discrete force network [Blo09] closely matches the non-conforming approximation of the equilibrium problem [Fra11]. In particular, they both point out the singular nature of the stress (which is "lumped" along mesh edges only) and the importance of a *reciprocal* diagram to the planar orthographic projection of the surface mesh. There are, however, a few significant differences. For instance, Block [Blo09] favors primal and dual graphs with corresponding edges intersecting at a fixed angle, but Fraternali [Fra11] uses a barycentric dual graph instead. This difference stems from the physical interpretation of the dual graph: while the latter only needs well-formed dual cells over which the stress can be *averaged* in order to weakly enforce equilibrium, the former associates a reciprocal diagram with the existence of edge-aligned forces that cancel out the gravitational loading on each vertex. Our approach, described next and based on numerical homogenization, can be seen as reconciling these two methods for simplicial masonry structures, as we will inherit the conceptual simplicity of TNA while still leveraging finite elements literature for rigorous convergence analysis and bounds on accuracy. In particular, we will introduce a discrete formulation for which both conditions in Equation (7.5) will be precisely enforced.

7.2.4 Upscaling of Self-Supporting Surfaces

We now introduce our finite-dimensional characterization of equilibrium for masonry structures. Derived from an edge-based discretization of symmetric tensors and leveraging a geometric interpretation of the diagram of forces at play, it will serve as the foundation of our computational approach to simplicial self-supporting form finding.

Masonry equilibrium as upscaled elliptic problem. In Section 7.2.2, we formulated the condition of equilibrium of masonry structures as an elliptic problem in the space of divergence-free, symmetric, negative, and semi-definite tensors (Equation (7.5)). This exact case (up to sign) appeared in the very different context of electric impedance tomography with rough conductivity coefficients in [DDO13], where a numerical homogenization (or upscaling) of the equations were proposed via a harmonic change of coordinates. In particular, it was shown that a symmetric tensor σ in 2D can be conveniently and rigorously discretized on a simplicial mesh T by a scalar value σ_{ij} for every (unoriented) edge ij using the piecewise linear basis functions ψ_i and ψ_j as:

$$\sigma_{ij} = -\int_{\mathcal{U}} \nabla\psi_i{}^t \sigma \nabla\psi_j \, dudv.$$

Properties of the continuous tensor σ were shown to carry over as simple properties on the coefficients σ_{ij}. For instance, the negative semi-definiteness of the continuous tensor corresponds to non-positive values of σ_{ij}. Moreover, due to the tensor σ being divergence-free, the authors proved that for every vertex i,

$$\sum_{j\in\mathcal{N}_i} \sigma_{ij}(u_j - u_i) = 0 \quad \text{and} \quad \sum_{j\in\mathcal{N}_i} \sigma_{ij}(v_j - v_i) = 0. \tag{7.7}$$

Finally, the elliptic operator in Equation (7.5) was then expressed in weak form on T as $\sum \sigma_{ij}(z_j - z_i)$. Consequently, a direct application of their finite elements methodology to our masonry case recovers the TNA formulation of Equation (7.6).

Connection to DEC. The use of edge values to encode the stress tensor fits well the formalism of discrete exterior calculus (DEC, Section 5.4), which will allow us to use a more compact notation expressing both the continuous and discrete equations at play. Indeed, discretizing a symmetric tensor with a value per *unoriented* edge complements the discrete version of a differential form (i.e., an antisymmetric tensor), which uses a value per *oriented* edge instead. In fact, σ_{ij} can be interpreted, up to a sign, as a discrete Hodge star deriving from the continuous metric σ. Notice that if σ is the negated identity matrix (i.e., the negated Euclidean metric on $\mathcal{U}$), one finds the well-known cotangent-based diagonal Hodge star widely used in geometry processing. (See [ZRS05] for a related expression in the context of quasi-conformal maps with piecewise-constant tensors.) More generally, the diagonal matrix $\star_1^\sigma = \{\star_{ij}^\sigma := -\sigma_{ij}\}_{ij\in E}$ is the discrete Hodge star for one-forms associated with the metric corresponding to the planar stress tensor σ. Note that the compressive nature of the stress implies that the discrete Hodge star values must all be non-negative. Moreover, the term in Equation (7.7) resulting from the divergence-freeness of the stress σ can be concisely rewritten as

$$\mathbf{d}_1^t \star_1^\sigma \mathbf{d}_0 \mathbf{u} = 0. \tag{7.8}$$

One can readily check this corresponds to a statement of *linear precision* for the σ-induced discrete Laplacian $\Delta^\sigma = \mathbf{d}_1^t \star_1^\sigma \mathbf{d}_0$ on $\mathcal{U}$, since this discretization of the elliptic operator $\nabla \cdot (\sigma \nabla)$ applied to the u and v coordinates of the mesh T returns zero.

Equilibrium equations. Finally, the equilibrium equations that must be satisfied by the height z can be given in a weak form, by integrating the differential equation over each planar dual cell of T:

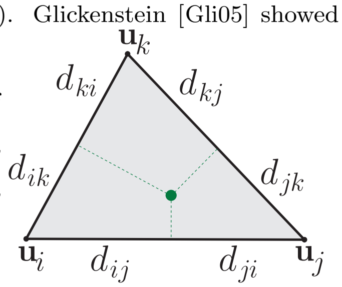
$$\mathbf{d}_1^t \star_1^\sigma \mathbf{d}_0 \, z = \star_0 \rho g, \tag{7.9}$$

where $\star_0 = \{\star_i\}_{i \in V}$ is the conventional discrete Hodge star for zero-forms on T (Equation (5.10)). Note the presence of *two different Hodge stars*: while we discussed the emergence of a stress-induced Hodge star $\star_1^\sigma$ for discrete one-forms that provides a geometric realization of the force diagram, the second Hodge star $\star_0$ is for zero-forms and represents an integration over each (barycentric, circumcentric, or otherwise) partition of the mesh. This latter Hodge star corresponds to the common practice of "mass lumping" in computational mechanics. With Equations (7.8) and (7.9), we have a discrete formulation mirroring the continuous conditions of Equation (7.5).

7.2.5 Stress-induced orthogonal dual diagram

The properties of the scalar values σ_{ij} we listed above also have a simple geometric implication due to the Maxwell-Cremona theorem [Cre90, WMKG07]: it implies that any *divergence-free* discrete stress tensor is associated with an *orthogonal* (reciprocal) Poincaré dual diagram to the simplicial planar mesh for which the supporting lines of primal and dual edges are perpendicular. Consequently, a simplicial surface is self-supporting if and only if there exists a *stress-induced orthogonal dual mesh*, corresponding to the TNA force diagram in [BO07, VHWP12]. Conversely, the requirement of a divergence-free stress tensor removes the need to consider non-orthogonal dual diagrams.

Reduced coordinates of orthogonal dual diagram. An orthogonal dual diagram to a given planar primal mesh can be defined by a set of dual vertex locations with added constraints to satisfy the orthogonality condition between primal and dual edges (Definition 5.1.4). Glickenstein [Gli05] showed that weighted triangulations (Definition 5.2.1) determine any orthogonal dual diagram for simply-connected triangle meshes. We instead introduce a set of reduced coordinates that encode *all possible* orthogonal dual diagrams for a primal mesh of *arbitrary topology*. Recall that orthogonal dual structure in a single triangle exists if and only if three perpendicular bisectors crossing the triangle edges have a common intersecting point. This condition, which must hold on every triangle of a given simplicial mesh, was explicitly written in [Gli05] as:



$$\left(d_{ji}^2 - d_{ij}^2\right) + \left(d_{kj}^2 - d_{jk}^2\right) + \left(d_{ik}^2 - d_{ki}^2\right) = 0, \tag{7.10}$$

where d_{ab} is the signed distance between the vertex $\mathbf{u}_a$ and the intersection of the lines supporting edge ab and its perpendicular bisector (as in Equation (7.13); see inset). We can rewrite this condition in the language of algebraic topology (using DEC) so as to uncover a parametrization of the entire space of orthogonal dual diagrams for a given triangulation. Indeed, we define a primal discrete one-form ω with value $\omega_{ij} = d_{ji}^2 - d_{ij}^2$ per oriented edge ij. Then Equation (7.10) states that an orthogonal dual mesh is fully defined by a *closed* primal one-form ω, i.e., $\mathbf{d}_1\omega = 0$: once such a closed form ω_{ij} is known, d_{ij} can be deduced from the given primal lengths l_{ij} (Equation (7.13)). Furthermore, Hodge decomposition (see, e.g., [DKT07]) can be invoked to rewrite this closed one-form as a function of a primal zero-form w and a (non-integrable) primal harmonic one-form η:

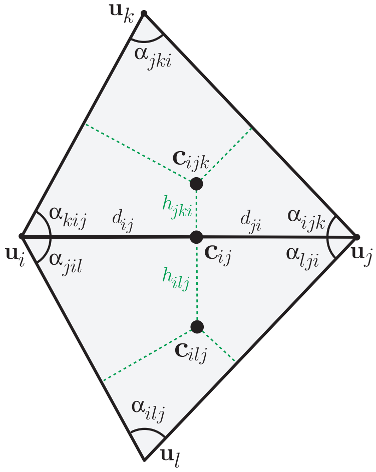
$$\boxed{\omega = \mathbf{d}_0 w + \eta,} \tag{7.11}$$

with normal and tangential boundary conditions for the two terms respectively. Intuitively, the one-form η corresponds to a harmonic vector field which cannot be written as the gradient of a scalar function, hence the term non-integrable. Finally, recall that harmonic one-forms admit a small set of β_1 basis one-forms for an arbitrary bounded planar triangulation, where β_1 is the first Betti number indicating the number of holes in the domain (see a simple computational procedure to compute this basis in, for instance, [TACSD06]). We can thus compactly encode the one-form η at an edge ij as a linear combination of these basis one-forms, i.e.,

$$\eta_{ij} = \sum_{q=1..\beta_1} c_q \gamma^q_{ij}, \tag{7.12}$$

where γ^q is a basis element for harmonic one-forms (with unit circulation around its associated hole) and c_q is a real-valued hole-indexed coefficient. An orthogonal dual mesh can thus be arbitrarily constructed by assigning a value w_i per vertex (zero-form) and a set of β_1 values $\{c_q\}_{q=1..\beta_1}$ defining a harmonic one-form. Note that the use of $\mathbf{d}_0$ makes the zero-form defined only up to a constant. The dimensionality of the full space of orthogonal dual meshes for a given mesh T discretizing a domain $\mathcal{U}$ is therefore $(|V|-1)+\beta_1$, i.e., the number of vertices of T (minus one to remove the additive constant of the zero-form) plus the number of holes β_1 of the domain $\mathcal{U}$. An example of how the dual diagram is changed by hole-indexed coefficients is in Figure 7.2; notice the concentration of stress near the hole, linked to the Saint-Venant principle of structural engineering. Even though we will only consider planar meshes in this work, this result is also valid for manifold triangulations of arbitrary genus.

