Available online at www.sciencedirect.com

SciVerse ScienceDirect

International Meshing Roundtable 00 (2016) 000–000

SAND2016-9097C

Procedia Engineering

www.elsevier.com/locate/procedia

25th International Meshing Roundtable

Removing low-frequency artefacts from topology-optimized surfaces

Salvatore S. Elder[a], William R. Quadros[b,*]

[a]*School of Applied and Engineering Physics, Cornell University, Ithaca, NY 14853, USA*
[b]*Sandia National Labs,[†] P O Box 5800, MS 0897, Albuquerque, NM 87185, USA*

Abstract

Topology optimization is an emerging technology capable of producing organic shapes subject to various physical objectives. One of the undesirable features present in the resulting triangle meshes is low-frequency surface rippling. Here we propose a method in which the ripple frequencies are estimated based on geometric mesh characteristics. The ripples can then be removed by direct projection to and from an orthogonal frequency basis derived from the discrete Laplace-Beltrami operator.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: topology, optimization, smoothing, low-frequency, ripples, Laplacian

1. Introduction

When a topology optimization problem is solved, the shape of the object is not known in advance. In order to explore the full space of possible topologies, therefore, the shape must be expressed in a fully general way. The usual representation is a scalar density field ρ, calculated on the nodal points of a volume mesh. Where $\rho = 1$, there is to be material. Where $\rho = 0$, there is not. Intermediate values are allowed so that the optimization can be formulated as a continuous problem [1]. When it comes time to manufacture the part, however, the densities must be binary: either a 3d printer *should* place material at a point, or it *should not.* Before working with the optimized shape, therefore, the analyst or manufacturer must first extract the boundary surface. Quadros et al. [2] describe a procedure for extracting, refining, and smoothing the $\rho = 0.5$ isosurface as a triangle mesh.

Topology optimization codes penalize intermediate values of ρ, so that there tends to be a sharp boundary between high and low material density [3]. Although a sharp boundary is desired, we believe that the penalization is responsible for the low-frequency artefacts observed in the extracted triangle meshes. If nodal density values tend to snap toward 0 and 1, then the isosurface will tend to cross through the midpoints of the volume elements' edges, even if the ideal surface does not lie here. This effect is illustrated in Fig. 1. If the volume elements are relatively coarse, then these deflections will be significant. It may be possible to modify the topology optimization scheme itself to alleviate these

[†] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000
* Corresponding author. Email: wrquadr@sandia.gov; Tel.: +1-505-220-9458.

Peer-review under responsibility of the organizing committee of IMR 25.

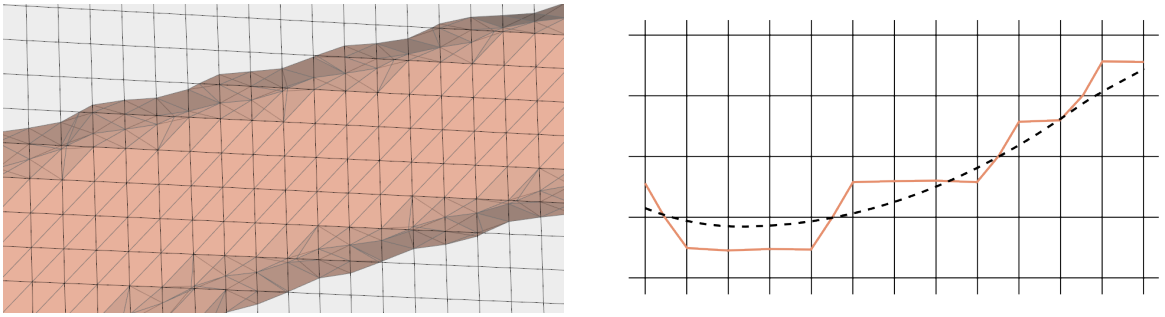

Fig. 1. **Left:** Close-up of an actual isosurface extraction of a topology-optimized bracket. The hex elements on which the density field was calculated are overlaid. Note the ripples which occur with about a four-element wavelength. **Right:** Exaggerated illustration of the way in which penalizing intermediate values of ρ may exaggerate or produce ripples on top of the ideal topology. If the values at the element corners (nodes) are nearly 0 or 1, then the $\rho = 0.5$ isosurface will tend to pass through the midpoints of the inter-element boundary edges. This is equally true in three dimensions.

