March 2003

A Local Convergence Proof for the Minvar Algorithm for Computing Continuous Piecewise Linear Approximations

Richard E. Groff
University of Michigan

Pramod P. Khargonekar
University of Florida

Daniel E. Koditschek
University of Pennsylvania, kod@seas.upenn.edu

Reprinted from *SIAM Journal on Numerical Analysis*, Volume 41, Issue 3, 2003, pages 87-93.
Publisher URL: http://dx.doi.org/10.1137/S0036142902402213

NOTE: At the time of publication the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

A Local Convergence Proof for the Minvar Algorithm for Computing Continuous Piecewise Linear Approximations

Abstract

The class of continuous piecewise linear (PL) functions represents a useful family of approximants because invertibility can be readily imposed, and if a PL function is invertible, then it can be inverted in closed form. Many applications, arising, for example, in control systems and robotics, involve the simultaneous construction of a forward and inverse system model from data. Most approximation techniques require that separate forward and inverse models be trained, whereas an invertible continuous PL affords, simultaneously, the forward and inverse system model in a single representation. The minvar algorithm computes a continuous PL approximation to data. Local convergence of minvar is proven for the case when the data generating function is itself a PL function and available directly rather than through data.

Keywords

piecewise linear, invertible approximation, moving mesh, triangulation

Comments

Reprinted from *SIAM Journal on Numerical Analysis*, Volume 41, Issue 3, 2003, pages 87-93.
Publisher URL: http://dx.doi.org/10.1137/S0036142902402213

NOTE: At the time of publication the author, Daniel Koditschek, was affiliated with the University of Michigan. Currently, he is a faculty member of the School of Engineering at the University of Pennsylvania.

SIAM J. NUMER. ANAL.
Vol. 41, No. 3, pp. 983–1007

A LOCAL CONVERGENCE PROOF FOR THE MINVAR ALGORITHM FOR COMPUTING CONTINUOUS PIECEWISE LINEAR APPROXIMATIONS*

RICHARD E. GROFF†, PRAMOD P. KHARGONEKAR†‡, AND DANIEL E. KODITSCHEK†

Abstract. The class of continuous piecewise linear (PL) functions represents a useful family of approximants because invertibility can be readily imposed, and if a PL function is invertible, then it can be inverted in closed form. Many applications, arising, for example, in control systems and robotics, involve the simultaneous construction of a forward and inverse system model from data. Most approximation techniques require that separate forward and inverse models be trained, whereas an invertible continuous PL affords, simultaneously, the forward and inverse system model in a single representation. The `minvar` algorithm computes a continuous PL approximation to data. Local convergence of `minvar` is proven for the case when the data generating function is itself a PL function and available directly rather than through data.

Key words. piecewise linear, invertible approximation, moving mesh, triangulation

AMS subject classifications. 41-04, 41A30, 65D15

DOI. 10.1137/S0036142902402213

1. Introduction.

In this paper, we present `minvar`, a novel algorithm for computing continuous multidimensional piecewise linear (PL) approximations to data. The algorithm takes advantage of the structure of PL functions to provide a computationally effective approximation technique. This paper provides a local convergence proof for the special case when the data generating function is itself PL and is available directly rather than through discrete data.

Our interest in the PL family is driven by applications that require approximation of both forward and inverse functions from data. In xerography, for example, the print engine's color space transformation is required to stabilize color reproduction, while its inverse is required to generate printer specific color mixture commands in response to inputs expressed in device independent color coordinates [21, 23]. The field of robotics is rife with examples where changes of coordinates play a key role: in mobile robot navigation [27, 32, 33]; in the representation of gaits [31, 34]; in sensor based manipulation [8]; as well as in calibration [42]. Since a change of coordinates is a continuous and continuously invertible function, building a custom change of coordinates amounts to a search for the appropriate forward and inverse function. Representations of scalar invertible functions are required for certain machine tool calibration problems [24], for certain automobile fuel control settings [17], as well as for probability density estimation [15]. In all such settings, most approximation techniques require the construction of distinct forward and inverse representations, because the approximations are not invertible in closed form. In addition to doubling the effective training effort, accuracy suffers since the approximation of the inverse is not exactly the inverse of the forward approximation. In contrast, invertibility of PL

*Received by the editors February 6, 2002; accepted for publication (in revised form) November 27, 2002; published electronically June 18, 2003. This work was supported in part by NSF Award ECS-96322801.

http://www.siam.org/journals/sinum/41-3/40221.html

†EECS Department, 1101 Beal Ave., University of Michigan, Ann Arbor, MI 48109 (regroff@umich.edu, kod@umich.edu).

‡Current address: ECE Department, University of Florida, 300 Weil Hall, P.O. Box 116550, Gainesville, FL 32611 (ppk@ufl.edu).

functions can be verified and even imposed geometrically, that is, by well characterized and computationally effective techniques arising from geometric insights. Moreover, if a PL function is invertible, it can be inverted in closed form. Thus a single PL approximation is ideal for applications requiring the approximation of a function and its inverse.

A substantial amount of mathematical literature on real function approximation (see, for example, [6, 9, 29]), largely concerned with linear-in-parameters techniques, deals extensively with algorithms, fundamental limits, convergence rates, and families of bases in approximating functions. Recent activity has been spurred by evidence that nonlinear-in-parameters function families offer improved approximation rates in higher dimensions as compared to linear-in-parameters representations [3]. Recently, approximation methods that employ collections of local approximations have received increasing attention [1, 16, 38]. However, very little of this linear- or nonlinear-in-parameters literature addresses the problem of function approximation under the constraint of invertibility.