Closed-form expressions of edge stress values. Another consequence of the Maxwell-Cremona theorem is that the values σ_{ij} (or, equivalently, the discrete Hodge star for one-forms $\star^\sigma_1$) are directly linked to the geometry of the primal-dual structure: each edge value σ_{ij} is (minus) the ratio of dual length to primal length of edge ij—generalizing the definition of the diagonal Hodge star in Section 5.4 to triangulations of arbitrary topology. We can then provide closed-form expression of the edge values corresponding to a divergence-free tensor σ. Similar to Section 5.1, we denote by $\mathbf{c}^\sigma_{ijk}$ the position of the stress-induced dual node to triangle ijk, and by $\mathbf{c}^\sigma_{ij}$ the position of the intersection of the supporting lines of the primal edge ij and of its stress-induced dual (see inset). Additionally, we define h_{jki} as the signed distance between $\mathbf{c}^\sigma_{ijk}$ and $\mathbf{c}^\sigma_{ij}$, with a positive sign if the triangle $(\mathbf{c}^\sigma_{ijk}, \mathbf{u}_i, \mathbf{u}_j)$ has the same orientation as the triangle $(\mathbf{u}_i, \mathbf{u}_j, \mathbf{u}_k)$, and negative otherwise. One finds, for triangle ijk, the following extensions of Equations (5.2) and (5.3):



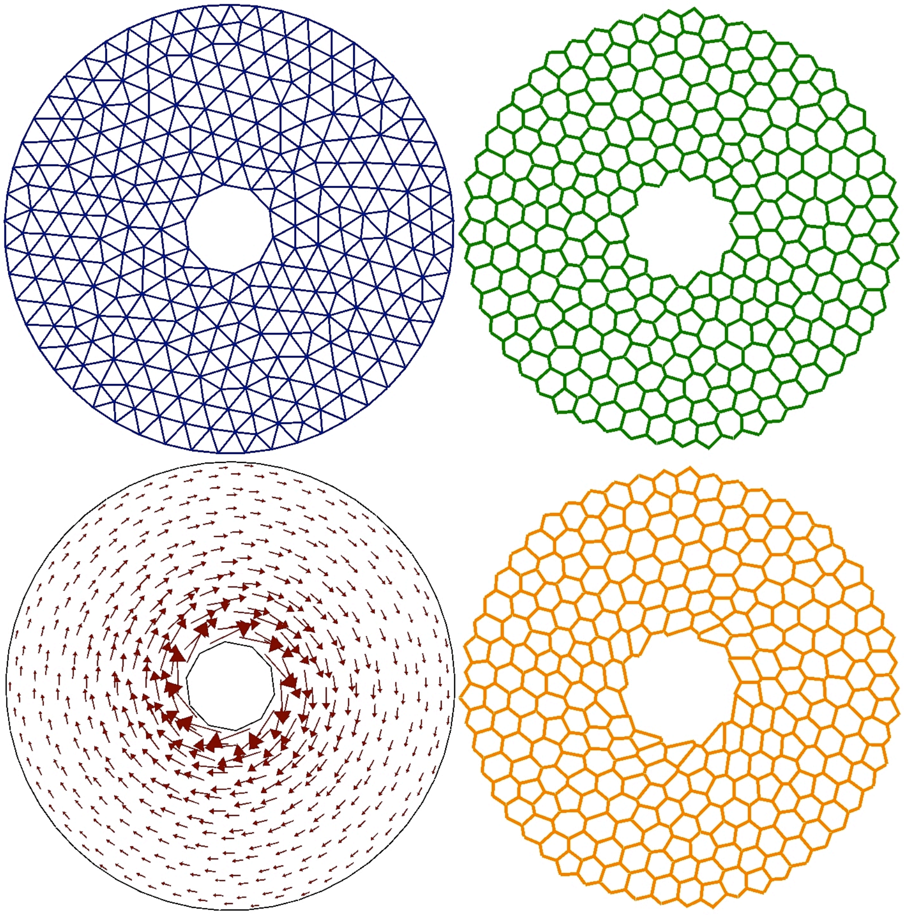


Figure 7.2: **Effect of harmonic one-forms.** If a harmonic one-form η is added to a planar triangulation with non-trivial genus (top left), the stress-induced dual diagram (top right) is displaced (bottom right) by a non-integrable vector field (bottom left), adding a valuable degree of freedom to find self-supporting shapes. Notice that the displacement is larger around the hole, and decreases rapidly due to its harmonic nature.

$$d^{\sigma}_{ij} = \frac{l^2_{ij} + \omega_{ji}}{2\, l_{ij}} \quad \text{and} \quad d^{\sigma}_{ji} = \frac{l^2_{ij} + \omega_{ij}}{2\, l_{ij}}, \tag{7.13}$$

$$h^{\sigma}_{jki} = \frac{l_{ij} \cot \alpha_{jki}}{2} + \frac{\cot \alpha_{kij}}{2\, l_{ij}} \omega_{kj} + \frac{\cot \alpha_{ijk}}{2\, l_{ij}} \omega_{ki}. \tag{7.14}$$

Then, from these functions of ω_{ij}, the stress-induced Hodge star value $\star^{\sigma}_{ij}$ generalizes Equation (5.9) as the ratio of the dual length A^{σ}_{ij} to the primal length of the stress-induced diagram edge ij:

$$\star^{\sigma}_{ij} = \frac{A^{\sigma}_{ij}}{l_{ij}} \quad \text{with } A^{\sigma}_{ij} = h^{\sigma}_{jki} + h^{\sigma}_{ilj}. \tag{7.15}$$

Notice from this expression that the σ-induced Hodge star has only non-negative coefficient (corresponding to pure compression) if and only if every dual edge has a non-negative length, i.e., if the primal mesh T is a *regular triangulation* [PS85].

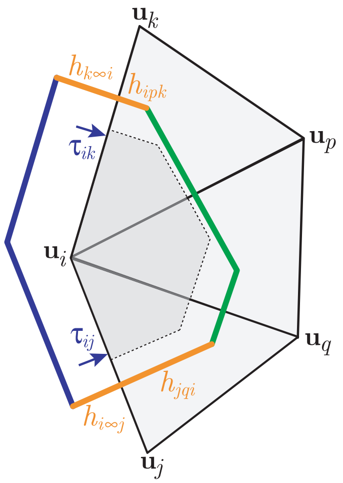
7.2.6 Boundary conditions

Proper handling of the boundaries is crucial to both the numerical treatment (i.e., to avoid overconstrained equations) and the quality of the results (i.e., to guarantee self-support). Our discrete setup leads to simple and flexible boundary conditions.

Stress on boundaries. First recall that since only compressive forces should be at play, the σ_{ij} edge values must all be negative; the boundary σ_{ij} values are no exception, and they define boundary dual edges on the stress-induced dual diagram. Moreover, since boundary vertices may be fixed (via a buttress or other construction artifacts), enforcing a divergence-free stress tensor at the boundary is necessary only for non-fixed vertices. In our setup, the divergence-free condition at the boundary is enforced in weak form by integrating the strong form over the planar region $\mathcal{V}_i^\diamond$ formed by the circumcentric dual cell associated to i and clamped at the boundary ∂T; it thus reads:

$$0 = \int_{\mathcal{V}_i^\diamond} \nabla \cdot \sigma = \int_{\partial \mathcal{V}_i^\diamond} \sigma \mathbf{n} = \sum_{j \in \mathcal{N}_i} \star_{ij}^{\sigma} (\mathbf{u}_i - \mathbf{u}_j) + \int_{\mathcal{V}_i^\diamond \cap \partial T} \sigma \mathbf{n}. \tag{7.16}$$

The first term of this equation matches the definition of divergence-freeness for interior points (Equation (7.8)), yet with an important difference: boundary edges have values A_{ij}^{σ} made out of only a partial dual edge (one of the two terms in Equation (7.15) is zero since there is only one triangle adjacent to ij). We thus complete these dual edges by adding boundary dual lengths $A_\infty^\sigma = \{h_{i\infty j}^{\sigma}\}_{ij \in \partial T}$. With these additional boundary variables, the resulting stress-induced Hodge star values $\star_1^\sigma$ (that must be positive to enforce compression) are well defined even at the boundary as dual-primal edge length ratios. Notice that the resulting stress-induced dual boundary edges represent the (rotated) *tangential* boundary forces (just like the internal dual edges represents the rotated edge-aligned forces at play in the domain). The compressive *normal* forces on the boundary are instead reflected in the second term of the equation above: if we define the (negated) normal stress at the boundary edge ij as $\tau_{ij} \geq 0$, the resulting normal force on ij is thus equal to $-\tau_{ij}\mathbf{n}_{ij}$ with $\mathbf{n}_{ij}$ being the length-weighted outward normal of ij. The integration of the normal forces along $\mathcal{V}_i^\diamond \cap \partial T$ with adjacent boundary vertices j and k (see inset) becomes: $-\left(\tau_{ij}\mathbf{n}_{ij} + \tau_{ki}\mathbf{n}_{ki}\right)/2$. Note that the two 1/2 coefficients come from our specific choice of $\mathcal{V}_i^\diamond$ as circumcentric dual cells, which split every primal edge into two equal parts; other cell decompositions would lead to different coefficients. These tangential and normal boundary terms offer a discrete analog to the continuous, general stress boundary condition [GG85], and bring a more flexible and general boundary handling in the design of self-supporting surfaces.



Boundary equilibrium. Consider now the balance equations, which now involve heights z_i at the boundary. There are only two types of boundary conditions for a boundary vertex:

- if a boundary vertex is *anchored*, its height z is fixed (generally to zero if the height is supposed to be on the ground), and no balance equation is needed for this vertex;
- if a boundary vertex is not anchored (i.e., *free*), the balance equation is then activated.

The physical and geometric justifications of these two cases are straightforward. Indeed, the equilibrium equation for the height at a boundary vertex i is written in weak form as before:

$$\int_{\mathcal{V}_i^\diamond} \Delta^\sigma z = \int_{\partial\mathcal{V}_i^\diamond} \mathbf{n}^t \sigma \nabla z = \sum_{j\in\mathcal{N}_i} \star_{ij}^\sigma (z_i - z_j) + \int_{\mathcal{V}_i^\diamond \cap \partial T} \mathbf{n}^t \sigma \nabla z.$$

Besides the $\star_{ij}^\sigma(z_i - z_j)$ terms which are also present for interior points, an additional integral of the boundary stress appears relating ∇z to the boundary values A_∞^σ and τ. As in the divergence-freeness case, the values A_∞^σ are incorporated in the dual lengths, while the normal component along the boundary segment $ij \cap \mathcal{V}_i^\diamond$ adjacent to the triangle ijk is discretized following [FSDH07] as:

$$\frac{1}{2}\tau_{ij}\Big(\cot\alpha_{ijk}(z_i - z_k) + \cot\alpha_{kij}(z_j - z_k)\Big).$$

Engineering considerations. Note that engineering design of masonry-like edifices requires full control of the stresses at play in the structure. So while we proposed to simply skip the divergence-freeness and equilibrium equations for anchored boundary vertices, one may also adopt a more thorough description of the stress field by adding these equations back, and including the respective boundary compressions as variables for each vertex. Then one can control these values to either make sure they do not exceed a critical threshold that the anchor could not bear, or simply to make sure that these forces are also sufficiently compressive as a margin of safety. Similarly, hole-indexed coefficients c and free boundary stresses τ can be either fixed based on engineering constraints such as prestressing, or simply optimized along with the other variables. While we do not explore all these specific engineering requirements, our formulation accommodates them naturally.

