Not so HOT Triangulations

Scott A. Mitchell[a,*], Patrick Knupp[b], Sarah Mackay[a], Michael F. Deakin[a]

[a]*Center for Computing Research, Sandia National Laboratories*
[b]*Dihedral, LLC.*

Abstract

We propose primal-dual mesh optimization algorithms that overcome shortcomings of the standard algorithm while retaining some of its desirable features. "Hodge-Optimized Triangulations" defines the "HOT energy" as a bound on the discretization error of the diagonalized Delaunay Hodge star operator. HOT energy is a natural choice for an objective function, but unstable for both mathematical and algorithmic reasons: it has minima for collapsed edges, and its extrapolation to non-regular triangulations is inaccurate and has unbounded minima. We propose a different extrapolation with a stronger theoretical foundation, and avoid extrapolation by recalculating the objective just beyond the flip threshold. We propose new objectives, based on normalizations of the HOT energy, with barriers to edge collapses and other undesirable configurations. We propose mesh improvement algorithms coupling these. When HOT optimization nearly collapses an edge, we actually collapse the edge. Otherwise, we use the barrier objective to update positions and weights and remove vertices. By combining discrete connectivity changes with continuous optimization, we more fully explore the space of possible meshes and obtain higher quality solutions.

Keywords: mesh quality, optimization, HOT Hodge-Optimized Triangulations

1. Introduction

"Primal-dual meshes" are distinguished by the significance of the geometric relationship between primal elements and their dual Voronoi cells, and by the quality of both types of elements being important at the same time. A well-centered mesh has all triangles containing their circumcenters [1]. Well-centered Delaunay-Voronoi pairs are an attractive ideal. There are some algorithms to construct them [2], but robustness is elusive, especially for bounded domains. Allowing points to be weighted extends (Delaunay mesh)—(Voronoi diagram) duality to (regular triangulation)—(power diagram) duality. Weights drastically affect power cell shape, and can change primal mesh connectivity. Weights represent additional degrees of freedom beyond positions, dramatically extending the types of constructions that are possible. [3]

The HOT (Hodge-Optimized Triangulations) energy is a particular measure of weighted primal-dual quality. Mullen et al. [4] introduced the HOT energy as an upper bound on the discretization error of the diagonalized Delaunay Hodge star operator from Discrete Exterior Calculus (DEC) [5]. This operator enables simulations involving primal and dual quantities, such as divergence and curl, e.g. electromagnetism. For fluid flow, flux divergence is stored on the primal edge, and vorticity curl is stored on the dual perpendicular edge. The simulation couples these quantities, and transfers energy between them. The quantities are typically stored at the midpoint of each edge. If these midpoints are far apart, the transfer introduces error.

A *completely well-centered* mesh requires all dimensions of cells to be well-centered: e.g. edges contain their (weighted) midpoints; and, in 3D, tetrahedra contain their circumcenters. Since the beginning of DEC it was known that a completely well-centered mesh is sufficient, but perhaps not necessary, for a diagonal discretization of the Hodge star operator [5]. For some meshes, it is not possible to adjust the node positions to achieve a well-centered mesh while maintaining the original connectivity [1]. Introducing vertex weights can sometimes produce a completely well-centered mesh. However, even in 2D, for some fixed positions and connectivities, no such weights exist. Op-

⋆This paper is extended from an earlier version awarded the *best paper* of the International Meshing Roundtable (IMR) 2021.

*Corresponding author
Email address: samitch@sandia.gov (Scott A. Mitchell)

timizing for well-centeredness is not the same as optimizing for the diagonalized Hodge star or other operators. Lowering HOT energy also improves many geometric processing algorithms in computer graphics [3], some of which crossover to engineering, such as building self-supporting structures [6].

Mullen et al. [4] demonstrated using the HOT energy as the objective function in mesh position and weight optimization steps. Although the Hodge star operator is undefined for non-regular meshes, the HOT energy formula is continuously extrapolated whenever the optimization algorithm destroys regularity. Edge flips periodically restore regularity throughout the optimization. They state the necessity of preconditioning the mesh, optimizing the position before the weight, and treating the boundary exceptionally. While these improve robustness, they may limit the range of final configurations. They did not fully describe *why* these are needed.

1.1. Contribution

We study both the mathematics and algorithms for primal-dual mesh quality. We focus on 2D planar meshes and demonstrate

- *why* optimizing HOT is not robust;
- using the HOT energy to select *discrete changes* to the mesh;
- *new metrics* that are simple, well-behaved under optimization, and require no preconditioning;
- *algorithms* that combine these concepts.

HOT energy is a poorly behaved optimization objective for both mathematical and algorithmic reasons. We describe the functional form and numerical features of the HOT energy in detail. This illuminates why preconditioning (as in Mullen et al. [4]) or extra length constraints (as in de Goes et al. [3]) were needed, and why weight optimization is especially ill behaved and was deferred to later iterations. In particular, HOT position optimization can nearly collapse edges, moving nodes very close to one another, and can even invert triangles, depending on the extrapolation used. HOT weight optimization of a node can have unbounded solutions, with energy approaching negative infinity for large weights. For a single vertex patch, HOT can be nonconvex with multiple minima. These facts will help researchers design new metrics and algorithms, and we present several of our own.

When HOT would nearly-collapse an edge, we explicitly collapse it. If removing a vertex would improve our new metrics, we remove it. This works very well, leading to higher quality meshes with smooth size transitions. We propose a different extrapolation to non-regular meshes, which eliminates the unbounded solutions for weight optimization. We may avoid extrapolation altogether, by bounding the position or weight optimization step to just barely cause a flip, then restoring regularity before continuing the optimization. We show that this further improves stability. In some cases, the flip threshold is a local optimum, and extrapolation causes oscillations around it. Our new barrier metric prevents collapsed edges and inverted triangles, keeps orthocenters and edge midpoints centered, is invex and scale invariant. It is simple, with only two terms, yet captures the primal-dual geometry. Our pseudo-barrier relaxation can handle an orthocenter that is outside its triangle. We demonstrate an optimization framework that combines vertex removal with position optimization and weight optimization, and wraps these in an outer loop that performs edge flips to restore regularity.

1.2. Prior Work

We identified some HOT shortcomings previously in a technical report [7]. The HOT mathematical formulas had undesirable features, such as troughs that led to inverted meshes and collapsed elements. Some of these issues are fundamental, and some stemmed from how HOT was extrapolated from regular meshes to non-regular meshes. We proposed a new extrapolation grounded in optimal transport theory. We considered introducing barriers by normalizing HOT by the square of the triangle area. While these helped, they did not provide barriers to all types of primal-dual degeneracies we wish to avoid.

Two ad hoc smoothing operators are popular for primal-dual meshes. Optimal Delaunay Triangulations (ODT) [8] repositions primal vertices to the weighted mean of adjacent triangle circumcenters. This can be extended to weighted triangulations [9]. Centroidal Voronoi Tessellations (CVT) [10] reposition vertices at Voronoi cell centroids [11]; this can be sped up by directly minimizing a particular energy [12]. Räbinä [13] uses mixed elements matched to the boundary via tetmesh HOT-based optimization.

Besides DEC, primal-dual meshes are important for mimetic formulations [14] including finite-differences/volumes [15, 9] and staggered discontinuous Galerkin. In mimetic discretizations of fluid flow, velocity is naturally associated with primal edges, while vorticity and conserved scalar quantities are associated with the dual grid. These discretely mimic continuum conservation laws.

The community has moved away from directly optimizing HOT and using the diagonal Hodge star, and has developed alternative metrics and methods for particular applications [9, 3, 16, 17, 18].

Engwirda [9] developed a system for primal-dual meshing of ocean coasts for a mimetic climate simulation. It achieves many of the same objectives we seek in the Hodge context. He designs metrics based on distances between primal and dual points. He alternates between optimizing primal and dual metrics. His metrics are unitless and scale invariant, an important improvement over raw HOT. While the metrics themselves have no barriers to triangle inversion, nor to a weighted edge-center appearing outside its edge, his optimization procedure improves the worst quality at every step so such defects are not introduced provided the original mesh has none. Regularity is restored when nodes or weights move too far. His system includes edge collapses and splitting.

For mimetics, the discretization becomes non-interpolatory when adjacent primal and dual edges fail to intersect. DEC formulations desire primal and dual edges to intersect at their midpoints, and often a constraint is added to require the affine hull of the dual edge to cross the interior of the primal edge [3].

In 2D regular triangulations, primal and dual edges are always orthogonal. Using non-orthogonal primal and dual edges with the diagonal Hodge star introduces error. However, one could include orthogonality as an optimization objective instead of a constraint.

An active research topic is finding alternative Hodge operators with good numerical properties. The competing goals include efficient computation, accurate computation, and indifference to mesh quality. That is, in principle, numerical issues can be addressed by changing the operator, changing the mesh, or some combination of the two. Many recent proposals do not depend on orthogonality, with dual vertices at element centroids, incenters, or other locations.

El Ouafdi et al. [16] proposes an operator that *relaxes the orthogonality condition* and instead adapts the operator to the angle between the primal and dual edge. It replaces the simplex circumcenter by a point that is guaranteed to lie inside the simplex, computed by solving an optimization problem. No mesh modification is needed.

Ayoub et al. [17] proposes a *non-diagonal*, non-symmetric discrete Hodge operator that allows dual vertices to be anywhere, typically at barycenters or incenters, as long as dual elements are not degenerate. The primal mesh is not modified. They conclude that the diagonal Hodge does not always give the most accurate results even when the mesh is well-centered.

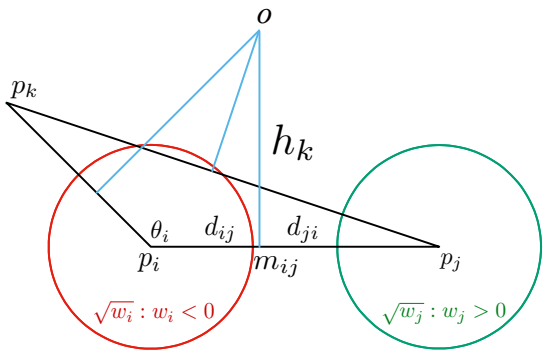


Figure 1: Distances in a triangle and its dual.

Mohamed et al. [18] compare different Hodge star operators, such as the Galerkin Hodge, using different types of dual point positions such as barycenters.

2. Notation and Definitions

2.1. Triangle Geometry

We consider the *regular triangulation* (a.k.a. weighted Delaunay triangulation) $\mathcal{T}$ of a set of weighted points $\mathcal{P} = \{p_i\}$. Here σ denotes a simplex: $\sigma_i = p_i$ is a vertex, $\sigma_{ij} = \overline{p_i p_j}$ an edge, and $\sigma_{ijk} = \triangle(p_i, p_j, p_k)$ a triangle. Recall the weighted distance $D(p, x)$ from weighted point p to unweighted point x is defined by

$$D^2(p_i, x) = |p_i - x|^2 - w_i,$$

where $|p_i - x|$ is Euclidean distance and w_i is the weight of p_i. A weight could be positive or negative, and has units of distance-squared. Unless we explicitly state that a distance is a weighted distance, we mean the Euclidean distance. Triangle σ_{ijk} has an *orthocenter* (a.k.a. weighted circumcenter) o, whose weighted distance to each of the p are identical. The weighted midpoint of edge σ_{ij} is m_{ij}, and is the closest point to o on the affine hull of edge ij, aff(σ_{ij}). See fig. 1.

Closed formulas for o and other quantities are easily computed. The Euclidean distance from o to aff(σ_{ij}) is $|h_k|$, where $h_k > 0$ if o and p_k are on the same side of edge σ_{ij}. Here m_{ij} is at signed distance d_{ij} from p_i towards p_j, with $w_{ij} = w_i - w_j$ and

$$d_{ij} = \frac{\ell_{ij}}{2} + \frac{w_{ij}}{2\ell_{ij}}$$

with $\ell_{ij} = |\sigma_{ij}|$. Since $d_{ij} + d_{ji} = |\sigma_{ij}|$, at most one of d_{ij} and d_{ji} can be negative. Thus, points with positive

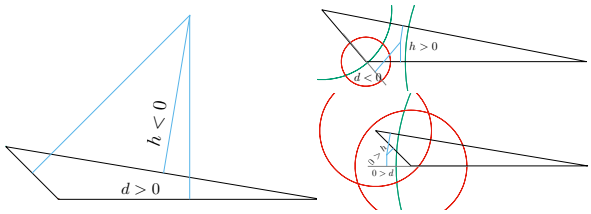


Figure 2: An unweighted obtuse triangle has one $h < 0$, because the circumcenter lies outside it. An unweighted triangle has all $d > 0$. In a weighted triangle, we can have some $d < 0$ even if all $h > 0$. We can have both a $d < 0$ and an $h < 0$. For three non-colinear points, one can place the orthocenter anywhere in the plane and find weights which realize it.

weight have larger power cells and "push" the midpoint and orthocenter farther away. See fig. 2.

The power diagram (a.k.a weighted Voronoi diagram) is the dual of the triangulation. For triangles σ_{ijk} and $\sigma_{ji\ell}$, edge σ_{ij} has dual edge $*\sigma_{ij} = \overline{o_{ijk}o_{ji\ell}}$ with length $h_k + h_\ell$. If the dual length is negative, i.e. the circumcenters are out-of-order, then the triangulation is not regular and the primal edge needs to be flipped to restore regularity. (A triangulation may have all positive dual-length edges and still be non-regular, e.g. it may have inside-out triangles with reversed orientation, and overlapping triangles.)

2.2. HOT Energy

Mullen et al. [4] defines the HOT energy function of a regular mesh $\mathcal{M}$. For $s = 0, 1, 2, \ldots D$,

$$\star^s\mathrm{HOT}_{p,p}(\mathcal{M}) = \sum_{\sigma\in\Sigma^s} |*\sigma||\sigma| W_p(\mu_\sigma, \mu_{*\sigma})^p \qquad (1)$$

where D is the dimension of the mesh and Σ^s are the s-dimensional simplices $\{\sigma\}$, and $\{*\sigma\}$ their $(D-s)$-dimensional duals. Herein we exclusively consider $D = 2$. Here $|\cdot|$ is measure: area for 2-cells, length for 1-cells, and constant 1 for points. Also μ is the probability measure (uniform distribution) over σ. Note W_p is the p-Wasserstein distance between a cell and its dual, the optimal cost of transporting σ to $*\sigma$. Thus the energy is less when a cell is near its dual, such as primal and dual edges crossing near their midpoints, or an orthocenter lying at the centroid of its triangle, or a node lying at the centroid of its power cell. We exclusively consider $p = 2$, so drop its explicit mention.