PL functions have been addressed in a number of different settings. Algebraic topologists used PL homeomorphisms to classify topological spaces [36] but did not address computational considerations. The study of splines, piecewise polynomials with continuity and smoothness constraints, includes PL functions [7, 10, 35]. Splines are typically extended to multiple dimensions by means of tensor products. The domain partition is then a tensor product of partitions of the individual dimensions and the approximant is the sum of tensor products of scalar spline functions. A multidimensional linear spline is then multilinear, that is, linear in each variable separately, rather than truly linear. General splines enjoy no invertibility properties. Moreover, most of the spline literature assumes the domain partition to be fixed, in which case approximation of the best L_2 spline is a linear-in-parameters problem. Allowing the partition to change introduces a nonlinear-in-parameters problem. The multivariate adaptive regression spline (MARS) literature admits a limited nonlinear parameterization by allowing the basis to adapt but does not allow general motion of the domain partition [16, 38]. The piecewise polynomial literature addresses the problem of finding (possibly discontinuous) piecewise polynomial approximations to an explicitly known scalar function. In this setting, the domain partition is considered as part of the approximation's parameterization. For scalar functions, there are results for the existence of a best approximation by possibly discontinuous piecewise polynomials under certain generalized convexity conditions [4, 18]. Algorithms similar in flavor to the scalar specialization of `minvar` were introduced in [2, 25, 26]. A treatment of discontinuous piecewise polynomial approximations on two-dimensional triangulations is provided in [39]. Also, [40] provides an algorithm for a moving mesh finite element solution to variational problems. A specialization of this moving mesh algorithm is finding the best L^p, p finite and even, continuous piecewise polynomial approximation to a function. Both of these algorithms [39, 40], as well as the piecewise polynomial literature in general [2, 25, 26], assume that the function to be approximated is available directly, and the algorithms entail steps, such as root finding, that incorporate the function intrinsically. In contrast, the `minvar` algorithm is defined for arbitrary (finite) dimension and can either use a finite set of data or directly use the function to be approximated.

Motivated by applications that require the approximation of invertible functions, we have developed the `minvar` algorithm for computing PL approximations to a set of discrete data. In the context of these applications, PL approximations offer the

substantial benefit of closed form invertibility. When the domain partition is fixed, computing the best PL approximation is a linear-in-parameters problem that can be solved using classical techniques. Treating the partition as a component of the approximation's parameterization gives a much more powerful approximant, at the cost of entering the nonlinear-in-parameters problem domain. In nonlinear-in-parameters problems, one can generally expect only local, as opposed to global, convergence properties. Moving the domain partition of a PL function, or triangulation, as formally defined in the next section, has an added difficulty. A triangulation has both continuous and combinatorial parameters that interact in complex ways. Not all combinations of continuous and combinatorial parameters yield a proper triangulation. Triangulations in two and three dimensions have been studied extensively in the computational geometry literature [13, 30], but results for general dimension are more scarce, notwithstanding significant recent progress [5, 14, 28]. The price of using a family of finitely parameterized homeomorphisms, the PL approximations, is the cost of managing the combinatorial complexities of PL functions.

This paper is divided into five main sections. Section 2 provides a careful definition of the concept of a triangulation, relating it to the parameterization of PL functions. Section 3 introduces and defines the `minvar` algorithm. Section 4 provides a local convergence proof for the `minvar` algorithm when the data generating function is piecewise linear. Section 5 presents a numerical example.

2. Triangulations and PL functions. The ability to check invertibility of a PL function, and to invert it in closed form, derives from the interplay between the PL function's combinatorial and continuous parameters. This interplay provides much power but also creates potential pitfalls. For example, changing the continuous parameters inappropriately with respect to the combinatorial structure can cause "tangles" in the domain partition. Triangulations in general dimension, the key concept in understanding PL functions, are still an area of active research in computational geometry. While the `minvar` algorithm can be stated using only an intuitive notion of triangulation, further analytical insight, such as the local convergence proof provided in section 4, is limited without a much more careful definition. This section provides definitions of triangulations and PL functions to facilitate the exposition. For further background, see [41] for an introduction to concepts in convexity and [13] for an introduction to the geometric concept of triangulations.

2.1. Simplices. An *affine subspace* $V \subseteq \mathbb{R}^d$ is a linear subspace $L \subseteq \mathbb{R}^d$ translated by some $x_o \in \mathbb{R}^d$, i.e., $V = L + x_o$. The dimension of V is $\dim(V) := \dim(L)$. The *affine hull* of a set $\mathcal{U} \subseteq \mathbb{R}^d$, $\mathrm{aff}(\mathcal{U})$, is the smallest affine subspace containing $\mathcal{U}$. A finite set of points $\mathcal{U} \subseteq \mathbb{R}^d$ is *affinely independent* if for $i = 1, \ldots, d$, no affine subspace of dimension i contains more than $i+1$ points from $\mathcal{U}$. The *convex hull* of a set $\mathcal{U} \subseteq \mathbb{R}^d$, $\mathrm{conv}(\mathcal{U})$, is the smallest convex set containing $\mathcal{U}$. A *simplex*, s, is the convex hull of a (finite) set $\mathcal{V} \subseteq \mathbb{R}^d$ of affinely independent points, $s = \mathrm{conv}(\mathcal{V})$. The set of the extreme points, or *vertices*, of s, $\mathcal{V} = \mathrm{vert}(s)$, uniquely defines s.[1] The dimension of s is $l = \dim(s) = \dim(\mathrm{aff}(s)) = \mathrm{card}\,(\mathcal{V}) - 1$, where $\mathrm{card}\,(\mathcal{V})$ is the cardinality of $\mathcal{V}$, and s is called an l-simplex. There can be at most $d+1$ affinely independent points in $\mathbb{R}^d$, and thus there are simplices of dimension $-1, 0, 1, \ldots, d$, where by convention $\emptyset$ is

[1] vert is a pseudoinverse of conv, not a true inverse, since if $\mathcal{U} \subseteq s$, then $s = \mathrm{conv}(\mathcal{U} \cup \mathrm{vert}(s))$. The concept of vertices of a simplex is a special case of the more general notion of extreme points of a convex body. The Krein–Milman theorem [41] states that any convex compact set in $\mathbb{R}^d$ is the convex hull of its extreme points.

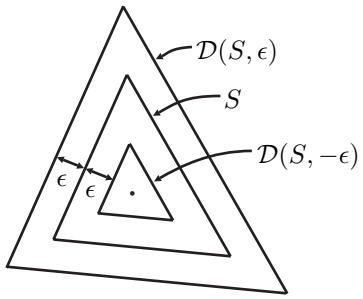

FIG. 2.1. *The 2-simplex S with its ϵ and $-\epsilon$ dilations. The point at the center is where dilation degenerates to a single point.*

considered a simplex with $\dim(\emptyset) := -1$. We may apply a partial order to simplices. Given two simplices s_1, s_2, we say that $s_1 \leq s_2$ if and only if $\text{vert}(s_1) \subseteq \text{vert}(s_2)$, in which case we call s_1 a face of s_2. If $\dim(s_2) = d$ and $\dim(s_1) = d-1$, then we also call s_1 a *facet* of s_2. In this paper we will predominantly be interested in d-simplices, so we adopt the convention that a capital S indicates a d-simplex, while a lowercase s denotes a simplex of any dimension.