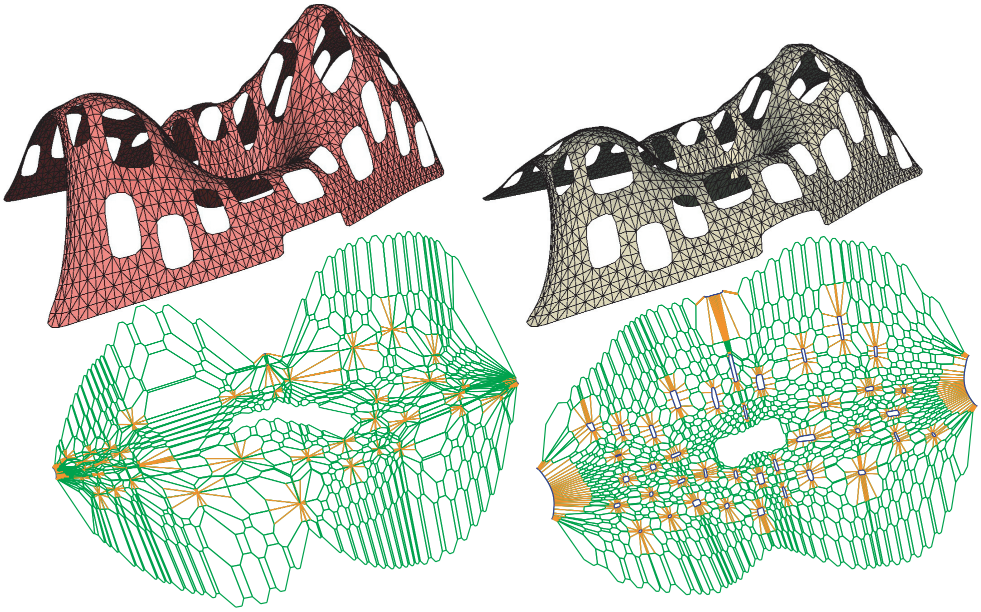


Figure 7.3: **Boundary conditions.** The normal stress imposed around holes can significantly affect the shape of a masonry structure; here, a shape from [VHWP12] with several holes, where the boundary normal stresses are either set to zero (left; notice that the boundary (orange) dual edges meet at a point) or not (right; the normal forces are in dark blue).

Special cases. Finally, we point out that our approach reproduces as special cases the boundary treatments presented in previous work. Employing Dirichlet or Neumann boundary conditions on the Airy function as in [Fra10, ABF12] (enforcing both is known to be overconstrained) corresponds to enforcing the same conditions on the zero-form values w_i at the boundary. Instead, free boundary dual edges intersecting in a single point as in [Blo09, VHWP12] corresponds to setting normal stresses τ to zero (Figures 7.3 and 7.4). Note that the latter boundary condition pushes the extra boundary term in the equilibrium equation to zero as well. While these choices are valid, they only correspond to a subset of all possible boundary conditions.

7.2.7 Discussion

Our discretization provides a formal backdrop to the equations used in thrust network analysis [BO07], and inherits the convergence and accuracy analysis tools available in the finite elements literature [DDO13]. Moreover, we managed to completely characterize the space of valid discrete equilibrium solutions, which will allow us to solve for self-supporting structures using the reduced set of variables w and c that fully describe the set of valid discrete divergence-free symmetric tensors σ. Many remarks are in order, as our results relate to previous work not only in masonry design, but also in computational geometry.

Weighted versus regular triangulations. We parameterized the set of all orthogonal duals of a simplicial mesh with a primal zero-form and a harmonic one-form. Note that if the mesh T is simply connected, there exists no harmonic one-forms, so only vertex values are needed to span the space of orthogonal dual diagrams for a simply connected mesh. This special topology case corresponds to weighted triangulations (Section 5.2), with the zero-form w being referred to as *vertex weights*. In particular, observe that the displacement gradient field ∇w used in Equation (5.1) is equivalent to an interpolation of the exact 1-form $\mathbf{d}_0 w$ through Whitney functions [Whi57]. However, the case of an arbitrary domain topology was not characterized before, probably due to the fact that meshes in computational geometry are often studied as projections of higher-dimensional convex polytopes—which forbids the presence of holes. It should be noticed that, as a consequence, weighted Delaunay triangulations and regular triangulations are often assumed to be equivalent, but this statement ignores the β_1 additional dimensions available for the latter due to topology (Figure 7.2).

Airy function. Although our formulation drastically differs from Fraternali's work, a formal connection to the Airy stress function is easily made, once again for the special case of simply connected domains. The Airy function comes from the integrability of the stress tensor seen as the (rotated and negated) Hessian of a scalar function [GZ02]. In the discrete setting, this function corresponds to the integrable part of the orthogonal dual mesh associated to the discrete stress values σ_{ij} and it thus is defined through the zero-form w via $\zeta(\mathbf{u}) = \frac{1}{2}\sum_i \left(\|\mathbf{u}_i\|^2 - w_i\right)\psi_i(\mathbf{u})$. Geometrically, the Airy function can be seen as the lifting of the 2D mesh T to a paraboloid of height $\psi(\mathbf{u})$ as in Section 2.3; see Figure 7.1. This geometric picture was previously used to construct power diagrams [AHA98] and is related to the convex potential function found in optimal transport [Mér11]. Hence, we have formally established that the discrete Airy values ψ_i in Fraternali's work are related to weights of the resulting weighted Delaunay triangulation through: $\psi_i = \frac{1}{2}\|\mathbf{u}_i\|^2 - \frac{1}{2}w_i$. Besides extending his approach to arbitrary topology (i.e., non-zero β_1), our analysis offers a valuable discrete notion of divergence-freeness that his work did not exploit, and links the convex hull procedure he advocated to the construction of a weighted Delaunay triangulation.

Generalized Airy function. In the broader context of elastostatics, Fosdick and Schuler [FS03] introduced a *continuous* generalization of the Airy stress function to domains with holes by incorporating the symmetric part of the derivative of smooth vector fields. Remarkably, our reduced set of coordinates for orthogonal dual diagrams provides a principled and complete *discretization* of such generalized representation of stress in the case of divergence-free tensors on simplicial meshes of arbitrary topology.

Lumped mass matrix. Finally, we point out that the specific choice of lumped mass matrix (in our case, $\star_0$) varies across the literature. While Fraternali [Fra11] uses the integral of the density over each barycentric dual cell of T, Vouga et al. [VHWP12] integrate over circumcentric dual cells. We also adopt circumcentric dual cells for simplicity, but alternatives are easily incorporated. One may even consider using the space of all possible partitions of the domain as yet another set of degrees of freedom. Besides the changes in local dead load that other choices of dual cells generate, the only other difference that one needs to address is the treatment of boundary conditions: using two sub-variables τ_{ij} and τ_{ji} per boundary edge ij representing the integration of the normal stresses on $\mathcal{V}_i^\diamond \cup ij$ and $\mathcal{V}_j^\diamond \cup ji$ respectively allows for a more general setup with minimal code modification.

7.3 Variational Formulation

Variational formulations for equilibrium equations have a long history in mechanics, and the specific case of masonry structures is no exception [GG85, FAF02, FS03, Fra11]. Our setup involving a primal simplicial mesh and a stress-induced orthogonal dual diagram turns out to also offer a convenient variational formulation, but now at the discrete level, which can then be used for computations.

Equilibrium functional. We introduce an energy $\mathcal{H}$ that is a function of both the zero-form ("weights") $w = \{w_i\}_{i \in V}$ and the harmonic-form coefficients $c = \{c_q\}_{q=1..\beta_1}$ as follows:

$$\mathcal{H}(w,c) = \sum_i z_i \left[\frac{1}{2} \sum_{j \in \mathcal{N}_i} d^\sigma_{ij}\, A^\sigma_{ij} \right] - \sum_i w_i \left(\star_i\, \rho_i\, g \right). \tag{7.17}$$

Based on Equations (7.13) and (7.14), we note that this functional is quadratic in both variables w and c. We also point out that the term in brackets is an analytical expression of the area of the stress-induced dual cell V_i^σ for vertex i (Equation (5.5)), while the second term depends on the vertex load $\star_i\, \rho_i\, g$. To account for the boundary equilibrium conditions discussed in Section 7.2.6, we also incorporate into our energy $\mathcal{H}$ an extra term for every free boundary edge ij:

$$h^\sigma_{i\infty j} \left(d^\sigma_{ij}\, z_i + d^\sigma_{ji}\, z_j \right) + \frac{1}{2} \tau_{ij} \left(w_i + w_j \right) \Big(\cot \alpha_{ijk} (z_i - z_k) + \cot \alpha_{kij} (z_j - z_k) \Big).$$

Derivatives. We first compute derivatives of d^σ_{ij} and h^σ_{jki} with respect to weights w_i and harmonic coefficients c_b:

$$\left\{\begin{array}{l} \nabla_{w_i} d^\sigma_{ij} = \dfrac{1}{2l_{ij}} \\[2ex] \nabla_{w_i} d^\sigma_{ji} = -\dfrac{1}{2l_{ij}} \\[2ex] \nabla_{c_b} d^\sigma_{ij} = \dfrac{\gamma^b_{ji}}{2l_{ij}} \\[2ex] \nabla_{c_b} d^\sigma_{ji} = -\dfrac{\gamma^b_{ji}}{2l_{ij}} \end{array}\right. \qquad \left\{\begin{array}{l} \nabla_{w_i} h^\sigma_{jki} = \dfrac{\cot\alpha_{ijk}}{2l_{ij}} \\[2ex] \nabla_{w_i} h^\sigma_{ijk} = \dfrac{\cot\alpha_{jki}}{2l_{ki}} \\[2ex] \nabla_{w_i} h^\sigma_{kij} = -\dfrac{l_{jk}}{4a_{ijk}}. \\[2ex] \nabla_{c_b} h^\sigma_{jki} = \dfrac{\cot\alpha_{ijk}}{2l_{ij}}\gamma^b_{ki} + \dfrac{\cot\alpha_{kij}}{2l_{ij}}\gamma^b_{kj}. \end{array}\right. \tag{7.18}$$

Based on these terms, one finds that the gradient of $\mathcal{H}$ is of the form:

$$\left\{\begin{array}{l} \nabla_{w_i}\mathcal{H} = \displaystyle\sum_{j\in\mathcal{N}_i} \star^\sigma_{ij}\,(z_i - z_j) - (\star_i\,\rho_i\,g)\,, \\[3ex] \nabla_{c_b}\mathcal{H} = \dfrac{1}{2}\displaystyle\sum_{ij} \star^\sigma_{ij}\,(z_j - z_i)\,\gamma^b_{ij} = \tfrac{1}{2}\langle \mathbf{d}_0 z, \gamma^b\rangle_\sigma. \end{array}\right. \tag{7.19}$$

The second derivatives are also easily computed as:

$$\left\{\begin{array}{l} \nabla_{w_i,w_j}\mathcal{H} = \dfrac{\cot\alpha_{kij}}{2\,l^2_{ij}}\,(z_k - z_j) + \dfrac{\cot\alpha_{ijk}}{2\,l^2_{ij}}\,(z_k - z_i) + \dfrac{\cot\alpha_{jil}}{2\,l^2_{ij}}\,(z_l - z_j) + \dfrac{\cot\alpha_{lji}}{2\,l^2_{ij}}\,(z_l - z_i) \\[3ex] \nabla_{w_i,w_i}\mathcal{H} = -\displaystyle\sum_{j\in\mathcal{N}_i} \nabla_{w_i}\nabla_{w_j}\mathcal{H} \\[3ex] \nabla_{w_i,c_b}\mathcal{H} = \dfrac{1}{2}\displaystyle\sum_{j\in\mathcal{N}_i}\left(\dfrac{z_i - z_j}{l_{ij}}\right)\nabla_{c_b} A^\sigma_{ij}. \end{array}\right. \tag{7.20}$$

Extremization conditions. We immediately deduce from Equation (7.19) that a critical point for weights of $\mathcal{H}$ enforces exactly the balance equations on all interior vertices, and similarly for free boundary vertices if we include the additional boundary terms. Therefore, finding a self-supporting structure for a fixed set of heights, boundary anchors, and harmonic-form coefficients c amounts to extremizing $\mathcal{H}$ subject to the linear inequality constraints $\star^\sigma$, $\tau \geq 0$ that enforce pure compression throughout the structure, plus the divergence-free condition at the free boundary vertices.