errors. Increasing the mesh resolution used by the optimizer may also help. In this note, however, we describe a method to remove the ripples "downstream" in the workflow, once they have already been introduced into a triangle mesh.

A single undulation may occur over several volume elements; the ripple error, therefore, extends into the *low-frequency* end of the Fourier spectrum. Typical mesh smoothing algorithms remove high-frequency noise while preserving low-frequency shape components [4, 6–8]. This is the desired behavior in applications such as MRI and 3d scanning, since the errors associated with measured data are presumably *random noise.* In our application, however, the error associated with the surface ripples is neither *random*, nor is it *noise* in the traditional sense. Randomly distributed noise can easily be "averaged out" by Gaussian and other smoothing schemes, but sufficiently wide ripples *are smooth* even though they are undesired.

To explain why existing methods fall short of the desired output, and to explain our proposed approach, it is necessary to define the notion of frequency for an arbitrary triangle mesh.

2. Frequency formulation

A Fourier series on the Cartesian plane is nothing more than the projection of a function into a special function basis. These are complex exponentials, $\exp[\sqrt{-1}(\xi x + \eta y)]$. The reason that these functions are special is that they are eigenfunctions of the Laplacian, $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, subject to periodicity constraints. The eigenvalue of each of these signals is $-\xi^2 - \eta^2$, which is a function of the spatial frequency (ξ, η).

There are several discrete extensions of the Laplacian to graphs and to surface meshes [5]. Suppose the mesh vertices are indexed by $V = \{1, 2, \ldots, N\}$. Also suppose that the value of an arbitrary graph signal $\mathbf{g}$ at each node i is g_i. Let us consider only *linear* graph Laplacians, $\mathbf{L}$. That is, we assume that $\mathbf{L}$ can be described by constant coefficients $L_{ij},\ i, j = 1, 2, \ldots, N$ such that the value of $\mathbf{Lg}$ at any node i is given by $(\mathbf{Lg})_i = \sum_{j=1}^{N} L_{ij}\, g_j$. Therefore, it is natural to think of signals $\mathbf{g}$ as coulmn vectors, and the Laplacian as a matrix.

The critical leap made by Taubin and others [6–10] in decomposing a signal into frequency components is that the eigenvectors of a suitable graph Laplacian will correspond to oscillatory modes. Furthermore, the greater the eigenvalue of each eigenmode, the greater its spatial frequency. Conceptually, the graph Laplacian helps us to smooth signals defined on the mesh *topology*. In our case, the signal to be smoothed is the *geometry* itself. If these eigenpairs are $\mathbf{v}_k, \lambda_k,\ k = 1, 2, \ldots, N$, then the Fourier decomposition of a mesh geometry $\mathbf{R} = [\mathbf{x}\ \mathbf{y}\ \mathbf{z}]$ can be written as

$$\mathbf{x} = \sum_{k=1}^{N} \hat{x}_k\, \mathbf{v}_k, \tag{1}$$

with similar expressions for $\mathbf{y}$ and $\mathbf{z}$. Again, the ith components of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ give the coordinates of the ith vertex, (x_i, y_i, z_i).

Fig. 2. **Left:** a full view of the isosurface shown up-close in Fig. 1. **Right:** The result of removing medium-low frequencies with an implicit method extending the approach of Taubin. The ripples are removed, but so is meaningful geometry! It is very difficult to design polynomial filters that *selectively* remove narrow frequency bands.