Let S be a d-simplex with $\text{vert}(S) = \{p_1, \ldots, p_{d+1}\}$. The ϵ dilation of S, written $\mathcal{D}(S,\epsilon)$, is defined as

$$\mathcal{D}(S,\epsilon) := \left\{ x = \sum_{j=1}^{d+1} \alpha_j p_j \,\middle|\, \sum_{j=1}^{d+1} \alpha_j = 1 \;\forall j, \alpha_j \geq \frac{-\epsilon}{\delta(p_j, \text{aff}(\text{vert}(S) - \{p_j\}))} \right\}, \tag{2.1}$$

where $\delta(p,\mathcal{U})$ is the distance from point p to the nonempty set $\mathcal{U}$. Figure 2.1 illustrates the dilation of a 2-simplex. $\mathcal{D}(S,\epsilon)$ is well defined for

$$\epsilon \geq -\left(\sum_{j=1}^{d+1} 1/\delta(p_j, \text{aff}(\text{vert}(S) - \{p_j\})) \right)^{-1}.$$

When equality holds, $\mathcal{D}(S,\epsilon)$ is a single point; otherwise it is a d-simplex with facets parallel to S_i, but distance $|\epsilon|$ away, with $S \subseteq \mathcal{D}(S,\epsilon)$ for $\epsilon > 0$, and $\mathcal{D}(S,\epsilon) \subseteq S$ for $\epsilon < 0$. (See Claim 3 in Appendix A for the dilation's properties.)

2.2. Triangulation. An *abstract simplicial complex* is a collection of finite sets $\mathcal{S}$ satisfying the following: if $\alpha \in \mathcal{S}$ and $\beta \subseteq \alpha$, then $\beta \in \mathcal{S}$. The *vertex set* of an abstract simplicial complex is the set $\left\{x \middle| x \in \alpha, \alpha \in \mathcal{S}\right\}$.

A *geometric simplicial complex* is a collection $\mathcal{K}$ of simplices in $\mathbb{R}^d$ satisfying

1. $s_1 \in \mathcal{K}$ and $s_2 \leq s_1 \implies s_2 \in \mathcal{K}$,
2. $s_1, s_2 \in \mathcal{K} \implies s_1 \cap s_2 \leq s_1, s_2$.

The *vertex set* of a geometric simplicial complex is $\text{vert}(\mathcal{K}) := \bigcup_{s\in\mathcal{K}} \text{vert}(s)$. The *underlying space* of a geometric simplicial complex is $|\mathcal{K}| := \bigcup_{s\in\mathcal{K}} s$.

A *subcomplex* is a subset of a simplicial complex that is itself a simplicial complex. The *closure* of a subset $\mathcal{L} \subseteq \mathcal{K}$ is the smallest subcomplex that contains $\mathcal{L}$,

$$\text{Cl}\,\mathcal{L} := \left\{\alpha \in \mathcal{K} \middle| \alpha \leq \beta, \beta \in \mathcal{L}\right\}.$$

The *star* of a simplex s is the set of all simplices that contain s,

$$\text{St}\, s := \left\{s' \in \mathcal{K} \middle| s \leq s'\right\}.$$

The star is not in general a subcomplex.

We can parameterize[2] a geometric simplicial complex $\mathcal{K}$ in $\mathbb{R}^d$ by the pair $(P, \mathcal{S})$, where P is an indexed set of n unique points in $\mathbb{R}^d$,

$$P = \{p_1, \quad p_2, \quad \ldots, \quad p_n\},$$

and $\mathcal{S}$ is an abstract simplicial complex with vertex set $\{1, 2, \ldots, n\}$. Let[3]

$$\mathcal{K}(P, \mathcal{S}) = \{\text{conv}(P(\alpha)) | \alpha \in \mathcal{S}\}.$$

$\mathcal{K}(P, \mathcal{S})$ is a geometric simplicial complex if

1. for all $\alpha \in \mathcal{S}$, the points in $P(\alpha)$ are affinely independent,
2. $s_1, s_2 \in \mathcal{K}(P, \mathcal{S}) \Longrightarrow s_1 \cap s_2 \leq s_1, s_2$,

and, moreover, if these properties hold, then $\text{vert}(\mathcal{K}(P, \mathcal{S})) = P$. Proofs of these properties are provided in [20].

A *triangulation*[4] $\mathcal{T}$ is a geometric simplicial complex in $\mathbb{R}^d$ for which the underlying space is a k-manifold with boundary. Since a triangulation is a type of geometric simplicial complex, it can be parameterized in the same manner. We write $\mathcal{T}(P, \mathcal{S}) := \mathcal{K}(P, \mathcal{S})$ to indicate that the resulting geometric simplicial complex generated by the pair $(P, \mathcal{S})$ is a triangulation.

In this paper, we will deal only with triangulations which are d-manifolds with boundary that have a simply connected underlying space.

For notational convenience, we assume that the triangulation $\mathcal{T} = \mathcal{K}(P, \mathcal{S})$ has N d-simplices that have been indexed and named S_i, $i = 1, \ldots, N$. Let $S_i, S_j \in \mathcal{T}$. We then define

$$d_{i,j} := \dim(S_i \cap S_j) = \text{card}\,(\text{vert}(S_i \cap S_j)) - 1,$$

the dimension of the face shared by S_i and S_j. Let N_i be the number of d-simplices in $\text{St}\{p_i\}$,

$$N_i = \sum_{S_j \in \text{St}\, p_i} 1.$$

2.3. PL functions. A continuous PL function $f_{\mathcal{P}} : D \subseteq \mathbb{R}^d \to \mathbb{R}^d$ is parameterized by a triplet $\mathcal{P} = (P, Q, \mathcal{S})$. P is an indexed set of n points in the domain and Q is an indexed set of n points in the codomain,

$$P = \{p_1, \quad p_2, \quad \ldots, \quad p_n\}, \qquad Q = \{q_1, \quad q_2, \quad \ldots, \quad q_n\}.$$

[2]This is not formally a parameterization, because there are some pairs $(P, \mathcal{S})$ for which $\mathcal{K}(P, \mathcal{S})$ is not a geometric simplicial complex. However, for any geometric simplicial complex $\mathcal{K}$, we can write down a pair $(P, \mathcal{S})$ such that $\mathcal{K} = \mathcal{K}(P, \mathcal{S})$.