7.4 Computational Form Finding Algorithms

With a clear understanding of the geometry of the problem and of the variables at play, we can now provide an approach to the generation of self-supporting simplicial structures. We assume that an initial pointset $(\mathbf{u}_i, z_i)$ is given, as well as a connectivity that defines the planar mesh T with a non-overlapping orthographic projection, and with tags on boundary vertices indicating whether an anchored or free boundary condition is desired. In case the surface can not stand by itself with the given inputs, we need to (minimally) alter its shape to make it self-supporting. We give pseudocode of our overall form finding procedure in Algorithm 3, and go through the numerical details of each step. We then discuss possible variants and design tools derived from this basic approach.

Algorithm 3 Form-finding solver

```
Input: initial mesh $T$ + associated heights $z_i$ + boundary conditions
repeat
    // Find best force diagram
    Perform Stress Optimization (Section 7.4.1)
    Perform Hole-induced Stress Optimization (Section 7.4.2)
    // Alter heights (and optionally, move vertices)
    Perform Shape Optimization (Section 7.4.3)
until Equilibrium criterion met
```

7.4.1 Stress optimization

We designed in Section 7.3 a variational principle whose gradient with respect to weights measures how self-supporting a structure is: if we are at a critical point (gradient equals zero), the structure is at equilibrium. Unfortunately, this energy is not necessarily convex depending on the input mesh (Equation (7.20) may be indefinite); furthermore, there may not even be weights that enforce equilibrium for a given arbitrary shape while satisfying the inequality constraints. However, we previously noted that this energy is quadratic in the weights. So from our variational principle, we can find vertex weights and boundary values that best enforce equilibrium of non-anchored vertices in the L_2 sense by solving for:

$$\operatorname{argmin}_{w,A^{\sigma}_{\infty},\tau} \|\nabla_w \mathcal{H}\|_2^2 \text{ such that } \star_1^{\sigma}, \tau \geq 0, \, (\nabla \cdot \sigma = 0)_{\text{boundary}} . \tag{7.21}$$

Note that the divergence-free constraint is necessary only for free boundary vertices due to the extra boundary terms described in Section 7.2.6. This quadratic energy with linear constraints will reach zero if and only if there exists a set of weights and boundary values defining a valid stress-induced orthogonal dual for which the heights z_i satisfies the equilibrium equations. In the likely case there are no such weights (as the user may start from a very bad configuration), we find the weights which make the structure as close to equilibrium as possible.

Our approach is similar in spirit to [VHWP12], but we now compute the best stress configuration using a smaller set of equations. In fact, working in reduced coordinates enforces the divergence-free equations (Equation (7.8)) exactly for all interior vertices—thus we only have to solve for the remaining equilibrium equations (Equation(7.9)), corresponding to two thirds fewer equations. As the size of the system is considerably reduced, solving this optimization to convergence is much more efficient: even on relatively small meshes, we have a 5x speed up. Our method also resembles the work of [BL11]; however, while the latter computes a space of reduced coordinates through Gauss elimination of the TNA equations (Equation (7.6)), which has cubic complexity in the mesh size, our approach provides the reduced coordinates in closed form (Equations (7.13) and (7.14)). More importantly, our boundary treatment in Section 7.2.6 is more general and turns previous approaches such as [BL11, VHWP12] from potentially overconstrained to systematically underconstrained system.

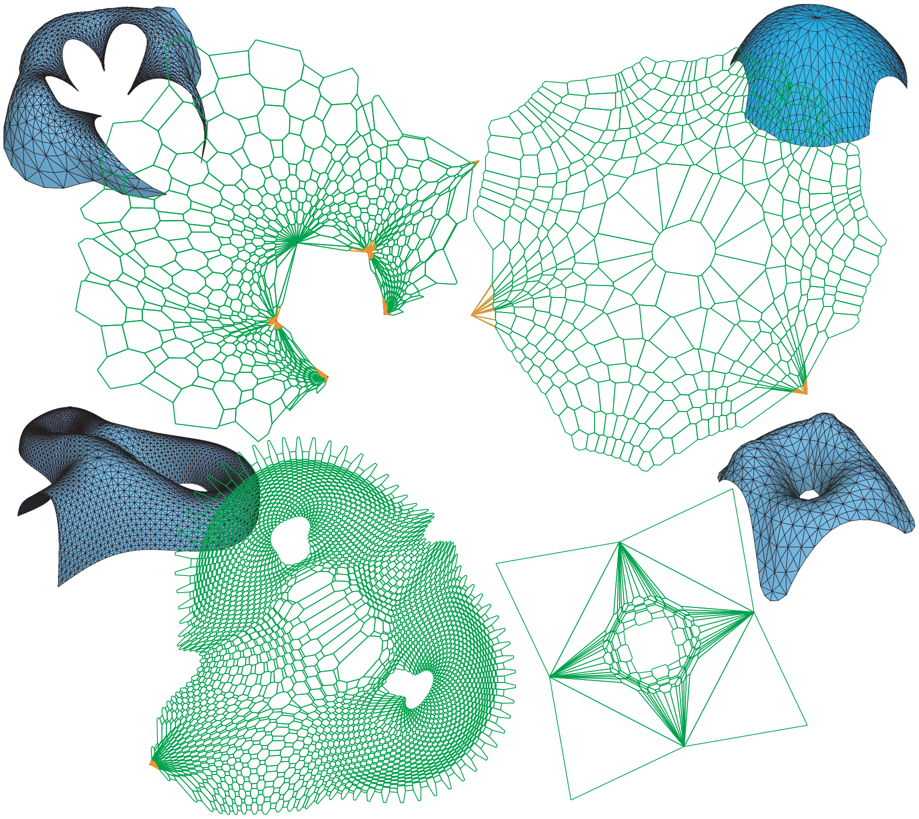


Figure 7.4: **Stress-induced Dual.** Self-supporting meshes and their stress-induced dual diagrams computed via stress and hole-induced stress optimization (Section 7.4.1 and 7.4.2) for a series of meshes of arbitrary topology from [VHWP12], with boundary normal stresses set to zero ($\tau = 0$).

7.4.2 Hole-induced stress optimization

Once the vertex weights and boundary values have been optimized, we can then further improve balance (if it is not already enforced) by optimizing for the harmonic-form coefficients c. This is only performed if the domain has at least one hole ($\beta_1 > 0$), and we proceed as in the previous case by performing a solve for:

$$\text{argmin}_{c, A^\sigma_\infty, \tau} \|\nabla_w \mathcal{H}\|_2^2 \;\text{ such that } \star_1^\sigma, \tau \geq 0,\, (\nabla \cdot \sigma = 0)_{\text{boundary}} \,. \tag{7.22}$$

Because the above energy is also quadratic in the coefficients c, this optimization is particularly simple: we have a quadratic form of only β_1 coefficients to minimize under constraints. This allows us to further adjust the stress-induced dual diagram in order to improve balance, a step ignored by all previous approaches. See a few results of stress and hole-induced stress optimization on various input meshes in Figures 7.3, 7.9 and 7.4.

7.4.3 Shape optimization

Assuming that optimizing weights, boundary values, and harmonic components have not yielded equilibrium, we must modify the assigned height values z_i to find a self-supporting surface. This shape optimization step can take on various forms. Vouga et al. [VHWP12] advocated a change of heights $\{z_i\}$ in concert with a change of positions $\{\mathbf{u}_i\}$ to reach a L_2 minimum of the equilibrium conditions. A change in node positions was beneficial in their case because it helped enforce the divergence-free condition which, unlike in our approach, was *not* satisfied by default. This particular approach has the inconvenience of not separating shape control and mesh quality, which can change the initial shape quite significantly in the process, as demonstrated in Figure 7.5. Methods enforcing divergence-freeness through the Airy stress function (for simply connected domains) also proposed shape optimization of various forms. For instance, Fraternali [Fra11] was removing equilibrium-violating vertices altogether until an equilibrium was reached, but this form finding approach may destroy the mesh quality. Instead, Angelillo et al. [ABF12] proposed to freeze the parts of the surface already satisfying equilibrium and solve for the optimal remaining heights. However, this binary update reduces the smoothness of the results, and ends up taking more time to converge.

Here again, we leverage our variational approach to provide a simple and robust numerical approach for smoothly modifying the shape to become self-supported while staying as faithful as possible to the original input shape. To achieve this effect, we use a minimization with soft constraints: we minimize with respect to each z_i the norm of the residual of the balance equations, where each vertex i is weighted inversely proportional to its local balance residual. In other words, we optimize the heights by giving more leeway to vertices that are significantly violating the equilibrium condition, while vertices already near equilibrium will have their heights almost unchanged. This weighted-L_2 residual minimization has the advantage of only affecting the shape in parts that are significantly not self-supporting: Figure 7.5 shows that while concave parts of the original shape are necessarily altered to reach self-support, the rest of the surface is mostly unaffected. Other weighting strategies can of course be designed based on user preference.

As part of the shape optimization procedure, we also introduce an optional step of mesh smoothing: based on the current vertex positions $\mathbf{u}_i$ in the plane, we compute an update of the Optimal Delaunay Triangulation optimization introduced in [ACSYD05]: we move the coordinates of the mesh T such that the mesh elements *on the surface* are more equilateral. Note that this may actually reduce the quality of the shape of triangles in the plane, but the simplicial masonry structure will be better geometrically discretized. This optional step serves several purposes: first, it guarantees that our algorithm does not create degenerate elements; it also helps getting smoother resulting shape since the sampling quality is improved; finally, it favors the creation of concavities on free boundaries.

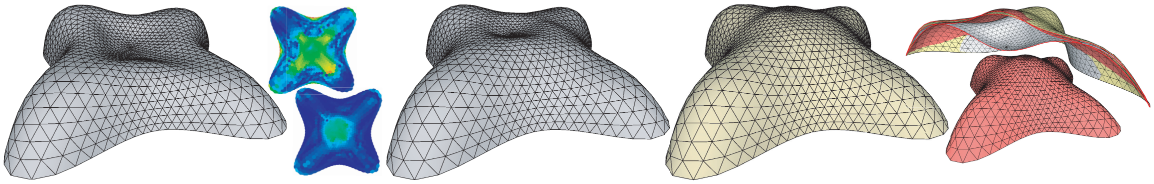


Figure 7.5: **Height optimization.** An input shape from [VHWP12] is optimized, improving the residual of the equilibrium equations along the way (colormapped insets). Our form finding procedure removes the aphysical, concave center part of the shape. Compared to Vouga et al.'s result (red), we find a self-supporting configuration with a normalized Hausdorff and a L_2 distance to the initial shape of 0.04975 and 0.00758 respectively, while theirs are 0.04996 and 0.00831: their result suffers "sagging" of the initial shape near the anchored boundary, with less steep walls (see cut for comparisons).