Even if we do not explicitly know $\mathbf{v}_k$ and $\hat{x}_k$, we do know that the decomposition in Eq. (1) *exists*. We can also compute $(a\mathbf{I} + b\mathbf{L})\,\mathbf{x}$ in-place, without an explicit matrix–vector representation in code. ($\mathbf{I}$ is the identity operator.) Doing so automatically filters $\mathbf{x}$ by its eigenmode components:

$$(a\mathbf{I} + b\mathbf{L})\,\mathbf{x} = \sum_{k=1}^{N} (a + b\lambda_k)\,\hat{x}_k\,\mathbf{v}_k. \tag{2}$$

In [7] and [8], this step is repeated with different a, b at each iteration in order to implement *polynomial transfer functions* on mesh frequency components, without ever calculating any eigenmodes!

Kim and Rossignac [9] use a slightly different approach, in which $\mathbf{L}^{-1}$ is calculated and used to extend the domain of possible transfer functions to include *rational functions* as well.

A simple low-pass filter $f_{\text{lp}}(\lambda)$ is introduced in [7]. We have experimented with a transfer function of the form $f_{\text{lp}}(\lambda)(\lambda - r)^2$, which is meant to remove ripple frequencies near $\lambda = r$. Polynomial (and even rational) transfer functions, however, are difficult to control when a sharp and narrow band-stop region is desired. Distorted results of this experiment are shown in Fig. 2. Although this filter shows some promise, we believe this and similar implicit methods are probably unsuitable for our application. Thus we turn to calculating $\mathbf{v}_k$ and $\hat{x}_k$ explicitly. This will afford us total freedom in filter design.

3. Explicit frequency filtering

Although the oscillatory modes were identified as eigenmodes of a matrix, several authors [6–9] avoided explicit calculation of these eigenmodes because the dimension of the Laplacian matrix was simply too large even for reasonably sized meshes. In 2008, however, Vallet and Lévy [10] made it possible to efficiently calculate and filter eigenmodes of meshes of up to a million vertices. In this approach, a large subset of $\mathbf{v}_k$ and $\hat{x}_k$ are known and can be inverted to reconstruct $\mathbf{R}$. The primary advantage of taking the extra processing power to perform these calculations is that *arbitrary* filters can be accomplished. Once the vector of $\hat{x}_k$ components has been obtained, one can loop through its entries and modify them in any way one desires, then invert to obtain the filtered mesh geometry. This approach admits low-pass filters to remove noise as a special case, but it also allows for the targeted removal of ripples, which is a more difficult problem.

The framework of [6] chooses a discretization of the Laplacian based on cotangent weights. As the triangle sizes shrink, this Laplacian is proven in [5] to converge to the true Laplace-Beltrami operator defined on manifold surfaces. Then the eigenvalues are related to the true geometric frequencies of their respective eigenmodes. A symmetric version of this operator is used by [10] to ensure orthogonality of its eigenvectors.

3.1. Filter determination

It would be desirable to select parameters of the frequency filter automatically. The *spatial* properties of the ripples are almost certainly more difficult to estimate directly than their *geometric* properties, however. In a prior work [2]

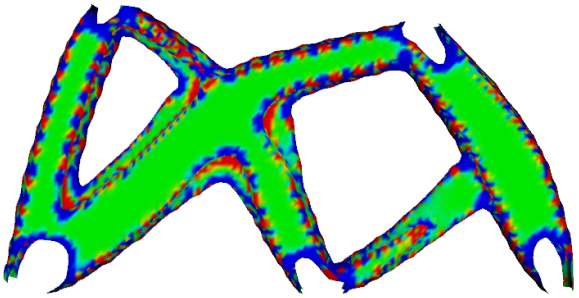

Fig. 3. The Gaussian curvature of a topology optimized surface without smoothing. Red regions correspond to positive curvature (hills and valleys), blue regions to negative curvature (saddle points), and green regions to relatively flat areas (curvature ≈ 0).