The energy $\star^s\mathrm{HOT}(\mathcal{M})$ gives an upper bound on the error of the discrete diagonal approximation of the Hodge star operator $\star^s$. The operator $\star^s$ intakes a differential s-form and outputs a $(D-s)$ form. More precisely, given an s-form ω, for each s-cell in $\mathcal{M}$, Discrete Exterior Calculus (DEC) uses the approximation

$$\int_{*\sigma} \star^s(\omega) \approx \frac{|*\sigma|}{|\sigma|} \int_\sigma \omega.$$

The energy $\star^s\mathrm{HOT}(\mathcal{M})$ gives an upper bound on the total approximation error, which is defined as a weighted p-norm of the integral approximation errors, with weights determined by the volumes of the cell and its dual [4]. (Not to be confused with point weights.) Equation (1) reduces to

$$\star^{s/2}\mathrm{HOT}(ijk) = \alpha_s d_{ij}^3 h_k + \beta_s d_{ij} h_k^3, \qquad (2)$$

where the energy of each triangle σ_{ijk} is the sum of $\star^{s/2}\mathrm{HOT}$ over all six permutations of ijk, and the energy of a triangulation is the sum of triangle energies.

All the $\star^{s/2}\mathrm{HOT}$ have the same algebraic form, just different coefficients for each s. For $\star^0$ we have $\alpha_0 = 1/4$ and $\beta_0 = 1/12$; next $\star^1$ has $\alpha_1 = \beta_1 = 1/3$; and $\star^2$ has $\alpha_2 = 1/12$ and $\beta_2 = 1/4$. Despite this seemingly mild difference, these coefficients produce qualitatively different energy curve shapes, with different numbers of minima and different asymptotes.

Although one can compute the energy of a triangulation by summing the terms in any order, the terms have a different geometric interpretation depending on $\star^s$. We call $\star^{1/2}\mathrm{HOT}(ijk)$ the *half-edge* energy.

$$\star^1\mathrm{HOT}(ijk) = \star^{1/2}\mathrm{HOT}(ijk) + \star^{1/2}\mathrm{HOT}(jik)$$

is the *edge energy* of σ_{ij} with respect to triangle σ_{ijk}. For an edge σ_{ij} in triangles σ_{ijk} and σ_{jil}, the sum of the two edge energies, $\star^1\mathrm{HOT}(ijk) + \star^1\mathrm{HOT}(ji\ell)$, is the metric for transporting the edge to its dual, which we call the *full edge* energy. For $\star^2$ for a triangle, summing all permutations of $\star^{2/2}\mathrm{HOT}(ijk)$ is the metric for transporting it to its orthocenter. For $\star^0$ for a point i, summing $\star^{0/2}\mathrm{HOT}(ijk)$ over its incident triangles is the metric for transporting its power cell to it. We focus on $\star^1$, the edge energy, because it is the least well behaved and thus our paper contributes the most towards it. But all are important because there exist simulations that use each type of $\star^s$ to store primal and dual simulation quantities.

2.3. Quality Metrics

The literature [4] used $E_{\mathrm{F}}^s = \star^s\mathrm{HOT}$ directly as a mesh optimization metric, where eq. (2) is applied even when the mesh is not regular and dual edges have negative length. We call this *functional extrapolation*.

Mousley et al. [7] proposed an optimal-transport inspired *theory extrapolation* E_{T}. Here $E_{\mathrm{T}} = E_{\mathrm{F}} = \star\mathrm{HOT}$ when the mesh is regular, but modified to treat negative dual lengths as positive:

$$E_{\mathrm{T}}^{s} = \mathrm{sgn}(h_k + h_\ell) \star^{s} \mathrm{HOT} \tag{3}$$

This ensures the full edge energy is always positive. In this paper we test this approach. In our numerical experiments the extrapolation is closer to the true energy after flips restore regularity. These avoid spurious unbounded solutions during the optimization.

For a bounded domain, there is a choice of how to define the extent of the dual of a boundary edge σ_{ij}. We define $h_\ell = 0$ for the non-existent triangle $\sigma_{ij\ell}$ outside the boundary for all our energies. For E_{T}, $\mathrm{sgn}(h_k + h_\ell)$ reduces to $\mathrm{sgn}(h_k)$ in eq. (3), and h_k is replaced by $|h_k|$ in eq. (2).

We propose $E_{\partial\mathrm{T}}$ which uses E_{F} for interior edges, and E_{T} for domain boundary edges. We explore $E_{\partial\mathrm{T}}$ as a compromise, a minimal change to E_{F} that removes an undesirable boundary effect, but still allows negative edge energies for internal edges.

We propose the following three metrics with barriers to triangle inversion and edge collapses; see section 4 for details.

The *hard barrier* half energy E_{BH} is inspired by normalizing eq. (2) by the squared area of the corresponding sub-triangle in fig. 5, specifically by $d_{ij}^2 h_k^2$. Additionally, to actually provide a barrier, one normalizing d_{ij} term is replaced by $\sqrt{d_{ij}d_{ji}}$. It requires $h_k > 0$ and $d_{ij}d_{ji} > 0$.

$$E_{\mathrm{BH}}^{s/2} = \frac{\alpha d_{ij}^2 + \beta h_k^2}{\chi_{ij} h_k} \tag{4}$$

$$\chi_{ij} = \sqrt{d_{ij}d_{ji}} \tag{5}$$

The *shifted hard barrier* half energy E_{BHs} modifies one normalizing h term to allow unweighted non-acute triangles, where $h_k \le 0$. We choose constant $h_k^o \le 0$ so $h_k > h_k^o$ and still require $d_{ij}d_{ji} > 0$.

$$E_{\mathrm{BHs}}^{s/2} = \frac{\alpha d_{ij}^2 + \beta h_k^2}{\chi_{ij}(h_k - h_k^o)} \tag{6}$$

Note $E_{\mathrm{BHs}}^{s/2} \to \infty$ as $h_k \to h_o$. In contrast, we design the *pseudo barrier* half energy E_{BP} to become large, but not infinite, as h_k decreases. We choose a constant γ, require $d_{ij}d_{ji} > 0$, but h_k can be arbitrary.

$$E_{\mathrm{BP}}^{s/2} = \frac{\alpha d_{ij}^2 + \beta h_k^2}{\chi_{ij}\phi_k} \tag{7}$$

$$\phi_k = \frac{1}{2}\left(h_k + \sqrt{h_k^2 + \gamma^2}\right) \tag{8}$$

2.4. Calculating Flip Threshold

We seek to calculate the limit of how much we can change the position or weight of a vertex before the connectivity of the regular triangulation changes. Triangle σ_{ijk} exists if its power product with respect to any other vertex p_ℓ is positive. The power product is the determinant of the following matrix; see for example the Regular_triangulation_2 section of the CGAL manual [19].

$$\mathrm{pwr}(\sigma_{ijk}, p_\ell) = \begin{vmatrix} 1 & x_i & y_i & v_i \\ 1 & x_j & y_j & v_j \\ 1 & x_k & y_k & v_k \\ 1 & x_\ell & y_\ell & v_\ell \end{vmatrix} \tag{9}$$

where $v_i = x_i^2 + y_i^2 - w_i$. Let p_ℓ be the patch vertex we are optimizing. Expanding the determinant along the bottom row, the power test is

$$-A + x_\ell B - y_\ell C + v_\ell D > 0, \tag{10}$$

$$A = \begin{vmatrix} x_i & y_i & v_i \\ x_j & y_j & v_j \\ x_k & y_k & v_k \end{vmatrix}, \quad B = \begin{vmatrix} 1 & y_i & v_i \\ 1 & y_j & v_j \\ 1 & y_k & v_k \end{vmatrix},$$

$$C = \begin{vmatrix} 1 & x_i & v_i \\ 1 & x_j & v_j \\ 1 & x_k & v_k \end{vmatrix}, \quad D = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix}.$$

A flip occurs when the determinant is zero. WLOG let p_ℓ's initial position be the origin and weight be zero. When optimizing the weight, a flip occurs at

$$w = -A/D.$$

Only weight thresholds in the downhill direction are relevant. E.g., if increasing w improves quality, then we are searching for the optimal positive weight, and we do not care about any negative weight thresholds.

When optimizing the position, we move in the direction of the gradient $[x, y]$ parameterized by t. Therefore, a flip occurs for t at the roots of

$$-A + t(xB - yC) + t^2(x^2 + y^2)D,$$

which can be determined by the quadratic formula. One root occurs when the gradient line is tangent to the orthocircle of σ_{ijk}, two when the line intersects the circle

twice, and zero when the line misses the circle. As with weights, some of these roots may occur in the uphill direction and be irrelevant.

When optimizing a patch center, we consider the thresholds that cause each patch triangle to cease to exist. For each patch-triangle edge, we compute the power with respect to the far vertex of the other triangle sharing that edge. (Two of the other triangles are also patch triangles; the third is immediately outside the patch, or does not exist if the edge is on the domain boundary.) The limit is the minimum-magnitude threshold over all patch triangles.

3. HOT Shortcomings

HOT energy may be a stellar way of bounding the discretization error, but using it as the objective in an optimization algorithm has shortcomings, especially if positions or weights change enough that the triangulation is no longer regular.

- HOT optimization can collapse triangle edges.
- Extrapolating HOT to non-regular meshes is inaccurate, and optima can be bad.
- Extrapolating to bounded domains takes care.
- HOT energy depends on scale.

HOT has no barrier to an edge collapse. Indeed, the energy of a triangle approaches zero as its edge is collapsed, a local minimum for $\star^1$HOT. Collapsing is a problem even if the optimization algorithm maintains the initial regular triangulation. HOT for a non-regular triangulation is an extrapolation because the Delaunay Hodge star is only defined for regular triangulations. After restoring regularity the energy may be quite different: increase or decrease. Worse, the extrapolation may lead to nonsensical optimal configurations. HOT depends on length to the fourth power, so the energy can go down if a change makes the biggest element smaller, even if all elements' shapes gets worse.

We demonstrate these facts in the next subsections, and show their effects on optimization in section 6.

3.1. No Barriers, Non-convex

3.1.1. Zero Energy for Collapsed Edges

Figure 3 plots the E_F HOT energy for a triangle as one of the vertices is moved. None of the points are weighted. None of the HOT energies are convex. For the $\star^1$ energy E^1_F the optimal triangle is not equilateral: there is a saddle point for an obtuse isosceles triangle, which is a local minima if we just vary y, but a local max if we vary x. Moving the free vertex to either of the other two vertices is a local minimum, with $E^1_\mathrm{F} = 0$. Otherwise, moving the free vertex towards the edge increases the energy to infinity. The contour plot for $\star^2$ is qualitatively similar to $\star^1$'s. [7].

The $\star^0$ energy E^0_F is also 0 for a collapsed edge. However, there is no saddle, and the energy increases with increasing y coordinate. Worse, the energy is *negative* for points close to $y = 0$, and collapsing the node into the middle of the opposite edge is a local minimum.

What does it even mean to have negative energy? The energy is defined in eq. (1) in terms of positive probability distributions and the positive cost of transporting points positive distances. This presents a theoretical conundrum, resolved by realizing this is an artifact of how we have chosen to separate the energy into sub-terms and how we deal with bounded domains. Recall the $\star^0$ energy is the transport cost from Voronoi cells to nodes. For E_F we have not said how we are defining the extent of the Voronoi cells for the bounded domain of a single triangle. In an unbounded mesh the sum of contributions for the entire Voronoi cell around each node would be positive. This resolves the immediate challenge, but sets up the numerical pitfalls of extrapolating the HOT function to non-regular meshes and defining the dual for the domain boundary.

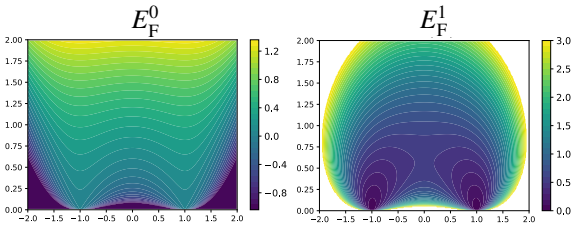


Figure 3: E_F HOT contour plots for a triangle with a single free vertex, and fixed vertices (-1,0) and (1,0), courtesy of Mousley et al. [7].

3.1.2. Unbounded Energy by Weight

Figure 4 shows E_F as the weight of a vertex of a single triangle is varied. For E^1_F there is a desirable local minima, but globally the energy is unbounded: $E^1_\mathrm{F} \to -\infty$ as $w \to -\infty$. Triangles with other angles and side lengths are qualitatively similar. The local minima is shallower the larger the angle. (While we depict an isolated triangle in fig. 4 center, it could be embedded in a regular mesh with large nearby triangles, and that weight could be a local minimum.)

E^2_F is convex and *not* unbounded. At the local minimum, often the orthocenter is inside the triangle, but

the problem is we can have $d_{ij} < 0$ if the triangle has a large angle; see fig. 4 top right, where the leftmost blue perpendicular does not intersect the leftmost edge. (It merely intersects the affine hull of the leftmost edge.) If we simply constrain $d_{ij} > 0$, then the energy keeps decreasing as $d_{ij} \to 0$, which can have algorithmic consequences.

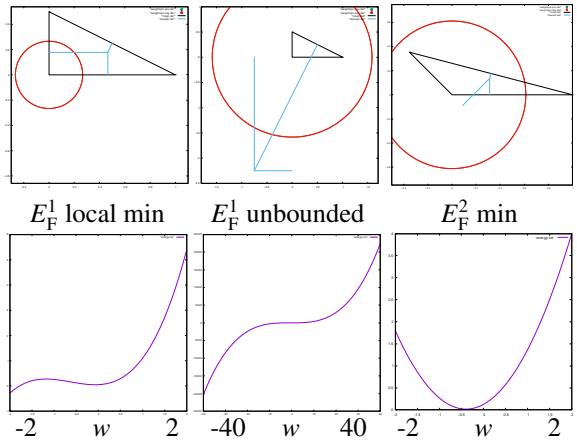


Figure 4: Bottom: E_F energies for varying the weight of one vertex in a triangle with side lengths 0.5 and 1. Top: $\triangle$ at local min and approaching global min.

3.2. Extrapolation to Non-Regular

If the triangulation is non-regular we have the theoretical challenge that the Delaunay Hodge star is not defined, so any definition of HOT is an extrapolation. The original HOT paper [4] uses E_F extrapolation [via personal communication with the authors]. This presents the numerical challenge that the energy could indeed be negative; we cannot rely on positive terms cancelling out negative terms.

For a single edge that should be flipped, the two adjacent triangles' orthocenters are in reverse order and the dual edge has negative $(h_k + h_\ell)$ length. Clearly, after flipping the edge, both the primal and dual replacements will have non-negative length. The length of the replacement is monotonic in the magnitude of the negative length, so we propose using Mousley et al.'s [7] extrapolation E_T that *increases* with increasing magnitude, which is eq. (3). This also makes sense for optimal transport theory if we consider the negative-length dual a proxy for its replacement after flipping. However, the extrapolation could be an over-estimate or an under-estimate, depending on the shape of the triangles and the weights of points, and we have no bounds on its accuracy. This also does not resolve the issue that the other edges of each triangle are using a "wrong" orthocenter and hence an extrapolated h_k, nor the fact that sometimes multiple non-local edge flips are needed to restore regularity. It appears difficult to come up with an accurate extrapolation short of actually restoring regularity and computing the true new energy.