7.4.4 Variants

Many possible variants of our algorithm can be implemented. In particular, margin of errors are important in engineering to allow for small construction errors. One can modify the solvers to enforce a number of physical properties. For instance, the constraint that each σ_{ij} be zero or negative can be changed to be bounded away from zero (to enforce non-negligible, but material-adequate compression), which we accommodate with ease by changing the constraints in our optimization. Similarly, the boundary forces can be assigned or optimized based on engineering needs. One could enforce constraints on the force diagram as well, by either bounding the maximum dual edge lengths or penalizing wild variations in dual lengths. We could also prescribe the normal stress for free boundary vertices, to deal with cases where a hole is attached to, say, a pole, in order to suspend the whole masonry structure. By setting the normal stress terms to zero, we can further enforce the boundary dual edges to intersect in a single point, as done in [Blo09, VHWP12] (Figures 7.3 and 7.4). A user-defined varying mass density can also be defined if the masonry structure is supposed to withstand an extra load (Figure 7.6). We point out that we assumed a given connectivity in our approach. However, we may also alter the connectivity either based on user-guidance, or during the form finding procedure to optimize for, e.g., maximum stress [LHS$^+$13]. In all cases, our knowledge of the reduced coordinates for which a structure is self-supporting leads to more compact equations and faster solves.

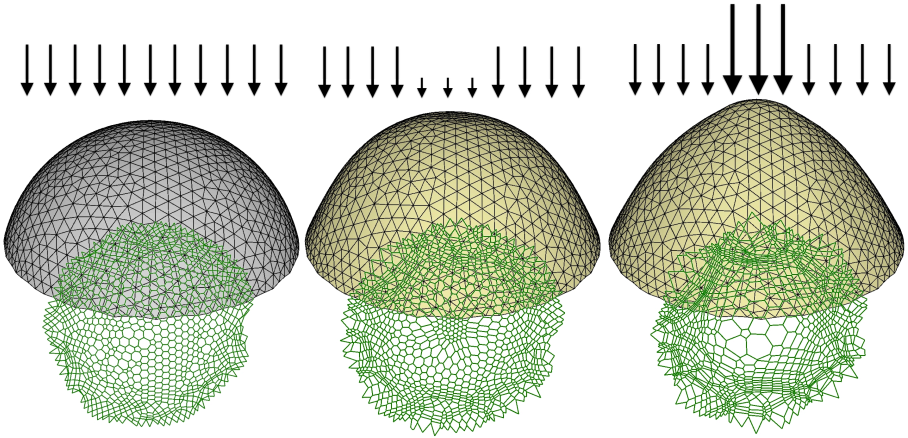


Figure 7.6: **Load bearing.** By changing the load in a small region at the top of a half-sphere shape, various dome shapes can be generated. In order: constant mass, lighter (×0.1), and heavier (×5).

7.4.5 Timing and accuracy

We used CGAL [CGA14] as our mesh library, and IPOPT [WB06] as our numerical optimization library. Equilibrium of all our examples was enforced by ensuring that the L_∞ norm of the gradient of the residual of non-anchored vertices is below 1e-6, which took between 5 and 20 iterations of our form-finding iterative solver depending on the input model size. Typical timings for the various form finding tests we made were at most two minutes. We also compared in Figure 7.7 our stress optimization timing to Vouga et al. [VHWP12] and found a systematic improvement varying from a factor of 2 to 7 for meshes of various sizes.

Model	V ; E ; F	[VHWP12]	Ours
Cheese-model (Figure 7.3)	2348 ; 6218 ; 3832	13.078	9.8
Free vault (Figure 7.4)	360 ; 1000 ; 640	0.387	0.18
Dome with doors (Figure 7.4)	577 ; 1600 ; 1024	0.877	0.368
Video-surface (Figure 7.4)	1131 ; 3194 ; 2064	3.55	0.98
Cas-model (Figure 7.4)	5951 ; 17472 ; 11520	90.1	18.0
Lilium (Figure 7.5)	1201 ; 3504 ; 2304	4.136	0.887
Variable load (Figure 7.6)	1156 ; 3368 ; 2213	3.65	0.77
Shifted barrel vault (Figure 7.9)	310 ; 836 ; 527	0.24	0.15
Moebius igloo (Figure 7.9)	702 ; 2006 ; 1304	1.3	0.4
Dome with hole (Figure 7.9)	1656 ; 4842 ; 3186	7.7	1.5
Groin vault (Figure 7.9)	2943 ; 8569 ; 5627	23.6	4.1

Figure 7.7: **Timing.** Comparisons between dual optimization timings (in seconds) from [VHWP12] and our approach. All results were clocked on an Intel Core i7 2.2 GHz laptop with 4GB RAM.

7.5 Isotropic Constant Mean Curvature Surfaces

Vouga et al. [VHWP12] showed that the ratio of a vertex load ρ_i and its dual cell area V_i^σ induced by the stress σ is equivalent to the concept of relative mean curvature borrowed from the theory of isotropic geometry [PGM09]. Based on this observation and on Equation (5.5), Liu et al. [LHS$^+$13] proposed to generate smooth self-supporting meshes through an iterative process that estimates vertex weights by approximating $V_i^\sigma = \rho_i/H$ for every vertex i, where H indicates a constant relative mean curvature. We point out that computing weights w from a given load function ρ and constant relative mean curvature H is akin to the capacity-constrained problem described in Section 6.4.1. We can indeed set $\overline{V}_i \equiv \rho_i/H$ and solve for the dual metric w associated to a planar primal mesh by minimizing the energy $\mathcal{C}$ (Equation (6.6)) through an efficient Newton's solver [NW99]. We can further enforce positive dual lengths either by adding them as inequality constraints (Section 7.4) or by performing edge flips (as in [LHS$^+$13]). Once the 2D primal-dual mesh is found, we can finally construct a self-supporting height function by solving Equation (7.9). As illustrated in Figure 7.8, our approach offers a novel tool for modeling self-supporting meshes based on the prescribed distribution of vertex loads. Our implementation utilizes the Ipopt library [WB06] (with default parameters) and took three iterations to convergence.

7.6 Conclusions

Our work on simplicial masonry provides a discrete theory of equilibrium for purely compressive structures that can support their own weight. We showed that the well-known continuous equations from the rich mechanical engineering literature find simple, discrete equivalents. We also exploited these properties to formulate a set of reduced coordinates in order to encode equilibrium and boundary conditions. Finally, we leveraged this tight formulation to produce a computational form finding procedure to alter a reference shape into a free standing simplicial masonry structure. As future work, our discretization could be used to predict or simulate the development of cracks over time based on the stress field's principle directions as postulated in [Fra11]. It may also be interesting to apply our setup to the "opposite" case of wrinkled membranes, for which the forces at play are tensile instead of compressive [WP06]. Many of the properties we mention should remain valid as is, with an opposite sign. Generalization to arbitrary structures, including reinforced concrete, may also bring a complementary set of computational techniques to the traditional finite elements tools currently used in engineering firms.

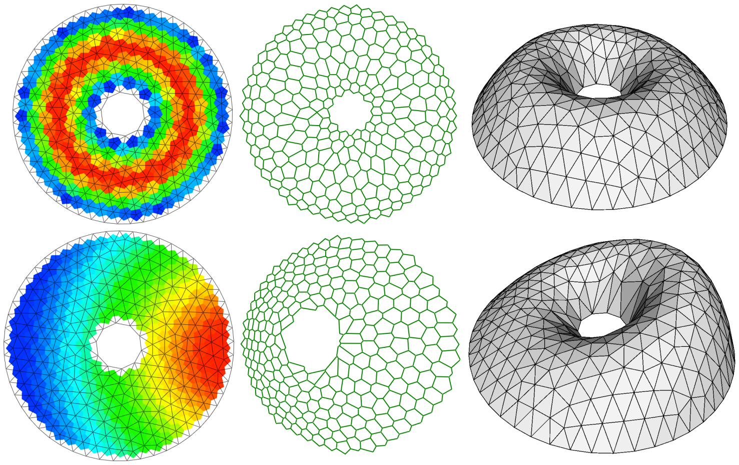


Figure 7.8: **Modeling self-supporting meshes.** Based on the distribution of vertical loads on a planar mesh (left, color ramp indicates load intensity ranging from blue to red) and setting a target relative mean curvature ($H=5$ for both examples), we construct self-supporting meshes by first recovering a dual metric w that minimizes the energy $\mathcal{C}$ in Equation (6.6) (center), and then recovering a height function from Equation (7.9) (right).

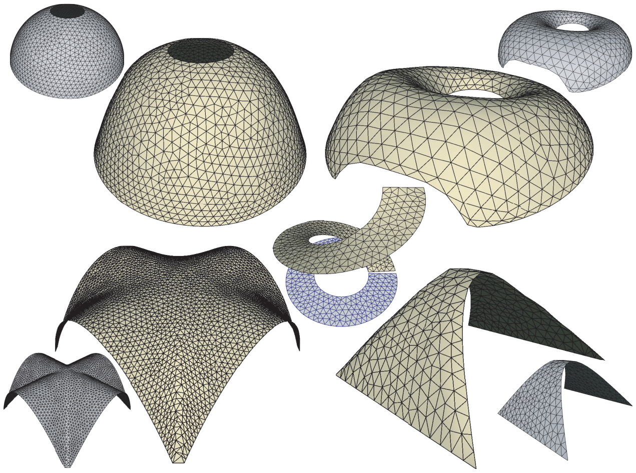


Figure 7.9: **Gallery.** Free-standing shapes can be obtained with our approach. While the Moebius igloo was our design (top right), the dome with a circular oculus (top left, similar to Rome's Pantheon; notice how the opening dilates during optimization to enforce equilibrium), the groin vault (bottom left), the shifted barrel vault (bottom right), and the spiral staircase (center) are all classic masonry structures (insets show the initial meshes).

Chapter 8

Conclusion

In this thesis, we advocated the use of primal-dual meshes, formed by weighted triangulations and power diagrams, in order to offer additional degrees of freedom in the discretization of differential operators and thus retain geometric structures of the smooth setting in the discrete realm. In particular, we developed new discrete differential computational tools on cell complexes and triangulations. The key addition of a scalar value (namely, the weight) per vertex leads to versatile operators, efficiently represented through sparse matrices, that are reliable even on coarse discretizations. We applied our computational framework on a range of applications including image stippling, fluid simulation, surface meshing, and structural analysis.

Our geometric approach for computational modeling offers promising avenues for future work. Most notably, our characterization of divergence-free tensor fields can be further extended to discretize arbitrary tensor fields on triangulations, thus providing a discrete framework for tensor calculus. We are also investigating how to leverage our tensor representation as a means to compute covariant derivatives on triangulations. The generalization of these operators to polygonal meshes is also of interest. Additionally, extensions of the power particle formulation in Chapter 4 are likely to be relevant to a broad family of physical simulation techniques, including incompressible elasticity. Finally, we point out that further investigations in the construction of non-orthogonal primal-dual meshes may lead to generalizations of our framework.

Bibliography

[AAT13] N. Akinci, G. Akinci, and M. Teschner. Versatile surface tension and adhesion for SPH fluids. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 32(6), 2013.