[3]Formally, an indexed set P of n points in $\mathbb{R}^d$ is a map $P : \{1, 2, \ldots, n\} \to \mathbb{R}^d$. The ith member of P is $P(i)$, which we generally write as p_i for notational convenience. Here we extend the notion of P sets. Let $\alpha \subseteq \{1, 2, \ldots, n\}$; then $P(\alpha) := \{P(i) | i \in \alpha\}$.

[4]There is no formal definition of triangulation in geometry [13]. The definition of triangulation used here is slightly more general than the one used in [14], which requires that the underlying space be the convex hull of the vertex set. A triangulation as defined here which has a simply connected underlying space may be transformed to a triangulation as defined in [14] by a PL homeomorphism. The key concept in our definition is that a triangulation has good local volume properties everywhere. The underlying space has no "thin" spots. In topology, a triangulation of a topological space $\mathcal{X}$ is formally defined as a geometric simplicial complex $\mathcal{K}$ coupled with a homeomorphism between $|\mathcal{K}|$ and $\mathcal{X}$. The definition of triangulation used in this paper is more narrow than the topological notion.

$\mathcal{S}$ is an abstract simplicial complex of indices with $\text{vert}(\mathcal{S}) = \{1, \ldots, n\}$ such that $\mathcal{T}(P, \mathcal{S})$ is a triangulation and $|\mathcal{T}(P, \mathcal{S})| = D$. This defines a continuous PL function $f_{\mathcal{P}}$ such that $f_{\mathcal{P}}(p_i) = q_i$, and for any $S \in \mathcal{T}(P, \mathcal{S})$, $f_{\mathcal{P}}(x)$ is affine on S. For a d-simplex $S_i \in \mathcal{T}(P, \mathcal{S})$ with $\text{vert}(S_i) = \{p_{i_1}, p_{i_2}, \ldots, p_{i_{d+1}}\}$, the PL function $f_{\mathcal{P}}(x)$ for $x \in S_i$ is given by

$$f_{\mathcal{P}}\big|_{S_i}(x) = \begin{bmatrix} q_{i_1} & q_{i_2} & \cdots & q_{i_{d+1}} \end{bmatrix} \begin{bmatrix} p_{i_1} & p_{i_2} & \cdots & p_{i_{d+1}} \\ 1 & 1 & & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ 1 \end{bmatrix}. \tag{2.2}$$

Equation (2.2) uses a homogeneous representation for the rightmost two factors, though $f_{\mathcal{P}}|_{S_i}$ can be equivalently expressed in the more typical form as $A_i x + b_i$. The d-simplices of $\mathcal{T}(P, \mathcal{S})$ are a cover for D. If $S_i \cap S_j \neq \emptyset$, and $f_{\mathcal{P}}\big|_{S_i}(x) = A_i x + b_i$ and $f_{\mathcal{P}}\big|_{S_j}(x) = A_j x + b_j$, then $A_i x + b_i = A_j x + b_j$ for $x \in S_i \cap S_j$. This follows from Claim 5 in Appendix A, which states that $(A_i - A_j)$ has a null space of dimension $d_{i,j}$ parallel to $\text{aff}(S_i \cap S_j)$.

One of the most compelling properties of PL functions is the ability to check invertibility and invert in closed form. Let $f_{\mathcal{P}}$ be a PL function parameterized by $\mathcal{P} = (P, Q, \mathcal{S})$. If $\mathcal{T}(Q, \mathcal{S})$ is a triangulation, then the PL function is invertible, and the inverse ${f_{\mathcal{P}}}^{-1}$ is a PL function parameterized by $\mathcal{P}^{-1} = (Q, P, \mathcal{S})$. This is proven in Claim 7 in Appendix A.

Another important fact applied in proving the main result of section 4 is the continuity of a PL function in its continuous parameters. This claim, stated formally in Claim 6 in Appendix A, establishes that two PL functions with the same combinatorial structure are close in the L_∞ sense if their vertices are close in the Euclidean sense.

We call a PL function $f_{\mathcal{P}}$ parameterized by $\mathcal{P} = (P, Q, \mathcal{S})$ *nondegenerate* if for all $p_i \in P$ such that $p_i \notin \partial\,|\mathcal{T}(P, \mathcal{S})|$ the matrix H_i is full rank, where

$$H_i = \left(\frac{1}{N_i} \sum_{S_j \in \text{St}\{p_i\}} {A_j}^{\mathrm{T}} A_j \right) - \overline{A^i}^{\mathrm{T}} \overline{A^i}, \qquad \text{where } \overline{A^i} = \frac{1}{N_i} \sum_{S_j \in \text{St}\{p_i\}} A_j.$$

Intuitively, nondegeneracy of $f_{\mathcal{P}}$ requires that for any $p_i \notin \partial\,|\mathcal{T}(P, \mathcal{S})|$ not all of the affine functions that $f_{\mathcal{P}}$ takes in the surrounding d-simplices are parallel.

3. The `minvar` algorithm. The `minvar` algorithm is an iterative scheme to generate a locally good PL approximation to data. Similar to algorithms proposed in the possibly discontinuous piecewise polynomial approximation literature [2, 25, 26, 39], `minvar` takes advantage of the structure of PL functions. Let $\mathcal{Z} = \{(x_i, y_i)\}_{i=1}^{N_s}$, where $x_i \in D \subseteq \mathbb{R}^d$ and $y_i \in \mathbb{R}^d$, be the set of input-output data to be approximated. The `minvar` algorithm iteratively improves a PL approximation to the data, $f_{\mathcal{P}}^{(k)}$, parameterized by $\mathcal{P}^{(k)} = (P^{(k)}, Q^{(k)}, \mathcal{S})$, such that $\left|\mathcal{T}(P^{(k)}, \mathcal{S})\right| = D$. (The superscript in parentheses indicates the iteration number.) The algorithm breaks down into two stages. The first stage partitions the data according to the d-simplices of $\mathcal{T}(P^{(k)}, \mathcal{S})$ and computes the least squares linear approximations for each subset of the data. This set of linear approximations is the optimal possibly discontinuous PL approximation on the partition $\mathcal{T}(P^{(k)}, \mathcal{S})$. The second stage chooses $(P^{(k+1)}, Q^{(k+1)})$ to make $f_{\mathcal{P}}^{(k+1)}$, a continuous PL function, be "close" to the discontinuous approximation from the first stage. The stages are then iterated.