One must also decide how to compute the energy of inverted triangles, with reversed vertex orientation. If they are treated the same as non-inverted triangles, then the sum of triangle areas is greater than the domain, and HOT's length-to-the-fourth-power scale dependency tends to produce large energies, which discourages inverted triangles during optimization. This is what we have chosen to do in our implementation, but have demonstrated cases where HOT optimization produces inverted triangles nonetheless; see section 6.5.2. For weighted points, one must also decide how to extrapolate energies for a "hidden vertex" that will have no power cell and will not appear in the triangulation after regularity is restored. We explore this issue in section 6.6.2.

3.2.1. Energy on the Boundary

A domain-boundary edge has only one triangle, which defines only one of its dual-edge vertices. We recommend defining the other dual vertex to be the edge's weighted midpoint and using our theory-based extrapolation. This is theoretically reasonable, and eminently practical. We simply use $|h_k|$ in the boundary-edge terms. It improves quality over functional extrapolation by discouraging large obtuse angles on the domain boundary; see section 6.5. One could also use some other fixed fictitious orthocenter. Mullen et al. mirrors the triangle [4].

3.2.2. Flips can Increase Energy

Many mesh quality metrics are globally non-convex without serious consequences. However, we note that because of the imperfect extrapolation, the energy can go *up* when we do flips to restore regularity. This happens when we optimize the "dense horseshoe" (Figure 11). If we re-optimize after flipping we get oscillatory behavior, cycling between two optimal meshes; see section 6.6.1.

3.3. HOT is not Scale Invariant

HOT strongly favors small elements, due its dependence on length-to-the-fourth-power. In an optimization of a fixed domain and fixed mesh connectivity, the only thing keeping an element from collapsing to a point is the increase in size of neighboring elements, and keeping domain-boundary nodes fixed.

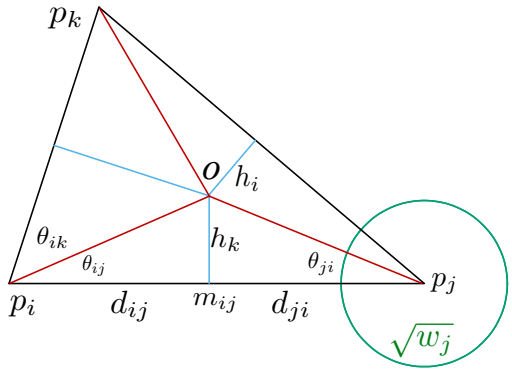


Figure 5: Barrier metric triangle distances.

HOT energy depends not only on the shape of the elements, but indirectly on the number of elements. This causes problems in some optimization loops. This is counter-productive when deciding if a vertex should be removed or inserted to improve quality, and whether the solution is converged. In 2D, or 3D with fixed connectivity, one can get away with it because the number of each type of element (and hence, energy terms) remains constant. However, in 3D with flips to restore regularity, or in 2D with our vertex removal, the number of elements changes. Optimizing the average-per-element HOT energy is an insufficient fix.

4. Barrier and Pseudo-barrier Energies

We propose energies with barriers to triangle inversion and bad primal-dual edge configurations. This section does not consider flips or collapses but only metrics for node movement and weight optimization.

Barrier metrics (energies) have been used with success in primal mesh optimization for some time to enforce the condition that the Jacobian determinant remains positive during optimization (provided the initial determinant is positive). This is an essential requirement for performing computations on primal meshes. In the Target Matrix Optimization Paradigm such barriers are widely used, with the Condition Number metric being a prominent example [20].

As a casual definition, a barrier metric is a non-negative metric defined on some domain $\mathcal{D}$ such that (1) the value of the metric approaches infinity as one approaches the boundary of $\mathcal{D}$ from inside $\mathcal{D}$ and (2) there exists at least one global minimum on $\mathcal{D}$. A barrier metric is undefined for points outside $\mathcal{D}$.

Primal barriers restrict the position of a vertex during optimization. A similar approach can be used in the primal-dual setting to create barrier metrics with additional restrictions that control the orthocenter so that $h_k > 0$ and $d_{ij} > 0$. First, let us focus on $d_{ij} > 0$ and consider edge σ_{ij} of triangle σ_{ijk}. The HOT energy does not naturally enforce the condition that the line passing through the dual edge must intersect the primal edge in its interior i.e. we must explicitly require $\text{aff}(*\sigma_{ij}) \cap \text{int}(\sigma_{\text{ij}}) \neq \emptyset$ [3]. This condition is important in several discretization schemes. This is equivalent to requiring $0 < d_{ij} < \ell_{ij}$ where $\ell_{ij} = |\sigma_{ij}|$. Further, if $0 < d_{ij} < \ell_{ij}$, then $0 < d_{ji} < \ell_{ij}$, and $0 < \chi_{ij} < \frac{\ell_{ij}}{2}$ from eq. (5). If all the weights are zero, then $d_{ij} = \frac{\ell_{ij}}{2}$ and these conditions are satisfied.

Second, let us consider how to avoid triangle inversion, which is related to $h_k > 0$. Let us define the sign of h_i in terms of the right-handed orientation of σ_{ij} to o. If all h are positive then the triangle cannot be inverted. This is sufficient, but not necessary. (If we define h_i to be positive if o and σ_k lie on the same side of $\text{aff}(\sigma_{ij})$ then positive h is not sufficient.)

To create a barrier, we can thus use the product $h_k\chi_{ij}$ in the denominator of the energy of the ij^{th} edge. For this edge metric, the abstract domain is

$$\mathcal{D} = \left\{(d_{ij}, h_k) \mid 0 < d_{ij} < \ell_{ij}\,, h_k > 0\right\},$$

which is open, semi-infinite, and convex. To keep the coefficients simple, only the $\star^1$ case $\alpha = \beta$ is considered. The edge energy is the sum of the two half edge energies given in eq. (4). Thus the *hard barrier* edge energy for σ_{ij} in triangle σ_{ijk} is

$$E^1_{\text{BH}} = \frac{d_{ij}^2 + (\ell_{ij} - d_{ij})^2 + 2h_k^2}{h_k\chi_{ij}}, \qquad (11)$$

where recall $\chi_{ij} = \sqrt{d_{ij}d_{ji}}$ from eq. (5).

This energy is invariant to scaling, i.e., it has dimensions of length-to-the-zero power, in contrast to the HOT energy. This metric most likely cannot be derived from optimal transport theory.

As the point $(d_{ij}, h_k) \in \mathcal{D}$ approaches the boundary given by $h_k = 0$, the barrier energy goes to infinity. The same is true for points approaching the boundaries $d_{ij} = 0$ and $d_{ij} = \ell_{ij}$. Finally, as $\frac{h_k}{\ell_{ij}}$ goes to infinity within $\mathcal{D}$, as would happen if edge σ_{ij} was collapsing, the barrier edge energy also approaches infinity. Thus the edge energy satisfies condition (1) for being a barrier metric. (In contrast, if we attempt to normalize HOT by the square of the triangle area, as we proposed in our technical report [7], in the final case the energy goes to zero and there is no barrier.)

Considering condition (2), observe that $E^1_{\mathrm{BH}} \geq 4$ for any point in $\mathcal{D}$. This can be demonstrated using the following inequality:

$$\left[d_{ij} - (\ell_{ij} - d_{ij})\right]^2 + 2\left[h_k - \sqrt{d_{ij}(\ell_{ij} - d_{ij})}\right]^2 \geq 0$$

By inspection, the barrier edge energy is a continuous and differentiable function on $\mathcal{D}$. Thus, any critical points must occur at the stationary points, i.e., at the set of points for which $\frac{\partial E^1_{\mathrm{BH}}}{\partial d_{ij}} = 0$ and $\frac{\partial E^1_{\mathrm{BH}}}{\partial h_k} = 0$. There are several solutions to this pair of equations but only one of them lies inside $\mathcal{D}$. That solution is $(d_{ij}, h_k) = (\frac{\ell_{ij}}{2}, \frac{\ell_{ij}}{2})$. At this stationary point, the energy is equal to 4, therefore it is a global minimizer and, further, the global minimizer is unique. This satisfies condition (2). So, the barrier edge energy satisfies the definition of a barrier metric.

Figure 6 shows a contour plot of the edge barrier energy on a subset of $\mathcal{D}$. The contours suggest that the energy is a convex function of (d_{ij}, h_k), but we have counter-examples to show it is not. This leaves open the possibility that the energy is a function that obeys some sort of generalized convexity such as quasi-convexity. At the very least, it has been shown above that the energy is an invex function, i.e., a function for which every stationary point is a global minimizer.

Figure 6: Contour Plot of E^1_{BH} edge energy as a function of (d_{ij}, h_k).

Proposition.
The hard barrier edge energy eq. (11) is a quasi-convex function of (d_{ij}, h_k).
Proof.
For clarity, (d_{ij}, h_k) are replaced with (d, h) in this proof. Consider two points in $\mathcal{D}$: (d_0, h_0) and (d_1, h_1). Let $\lambda \in (0, 1)$ and define $d_\lambda = (1-\lambda)d_0 + \lambda d_1$, $h_\lambda = (1-\lambda)h_0 + \lambda h_1$. Define

$$\begin{aligned} T(d,h) &= d^2 + (\ell - d)^2 + 2h^2 \\ \chi(d) &= \sqrt{d(\ell - d)} \\ B(d,h) &= h\chi(d) \end{aligned}$$

so that the hard barrier edge energy is $e = \frac{T}{B}$. The function $T(d, h)$ is strictly convex on $\mathcal{D}$ because it is the sum of three strictly convex functions. Thus

$$T(d_\lambda, h_\lambda) < (1-\lambda)T(d_o, h_o) + \lambda T(d_1, h_1) \tag{12}$$

The Hessian of $B(d, h)$ is

$$\begin{pmatrix} \frac{\partial^2 B}{\partial d^2} & \frac{\partial^2 B}{\partial d \partial h} \\ \frac{\partial^2 B}{\partial d \partial h} & \frac{\partial^2 B}{\partial h^2} \end{pmatrix} = \begin{pmatrix} -\frac{h\ell^2}{4\chi^3} & \frac{\ell - 2d}{2\chi^2} \\ \frac{\ell - 2d}{2\chi^2} & 0 \end{pmatrix}$$

Since the diagonal elements of the Hessian are non-positive and the determinant of the Hessian is negative, the Hessian is negative semi-definite. Therefore

$$(1-\lambda)B(d_o, h_o) + \lambda B(d_1, h_1) \leq B(d_\lambda, h_\lambda) \tag{13}$$

From eqs. (12) and (13),

$$e(d,h) < \frac{(1-\lambda)T(d_o, h_o) + \lambda T(d_1, h_1)}{(1-\lambda)B(d_o, h_o) + \lambda B(d_1, h_1)} \tag{14}$$

The right-hand-side of eq. (14), as a function $f(\lambda)$, has no stationary points. Therefore,

$$\begin{aligned} \max_{0<\lambda<1} f(\lambda) &= \max\left[f(0), f(1)\right] \\ &= \max\left[\frac{T(d_o, h_o)}{B(d_o, h_o)}, \frac{T(d_1, h_1)}{B(d_1, h_1)}\right] \\ &= \max\left[e(d_o, h_o), e(d_1, h_1)\right] \end{aligned}$$

§

Quasi-convexity for Weights. In terms of the weight optimization algorithm, the energy is not minimized as a function of d_{ij} and h_k, but rather as a function of the weights w_i, w_j, w_k (with fixed vertex coordinates). Define the weight-difference $w_{ij} = w_i - w_j$ so that

$$\begin{aligned} d_{ij} &= \frac{\ell_{ij}}{2} + \frac{w_{ij}}{2\ell_{ij}} \\ h_k &= \frac{\ell_{ij}\cot\Theta_k}{2} + \frac{\cot\Theta_i}{2\ell_{ij}} w_{jk} + \frac{\cot\Theta_j}{2\ell_{ij}} w_{ik} \end{aligned} \tag{15}$$

Because $w_{ij} + w_{jk} + w_{ki} = 0$, the three weight differences are not independent. Choose w_{ik} and w_{jk} to be independent and $w_{ij} = w_{ik} - w_{jk}$. Then

$$d_{ij} = \frac{\ell_{ij}}{2} + \frac{w_{ik}}{2\ell_{ij}} - \frac{w_{jk}}{2\ell_{ij}} \tag{16}$$

Define $\mathcal{D}_w$ to be the set of points (w_{ik}, w_{jk}) such that the corresponding points $(d_{ij}, h_k) \in \mathcal{D}$. Then there is an affine transformation from $\mathcal{D}_w$ to $\mathcal{D}$ given by eqs. (15)

and (16) whose Jacobian is the matrix

$$J_{ij} = \begin{pmatrix} 1 & -1 \\ \cot\Theta_j & \cot\Theta_i \end{pmatrix}$$

The determinant of J_{ij} is non-zero provided $\cot\Theta_i + \cot\Theta_j \neq 0$. In that case, the map is invertible and there is a one-to-one and onto correspondence between the points in $\mathcal{D}_w$ and $\mathcal{D}$. Thus, there is a unique point in $\mathcal{D}_w$ at which the barrier edge energy is minimized. That point is easily found by setting $d_{ij} = h_k = \frac{\ell_{ij}}{2}$. One obtains

$$w_{ik} = w_{jk} = \frac{1 - \cot\Theta_k}{\cot\Theta_i + \cot\Theta_j}\ell_{ij}^2$$

At this minimizing point, $w_{ij} = 0$. Because affine maps transform straight lines to straight lines, $\mathcal{D}_w$ is open, semi-infinite, and convex when the determinant is non-zero. Moreover, the affine map guarantees that if the level sets of the barrier edge energy on $\mathcal{D}$ are convex, then the level sets on $\mathcal{D}_w$ are also convex.