[ABF12] M. Angelillo, E. Babilio, and A. Fortunato. Singular stress fields for masonry-like vaults. *Continuum Mechanics and Thermodynamics*, pages 1–19, 2012.

[ACSYD05] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Transactions on Graphics (SIGGRAPH)*, 24(3), 2005.

[AHA98] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20(1):61–76, 1998.

[AMR88] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, tensor analysis, and applications*, volume 75 of *Applied Mathematical Sciences*. Springer-Verlag, second edition, 1988.

[AO11] I. Alduan and M. A. Otaduy. SPH granular flow with friction and cohesion. In *Symposium on Computer Animation*, pages 219–228, 2011.

[APKG07] B. Adams, M. Pauly, R. Keiser, and L. J. Guibas. Adaptively sampled particle fluids. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3), 2007.

[Arn13] D. N. Arnold. Spaces of finite element differential forms. In U. Gianazza, F. Brezzi, P. Colli Franzone, and G. Gilardi, editors, *Analysis and Numerics of Partial Differential Equations*, pages 117–140. Springer, 2013.

[ATW13] R. Ando, N. Thürey, and C. Wojtan. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.

[Aur87a] F. Aurenhammer. A criterion for the affine equivalence of cell complexes in R^{d} and convex polyhedra in $\mathrm{R}^{\mathrm{d}+1}$. *Discrete & Computational Geometry*, 2:49–64, 1987.

[Aur87b] F. Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal of Computing*, 16(1):78–96, 1987.

[Bal09] M. Balzer. Capacity-constrained Voronoi diagrams in continuous spaces. In *Int. Symposium on Voronoi Diagrams*, pages 79 –88, June 2009.

[BB08] C. Batty and R. Bridson. Accurate viscous free surfaces for buckling, coiling, and rotating liquids. In *Symposium on Computer Animation*, pages 219–228, 2008.

[BB12] L. Boyd and R. Bridson. MultiFLIP for energetic two-phase fluid simulation. *ACM Transactions on Graphics*, 31(2), 2012.

[BBB10] T. Brochu, C. Batty, and R. Bridson. Matching fluid simulation elements to surface geometry and topology. *ACM Transactions on Graphics (SIGGRAPH)*, 29(4), 2010.

[BDL05] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi treemaps for the visualization of software metrics. In *Symposium on Software Visualization*, pages 165–172. ACM, 2005.

[BDWR12] O. Busaryev, T. K. Dey, H. Wang, and Z. Ren. Animating bubble interactions in a liquid foam. *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.

[Ber10] M. Bergou. *Discrete Geometric Dynamics and Artistic Control of Curves and Surfaces.* PhD thesis, Columbia University, 2010.

[BH08] M. Balzer and D. Heck. Capacity-constrained Voronoi diagrams in finite spaces. In *Int. Symposium on Voronoi Diagrams*, pages 44–56, September 2008.

[BL11] P. Block and L. Lachauer. Closest-fit, compression-only solutions for free form shells. In *IABSE/IASS London Symposium, Int. Assoc. Shell Spatial Structures*, 2011.

[Blo09] P. Block. *Thrust Network Analysis: Exploring Three-dimensional Equilibrium.* PhD thesis, Department of Architecture, Massachusetts Institute of Technology, 2009.

[BLS12] K. Bodin, C. Lacoursiere, and M. Servin. Constraint Fluids. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):516–526, 2012.

[BO07] P. Block and J. Ochsendorf. Thrust network analysis: A new methodology for three-dimensional equilibrium. *J. Int. Assoc. Shell and Spatial Structures*, 48(3):167–173, 2007.

[Bos98] A. Bossavit. *Computational Electromagnetism.* Academic Press, Boston, 1998.

[BPS10] A. Bobenko, U. Pinkall, and B. Springborn. Discrete conformal maps and ideal hyperbolic polyhedra, 2010. arXiv:1005.2698.

[BR86] J. U. Brackbill and H. M. Ruppel. FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions. *Journal of Computational Physics*, 65(2):314–343, 1986.

[Bri07] R. Bridson. Fast Poisson disk sampling in arbitrary dimensions. In *ACM SIGGRAPH sketches*, 2007.

[Bri08] R. Bridson. *Fluid Simulation for Computer Graphics.* AK Peters/CRC Press, 2008.

[BS03] A. I. Bobenko and B. A. Springborn. Variational principles for circle patterns and Koebe's theorem. *Transactions of the American Math. Society*, 356:659–689, 2003.

[BSD09] M. Balzer, T. Schlömer, and O. Deussen. Capacity-constrained point distributions: A variant of Lloyd's method. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3), 2009.

[BSV10] A. Buffa, G. Sangalli, and R. Vázquez. Isogeometric analysis in electromagnetics: B-splines approximation. *Computer Methods in Applied Mechanics and Engineering*, 199(1720):1143 – 1152, 2010.

[BT07] M. Becker and M. Teschner. Weakly compressible SPH for free surface flows. In *Symposium on Computer Animation*, pages 209–217, 2007.

[BvdPPH11] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich. Displacement interpolation using Lagrangian mass transport. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 30(6), 2011.

[BWWM10] J. Bowers, R. Wang, L.-Y. Wei, and D. Maletz. Parallel Poisson disk sampling with spectrum analysis on surfaces. *ACM Transactions on Graphics*, 29(6), 2010.

[BXH10] C. Batty, S. Xenos, and B. Houston. Tetrahedral embedded boundary methods for accurate and flexible adaptive fluids. In *Eurographics*, 2010.

[CBP05] S. Clavet, P. Beaudoin, and P. Poulin. Particle-based viscoelastic fluid simulation. In *Symposium on Computer Animation*, pages 219–228, 2005.

[CG11] G. D. Canas and S. J. Gortler. Orphan-free anisotropic voronoi diagrams. *Discrete & Computational Geometry*, 46(3):526–541, 2011.

[CG12] G. D. Canas and S. J. Gortler. Duals of orphan-free anisotropic voronoi diagrams are embedded meshes. In *Symposium on Computational Geometry*, pages 219–228, 2012.

[CGA14] CGAL. Computational Geometry Algorithms Library, 2014. http://www.cgal.org.

[CIPT14] J. Cornelis, M. Ihmsen, A. Peer, and M. Teschner. IISPH-FLIP for incompressible fluids. *Comp. Graph. Forum*, 2014.

[CL03] B. Chow and F. Luo. Combinatorial Ricci flows on surfaces. *J. Differential Geometry*, 63(1):97–129, 2003.

[Col91] Y. Colin de Verdière. Un principe variationnel pour les empilements de cercles. *Inventiones Mathematicae*, 104:655–669, 1991.

[Coo86] R. L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1):51–72, 1986.

[CR99] S. J. Cummins and M. Rudman. An SPH projection method. *Journal of Computational Physics*, 152(2):584–607, 1999.

[Cre90] L. Cremona. *Graphical Statics.* Oxford University Press, 1890.

[Cro77] F. C. Crow. The aliasing problem in computer-generated shaded images. *ACM Communications*, 20(11):799–805, 1977.

[CSHD03] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen. Wang tiles for image and texture generation. In *ACM SIGGRAPH*, 2003.

[CYC$^+$12] Z. Chen, Z. Yuan, Y.-K. Choi, L. Liu, and W. Wang. Variational blue noise sampling. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1784–1796, 2012.

[Dav59] C. Davis. The set of nonlinearity of a convex piecewise-linear function, 1959.

[Dav11] T. A. Davis. Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Transactions on Mathematical Software*, 38(1):8:1–8:22, 2011.

[DDO13] M. Desbrun, R. Donaldson, and H. Owhadi. Modeling across scales: Discrete geometric structures in homogenization and inverse homogenization. In M. Z. Pesenson, editor, *Multiscale* *analysis and nonlinear dynamics: from genes to the brain*, volume 8 of *Reviews of Nonlinear Dynamics and Complexity*. Wiley, 2013. *Extended version of arXiv:0904.2601 [math.AP], 2009.*

[DFG99] *Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms.* SIAM Review*, 41(4):637–676, 1999.*

[DG96] *M. Desbrun and M.-P. Gascuel. Smoothed Particles: A new paradigm for animating highly deformable bodies. In* EG Workshop on Computer Animation and Simulation*, pages 61–76, 1996.*

[dGAOD13] *F. de Goes, P. Alliez, H. Owhadi, and M. Desbrun. On the Equilibrium of Simplicial Masonry Structures.* ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.*

[dGBOD12] *F. de Goes, K. Breeden, V. Ostromoukhov, and M. Desbrun. Blue Noise through Optimal Transport.* ACM Transactions on Graphics (SIGGRAPH Asia)*, 31(6), 2012.*

[dGCSAD11] *F. de Goes, D. Cohen-Steiner, P. Alliez, and M. Desbrun. An Optimal Transport Approach to Robust Reconstruction and Simplification of 2D Shapes.* Computer Graphics Forum (SGP)*, 30(6), 2011.*

[DH06] *D. Dunbar and G. Humphreys. A spatial data structure for fast Poisson-disk sample generation.* ACM Transactions on Graphics*, 25(3):503–508, July 2006.*

[DHOS00] *O. Deussen, S. Hiller, C. Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings.* Computer Graphics Forum*, 19(3):40–51, 2000.*

[Dim12] *N. Dimitrov. Positively weighted Delaunay triangulations and their circle patterns as critical points of the hyperbolic volume functional, 2012. unpublished.*

[DKT07] *M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In A. Bobenko and P. Schröder, editors,* Discrete Differential Geometry*. Springer, 2007.*

[DMA02] *M. Desbrun, M. Meyer, and P. Alliez. Intrinsic Parameterizations of Surface Meshes.* Computer Graphics Forum (Eurographics)*, 21, 2002.*

[DS09] *R. Dyer and S. Schaefer. Circumcentric dual cells with negative area. Technical Report TR2009-06,* School of Computing Science, Simon Fraser University*, 2009.*

[DS11] *T. K. Dey and A. G. Slatton. Localized delaunay refinement for volumes.* Computer Graphics Forum (SGP)*, 30(5):1417–1426, 2011.*

[DW85] *M. A. Z. Dippé and E. H. Wold. Antialiasing through stochastic sampling. In* ACM SIGGRAPH*, 1985.*

[EDP$^+$11] *M. S. Ebeida, A. A. Davidson, A. Patney, P. M. Knupp, S. A. Mitchell, and J. D. Owens. Efficient maximal Poisson-disk sampling.* ACM Transactions on Graphics (SIGGRAPH)*, 30(4), 2011.*

[EMB11] *K. Erleben, M. K. Misztal, and J. A. Bærentzen. Mathematical foundation of the optimization-based fluid animation method. In* Symposium on Computer Animation*, pages 101–110, 2011.*

[ES85] *H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. In* Symposium on Computational Geometry*, pages 251–262, 1985.*

[ESn07] *M. Ellero, M. Serrano, and P. Espa nol. Incompressible Smoothed Particle Hydrodynamics.* Journal of Computational Physics*, 226(2):1731–1752, 2007.*

[ETK$^+$07] *S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun. Stable, circulation-preserving, simplicial fluids.* ACM Transactions on Graphics*, 26, 2007.*

[FAF02] *F. Fraternali, M. Angelillo, and A. Fortunato. A lumped stress method for plane elastic problems and the discrete-continuum approximation.* International Journal of Solids and Structures*, 39:6211–6240, 2002.*