Recall from the previous section that the domain of a PL function is $|\mathcal{T}(P, \mathcal{S})|$, so moving a vertex $p_i \in \partial\,|\mathcal{T}(P, \mathcal{S})|$ will change the domain of definition of the PL

function. Since we desire a fixed domain for the PL function, the present exposition considers vertices on the domain boundary to be fixed. This can be relaxed to allow boundary vertices that are not extreme points to move in appropriately chosen affine subspaces using a constrained version of the cost function from step 3 of the `minvar` algorithm. Computational details of constrained motion will be presented in a subsequent paper on engineering applications of `minvar`.

From an initial parameterization $\mathcal{P}^{(0)} = (P^{(0)}, Q^{(0)}, \mathcal{S})$, the `minvar` algorithm generates a sequence of parameterizations, $\mathcal{P}^{(k)} = (P^{(k)}, Q^{(k)}, \mathcal{S})$, as follows:

1. Partition the data set $\mathcal{Z}$ into subsets $\mathcal{Z}_j$ corresponding to the d-simplices, $S_1, \ldots, S_N$ of $\mathcal{T}(P^{(k)}, \mathcal{S})$, breaking multiple memberships (data points that lie on the boundary between d-simplices) systematically.
2. Compute the least squares affine approximation, $L_j(x)$, for each subset $\mathcal{Z}_j$.
3. Update the vertex locations,

$$p_i^{(k+1)} = \arg\min_{x \in \mathbb{R}^d} \operatorname{var} L^i(x) + \lambda \left\| x - p_i^{(k)} \right\|^2 \qquad \forall\, p_i^{(k)} \notin \partial \left| \mathcal{T}(P^{(k)}, \mathcal{S}) \right|, \tag{3.1}$$

$$p_i^{(k+1)} = p_i^{(k)} \qquad \text{otherwise}, \tag{3.2}$$

$$q_i^{(k+1)} = \frac{1}{N_i} \sum_{S_j \in \mathrm{St}\{p_i\}} L^i\left(p_i^{(k+1)}\right) \qquad \forall\, i, \tag{3.3}$$

where $L_j(x) = A_j x + b_j$ and

$$\operatorname{var} L^i(x) = \sum_{S_j \in \mathrm{St}\{p_i\}} \left(L_j(x) - \frac{1}{N_i} \sum_{S_k \in \mathrm{St}\{p_i\}} L_k(x) \right)^2 .$$

4. If the vertices are not converged, then $k \leftarrow k+1$, go to 1.

Notice that (3.1) is a positive definite quadratic function of x, and thus can be minimized in closed form by "completing the square," with the solution given by

$$p_i^{(k+1)} = -{H_i}^{-1} h_i, \tag{3.4}$$

where

$$H_i = \left[\frac{1}{N_i} \sum_{S_j \in \mathrm{St}\{p_i\}} {A_j}^{\mathrm{T}} A_j \right] - \overline{A^i}^{\mathrm{T}} \overline{A^i} + \lambda I,$$

$$h_i = \left[\frac{1}{N_i} \sum_{S_j \in \mathrm{St}\{p_i\}} {A_j}^{\mathrm{T}} b_j \right] - \overline{A^i}^{\mathrm{T}} \overline{b^i} - \lambda p_i^{(k)},$$

$$\overline{A^i} = \frac{1}{N_i} \sum_{S_j \in \mathrm{St}\{p_i\}} A_j \qquad \overline{b^i} = \frac{1}{N_i} \sum_{S_j \in \mathrm{St}\{p_i\}} b_j.$$

The nonnegative quantity $\operatorname{var} L^i(x)$ measures, as a function of location in the domain x, how tightly clustered the range values generated from the least squares approximations on d-simplices in $\mathrm{St}\{p_i\}$ are. In the case of a scalar domain, an interior vertex p_i is in at most two 1-simplices. Thus, if the least squares approximations are not parallel, then $\operatorname{var} L^i(x_c) = 0$ at and only at x_c, the domain value of the point

at which the least squares approximations intersect. In the scalar case, the `minvar` algorithm with $\lambda = 0$ moves the domain and codomain vertices to the intersection point of the least squares approximations. We called our initial scalar algorithm the "Graph Intersection" algorithm [22] due to this fact. For dimensions higher than 1, there is generically no unique intersection point for the least squares approximations surrounding p_i, due to the geometry of triangulations. Rather than the intersection point, `minvar` with $\lambda = 0$ picks the point where the range values are most tightly clustered.

The λ term in (3.1) is a regularization. It guarantees that (3.1) will have a unique minimum, even if all the least squared approximations are parallel. More importantly, in the implementation of `minvar` the λ parameter can be tuned to prevent a vertex from jumping long distances and creating a "tangle" in the domain triangulation of the approximation. A tangle is when movement of the vertices causes $\mathcal{T}(P, \mathcal{S})$ to no longer be a geometric simplicial complex. That is, either a simplex has been "flattened" or there are simplices whose intersection is not another simplex from the complex. This generally occurs when a domain vertex moves through one of its opposing faces. Methods for detecting and correcting triangulation tangles will be covered in a subsequent paper on using `minvar` in engineering applications.

In this exposition, `minvar` does not modify the combinatorial structure, $\mathcal{S}$, of the PL approximation. Heuristics for adapting the domain triangulation of a two-dimensional PL function are presented in [12, 11] for interpolation and [39] for approximation. These heuristics flip edges in the domain triangulation to improve a local goodness criterion, similar to a method for computing the planar Delaunay triangulation. Generalizing these heuristics to higher dimensions is difficult because local topological changes of the triangulation in dimensions greater than two are more complex than edge flipping [28, 14]. Nonetheless, we find that adaptation of the combinatorial parameters of the PL function via topological flipping provides significant benefit in practice. Techniques for adapting the combinatorial structure will be presented in a subsequent paper on engineering applications of `minvar`.

4. A local convergence proof for the `minvar` algorithm. We turn now to the central result: a local convergence proof for the `minvar` algorithm. The result is for the "approximation," as opposed to "estimation," version of the `minvar` algorithm. That is, the data generating function is considered to be directly available in closed form, rather than through a set of discrete data. In this case, the least squares approximations from step 2 become L_2 orthogonal projections of $f^*_{\mathcal{P}}|_{S_i}$, the data generating function restricted to S_i, to the space of affine functions. Since the data or data generating function only appear in step 2, the approximation version may be viewed as the limit behavior of the estimation version when provided with an unbounded quantity of uniformly distributed data.