Starting Inside the Barrier. An important prerequisite to using the barrier energy is that in the initial mesh all abstract points (d_{ij}, h_k) must belong to $\mathcal{D}$. If one uses zero for the initial weights, then $0 < d_{ij} < \ell_{ij}$ is guaranteed, so the d_{ij} are good. However, in the unweighted case, $h_k = \frac{\ell_{ij}\cot\Theta_k}{2}$ and the condition $h_k > 0$ is equivalent to the initial mesh being acute, which is rarely satisfied. This is a problem for the hard edge barrier eq. (11) because the metric is undefined when $h_k \leq 0$. This kind of difficulty occurs also when using barrier metrics to optimize primal meshes, and two work-arounds are commonly used. The first is the *shifted barrier* technique which moves the domain so that the initial mesh lies inside $\mathcal{D}$ [21]. For example, in the present case, replace h_k in the denominator with $h_k - h_k^o$, where $h_k^o = \min(h_{min} - \epsilon, 0)$, with h_{min} being the minimum of h_k over the initial mesh and ϵ a small positive number. The shifted domain of the metric is

$$\mathcal{D}_s = \left\{(d_{ij}, h_k) \mid 0 < d_{ij} < \ell_{ij}\,, h_k > h_k^o\right\}$$

and the shifted hard barrier edge metric is

$$E_{\text{BHs}} = \frac{d_{ij}^2 + (\ell_{ij} - d_{ij})^2 + 2h_k^2}{(h_k - h_k^o)\chi_{ij}} \tag{17}$$

Although h_k^o is computed on the initial mesh, it can be recomputed at later iterations, moving the barrier closer to zero. We have analyzed the shifted barrier energy and established that it is both invex and quasi-convex. The minimum occurs at $d_{ij} = \frac{\ell_{ij}}{2}$ and $h_k = h_k^o + \sqrt{(h_k^o)^2 + \left(\frac{\ell_{ij}}{2}\right)^2}$. The minimum lies in $\mathcal{D}$ for all $h_k^o \leq 0$.

The second work-around, explored more thoroughly in the present work, is the pseudo-barrier technique [22]. The quantity h_k is replaced with $\phi(h_k)$, where ϕ was defined in eq. (8) in section 2.3: $\phi_k = \left(h_k + \sqrt{h_k^2 + \gamma^2}\right)/2$. It is important to note that $\phi(h_k) > 0$ for any h_k regardless of $\gamma > 0$. In introducing ϕ into the metric, there is no longer a hard barrier on h_k in the pseudo-barrier edge energy.

$$E_{\text{BP}}^1 = \frac{d_{ij}^2 + (\ell_{ij} - d_{ij})^2 + 2h_k^2}{\phi(h_k)\chi_{ij}} \tag{18}$$

The pseudo-barrier metric has domain

$$\mathcal{D}_{BP} = \left\{(d_{ij}, h_k) \mid 0 < d_{ij} < \ell_{ij}\right\}$$

and can thus be used even for $h_k < 0$. We call it a *pseudo-barrier* because it does not go to infinity as $h_k \to 0$, but only becomes a very large but finite number. How large is controlled by the parameter γ. Using $\gamma = h_k^o$ may be suitable. As h_k becomes more and more negative, the energy continues to increase. So, the pseudo-barrier does not prevent h_k from being negative, but encourages h_k to be positive during the numerical optimization.

We implemented both work-arounds in our primal-dual optimization code. In limited experiments they performed about the same. For simplicity we focus on the pseudo-barrier when presenting numerical results in section 6. Figure 7 shows a contour plot of the pseudo-barrier edge energy on a subset of $\mathcal{D}_{BP}$ as a function of (d_{ij}, h_k); the contours again suggest a unique minimum for the pseudo-barrier energy. One can show that the Hessian of $\phi(h)\chi$ is negative definite for $\gamma^2 < h^2$. With that, the Proposition above can be modified to show that the pseudo-barrier edge energy is also a quasi-convex function.

Figure 7: Contour plot of E_{BP} edge energy as a function of (d_{ij}, h_k).

One problem with both the shifted and pseudo-barrier metrics, in terms of ease of use, is that parameters h_k^o and γ must be chosen. To maintain scale invariance,

they should be chosen to scale with the mesh.

5. Algorithm

The pseudocode appears in algorithms 1–5.

Algorithm 1 Optimize Triangulation $\mathcal{T}$

Parameters:
E_o energy to optimize, e.g., E_{BP}
E_c energy for collapses $\in \{E_{\text{F}}, E_{\text{T}}, E_{\partial\text{T}}\}$
i_c first iteration to consider collapses
i_w first iteration to optimize weights
function Optimize($\mathcal{T}$ with ∂)
    repeat ▷ outer iterations
        repeat ▷ inner iterations
            for point $p \in \mathcal{T}$ **do**
                Gather patch $P(p)$ with
                  ordered perimeter $\{q_j\}$
                if not Collapse(E_c, $P(p)$) **then**
                    p = Optimize position(E_o, $P(p)$)
                    p = Optimize weight(E_o, $P(p)$)
                    if $E_o(P \setminus p) < E_o(P)$ **then**
                        remove p from $\mathcal{T}$
            shift weights so p_{anchor} has weight = 0
            increment iteration i
        until $E(\mathcal{T})$ converged and no collapses
        Flip $\mathcal{T}$ to restore regularity
    until no flips were needed

Algorithm 2 Collapse Point p

function Collapse(E_c, $P(p)$, i, i_c)
    if ($p \in \partial\mathcal{T}$) or ($i < i_c$) **then return** `False`
    p' = Optimize position(E_c, $P(p)$)
    q_i = closest point(p', $\{q_j\}$)
    if $|\overline{p'q_i}| < 0.1 \min(|\overline{q_{i-1}q_i}|, |\overline{q_iq_{i+1}}|)$ **then**
        $\mathcal{T}$.remove(p)
        return `True`
    return `False`

5.1. HOT Collapses

We called HOT-optimization collapsing an edge a shortcoming, but we can use this as a strength. Our thesis is that this means the mesh would be improved by making a discrete mesh change and actually collapsing the edge. We implemented this option. We show that it obviates the need for preconditioning. We show it can dramatically improve the optimized mesh. It is powerful enough on its own to give us a good mesh even when we only optimize positions and not weights; see section 6.6.1.

Algorithm 3 Optimize Position p

function Optimize Position(E, $P(p)$)
    if $p \in \partial\mathcal{T}$ **then return** p
    return Optimize(E, $P(p)$, pos)

Algorithm 4 Optimize Weight p

function Optimize Weight(E, $P(p)$, i, i_w)
    if $i < i_w$ **then return** p
    return Optimize(E, $P(p)$, weight)

We implemented the following simple rule; see algorithm 2. We first optimize point p using HOT without a barrier: E_{F}, E_{T}, or $E_{\partial\text{T}}$. If this makes the shortest edge connecting p to the perimeter short compared to the adjacent perimeter edges, we remove p. Otherwise, we continue with p's original position.

After optimizing the position or weight of p, we can check if the energy of its patch improves by removing it. This often removes vertices in the same cases as the short-edge test, and many other cases. See section 6.6.4 for details.

5.2. Algorithm Details

We discuss the effect of the essential parameters and choices in algorithm 1, and provide some implementation information for reproducibility.

The patch $P(p)$ consists of the triangles containing p. When using E_{T}, we also need the orthocenters of the triangles sharing an edge with the patch perimeter to determine if any dual edges have negative length.

We use the most-recent position and weight for each point in a patch. A variation is to use the state at the beginning of the iteration, which removes order-dependence, but was unstable in practice and made the triangulation non-regular very quickly. A variation optimizes the positions for all points, then the weights for all points; this made little difference. Domain-boundary

Algorithm 5 Optimize

Require: E has no broken barriers
function Optimize(E, $P(p)$, position or weight)
    $u = \nabla E(P(p))$ ▷ finite differences, pos or w
    line search for p^* ▷ golden section
      in direction u
      limit distance to k*flip-threshold (optional)
    return p^*

nodes have fixed position but we optimize their weight. At the end of each inner iteration, we shift all weights so that an arbitrary "anchor point" always has weight 0.

We may skip collapses and weight optimization in early inner iterations. In our experiments we typically collapse starting immediately at iteration 0, or never. We typically start weight optimization at iteration 0, 2, 4, after positions are converged, or never. We perform flips only at the end of an inner iteration. Flipping after each point is optimized is costly but improves stability. It has an unpredictable effect on the final mesh.

We consider the inner loop converged if the latest iteration did no collapses, and the average energy per element decreased by less than 1.0e-6.

For positions, since the gradient is non-scalar, we optimize thrice. For both positions and weights, we optimize using finite differences for the gradient, and a golden section line search for the optimum. The maximum search distance is the patch radius. (Scaling the problem so the patch radius is 1 can improve the numerics.) For weights with shallow local minima, the line search can leave the local-minimum neighborhood and continue in the unbounded direction up to the maximum search distance. This can create triangles whose subsequent optimal positions invert the triangle.

We implemented our algorithm in C++ using CGAL 2D Triangulation [19] for the regular triangulation construction, collecting the patch, and performing flips to restore regularity. We use exact predicates and inexact constructions. We use a std::map to represent changed vertex coordinates and weights within an inner iteration. CGAL does not have constrained regular triangulations, which is expected considering that theory is underdeveloped. Instead, we represent non-convex boundaries by introducing vertices outside the domain that protect the boundary, marking certain triangles as outside the domain, as needed on an example-by-example basis.

6. Numerical Results

6.1. Single Triangle

6.1.1. Optimal Position

Figure 8 shows the $\star^1$ energies for a triangle with a free vertex and a fixed edge. Note E_{F} and E_{T} have minima when the free vertex is coincident with one of the fixed vertices. For E_{BH} the minimum is achieved by an equilateral triangle. For E_{BP}, if $\gamma = 0$ then $E_{\mathrm{BP}} = E_{\mathrm{BH}}$. Otherwise, the optimum is isosceles, and increasing γ increases the optimal angle at the free vertex, up to about 80°. Also E_{BHs} with $h_k^o = 0$ is E_{BH}, but increasing h_k^o changes the optimum in a similar way. E_{BHs} and E_{BH} have no values in the white regions, whereas E_{BP} is just very large.

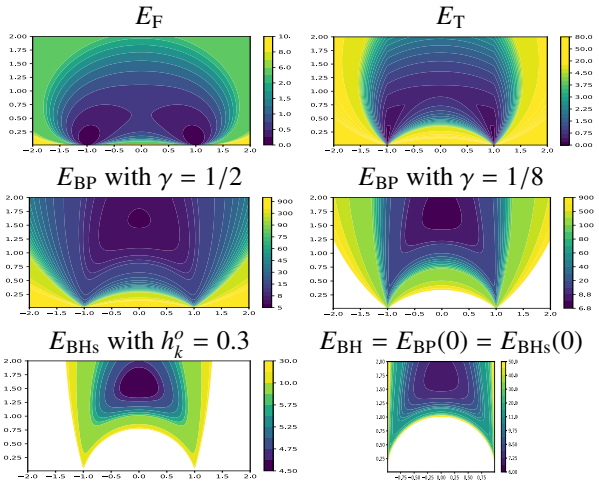


Figure 8: $\star^1$ energies for a triangle with two fixed vertices (-1,0) and (1,0) as a function of the free vertex coordinates (x,y).

6.1.2. Optimal Weight

We consider a fixed-position triangle with the freedom to vary the weight of one vertex, and find the optimal weight for that vertex; see also section 3.1.2. Triangles have varying angle θ between sides of length 0.5 and 1. Figure 9 shows the optimal triangles for each of our $\star^1$ energies. For E_{BH} and E_{BHs} for some large-angles there is no solution with only one weight: to get all $h_k > h_k^o$ at least one $d_{ij} < 0$. (With the freedom to change two weights, there is always a solution, since one can place the orthocenter anywhere, say at the incenter, and there are weights that achieve it.) Note $E_{\mathrm{T}} = E_{\mathrm{F}}$ whenever all $h_k > 0$. For E_{BP} we used $\gamma = 1$, the length of the horizontal edge. For E_{BHs} we used $h_k^o = 0.15$. E_{BP} and E_{BHs} can be encouraged towards well-centered orthocenters by setting smaller constants. For very obtuse triangles, E_{T} produces the best h_k, but sometimes with $d_{ij} < 0$.

Figure 10 shows how the optimal weights vary by angle for each energy. Note E_{BP}, E_{BHs}, and E_{BH} optimal energies increase for small triangles with sharp angles, whereas E_{F} and E_{T} optimal energies decrease. Figure 10 shows how the optimal weight, and $\min(d_{ij})$ and $\min(h_k)$ at optimality, compare for the different energies. Note E_{F}, E_{T}, and E_{BP} nearly agree for triangles without large or small angles. For extreme triangles, the order of preferring large h_k over large d_{ij} is $\{E_{\mathrm{BH}}, E_{\mathrm{T}}, E_{\mathrm{BHs}}, E_{\mathrm{F}}, E_{\mathrm{BP}}\}$.

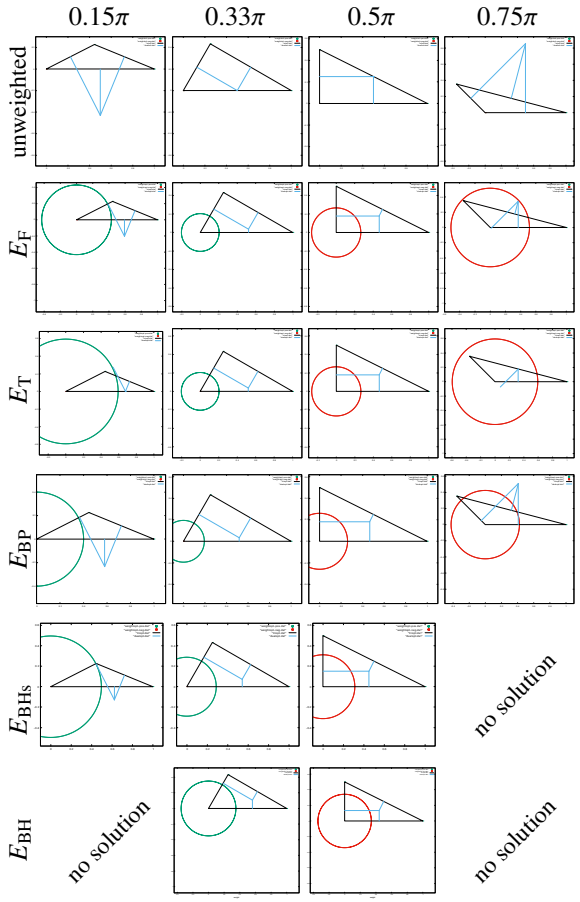


Figure 9: Optimal weight for one vertex of a triangle by angle θ and $\star^1$ energy function. Circle radii are $\sqrt{w}$, with green circles for positive weight and red for negative. For E_F we show the best local minima: its energy is unbounded for a large-magnitude weight.

6.2. *Weight and Valence Correlation*

Engwirda [9] observes a correlation between valence and weights when minimizing orthocenter to centroid distances. This is explained by observing the correlation between valence and angles.

"Low-valence nodes tend to have large negative weights." [9] Low valence correlates to large angles which means far away orthocenters. Negative weights simply move those orthocenters closer to the primal vertex and triangle centroid.

"High valence nodes tend to have large positive weights." [9] High valence is correlated with small angles. If the opposite edge is short, the centroid is far from the sharp angle. Positive weights move the orthocenter away from the vertex and towards the centroid.