[Fat11] *R. Fattal. Blue-noise point sampling using kernel density model.* ACM Transactions on Graphics (SIGGRAPH)*, 30(3), 2011.*

[Fra10] *F. Fraternali. A thrust network approach to the equilibrium problem of unreinforced masonry vaults via polyhedral stress functions.* Mechanics Research Communications*, 37(2):198–204, 2010.*

[Fra11] *F. Fraternali. A mixed lumped stress–displacement approach to the elastic problem of masonry walls.* Mechanics Research Communications*, 38:176–180, 2011.*

[FS76] *R. W. Floyd and L. Steinberg. An adaptive algorithm for spatial grey scale.* Proc. Soc. Inf. Display*, 17:75–77, 1976.*

[FS03] *R. Fosdick and K. Schuler. Generalized Airy stress functions.* Meccanica*, 38(5):571–578, 2003.*

[FSDH07] *M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe. Design of tangent vector fields. In* ACM SIGGRAPH*, 2007.*

[FSSB07] *M. Fisher, B. Springborn, P. Schröder, and A. I. Bobenko. An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing.* Computing*, 81(2-3):199–213, November 2007.*

[Gaw10] *E. Gawlik. Geometric, variational discretization of continuum theories, 2010.*

[GG85] *M. Giaquinta and E. Giusti. Researches on the equilibrium of masonry structures.* Archive for Rational Mechanics and Analysis*, 88:359–392, 1985.*

[Gli05] *D. Glickenstein. Geometric triangulations & discrete Laplacians on manifolds, 2005. arXiv.org:math/0508188.*

[Gli07] *D. Glickenstein. A monotonicity property for weighted Delaunay triangulations.* Discrete and Computational Geometry*, 38(4):651–664, 2007.*

[GM09] *M. N. Gamito and S. C. Maddock. Accurate multidimensional Poisson-disk sampling.* ACM Transactions on Graphics*, 29(1):8:1–8:19, 2009.*

[GP10] *L. J. Grady and J. R. Polimeni.* Discrete Calculus: Applied Analysis on Graphs*. Springer, 2010.*

[GSSP10] *P. Goswami, P. Schlegel, B. Solenthaler, and R. Pajarola. Interactive SPH simulation and rendering on the GPU. In* Symposium on Computer Animation*, pages 55–64, 2010.*

[Guo09] *R. Guo. Local rigidity of inversive distance circle packing, 2009. arXiv:0903.1401v2.*

[GZ02] *A. Green and W. Zerna.* Theoretical Elasticity*. Dover, 2002.*

[HA07] *X. Y. Hu and N. A. Adams. An incompressible multi-phase SPH method.* Journal of Computational Physics*, 227(1):264–278, 2007.*

[Har10] *D. Harmon.* Robust, Efficient, and Accurate Contact Algorithms*. PhD thesis, Columbia University, 2010.*

[Hey66] *J. Heyman. The stone skeleton.* International Journal of Solids and Structures*, 2(2):249–279, 1966.*

[Hir03] *A. Hirani.* Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003.*

[HKV13] *A. N. Hirani, K. Kalyanaraman, and E. B. VanderZee. Delaunay Hodge star.* Computer-Aided Design*, 45(2):540–544, 2013.*

[HLL+12] *X. He, N. Liu, S. Li, H. Wang, and G. Wang. Local Poisson SPH for viscous incompressible fluids.* Computer Graphics Forum*, 31(6):1948–1958, 2012.*

[HLW10] *E. Hairer, C. Lubich, and G. Wanner.* Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer Series in Computational Mathematics. Springer, 2010.*

[HPW06] *K. Hildebrandt, K. Polthier, and M. Wardetzky. On the convergence of metric and geometric properties of polyhedral surfaces.* Geometria Dedicata*, 123:89–112, 2006.*

[HSS00] *D. Hietel, K. Steiner, and J. Struckmeier. A finite-volume particle method for compressible flows.* Mathematical Models and Methods in Applied Sciences*, 10(9), 2000.*

[IABT11] *M. Ihmsen, N. Akinci, M. Becker, and M. Teschner. A parallel SPH implementation on multi-core CPUs.* Computer Graphics Forum*, 30(1):99–112, 2011.*

[IAGT10] *M. Ihmsen, N. Akinci, M. Gissler, and M. Teschner. Boundary handling and adaptive time-stepping for PCISPH. In* VRIPHYS*, pages 79–88, 2010.*

[ICS+13] *M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner. Implicit Incompressible SPH.* IEEE Trans. on Visualization and Computer Graphics*, 99:1, 2013.*

[IOS+14] *M. Ihmsen, J. Orthmann, B. Solenthaler, A. Kolb, and M. Teschner. Sph fluids in computer graphics.* State-of-the-Art Report*, 2014.*

[JKLG08] *M. Jin, J. Kim, F. Luo, and X. Gu. Discrete surface ricci flow.* IEEE Transactions on Visualization and Computer Graphics*, 14(5):1030–1043, 2008.*

[Jon06] *T. R. Jones. Efficient generation of Poisson-disk sampling patterns.* Journal of Graphics, GPU, & Game Tools*, 11(2):27–36, 2006.*

[KCODL06] *J. Kopf, D. Cohen-Or, O. Deussen, and D. Lischinski. Recursive Wang tiles for real-time blue noise.* ACM Transactions on Graphics*, 25(3), 2006.*

[Kha10] *L. Kharevych.* Geometric interpretation of physical systems for improved elasticity simulations. *PhD thesis, California Institute of Technology, 2010.*

[KSS06] *L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns.* ACM Transactions on Graphics*, 25(2), 2006.*

[LD06] *A. Lagae and P. Dutré. An alternative for wang tiles: Colored edges versus colored corners.* ACM Transactions on Graphics*, 25(4), 2006.*

[LD08] *A. Lagae and P. Dutré. A comparison of methods for generating Poisson disk distributions.* Computer Graphics Forum*, 27(1), 2008.*

[Lem09] *C. Lemieux.* Monte Carlo and Quasi Monte Carlo Sampling. *Springer, 2009.*

[Lew03] *A. Lew.* Variational time integrators in computational solid mechanics. *PhD thesis, California Institute of Technology, 2003.*

[LH10] *H. Lee and S. Han. Solving the shallow water equations using 2D SPH particles for interactive applications.* The Visual Computer*, 26(6-8):865–872, 2010.*

[LHS$^+$13] *Y. Liu, P. Hao, J. Snyder, W. Wang, and B. Guo. Computing self-supporting surfaces by regular triangulation.* ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.*

[LL06] *G. Lecot and B. Lévy. ARDECO: Automatic Region DEtection and COnversion. In* EG Symposium on Rendering*, pages 349–360, 2006.*

[Llo82] *S. Lloyd. Least squares quantization in PCM.* IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.*

[LNW$^+$10] *H. Li, D. Nehab, L.-Y. Wei, . Sander, and C.-W. Fu. Fast capacity constrained Voronoi tessellation. In* Symposium on Interactive 3D Graphics & Games*, pages 13:1–13:4, 2010.*

[LPRM02] *B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation.* ACM SIGGRAPH*, 21(3), 2002.*

[LS03] *F. Labelle and J. R. Shewchuk. Anisotropic voronoi diagrams and guaranteed-quality anisotropic mesh generation. In* Symposium on Computational Geometry*, pages 191–200, 2003.*

[LSLCO05] *Y. Lipman, O. Sorkine, D. Levin, and D. Cohen-Or. Linear rotation-invariant coordinates for meshes.* ACM Transactions on Graphics (SIGGRAPH)*, 24(3), 2005.*

[LSSF06] *F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. Multiple interacting liquids.* ACM Transactions on Graphics (SIGGRAPH)*, 25(3), 2006.*

[LTKF08] *F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw. Two-way coupled SPH and particle level set fluid simulation.* IEEE Transactions on Visualization and Computer Graphics*, 14(4):797–804, 2008.*

[Luc09] *V. Lucarini. Symmetry-break in Voronoi tessellations.* Symmetry*, 1(1):21–54, 2009.*

[Luo10] *F. Luo. Rigidity of polyhedral surfaces, III, 2010. arXiv:1010.3284v1.*

[LWL$^+$09] *Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, and C. Yang. On Centroidal Voronoi Tessellation - energy smoothness and fast computation.* ACM Transactions on Graphics*, 28(4), 2009.*

[LWSF10] *H. Li, L.-Y. Wei, P. V. Sander, and C.-W. Fu. Anisotropic blue noise sampling.* ACM Transactions on Graphics (SIGGRAPH Asia)*, 29(6), 2010.*

[Mac49] *R. MacNeal.* The Solution of Partial Differential Equations by means of Electrical Networks*. PhD thesis, Caltech, 1949.*

[MCG03] *M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In* Symposium on Computer Animation*, pages 154–159, 2003.*

[MCP$^+$09] *P. Mullen, K. Crane, D. Pavlov, Y. Tong, and M. Desbrun. Energy-preserving integrators for fluid animation.* ACM Transactions on Graphics (SIGGRAPH)*, 28(3), 2009.*

[MDSB02] *M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In* Proc. VisMath*, pages 35–57, 2002.*

[Mer01] *C. Mercat. Discrete Riemann surfaces and the Ising model.* Communications in Mathematical Physics*, 218(1):177–216, 2001.*

[Mér11] *Quentin Mérigot. A multiscale approach to optimal transport.* Compututer Graphics Forum (SGP)*, 30(5):1583–1592, 2011.*

[Mey04] *M. Meyer.* Discrete differential operators for computer graphics*. PhD thesis, California Institute of Technology, 2004.*

[MF92] *M. McCool and E. Fiume. Hierarchical Poisson disk sampling distributions. In* Proc. Graphics Interface '92*, pages 94–105, May 1992.*

[Mil82] *J. Milnor. Hyperbolic geometry: the first 150 years.* Bulletin of American Mathematical Society*, 6(1):9–24, 1982.*

[Mit87] *D. P. Mitchell. Generating antialiased images at low sampling densities. In* ACM SIGGRAPH*, 1987.*

[MM13] *M. Macklin and M. Müller. Position based fluids.* ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.*

[MMD11] *P. Memari, P. Mullen, and M. Desbrun. Parametrization of generalized primal-dual triangulations. In* Meshing Roundtable*, pages 237–253. 2011.*

[MMdGD11] *P. Mullen, P. Memari, F. de Goes, and M. Desbrun. HOT: Hodge-Optimized Triangulations.* ACM Transactions on Graphics (SIGGRAPH)*, 30(4), 2011.*

[Mon92] *J. J. Monaghan. Smoothed Particle Hydrodynamics.* Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.*

[Mon00] *J. J. Monaghan. SPH without a tensile instability.* Journal of Computational Physics*, 159(2):290 – 311, 2000.*

[Mon05] *J. J. Monaghan. Smoothed Particle Hydrodynamics.* Reports on Progress in Physics*, 68:1703–1759, 2005.*

[MSKG05] *M. Müller, B. Solenthaler, R. Keiser, and M. Gross. Particle-based fluid-fluid interaction. In* Symposium on Computer Animation*, pages 237–244, 2005.*

[Mul11] *P. G. Mullen.* Eulerian geometric discretizations of manifolds and dynamics*. PhD thesis, California Institute of Technology, 2011.*