Theorem 4.1 shows that, if the data generating function is a nondegenerate PL function and the approximation is initialized "close enough" to the data generating function, then the `minvar` algorithm with $\lambda = 0$ will cause the approximation to converge to the data generating function in the L_∞ sense. In this case, "close enough" means that the initial approximation shares the same combinatorial structure as the data generating function, and the vertices of the approximation start close to the corresponding vertices of the data generating function. Examining `minvar` when $\lambda = 0$ admits a simpler proof while capturing the essence of the algorithm. Similar results could be obtained for $\lambda > 0$, though the convergence rate would be slower. An additional technical condition, that the data generating function be nondegenerate,

is required when $\lambda = 0$ in order to guarantee existence of a unique solution to (3.1), whereas for $\lambda > 0$ the regularized variance minimization in (3.1) is guaranteed to have a unique solution.

In this paper, unless otherwise noted, vector norms are the standard Euclidean norm and matrix norms are the induced two norm.

THEOREM 4.1. *Let $f^*_{\mathcal{P}}$ be a nondegenerate PL data generating function parameterized by $\mathcal{P}^* = (P^*, Q^*, \mathcal{S}^*)$. Let $\epsilon_0 = \epsilon_0(\mathcal{P}^*)$ be given by (4.7). Let the initial approximation $f^{(0)}_{\mathcal{P}}$ be parameterized by $(P^{(0)}, Q^{(0)}, \mathcal{S}^*)$, satisfying for some $\epsilon < \epsilon_0$, $\|p^{(0)}_i - p^*_i\| < \epsilon$, for all i. Then application of the* `minvar` *algorithm with $\lambda = 0$ yields a sequence of approximations satisfying*

$$\lim_{j\to\infty} \left\| f^{(j)}_{\mathcal{P}} - f^*_{\mathcal{P}} \right\|_\infty = 0.$$

Proof. Proposition 4.5 shows that iteration of the `minvar` algorithm causes the vertices of the approximation to converge to the vertices of the data generating function. By Claim 6 in Appendix A, a PL function is continuous in its vertices. The theorem follows directly. ∎

The theorem follows readily from Proposition 4.5, which likewise follows readily from Proposition 4.4. The statements and proofs of the propositions and lemmas follow in the next subsections, but first we offer a short sketch of the structure of the proof. The essence of Proposition 4.4 is that when the distances between the vertices of the approximation and the corresponding vertices of the data generating function are bounded by ϵ, then after one iteration of the `minvar` algorithm the distances will be bounded by a constant times ϵ^2. This result is established by applying two lemmas corresponding to the two stages of the algorithm. Lemma 4.2 proves that if the distances between corresponding vertices are bounded by ϵ, then the perturbation of the least squares affine map over a given simplex of the approximation from the affine map in the corresponding simplex of the data generating function is bounded by a constant times ϵ^2. Lemma 4.3 proves that if the perturbation of the least squares affine map over a simplex of the approximation from the affine map in the corresponding simplex of the data generating function is bounded by Δ, then the variance minimization will place the new vertices of the approximation such that the distance between them and the corresponding vertices of the data generating function are bounded by a constant times Δ. The combination of Lemmas 4.2 and 4.3 provides Proposition 4.4.

The quadratic rate of convergence in Proposition 4.4 arises from the hypothesis that the data generating function is piecewise linear and close to the initial approximation. Without this assumption, Lemma 4.2 would fail to provide an ϵ^2 perturbation in the least squares affine approximations. In this case, we suspect the convergence rate of the algorithm to be linear. Convergence may be slower on fine triangulations, but since this algorithm is intended primarily for use with a discrete set of data, the fineness of the triangulation is inherently limited by the amount of data provided. In applications, `minvar` can run triangulations of practical size in a few minutes.

4.1. Lemmas and propositions. This section states the lemmas and propositions, while the proofs are provided in the following section. First, we introduce several reoccurring constants. These constants may be interpreted geometrically as minima or maxima of different measures of the "radii" of d-simplices in the triangulation $\mathcal{T}(P^*, \mathcal{S}^*)$ of the data generating function. The first measures the maximum

inter-vertex distance between "connected" vertices,

$$r_1 := \max_{\substack{S^* \in \mathcal{T}(P^*, \mathcal{S}^*) \\ p_i^*, p_j^* \in S^*}} \left\| p_i^* - p_j^* \right\|. \tag{4.1}$$

The second measures the minimum distance of a vertex to its opposing hyperplanes,

$$r_2 := \min_{\substack{S^* \in \mathcal{T}(P^*, \mathcal{S}^*) \\ p^* \in \mathrm{vert}(S^*)}} \delta(p^*, \mathrm{aff}(\mathrm{vert}(S^*) - \{p^*\})). \tag{4.2}$$

The third measures the d-simplex which can be dilated the least before it intersects simplices outside its immediate neighborhood.

$$r_3 := \min_{S^* \in \mathcal{T}(P^*, \mathcal{S}^*)} \sup \left\{ \epsilon \middle| \mathcal{D}(S^*, \epsilon) \subseteq |\mathrm{Cl}\,\mathrm{St} S^*| \right\}. \tag{4.3}$$

The first lemma shows that if the domain vertices of the approximation are close to the domain vertices of the data generating function, then least squares affine fit in a simplex S_i is a perturbation away from the affine function that the data generating function takes in S_i^*. Moreover, the perturbation is quadratic in the bound on the distance between the approximation and data generating function's domain vertices. We write $\Pi(f)$ to denote the L_2 orthogonal projection of the function f onto the space of affine functions.

LEMMA 4.2. *Let $f_{\mathcal{P}}^*$ be a PL data generating function parameterized by $\mathcal{P}^* = (P^*, Q^*, \mathcal{S}^*)$. Consider a PL approximation $f_{\mathcal{P}}$ parameterized by $\mathcal{P} = (P, Q, \mathcal{S}^*)$. Let $\epsilon < \epsilon_c$, where*

$$\epsilon_c := \min \left\{ \tfrac{1}{2(d+1)} r_2, \quad r_3, \quad 1 \right\}. \tag{4.4}$$

Consider the simplices S_i^* and S_i. Let $x_c \in S_i^*$. Let $f_{\mathcal{P}}^*|_{S_i^*}(x) = A_i^*(x - x_c) + b_i^*$. If $\left\| p_j - p_j^* \right\| < \epsilon$ for all $p_j^* \in S_i^*$, then the least squares approximation to $f_{\mathcal{P}}^*$ on S_i, $\Pi(f_{\mathcal{P}}^*|_{S_i})(x) = \hat{A}_i(x - x_c) + \hat{b}_i$, satisfies the property

$$\left\| \begin{bmatrix} \hat{A}_i{}^{\mathrm{T}} - A_i^{*\mathrm{T}} \\ \hat{b}_i{}^{\mathrm{T}} - b_i^{*\mathrm{T}} \end{bmatrix} \right\|_2 < c_{1,i} \epsilon^2, \tag{4.5}$$

where $c_{1,i} = c_{1,i}(\mathcal{P}^*)$ is given by (4.18).