we experimented with an ad-hoc method for removing surface ripples based on Gaussian curvature. In the process, we discovered the Gaussian curvature clearly highlights the ripples, as shown in Fig. 3. This has the effect of reducing the dimension of the problem by two (since vertex curvature is a scalar and vertex position is a vector). In particular, vertices can be classified as *bumps* or not bumps using a positive curvature tolerance. These high-curvature bumps are displayed in red in Fig. 3. (Hills and valleys both correspond to positive curvature, but we have mostly observed hills in our experience with topology-optimized surfaces.) The relationship between geometrical size and eigenvalue is explored in [9], although the authors focus on features which can be approximated by ellipsoids. This work could be extended to detect the characteristic distance between nearby bumps and estimate the eigenvalue of the associated oscillations.

We suggest another approach to eigenvalue estimation based on curvature, which would proceed as follows: Construct a signal vector $\mathbf{c}$, where $c_i = 1$ for i that belong to bumps, and $c_i = 0$ for i that do not. The spectrum of $\mathbf{c}$ can then be used to design the filter for $\mathbf{R}$. The soundness of this approach can be justified by two properties of the problem. First, mesh spectra are linear:

$$\sum_{k=1}^{N} (\hat{x}_k + \hat{x}'_k)\, \mathbf{v}_k = \mathbf{x} + \mathbf{x}'. \tag{3}$$

Second, even if the amplitude and overtones of $\mathbf{c}$ differ from those of the ripples, they will almost certainly have the same principal modal component. (Analogously, a violin and a tuning fork sounding an A4 share a fundamental frequency of 440 Hz.) Then the principal mode and bandwidth can be estimated from the spectrum of $\mathbf{c}$ and used to filter $\mathbf{R}$. The Results section also makes this approach plausible, albeit indirectly.

4. Results

We use Python (`http://www.python.org`) and SciPy (`http://www.scipy.org`) for these experiments. Although we do not use the algorithm of [10] to compute eigenmodes, we use the same Laplacian (up to a constant factor, which merely scales the eigenvalues); in principle, the only major difference is in implementation. After creating a triangulation on a grid of 225 points, we calculated the eigenmodes and associated eigenvalues of the Laplacian with cotangent weights as used in [10]. We created a cubic Bézier patch to represent a portion of a topology optimized surface, and superimposed a sinusoid on top of it with wavelength 0.2.[1] Figure 4 shows the result of removing all eigenmode components for which $0.5 \leq \lambda \leq 0.57$. Also shown is the spectrum of the ripples alone. Note the prominent peak shared by both spectra; this clear correspondence helps to justify the curvature approach described in Section 3.1.

[1] For simplicity, the Laplacian weights were calculated for $z = 0$, although in practice they would be calculated on top of the input geometry.