[Mun84] *J. R. Munkres.* Elements of Algebraic Topology*. Addison-Wesley, 1984.*

[Nie92] *H. Niederreiter.* Random Number Generation and Quasi-Monte-Carlo Methods*. SIAM, 1992.*

[NW99] *J. Nocedal and S. J. Wright.* Numerical optimization*. Springer Verlag, 1999.*

[O'D99] *D. O'Dwyer. Funicular analysis of masonry vaults.* Computers & Structures*, 73(1–5):187–197, 1999.*

[ODJ04] *V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties.* ACM Transactions on Graphics (SIGGRAPH)*, 23(3), 2004.*

[Ost07] *V. Ostromoukhov. Sampling with polyominoes.* ACM Transactions on Graphics (SIGGRAPH)*, 26(3), 2007.*

[Pav09] *D. Pavlov.* Structure-preserving discretization of incompressible fluids*. PhD thesis, California Institute of Technology, 2009.*

[PBSH13] *D. Panozzo, P. Block, and O. Sorkine-Hornung. Designing unreinforced masonry models.* ACM Transactions on Graphics (SIGGRAPH)*, 32(4), 2013.*

[Ped88] *D. Pedoe.* Geometry, a comprehensive course*. Dover Publications, 2nd edition, 1988.*

[PGM09] *H. Pottmann, P. Grohs, and N.J. Mitra. Laguerre minimal surfaces, isotropic geometry and linear elasticity.* Adv. Comp. Math*, pages 391–419, 2009.*

[PP93] *U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates.* Experimental Mathematics*, 2:15–36, 1993.*

[PP03] *K. Polthier and E. Preuss. Identifying vector field singularities using a discrete Hodge decomposition. In* Visualization and Mathematics III*, pages 113–134, 2003.*

[PS85] *F. P. Preparata and M. I. Shamos.* Computational Geometry: An Introduction*. Springer-Verlag, 1985.*

[PTB$^+$03] *S. Premože, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker. Particle-based simulation of fluids.* Computer Graphics Forum*, 22(3):401–410, 2003.*

[Reg61] *T. Regge. General relativity without coordinates.* Nuovo Cim.*, 19(3):558–571, 1961.*

[Riv94] *I. Rivin. Euclidean structures on simplicial surfaces and hyperbolic volume.* Annals of Mathematics*, 139(3), 1994.*

[RWT11] *K. Raveendran, C. Wojtan, and G. Turk. Hybrid Smoothed Particle Hydrodynamics. In* Symposium on Computer Animation*, pages 33–42, 2011.*

[SB12] H. Schechter and R. Bridson. Ghost SPH for animating water. ACM Transactions on Graphics (SIGGRAPH), 31(4), 2012.

[SBC$^+$11] B. Solenthaler, P. Bucher, N. Chentanez, M. Mller, and M. Gross. SPH-based shallow water simulation. In VRIPHYS, pages 39–46, 2011.

[SBH09] F. Sin, A. W. Bargteil, and J. K. Hodgins. A point-based method for animating incompressible flow. In Symposium on Computer Animation, pages 247–255, 2009.

[SD11] Thomas Schlömer and Oliver Deussen. Accurate spectral analysis of two-dimensional point sets. Journal of Graphics, GPU, and Game Tools, 15(3):152–160, 2011.

[Sec02] A. Secord. Weighted Voronoi stippling. In Symposium on Non-Photorealistic Animation and Rendering, pages 37–43, 2002.

[SEZ05] M. Serrano, P. Espanol, and I. Zuniga. Voronoi fluid particle model for Euler equations. Journal of Statistical Physics, 121:133–147, 2005.

[SF95] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In ACM SIGGRAPH, 1995.

[SGBW10] C. Schmaltz, P. Gwosdek, A. Bruhn, and J. Weickert. Electrostatic halftoning. Computer Graphics Forum, 29(8):2313–2327, 2010.

[SHD11] T. Schlömer, D. Heck, and O. Deussen. Farthest-point optimized point sets with maximized minimum distance. In Symposium on High Performance Graphics, pages 135–142, 2011.

[She02] J. R. Shewchuk. What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. In Proc. of the 11th Int. Meshing Roundtable, pages 115–126, 2002.

[SHWP09] A. Schiftner, M. Höbinger, J. Wallner, and H. Pottmann. Packing circles and spheres on surfaces. ACM Transactions on Graphics (SIGGRAPH), 28(5), 2009.

[SP08] B. Solenthaler and R. Pajarola. Density contrast SPH interfaces. In Symposium on Computer Animation, pages 211–218, 2008.

[SP09] B. Solenthaler and R. Pajarola. Predictive-corrective incompressible SPH. ACM Transactions on Graphics (SIGGRAPH), 28(3), 2009.

[Spr03] B. A. Springborn. Variational principles for circle patterns. PhD thesis, Technische Universitat Berlin, 2003.

[Spr08] B. Springborn. A variational principle for weighted Delaunay triangulations and hyperideal polyhedra. J. Differential Geometry, 78(2), 2008.

[SSP08] B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. ACM Transactions on Graphics (SIGGRAPH), 27(3), 2008.

[Ste03] K. Stephenson. Circle packing: A mathematical tale. Notices American Mathematical Society, 50(11):1376–1388, 2003.

[Ste09] A. Stern. Geometric discretization of Lagrangian mechanics and field theories. PhD thesis, California Institute of Technology, 2009.

[SZSY11] *M. S. Shadloo, A. Zainali, S. H. Sadek, and M. Yildiz. Improved incompressible Smoothed Particle Hydrodynamics method for simulating flow around bluff bodies.* Computer Methods in Applied Mechanics and Engineering*, 200(9-12), 2011.*

[TACSD06] *Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. In* Symposium on Geometry Processing*, pages 201–210, 2006.*

[Thu76] *W. Thurston. Geometry and topology of 3-manifolds, 1976.*

[TLHD03] *Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition.* ACM Transactions on Graphics (SIGGRAPH)*, 22(3), 2003.*

[Uli87] *R. Ulichney.* Digital Halftoning*. MIT Press, 1987.*

[VHGR10] *E. VanderZee, A. N. Hirani, D. Guoy, and E. Ramos. Well-centered triangulation.* SIAM Journal on Scientific Computing*, 31(6):4497–4523, 2010.*

[VHWP12] *E. Vouga, M. Höbinger, J. Wallner, and H. Pottmann. Design of self-supporting surfaces.* ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.*

[Vil09] *C. Villani.* Optimal Transport: Old and New*. Fundamental Principles of Mathematical Sciences, 338. Springer-Verlag, 2009.*

[Vor08] *G. Voronoi. Nouvelles applications des paramtres continus à la théorie des formes quadratiques.* Journal für die Reine und Angewandte Mathematik*, 133:97178, 1908.*

[Vou13] *P. E. Vouga.* Discrete Differential Geometry of Thin Materials for Computational Mechanics*. PhD thesis, Columbia University, 2013.*

[War06a] *M. Wardetzky.* Discrete Differential Operators on Polyhedral Surfaces - Convergence and Approximation*. PhD thesis, Freie Universitat Berlin, 2006.*

[War06b] *M. Wardetzky.* Discrete Differential Operators on Polyhedral Surfaces - Convergence and Approximation*. PhD thesis, Freie Universitat Berlin, 2006.*

[WB06] *Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.* Math. Program.*, 106(1):25–57, 2006.*

[WBLT12] *Y. Wang, B. B. Liu, and Y. Tong. Linear surface reconstruction from discrete fundamental forms on triangle meshes.* Comptuer Graphic Forum*, 31(8):2277–2287, 2012.*

[Wei08] *L.-Y. Wei. Parallel Poisson disk sampling.* ACM Transactions on Graphics (SIGGRAPH)*, 27(3), 2008.*

[Wei10] *L.-Y. Wei. Multi-class blue noise sampling.* ACM Transactions on Graphics (SIGGRAPH)*, 29(4), 2010.*

[Wes04] *M. West.* Variational integrators*. PhD thesis, California Institute of Technology, 2004.*

[Whi57] *H. Whitney.* Geometric Integration Theory*. Princeton University Press, 1957.*

[WMKG07] *M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete Laplace operators: No free lunch. In* Symposium on Geometry Processing*, pages 33–37, 2007.*

[WOD09] *E. Whiting, J. Ochsendorf, and F. Durand. Procedural modeling of structurally-sound masonry buildings.* ACM Transactions on Graphics*, 28(5), 2009.*

[WP06] *Y. Wesley Wong and S. Pellegrino. Wrinkled membranes Part II: analytical models.* Journal of Mechanics of Materials and Structures*, 1:25–59, 2006.*

[WSW$^+$12] *E. Whiting, H. Shin, R. Wang, J. Ochsendorf, and F. Durand. Structural optimization of 3D masonry buildings.* ACM Transactions on Graphics*, 31(6), 2012.*

[WW11] *L.-Y. Wei and R. Wang. Differential domain analysis for non-uniform sampling.* ACM Transactions on Graphics (SIGGRAPH)*, 30(4), 2011.*

[XG13] *J. Sun S.-T. Yau X. Gu, F. Luo. Variational principles for minkowski type problems, discrete optimal transport, and discrete monge-ampere equations, 2013. arXiv:1302.5472.*

[XHGL12] *Y. Xu, R. Hu, C. Gotsman, and L. Liu. Blue noise sampling of surfaces.* Computer & Graphics*, 36(4):232–240, 2012.*

[XLGG11] *Y. Xu, L. Liu, C. Gotsman, and S. J. Gortler. Capacity-constrained Delaunay triangulation for point distributions.* Computer & Graphics*, 35(3):510–516, 2011.*

[XXSH11] *Y. Xiang, S.-Q. Xin, Q. Sun, and Y. He. Parallel and accurate Poisson disk sampling on arbitrary surfaces. In* SIGGRAPH Asia Sketches*, 2011.*

[Yel83] *J. I. Yellott. Spectral consequences of photoreceptor sampling in the rhesus retina.* Science*, 221:382–385, 1983.*

[YGL$^+$09] *Y.-L. Yang, R. Guo, F. Luo, S.-M. Hu, and X. Gu. Generalized discrete Ricci flow.* Computer Graphics Forum*, 28(7):2005–2014, 2009.*

[YWLA11] *D.-M. Yan, K. Wang, B. Lévy, and L. Alonso. Computing 2D periodic centroidal Voronoi tessellation. In* Int. Symp. on Voronoi Diagrams in Science and Engineering*, pages 177–184, 2011.*

[YWLL13] *D.-M. Yan, W. Wang, B. Lévy, and Y. Liu. Efficient computation of clipped Voronoi diagram.* Computer-Aided Design*, 45(4):843 – 852, 2013.*

[ZB05] *Y. Zhu and R. Bridson. Animating sand as a fluid.* ACM Transactions on Graphics (SIGGRAPH)*, 24(3), 2005.*

[ZGLG12] *W. Zeng, R. Guo, F. Luo, and X. Gu. Discrete heat kernel determines discrete Riemannian metric.* Graphical Models*, 74(4):121–129, 2012.*

[ZHWW12] *Y. Zhou, H. Huang, L.-Y. Wei, and R. Wang. Point sampling with general noise spectrum.* ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012.*

[ZRS05] *R. Zayer, C. Rossl, and H.-P. Seidel. Discrete tensorial quasi-harmonic maps. In* Proceedings of Shape Modeling and Applications*, pages 278–287, 2005.*