The second lemma considers one set of affine functions that all intersect at a common point and another set of affine functions which are perturbations of the first set of functions. It is shown that performing the variance minimization, equivalent to (3.1) with $\lambda = 0$, on the second set of functions generates a point whose distance from the intersection point is linear in the norm of the perturbations.

LEMMA 4.3. *Let L^* be a set of N affine maps, $L_1^*, \ldots L_N^*$, such that all intersect at (p^*, q^*) and are written as $L_i^*(x) = A_i^*(x - p^*) + q^*$, and such that H^*, given by* (4.20)*, is full rank. Let L be a set of perturbed affine maps, $L_1, \ldots L_N$, expressed as $L_i(x) = \hat{A}_i(x - p^*) + \hat{q}_i$, which satisfy the property*

$$\left\| \begin{bmatrix} \hat{A}_i{}^{\mathrm{T}} - A_i^{*\mathrm{T}} \\ \hat{q}_i{}^{\mathrm{T}} - q^{*\mathrm{T}} \end{bmatrix} \right\| < \Delta \tag{4.6}$$

for $\Delta < \Delta_0$, where $\Delta_0 = \Delta_0(A_i^*, p^*, q^*)$ is given by (4.19). *Let p' and q' be given by*

$$p' = \arg\min_x \quad \text{var } L(x),$$
$$q' = \frac{1}{N}\sum_{i=1}^{N} L(p').$$

Then p' and q' satisfy

$$\|p' - p^*\| < c_2\Delta,$$
$$\|q' - q^*\| < c_3\Delta,$$

where $c_2 = c_2(A_i^*, p^*, q^*)$ and $c_3 = c_3(A_i^*, p^*, q^*)$ are given by (4.23) *and* (4.24).

The first proposition brings the two lemmas together to show that a single step of the `minvar` algorithm induces a quadratic change in the distance of the approximation vertices to the data generating function vertices.

PROPOSITION 4.4. *Let $f_{\mathcal{P}}^*$ be a nondegenerate PL data generating function parameterized by $\mathcal{P}^* = (P^*, Q^*, \mathcal{S}^*)$. Let $\epsilon < \epsilon_d$,*

$$\epsilon_d := \min\left\{\epsilon_c, \quad \sqrt{\frac{\Delta_0^m}{c_1}}\right\},$$

where $\epsilon_c = \epsilon_c(\mathcal{P}^*)$ is given by (4.4), *$\Delta_0^m = \Delta_0^m(\mathcal{P}^*)$ by* (4.25), *and $c_1 = c_1(\mathcal{P}^*)$ by* (4.27).

If the PL approximation $f_{\mathcal{P}}$ parameterized by $(P, Q, \mathcal{S}^*)$ satisfies $\|p_i - p_i^*\| < \epsilon$ for all i, then one iteration of the `minvar` *algorithm with $\lambda = 0$ gives the new approximation f_P' parameterized by $(P', Q', \mathcal{S}^*)$, which satisfies*

$$\|p_i' - p_i^*\| < c_4\epsilon^2,$$
$$\|q_i' - q_i^*\| < c_5\epsilon^2,$$

for all i, where $c_4 = c_4(\mathcal{P}^*)$ and $c_5 = c_5(\mathcal{P}^*)$ are given by (4.31) *and* (4.32).

The second proposition applies the first proposition to show that iteration of the `minvar` algorithm causes convergence of the vertices of the approximation to the vertices of the data generating function.

PROPOSITION 4.5. *Let $f_{\mathcal{P}}^*$ be a nondegenerate PL data generating function parameterized by $(P^*, Q^*, \mathcal{S}^*)$. Let*

$$\epsilon_0 = \min\left\{\epsilon_d, \ \frac{1}{c_4}\right\}, \tag{4.7}$$

where ϵ_d and c_4 are given in Proposition 4.4. *If for some $0 < \epsilon < \epsilon_0$ the initial PL approximation $f_P^{(0)}$ with parameterization $(P^{(0)}, Q^{(0)}, \mathcal{S}^*)$ satisfies $\|p_i^{(0)} - p_i^*\| < \epsilon$ for all i, then iteration of the* `minvar` *algorithm with $\lambda = 0$ gives a sequence of approximations $f_P^{(k)}$ satisfying*

$$\lim_{k\to\infty} \left\|p_i^{(k)} - p_i^*\right\| = 0,$$
$$\lim_{k\to\infty} \left\|q_i^{(k)} - q_i^*\right\| = 0$$

for all i.

4.2. Proofs of lemmas and propositions. This section presents proofs of the lemmas and propositions stated in the previous section.

Proof of Lemma 4.2. Let $\varphi_i(x) := A_i^*(x - x_c) + b_i^*$, the extension of $f_{\mathcal{P}}^*|_{S_i^*}$ to the entire domain. Let $\psi_i(x) := f_{\mathcal{P}}^*(x) - \varphi_i(x)$.

The orthogonal projection Π is a linear operator, and, moreover, for g affine, $\Pi(g) = g$. It follows that

$$\begin{aligned} \Pi(f_{\mathcal{P}}^*|_{S_i}) &= \Pi(\varphi_i|_{S_i}) + \Pi(\psi_i|_{S_i}) \\ &= \varphi_i + \Pi(\psi_i|_{S_i}). \end{aligned} \tag{4.8}$$