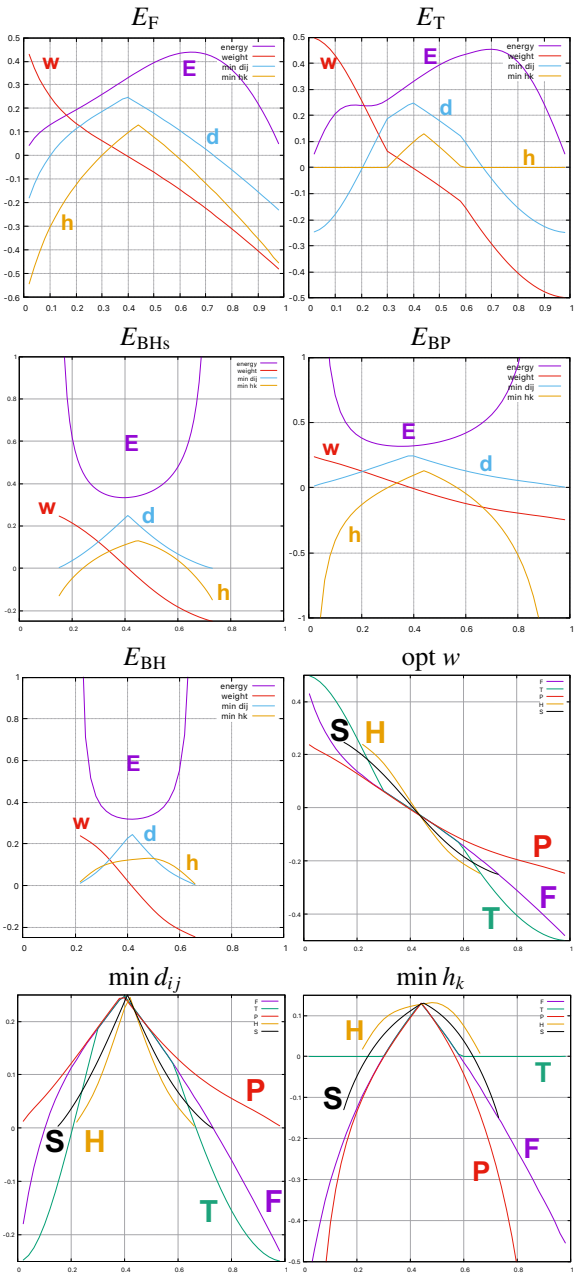


Figure 10: Optimal weight for a single triangle vertex, and $\min(d_{ij})$ and $\min(h_k)$ at optimality. Triangle has side lengths 0.5 and 1, and the x-axis is the angle between those sides divided by pi. E_F is unbounded but we plot the local minimum only. E_{BH} and E_{BHs} have no solutions outside the shown ranges. Energies are scaled for illustration, and figures have different y-scales. H=E_{BH}, S=E_{BHs}, P=E_{BP}, T=E_T, and F=E_F.

6.3. *Input Parameters*

Recall Algorithm 1 takes parameters energy function E, first iteration to start weight optimization i_w, and first

iteration to consider collapses i_c, and whether we continue optimizing after flips to restore regularity. We can turn off weights and collapses altogether. The first iteration is 0. In our examples we present a range of i_w, but $i_c = 0$ or collapses are turned off. For E, we restrict our presentation to a single star for brevity, and select $\star^1$ because it is the most challenging to obtain stable results.

6.4. Sparse Horseshoe

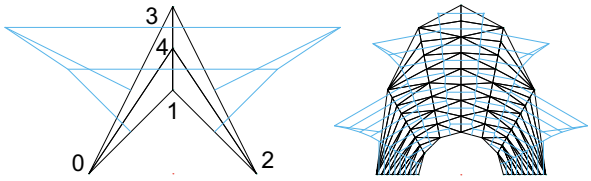


Figure 11: Sparse and dense horseshoe inputs.

We optimize the *sparse horseshoe* in fig. 11 left, with vertices $\{(-2,0),(0,2),(2,0),(0,4),(0,3)\}$. This is a challenge problem where Laplacian smoothing often inverts triangles. There is only one vertex, p_4, that is free to move, but all points can change weights. We consider both position and position+weight optimization. We consider discrete collapses or no collapses.

The hard barrier metric is unusable because there are no weights that satisfies all $d_{ij} > 0$ and all $h_k > 0$ for the input positions. (This is easy to observe: if $d_{ij} > 0$ for all edges, then $h_{401} > 0$ and $d_{10} > 0$ implies aff$(*\sigma_{04})$ intersects σ_{03} with y-coordinate below the y-coordinate of vertex p_4, so $h_{03} < 0$.)

See fig. 12 for position optimization. Energies $\{E_\mathrm{F}, E_{\partial\mathrm{T}}, E_\mathrm{T}\}$ all nearly-collapse edge σ_{41}, which is undesirable. (Depending on how the extrapolation for inverted triangles is done, optimization can send p_4's $y \to -\infty$.) Position optimization of E_BP and E_BHs produce superior meshes with no nearly-collapsed elements. If we enable discrete collapses, then *any* energy collapses σ_{41}, which is desirable because σ_{123} is better shaped than either of σ_{124} or σ_{423}.

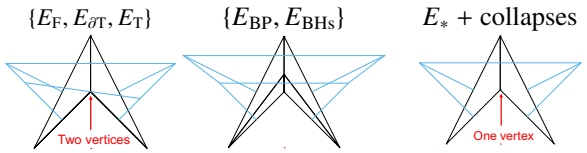


Figure 12: Position-only optimization. Right, with discrete collapses. Left and center, without. $\{E_\mathrm{F}, E_{\partial\mathrm{T}}, E_\mathrm{T}\}$ differences are minor. Note p_4's y-coordinate depends on $\{E_\mathrm{BP}, E_\mathrm{BHs}\}$'s parameters.

See fig. 13 for weight optimization. If we optimize weights starting in iteration 2 ($i_w = 2$), then E_F has the same problem we observed for a single triangle in section 3.1.2, that the energy is unbounded for a large magnitude weight, for triangles σ_{014} and σ_{124}. However, E_T and $E_{\partial\mathrm{T}}$ do not have this problem. E_F, E_T and $E_{\partial\mathrm{T}}$ all still nearly collapse edges if $i_w = 2$, but do not if $i_w = 0$ (depending on the order in which the optimization visits vertices); see fig. 14. I.e. if we optimize weights from the beginning, sometimes the weights will reshaped the duals enough that the energy no longer goes down when collapsing vertices. The barriers do not collapse vertices in either case.

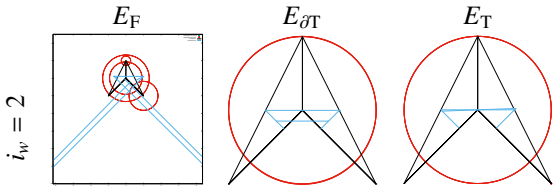


Figure 13: Deferred $i_w = 2$ weight optimization, no collapses. Without barriers, vertices 4 and 1 become coincident. E_T has nearly coincident orthocenters.

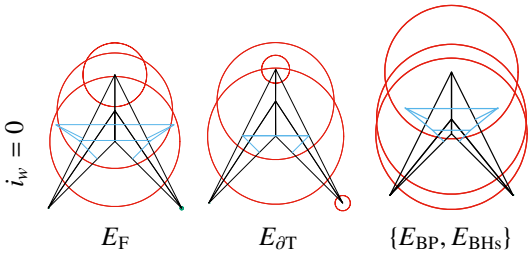


Figure 14: Immediate $i_w = 0$ weight optimization can deter near-collapses for $\{E_\mathrm{F}, E_{\partial\mathrm{T}}, E_\mathrm{T}\}$, but $\{E_\mathrm{BP}, E_\mathrm{BHs}\}$ do not create near-collapses regardless of i_w because of their barriers. E_T is similar to $E_{\partial\mathrm{T}}$.

Combining both weight optimization and explicit collapses, fig. 15 shows that collapses still take place if weight optimization is deferred to iteration 2. If we perform weight optimization from iteration 0, for $\{E_\mathrm{F}, E_{\partial\mathrm{T}}, E_\mathrm{T}\}$ we get results as in fig. 14: no undesirable near-collapses, but no desirable explicit collapses either. For E_BP using $i_w = 0$ the desirable explicit collapse still takes place as in fig. 15 right, just delayed.

6.5. Dense Horseshoe, No collapses

We optimize the dense horseshoe from fig. 11 right. The "parameterized horseshoe" is a classic challenge

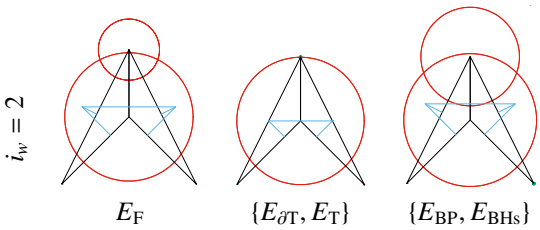


Figure 15: Explicit collapses with positions and delayed weight optimization ($i_w = 2$).

problem by one of the authors [Knupp].

$$x(a,b) = [(1-b)x_0 + bx_1]\cos(\pi(1-a))$$
$$y(a,b) = [(1-b)y_0 + by_1]\ \sin(\pi(1-a))$$

The vertices are defined by discrete values of a and b spaced uniformly from $[0,1]$. The number of vertices on each arch is the number of a values, and vertices on the feet b; and $\{x_0, x_1\}$ define the base of the feet, and $\{y_0, y_1\}$ the height of the arches. Our dense horseshoe has $x_i = \{1,2\}, y_i = \{1,4\}$, 7 values of a and 9 values of b. Larger y values make the problem more challenging, but $y_0 = 1$ was sufficient to demonstrate the problems we discovered and our solutions.

6.5.1. Position Only

Figure 16 shows $\star^1$ optimization with and without a barrier. Optimizing $E_{\partial T}$ produces nearly-coincident nodes. Optimizing E_{BP} does not, even though we use $\gamma = 1$, a large value. Optimizing E_{BP} gives a mesh where the E_F^1 energy is higher than it was in the initial mesh. Optimizing $E_{\partial T}$ dramatically reduces the circumcenters lying outside the domain. Optimizing E_F or E_T makes about the same nearly-coincident vertices as $E_{\partial T}$. Optimizing E_{BHs} produces a mesh similar to E_{BP}.

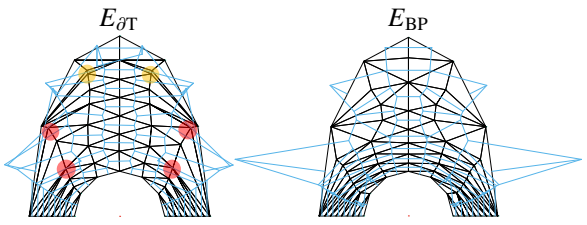


Figure 16: Position-only optimization, no weights, no discrete collapses. Optimal meshes before flips. Red highlights nearly-coincident nodes. Yellow highlights nodes that may become coincident in a few iterations.

6.5.2. Position and Weights

A HOT Mess. Optimizing E_F^1 using positions and weights produces a HOT mess on the dense horseshoe; see fig. 17. We quickly get large negative energies and inverted elements. Optimizing $E_{\partial T}^1$ is no better, since the tangled elements are interior to the domain. Collapsing and delaying weight optimization until positions are converged still produces HOT messes. These plus restricting weight changes can sometimes prevent a HOT mess.

Optimizing $\{E_T^1, E_{BP}^1, E_{BHs}^1\}$ produces a good mesh; see fig. 18. Using $i_w = 2$ to delay weight optimization makes little difference to the outputs, but they converges in fewer iterations than for $i_w = 0$.

6.6. Dense Horseshoe, Collapses and Flips

6.6.1. No Weights

No Barrier. Without weights, we demonstrate the effectiveness of edge collapse using the short-edge test even without a barrier, using $E_{\partial T}^1$. Once the positions and collapses (inner iterations) converge, we demonstrate flipping to restore regularity and re-optimizing (outer iterations). Figure 19 shows collapses within the first set of inner iterations. After a collapse, the next position-optimization step does *not* introduce large changes. This supports our thesis that actually collapsing nearly-collapsed vertices improves the mesh, because the configuration remains nearly-optimal and we have removed a sharp triangle. Figure 20 shows flips and outer iterations. Most collapses occur early in the inner iterations of each outer iteration. Note some large obtuse triangles are formed, second row left column, but flips remove these.

Outer Oscillation. The outer loop oscillates between two optimal meshes because each is no longer optimal after flipping. The inner optimization converges to one state that is non-regular. Restoring regularity, then optimizing again, converges to a second non-regular state. This repeats. The root cause is an imperfect extrapolation to non-regular meshes.

Let a be the interior vertex of the purple kite in the left arm of the horseshoe. From either regular state, when optimizing the patch around vertex a, a single line-search is sufficient to move vertex a to a non-regular state. Figure 21 shows this cycle and the extrapolated energy regime. For the left-most position for a, the regular triangulation of its patch has energy 1.14. Optimizing it to the right-most position gives an extrapolated energy of 0.986, which increases to 1.49 when regularity is restored. Optimizing it again produces the left-most position with extrapolated energy 1.09. The situation

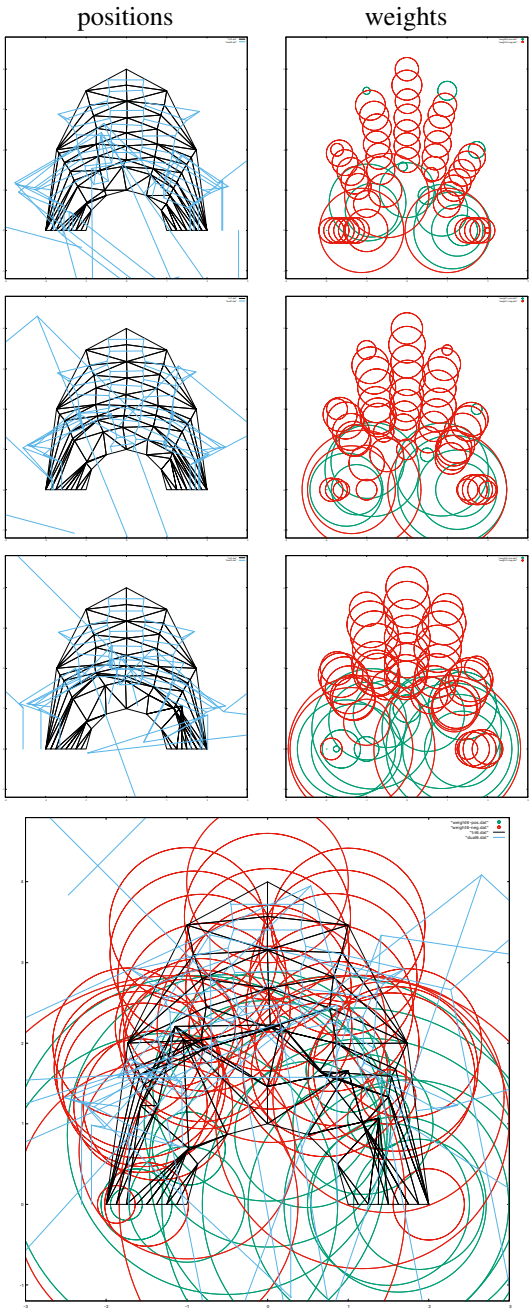


Figure 17: **HOT mess.** Position and weight optimization of E_{F}, with $i_w = 0$, no discrete collapses, produces a HOT mess for the dense horseshoe. For clarity we show mesh positions separate from the weights. Rows are iterations 0, 1, 2, and 6.

is similar for vertex b on the right arm of the horseshoe. (The absolute value of the patch energies in the two branches are not perfectly comparable because the patches have different numbers of triangles, and in each branch the other vertices are in different positions.)