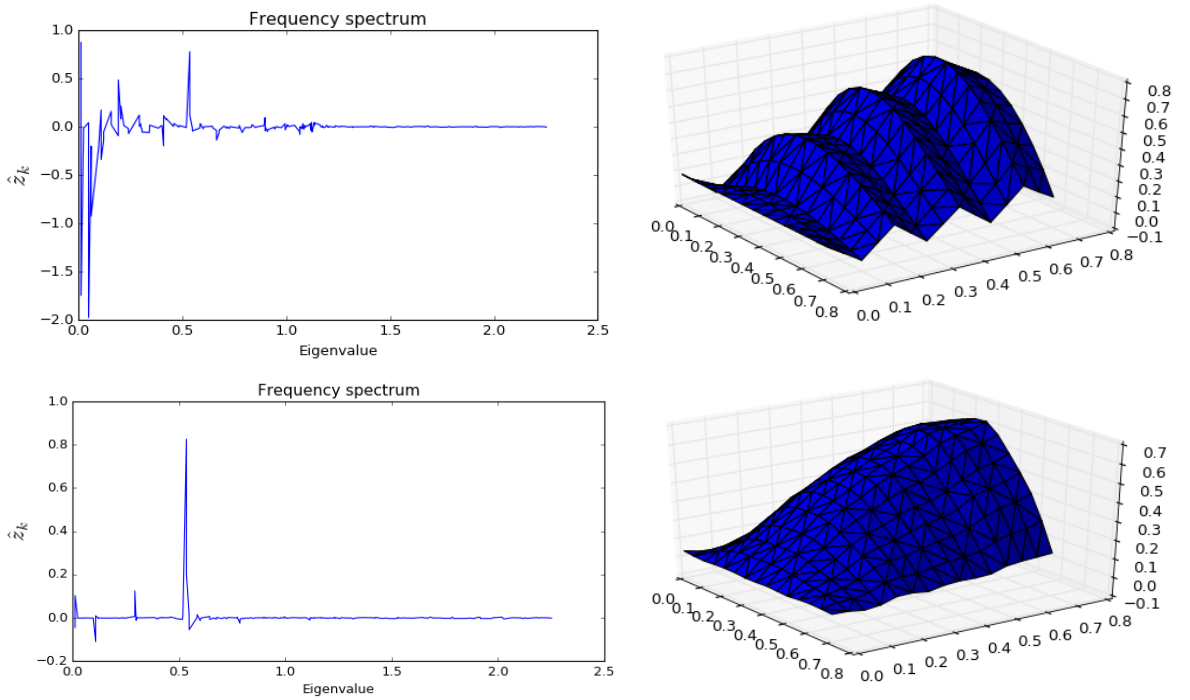

Fig. 4. **Top-right:** The mesh to be smoothed. We added the prominent ripples artificially. **Top-left**: The spectrum of the mesh. **Bottom-left:** The spectrum of the ripples we added. Note the prominent peak shared by both spectra. **Bottom-right:** The result of removing the aforementioned peak from the spectrum of the mesh, then inverting the frequency transform. The ripples have been removed, leaving us with a smooth mesh.

5. Conclusions

In this note we have identified systematic errors present in topology optimized surfaces, and explored a processing pipeline for removing them. We argue that typical, localized smoothing methods are for *fundamental reasons* unable to remove these low-frequency artefacts. We suggest the use of curvature as a means of locating the errors in frequency space, and remove undesired ripples from an example mesh. Future work would bring the ideas explored in this note to full maturity, with frequency detection and smoothing implemented for topology-optimized surfaces.

References

[1] Bendsøe MP. Optimal shape design as a material distribution problem. *Struct optimization* 1989;1(4):193–202.

[2] Quadros WR, Elder SS, Robbins J, Clark B. High-fidelity isosurface extraction for additive manufacturing. In *Solid Freeform Fabrication Symposium*, vol. 27, 2016.(accepted June 2016)

[3] Sigmund O, Petersson J. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct optimization* 1998;16(1):68–75.

[4] Belyaev A, Ohtake Y. A comparison of mesh smoothing methods. In *Israel-Korea Bi-national conference on geometric modeling and computer graphics* vol. 2, 2003.

[5] Xu G. Convergence of discrete Laplace-Beltrami operators over surfaces. *Comput Math Appl* 2004;48(3):347–360.

[6] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.

[7] Taubin G. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 351-358. ACM, 1995.

[8] Taubin G, Zhang T, Golub G. Optimal surface smoothing as filter design. In *European Conference on Computer Vision*, pp. 283–292. Springer Berlin Heidelberg, 1996.

[9] Kim BM, Rossignac J. Geofilter: Geometric selection of mesh filter parameters. In *Computer Graphics Forum*, vol. 24, no. 3, pp. 295–302. Blackwell Publishing, Inc, 2005.

[10] Vallet B, Lévy B. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, vol. 27, no. 2, pp. 251–260. Blackwell Publishing Ltd, 2008.