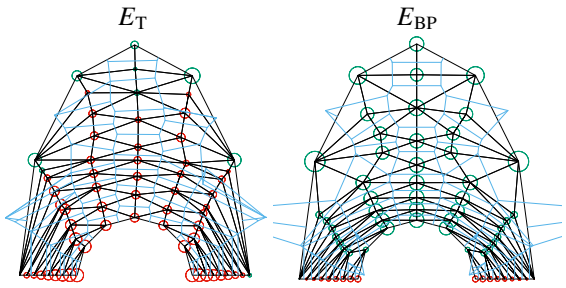


Figure 18: Positions and weights, optimal E_{T} and E_{BP} for $i_w = 0$, no discrete collapses, before flips for regularity. The E_{BHs} mesh is very similar to E_{BP}. Radii shown 1/4 scale for clarity.

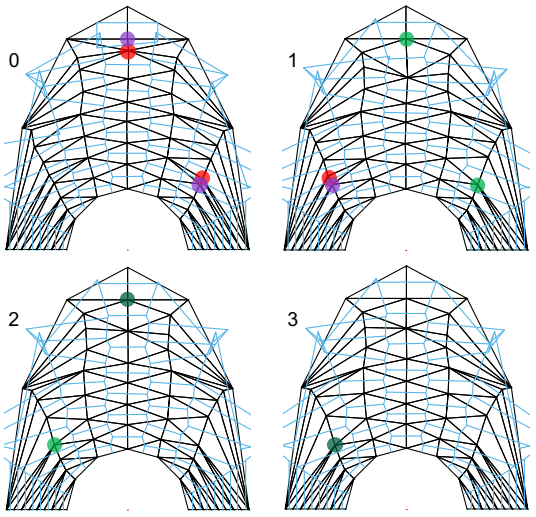


Figure 19: Positions and collapses, energy $E^1_{\partial \mathrm{T}}$, no weights. Inner iterations 0, 1, 2, 3. Illustration of position optimization nearly collapsing edges, followed by discrete collapses. A red vertex is removed because its E^1_{F}-optimal position is too close to the purple vertex. The remaining purple vertex is colored green in the next iteration, and dark-green in the iteration after that, to show that its position is stable. Figure 20 is a continuation of this run.

The solution is to stop the line searches as soon as the new position would cause a flip. More precisely, we calculate the threshold as in section 2.4 and limit the line search distance to some constant factor times it. We perform the necessary flips before continuing the optimization. This ensures that any significant movement is based on the actual energy. Using this strategy, the mesh converges to fig. 22. The global energy converges to 3.29, whereas the two oscillating states have global energies 3.41 and 4.08. Here a is located at the threshold of a position that causes a flip, the blue circle in fig. 21. On both sides of the flip, the two triangles of the kite have coincident orthocenters, the light-blue dual vertices slightly outside the lower part of the horseshoe

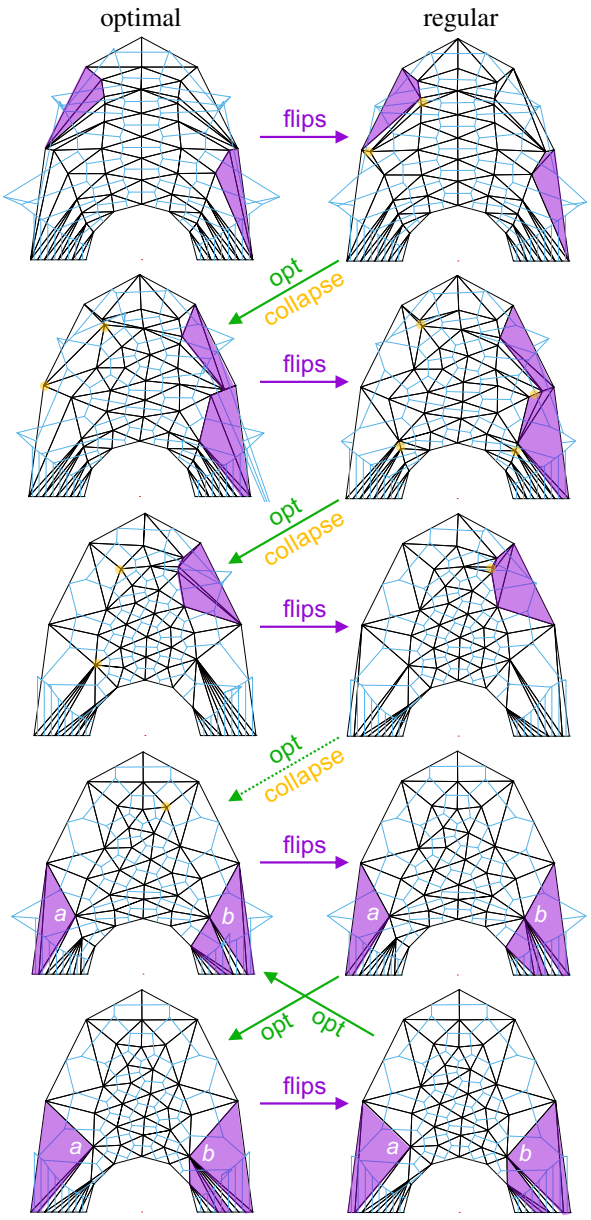


Figure 20: Outer iterations with positions and collapses, energy $E^1_{\partial T}$, no weights. The left column is the optimal mesh after the inner iterations converged, and the right column is that optimal mesh after flipping to restore regularity. Orange shows selected areas that collapse following the optimization arrow from the upper right to the lower left. Purple shows selected cavities that flip when going from the right to the left to restore regularity. The rows are the ends of outer iteration 0, 1, 2, 5, and 6. The last two rows repeat ad infinitum, flipping then optimizing vertices a and b to new positions that require flipping back. This is a continuation of the run started in fig. 19

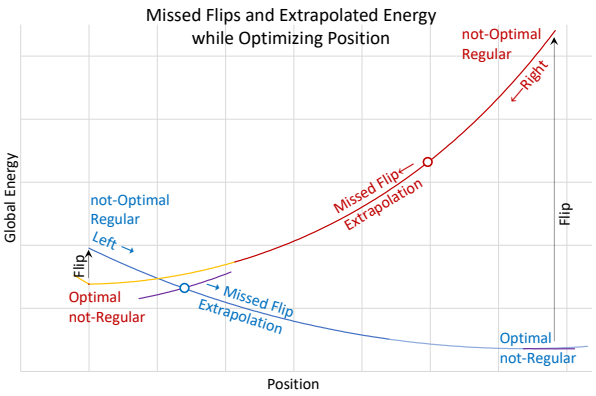


Figure 21: The $E_{\partial T}$ energy as vertex a is optimized and then the mesh is re-regularized by flips, as in the bottom four meshes in fig. 20. Blue shows starting from a's leftmost position and moving to the right through optimization. Each blue shade is one of three line searches; the first search stops at its limit, and the other two find locally-optimal positions. Red shows it returning via the same process and behavior, although the 2D route taken does not retrace the first route. In each direction the mesh becomes non-regular during the first line search, and most of the optimization is based on inaccurate extrapolated energies. The purple curve through the blue flip point shows the true energy if we perform the flip; the flip point is a local-minima of the true energy.

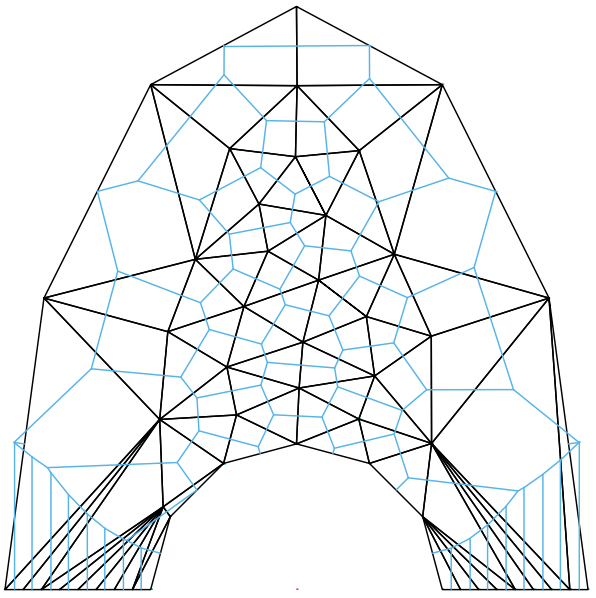


Figure 22: The optimal $E_{\partial T}$ mesh when the optimization uses the true energy and not an extrapolation. The mesh does not oscillate. As soon as a vertex moves to a position that would cause a flip, we perform that flip and continue optimizing from there. The starting configuration is the right-column last-row of fig. 20.

in fig. 22. Either diagonal of the kite contributes zero energy because its dual edge has zero length. That threshold is a local minima of the true energy: see the upper branches of the blue and purple energy curves through the blue flip point in fig. 21.

Position Thresholds. We experimented with various constant factors for the threshold, stopping *before*, *after*, or *at* the flip point. The best strategy was akin to simulated annealing, stopping *after* but with a decreasing overshoot. Initially we stop 10% *after* the flip point, that is, we allow movement up to $1.1\times$ the initial point's distance to the flip point. In each subsequent iteration we decrease the allowable overshoot by a factor of 10. Stopping *after* with too large of an overshoot led to worse final quality. Stopping *at* the flip point was unpredictable, because, due to numerical roundoff, either configuration might be regular. Stopping *before* converged slowly and prevented flips, so did not explore both sides of flip points.

6.6.2. Weight Optimization over Oscillatory Horseshoe

We investigate the stability and effect of $E^1_{\partial \mathrm{T}}$ weight optimization over the mesh that has outer oscillations under position-only optimization, bottom fig. 19. This appears to be a challenging problem, because position-and-weight optimization with deferred flips is a HOT mess.

We consider variations of when weights and connectivity are updated, and whether weights are limited with respect to the mesh becoming non-regular (e.g. needing flips) as in section 2.4. The terminology used for the variations, e.g. in figs. 23 and 24, is the following:

- "Immediate" updates means the current weight and regular connectivity of neighbors is used when optimizing each patch. That is, the inner and outer iterations are combined.
- "Deferred" updates means we use the weights or connectivity at the beginning of the iteration, and only update after all patches have been optimized. That is, we perform one inner iteration per outer iteration.
- "Unlimited" means we optimize the weights ignoring any flip boundaries, using extrapolated energy as needed.
- "Before" means we stop the weight optimization prior to the value where the regular connectivity would change, due to an edge would flip or a vertex becoming hidden.
- "After" is our annealing strategy, meaning we allow the weight to go just past the regularity threshold, with a decreasing limit each iteration.

As is well known in mesh optimization, when sweeping over a mesh, it is more stable to update a vertex immediately and use its new position and weight when optimizing its neighbors, as opposed using the initial weights and positions for its neighbors, and performing updates for all vertices simultaneously once all vertices are optimized. Deferring weight updates exhibits oscillations as expected.

Consider immediate weight & flip updates, the left column in figs. 23 and 24. Unlimited weights leads to 37 flips: a handful are the same vertex causing edges to flip back and forth. Many vertices become hidden, both in the interior and three on the boundary. Nonetheless, in this example the domain is preserved and the mesh is usable. "After" converges quickest. It has 13 flips; only two vertices cause more than one flip. "Before" is the most stable, and by design has no hidden vertices or flips.

Consider immediate weights but deferred flips, the middle column in figs. 23 and 24. Unlimited weights was chaotic. For "After," deferred flips resulted in a hidden vertex and lower energy. For "Before," since it prevents flips when combined with immediate weight updates, the results are independent of whether flips are immediate or deferred.

Consider deferring weights and flips, the right column in figs. 23 and 24. This resulted in oscillatory weights and the slowest convergence. Despite this, "After" and "Before" achieved lower energy than with immediate weights or flips, due to hiding vertices and eliminating skinny triangles on the base of the arms. Although the "After" and "Before" meshes are visually indistinguishable, the "After" mesh is minutely better.

Recommendations. In all cases the "After" weight threshold results in the best quality (lowest energy) mesh, and no other approach converged quicker. The exception is that if one wants to guarantee the connectivity does not change, no flips and no hidden vertices, then use "Before" with immediate weight updates. Unlimited weight changes based on extrapolated energies hid many vertices, so use with caution. These results depend on optimizing the normalized energy; since the un-normalized HOT energy scales as the size of an element to the fourth power, in most cases hiding vertices and coarsening the mesh would increase the HOT energy.

Position & Weight Optimization. Position and weight $E^1_{\partial \mathrm{T}}$ optimization with deferred flips leads to a HOT mess. Limiting position and weight updates to the $E^1_{\partial \mathrm{T}}$ flip threshold leads to a good mesh. Alternatively, using theory based extrapolation, E^1_{T}, also gives a good mesh, even without limits. See fig. 25 for the meshes.

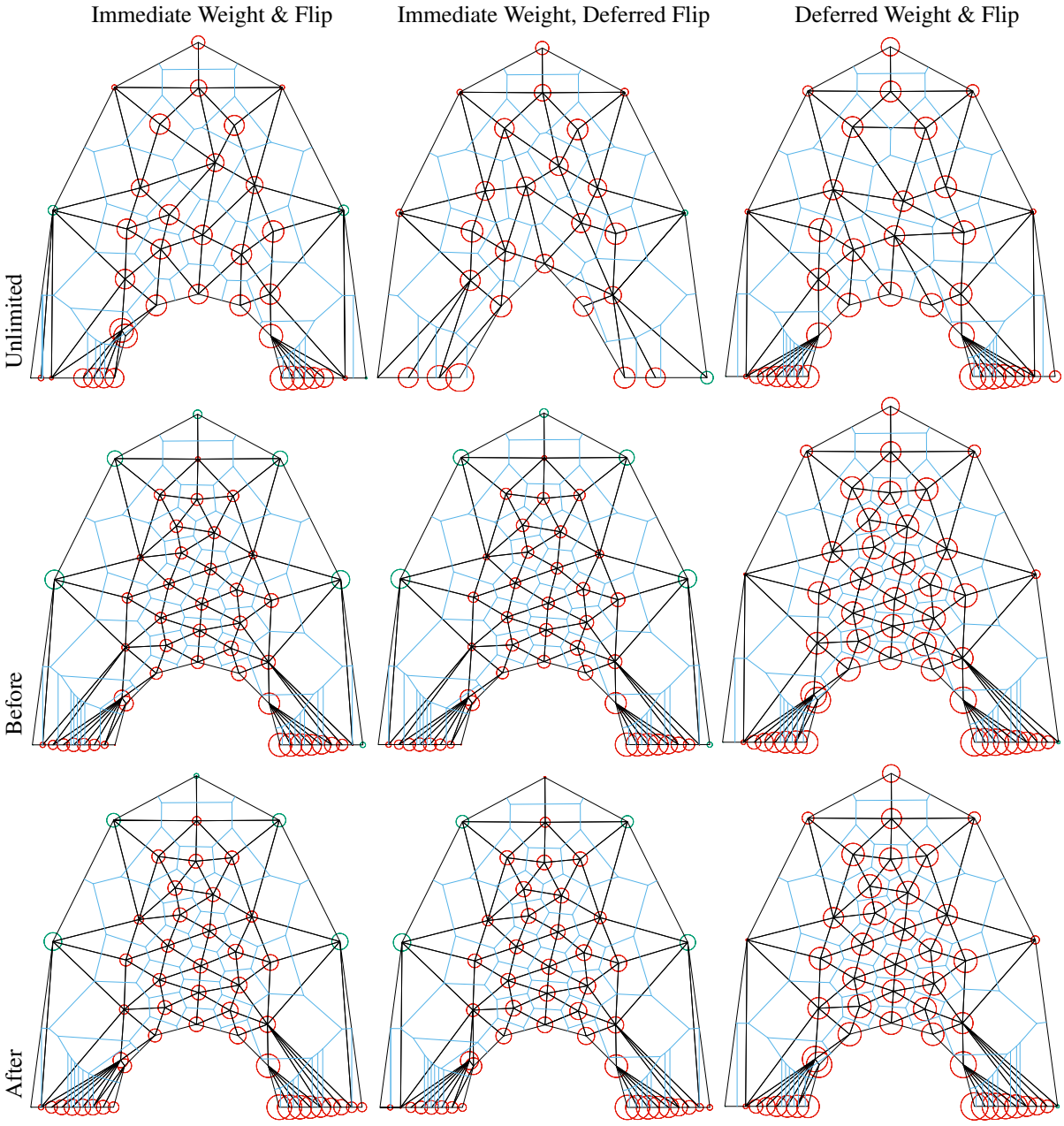


Figure 23: Final meshes for $E^1_{\partial T}$ weight optimization over the oscillatory horseshoe; weight circles drawn 1/4 radius for clarity. Note all dual vertices have moved inside the domain. In many cases vertices are hidden: in the mesh interior in the top row; and in the relative-interior of the domain boundary in the lower left arm of the horseshoe in middle-column bottom-row and right-column, and both arms in the upper left. The top-row middle-column has hidden vertices in the arch of the horseshoe, leading to a different domain. The starting meshes are identical.

6.6.3. *High Heat: Barrier, Collapses, Positions, Weights, and Flips*

We demonstrate that collapses can be combined with optimizing barrier energies, even though barriers by themselves prevent collapses. Recall we collapse if the optimal E_F position places two nodes close together, but that after that check we use E_{BP} or E_{BHs} optimization to actually move the node and reweight it. In this experiment, the first collapses occurred *after* several iterations of E_{BP} optimization, which is somewhat surprising. Here we allow collapsing immediately, but only start weight optimization after positions and collapses

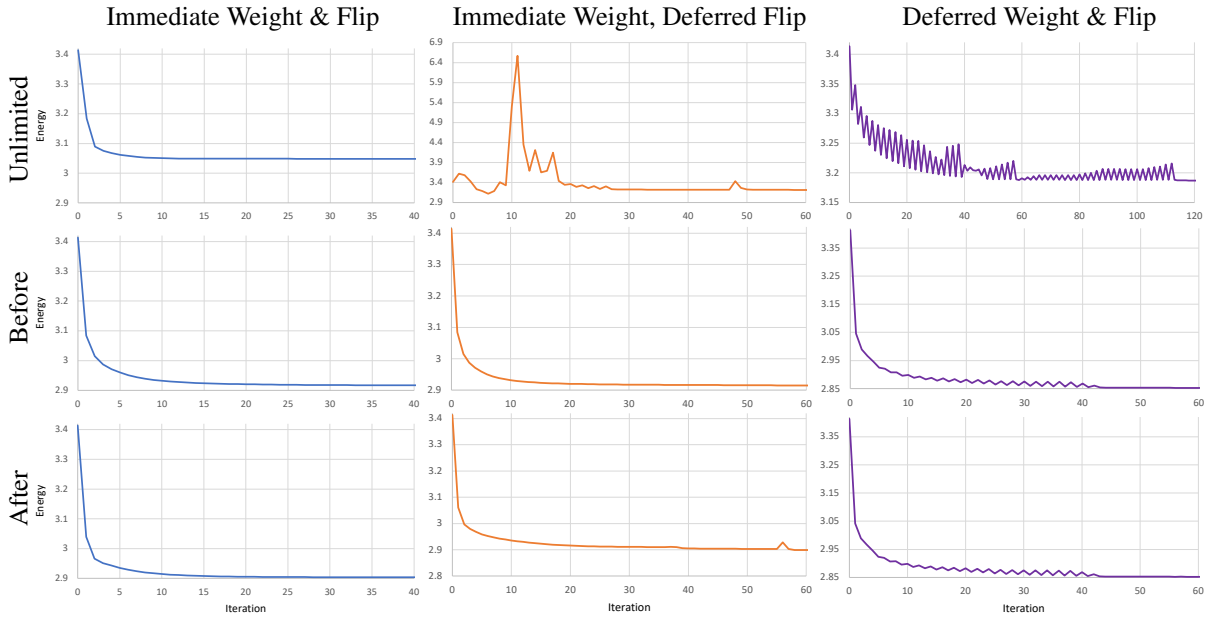


Figure 24: $E^1_{\partial \mathrm{T}}$ energy by iteration for weight optimization over the oscillatory horseshoe. The energy oscillations for the deferred weight updates (right column) arise because weight changes in the vertices of a triangle tend to be reinforcing, so the changes overshoot. The large jumps in energy tend to occur when a vertex becomes hidden. In the middle-column top-row, the hidden vertices change the domain area so the energy is not comparable to the other figures. Note the scale of some axes are different.

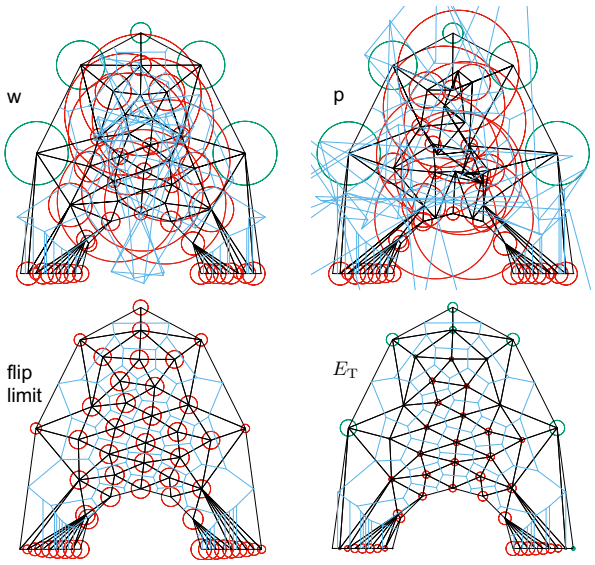


Figure 25: Position and weight $E^1_{\partial \mathrm{T}}$ optimization with deferred flips. Top row (circle radii to scale) the first weight iteration produces inverted dual elements, and the subsequent position iterations produces inverted primal elements as well. Bottom row (circle radii 1/4 scale) limiting weights and positions to the flip threshold, or using theory based extrapolation, E_{T}, produce good meshes.

have converged, $i_w = 40$. As in section 6.6.1, once the inner iterations converge, we flip for regularity and re-optimize (outer iterations).

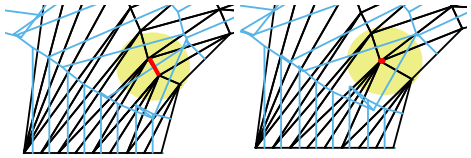
Figure 26 shows edge collapses for E_{BP}. Four collapses occurred in the first outer iteration. No collapses occurred in the second outer iteration, but a fifth collapse occurred immediately after flipping and starting the third outer iteration.

Figures 27 and 28 show progress through outer iterations. The outer iterations significantly improve the mesh near the bottom feet. We stopped after the seventh outer iteration because its optimal mesh was regular and did not need flips.

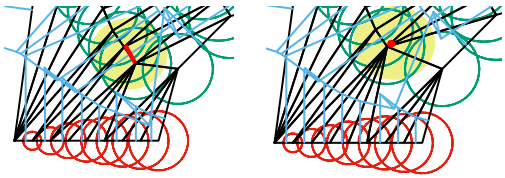
Figure 29 shows the final meshes for both E_{BP} and E_{BHs}. Each could be improved by collapsing the two 4-valent nodes in the center together; this is not caught by our simple collapse rule because the optimal E_{F} positions actually have the nodes farther apart.

6.6.4. Remove Patch Center

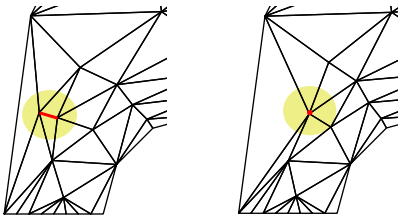
Patch Quality Test. We may remove a vertex if its patch has better quality with it removed. After optimizing the position or weight of a vertex, we calculate the mesh quality of its patch. We then triangulate the patch without the vertex. If its quality is better, we remove the vertex. Recall the triangulation will still contain the patch perimeter edges, so this change is purely local. (This follows from the property that a primal edge indicates intersecting power cells, and removing vertices can only



(a) Edge collapse when optimizing positions only.



(b) Another collapse when optimizing both positions and weights, similar to the position-only case.



(c) Edge collapse immediately after flipping. Only primal triangles shown for clarity.

Figure 26: High heat edge collapses for E_{BP}.

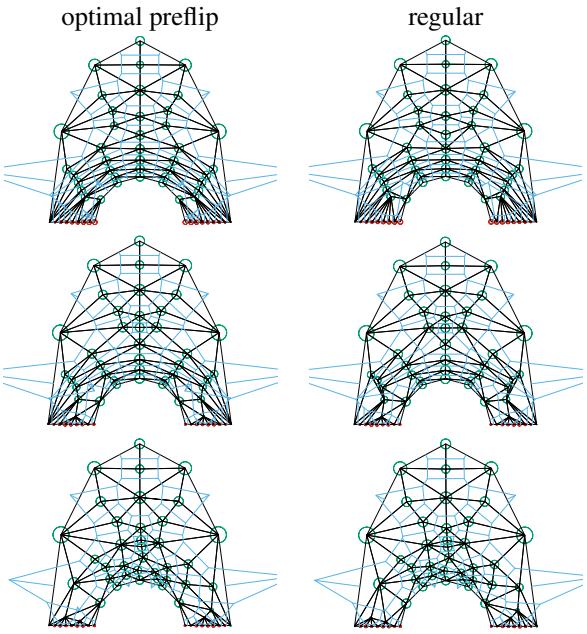


Figure 27: High heat outer iterations 0, 1, and 5 for E_{BP}. Weight circles 1/4 scale.

make the power cells bigger for the remaining vertices,

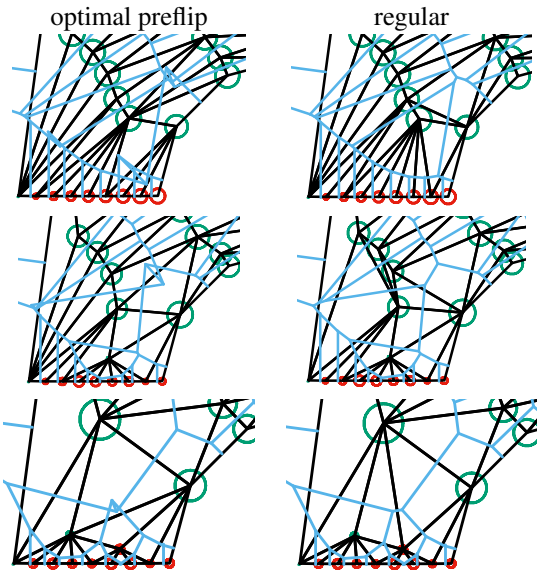


Figure 28: Zoom to show flips near the left foot in high heat outer iterations 0, 1, and 5 for E_{BP}.

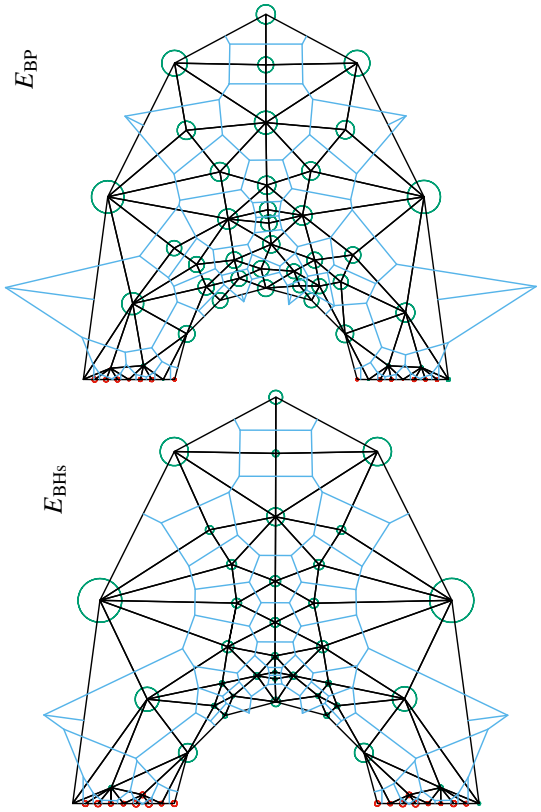


Figure 29: High heat optimal meshes with barriers, collapses, position, weights, and flips for E_{BP} and E_{BHs}. Weight circles scaled by 1/4 for clarity.

so their cells will still intersect.)

The patch quality test can be used independently, or with the short-edge collapse. It usually collapses the same vertices, plus many more. In experiments, vertices are removed both after position optimization, and after weight optimization.

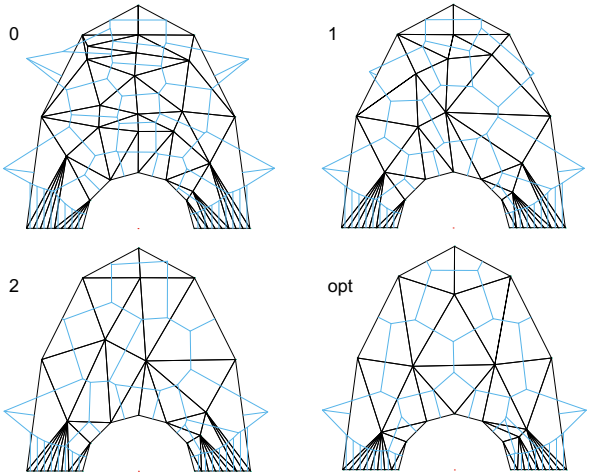


Figure 30: Vertices removed by the patch quality test, position-only E_{BP} optimization, by iteration count.

Figures 30 and 31 show optimizing the horseshoe with the patch quality test and E_{BP}, using immediate postion updates and flips to restore regularity. Coarse but high quality meshes result. Figure 30 optimizes only positions. Figure 31 optimizes positions and weights. Recall that with weight optimization, positions change less and fewer collapses are triggered by the explicit short-edge rule. The patch quality test triggers more collapses than the short-edge rule, and these collapses occur at later iterations than in the position-only case.

7. Conclusion

Recommendations. We recommend optimizing E_{BP} because it is stable in all combinations of optimizing weights, positions, flips, and collapses. We recommend allowing vertex removal because it improves stability and output quality for all combinations of other options. If you do not have the freedom to remove nodes, then only E_{BP} is robust. An exception to always using E_{BP} we suggest is $E_{\partial \mathrm{T}}$ when collapsing edges using the short-edge rule and optimizing positions with fixed weights, because collapses happen more easily than with E_{BP}. For removing vertices by the patch test, any metric can be used. We recommend optimizing weights only after a few position iterations. This is not

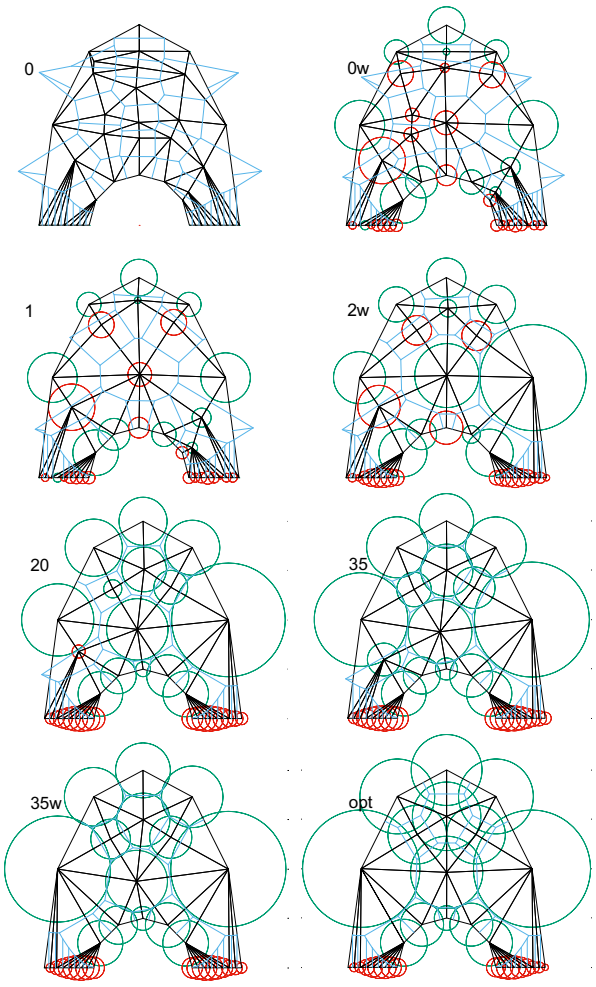


Figure 31: Vertices removed by the patch quality test, position and weight E_{BP} optimization. Weight circles to scale.

required for barrier energies, but it often allows more collapses, improves convergence speed and improves final quality. See fig. 32.

What Works. Using a barrier E_{BH} was sufficient to get a good mesh in all cases. We can easily combine barriers with vertex removal, especially by the patch test. Using theory-based extrapolation, E_{T}, together with collapses was sufficient to produce a good mesh. Using E_{F} or $E_{\partial \mathrm{T}}$ with collapses was sufficient to get a good mesh when optimizing positions only, but not when optimizing weights. Delaying weight optimization can affect the final mesh, but whether it is better or worse is problem dependent. It might be worth trying both immediate and deferred weight optimization and comparing results.

Our short-edge rule for edge collapses works for con-

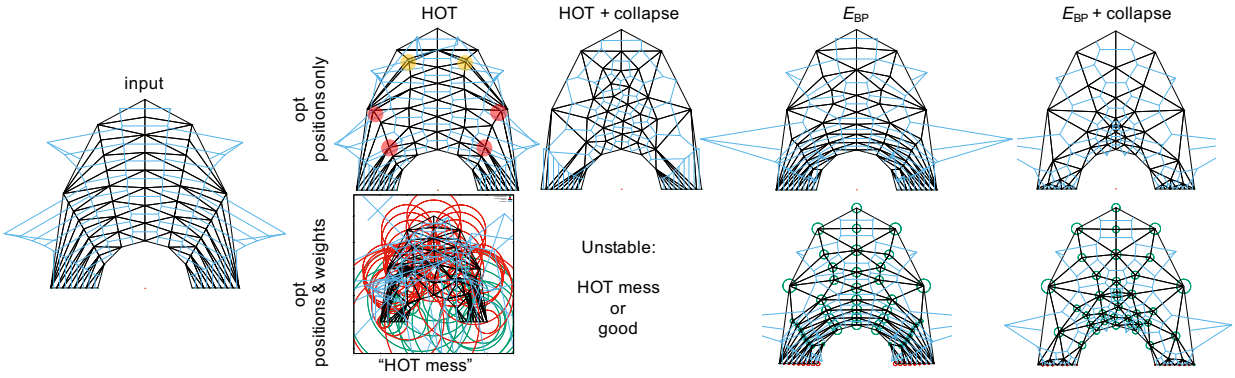


Figure 32: Summary. Optimizing HOT can collapse nodes, or produce a "HOT mess" with large-magnitude weights and inverted elements. Explicitly collapsing edges by removing vertices works when optimizing positions only, but can still produce a "HOT mess" when optimizing weights. The pseudo-barrier metric always works, and can be combined with discrete collapses.

figurations such as the highly non-convex patch in the sparse horseshoe. However, it misses some desirable collapses such as the 4-valent vertices in fig. 29. We have seen it fail to collapse small obtuse triangles and 3-valent vertices. The patch-quality test, comparing the patch energy with and without the center vertex, is successful at removing vertices in these cases. However, use caution for energies that depend on the number of elements.

What Does not Work. Optimizing positions with $\star^1\{E_{\mathrm{F}}, E_{\partial \mathrm{T}}, E_{\mathrm{T}}\}$ leads to nearly coincident points. Optimizing weights with $\{E_{\mathrm{F}}, E_{\partial \mathrm{T}}\}$ can be unstable and produce a HOT mess because of the presence of unbounded negative energies. In the prior literature the inputs considered were restricted: positions were preconditioned to start with well-spaced points, especially before optimizing weights, and domains were mostly convex. Perhaps these were to avoid the problems we described.

When optimizing, evaluating the quality of any position or weight change that makes the mesh non-regular is an extrapolation of the HOT energy. E.g. we can move a node anywhere and still evaluate the triangle quality metrics even if the mesh is now non-regular. But we cannot trust any of our metrics to be the same compared to actually performing the flip and evaluating the new quality. Using extrapolation, the metrics can get better and converge. But they can get worse, causing the optimization to go uphill, and leading to infinite cycles. It depends on the exact triangle shapes and weights.

In general, one might want to optimize something other than HOT; it is not the same as having well-centered orthocenters, because the restriction on $d_{ij} > 0$ is significant, as we have demonstrated. The best metric is likely application specific.

Further Exploration. The best way to intermix collapses, position optimization, weight optimization, and flips remains an open problem. One could combine different energy functions, such as optimizing $E_{\partial \mathrm{T}}$ for positions and collapses, then switching to E_{BP} when also optimizing weights and flips. The shifted barrier should be explored further. We believe parameter γ in ϕ should be chosen to be a small scale-invariant value, perhaps based on an edge length. It may be beneficial to adaptively shrink γ in E_{BP} or h_k^o in E_{BHs} towards zero as the mesh improves.

One could consider other criteria for discrete changes. It may be effective to use the HOT energy to insert nodes: e.g. the orthocenter of large-energy triangles. We do not know if the dual motivates connectivity changes. E.g. if optimizing HOT causes a d_{ij} to go negative, as sometimes happens for obtuse triangles, would the mesh be improved by flipping an edge or inserting a node to break up the obtuse triangle? What is the best place for that node: e.g. the circumcenter, orthocenter, or centroid? Vertices with large negative weights may be hidden when regularity is restored, so perhaps they should be removed explicitly.

The analysis showing that the barrier energy is a quasi-convex function of the weights needs to be extended. We wish to investigate the convexity (or lack thereof) of the energy, first with respect to the node positions, and second over a whole triangle. We have not studied $\star^0$ nor $\star^2$ to the extent of $\star^1$. We have not extended the barriers or collapses to 3D. We have not considered theoretical convergence rates nor practical efficiency. Our code is a research code but a production code may be of value to the community.

Open Problems. The open problem of a practical way of finding weights such that the regular triangulation of a given set of positions has a prescribed set of edges, or showing that no such weights exists, was recently solved [23]. It works because the constraints are linear and the problem size can be kept small by only adding violated constraints. We pose the open problem of finding weights such that all triangles have positive h_k and d_{ij}, if possible. Such a triangulation could be used as input to optimization of the hard barrier. (It is also open if we just require positive h_k. All positive d_{ij} is solved by weights = zero.)

Acknowledgements

We thank Prof. Mathieu Desbrun and Fernando de Goes for discussing their paper, Hodge stars, and optimal transport with us. We thank Darren Engwirda for discussing primal-dual meshes and their role in climate simulations. We thank Pavel Bochev for discussing mimetics and the role of primal-dual meshes in compatible discretizations.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Applied Mathematics Program. Sandia National Laboratories is a multi-program laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

[1] E. VanderZee, A. N. Hirani, D. Guoy, V. Zharnitsky, E. A. Ramos, Geometric and combinatorial properties of well-centered triangulations in three and higher dimensions, Computational Geometry 46 (6) (2013) 700 – 724.

[2] D. Yan, P. Wonka, Non-obtuse remeshing with centroidal voronoi tessellation, IEEE Transactions on Visualization and Computer Graphics 22 (9) (2016) 2136–2144. doi:10.1109/TVCG.2015.2505279.

[3] F. d. Goes, P. Memari, P. Mullen, M. Desbrun, Weighted triangulations for geometry processing, ACM Transactions on Graphics (TOG) 33 (3) (2014) 28.

[4] P. Mullen, P. Memari, F. de Goes, M. Desbrun, HOT: Hodge-optimized triangulations, ACM Transactions on Graphics (TOG) 30 (4) (2011) 103.

[5] A. N. Hirani, K. Kalyanaraman, E. B. VanderZee, Delaunay Hodge star, Computer-Aided Design 45 (2) (2013) 540 – 544, solid and Physical Modeling 2012.

[6] F. De Goes, P. Alliez, H. Owhadi, M. Desbrun, On the equilibrium of simplicial masonry structures, ACM Transactions on Graphics (TOG) 32 (4) (2013) 1–10.

[7] S. C. Mousley, M. Deakin, P. Knupp, S. A. Mitchell, Meshes optimized for discrete exterior calculus (DEC), Tech. Rep. SAND2018-2780O, Sandia National Laboratories, center for Computing Research Summer Proceedings 2017, A.D. Baczewski and M.L. Parks, eds. (2018).

[8] L. Chen, J. chao Xu, Optimal Delaunay triangulations, Journal of Computational Mathematics 22 (2) (2004) 299–308.

[9] D. Engwirda, Generalised primal-dual grids for unstructured co-volume schemes, Journal of Computational Physics 375 (2018) 155 – 176.

[10] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676.

[11] Q. Du, D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tesselations, Intl. J. Numer. Meth. Engr. 56 (2003) 1355–1373.

[12] Y. Liu, W. Wang, B. Lévy, F. Sun, D.-M. Yan, L. Lu, C. Yang, On centroidal Voronoi tessellation — energy smoothness and fast computation, ACM Transactions on Graphics 28 (4) (2009) 101:1–101:17.

[13] J. Räbinä, On a numerical solution of the Maxwell equations by discrete exterior calculus, Tech. Rep. 200, Univ. Jyväskylä (2014).

[14] P. B. Bochev, J. M. Hyman, Principles of mimetic discretizations of differential operators, in: D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, M. Shashkov (Eds.), Compatible Spatial Discretizations, Springer New York, New York, NY, 2006, pp. 89–119.

[15] T. Ringler, J. Thuburn, J. Klemp, W. Skamarock, A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids, Journal of Computational Physics 229 (9) (2010) 3065–3090.

[16] A. F. El Ouafdi, H. El Houari, D. Ziou, Adaptive estimation of Hodge star operator on simplicial surfaces, The Visual Computerdoi:10.1007/s00371-020-01879-5.

[17] R. Ayoub, A. Hamdouni, D. Razafindralandy, A new Hodge operator in discrete exterior calculus. application to fluid mechanics, arXiv preprint arXiv:2006.16930.

[18] M. S. Mohamed, A. N. Hirani, R. Samtaney, Comparison of discrete Hodge star operators for surfaces, Computer-Aided Design 78 (2016) 118 – 125, sPM 2016. doi:https://doi.org/10.1016/j.cad.2016.05.002.

[19] M. Yvinec, 2D triangulation, in: CGAL User and Reference Manual, 5.2 Edition, CGAL Editorial Board, 2020, https://doc.cgal.org/latest/Triangulation_2/index.html.

[20] L. Freitag, P. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, Intl. J. Numer. Meth. Engr. 53 (2002) 1377–1391.

[21] J. Tinoco-Ruiz, P. Barrera-Sánchez, Smooth and convex grid generation over general plane regions, Mathematics and Computers in Simulation 46 (2) (1998) 87–102.

[22] J. M. Escobar, E. Rodrıguez, R. Montenegro, G. Montero, J. M. González-Yuste, Simultaneous untangling and smoothing of tetrahedral meshes, Computer Methods in Applied Mechanics and Eng. 192 (25) (2003) 2775–2787.

[23] M. Alexa, Conforming weighted Delaunay triangulations, ACM Transactions on Graphics (TOG) 39 (6) (2020) 1–16.