Let $\hat{A}_i$ and $\hat{b}_i$ be such that $\Pi(f_{\mathcal{P}}^*|_{S_i})(x) = \hat{A}_i(x - x_c) + \hat{b}_i$. Then from (4.8) it follows that $\Pi(\psi_i|_{S_i}) = (\hat{A}_i - A_i^*)(x - x_c) + (\hat{b}_i - b_i^*)$. Moreover, since $\Pi(\psi_i|_{S_i}) = \Pi(f_{\mathcal{P}}^*|_{S_i} - \varphi_i|_{S_i})$, we can compute $(\hat{A}_i - A_i^*)$ and $(\hat{b}_i - b_i^*)$ using the formula for the L_2 orthogonal projection of $f_{\mathcal{P}}^*|_{S_i} - \varphi_i|_{S_i}$,

$$\begin{bmatrix} (\hat{A}_i - A_i^*)^{\mathrm{T}} \\ (\hat{b}_i - b_i^*)^{\mathrm{T}} \end{bmatrix} = {S_{xx,i}}^{-1} S_{xy,i},$$

$$S_{xx,i} = \int_{S_i} \begin{bmatrix} x - x_c \\ 1 \end{bmatrix} \begin{bmatrix} x^{\mathrm{T}} - {x_c}^{\mathrm{T}} & 1 \end{bmatrix} dx, \qquad S_{xy,i} = \int_{S_i} \begin{bmatrix} x - x_c \\ 1 \end{bmatrix} \left(f_{\mathcal{P}}^*(x) - \varphi_i(x)\right)^{\mathrm{T}} dx.$$

The submultiplicative property holds for the induced two norm,

$$\left\| \begin{bmatrix} (\hat{A}_i - A_i^*)^{\mathrm{T}} \\ (\hat{b}_i - b_i^*)^{\mathrm{T}} \end{bmatrix} \right\| \leq \left\| {S_{xx,i}}^{-1} \right\| \left\| S_{xy,i} \right\|,$$

so we can independently establish bounds on $\left\| {S_{xx,i}}^{-1} \right\|$ and $\|S_{xy,i}\|$. We will proceed to bound $\left\| {S_{xx,i}}^{-1} \right\|$. Since S_i is a d-simplex, it follows from calculus and the definition of $S_{xx,i}$ that $S_{xx,i}$ is a positive definite matrix. Let M_i^* be given by

$$M_i^* := \int_{\mathcal{D}(S_i^*, -\epsilon_c)} \begin{bmatrix} x - x_c \\ 1 \end{bmatrix} \begin{bmatrix} x^{\mathrm{T}} - {x_c}^{\mathrm{T}} & 1 \end{bmatrix} dx.$$

Since $0 < \epsilon < \epsilon_c$ by hypothesis and $\epsilon_c \leq \frac{1}{2(d+1)} r_2$ by definition, it follows from Claim 3 in Appendix A that $\mathcal{D}(S_i^*, -\epsilon_c)$ is a d-simplex. Thus M_i^* is also positive definite, and hence invertible. Since $\epsilon < \epsilon_c$ and the vertices or S_i are all less than ϵ away from the vertices of S_i^*, it follows that $\mathcal{D}(S_i^*, -\epsilon_c) \subseteq S_i$. Thus, $x^{\mathrm{T}} M_i^* x < x^{\mathrm{T}} S_{xx,i} x$ for all x, which implies that $\lambda_{\min}(M) < \lambda_{\min}(S_{xx,i})$. This provides the bound on $\left\| {S_{xx,i}}^{-1} \right\|$,

$$\left\| {S_{xx,i}}^{-1} \right\| < \left\| {M_i^*}^{-1} \right\|.$$

Now we proceed to $\|S_{xy,i}\|$. By the properties of norms,

$$\|S_{xy,i}\| \leq \int_{S_i} \left\| \begin{bmatrix} x - x_c \\ 1 \end{bmatrix} \right\| \|\psi_i(x)\| \, dx. \tag{4.9}$$

Let $\mathcal{C}(S_i, \epsilon) = \{x \in \mathbb{R}^d | \delta(x, S_i) \leq \epsilon\}$. By hypothesis, the vertices of S_i are less than ϵ away from the vertices of S_i^*, so $\operatorname{vert} S_i \subseteq \mathcal{C}(S_i^*, \epsilon)$. Moreover, since both S_i and

$\mathcal{C}(S_i^*, \epsilon)$ are convex, $S_i \subseteq \mathcal{D}(S_i^*, \epsilon)$. The integrand in (4.9) is nonnegative definite, so (4.9) is bounded by

$$\leq \int_{\mathcal{C}(S_i^*,\epsilon)} \left\| \begin{bmatrix} x - x_c \\ 1 \end{bmatrix} \right\| \|(\psi_i(x))\| \, dx.$$

By hypothesis $x_c \in S_i^*$. By the definition of r_1 and since $\epsilon < \epsilon_c$, it follows that for all $x \in \mathcal{C}(S_i^*, \epsilon)$, $\|x - x_c\| \leq \bar{r} := r_1 + 2\epsilon_c$. Thus, the integral above is further bounded by

$$\leq \sqrt{1 + \bar{r}^2} \int_{\mathcal{C}(S_i^*,\epsilon)} \|(\psi_i(x))\| \, dx. \tag{4.10}$$

Once again the integrand is nonnegative definite, so (4.10) can be bounded by integrating over $\mathcal{D}(S_i^*, \epsilon)$, since $\mathcal{C}(S_i^*, \epsilon) \subseteq \mathcal{D}(S_i^*, \epsilon)$,

$$\leq \sqrt{1 + \bar{r}^2} \int_{\mathcal{D}(S_i^*,\epsilon)} \|(\psi_i(x))\| \, dx \tag{4.11}$$

$$= \sqrt{1 + \bar{r}^2} \sum_{j=1}^{N} \int_{U_j} \|(\psi_i(x))\| \, dx, \tag{4.12}$$

where $U_j = \mathcal{D}(S_i^*, \epsilon) \cap S_j^*$ and N is the total number of d-simplices in the domain triangulation. Since $\psi_i(x) = 0$ on S_i^*, the term corresponding to $j = i$ in (4.12) is 0. By hypothesis $\epsilon < \epsilon_c \leq r_3$, so then following from the definition of r_3, $S_j^* \cap \mathcal{D}(S_i^*, \epsilon) \neq \emptyset$ if and only if $S_j^* \in \operatorname{St} S_i^*$. Thus the terms in (4.12) are only nonzero for j such that S_j^* is incident to S_i^*. Consider such a term,

$$\int_{U_j} \|\psi_i(x)\| \, dx = \int_{U_j} \left\| \left(A_j^* - A_i^*\right)(x - x_c) + b_j^* - b_i^* \right\| dx,$$

where $f_{\mathcal{P}}^*|_{S_j^*}(x) = A_j^*(x - x_c) + b_j^*$. By Claim 5 in Appendix A, there exists $x_O \in S_j^* \cap S_i^* \subseteq U_j$ such that $\left(A_j^* - A_i^*\right)(x_O - x_c) + b_j^* - b_i^* = 0$. Applying the change of coordinates $y = x - x_O$ gives

$$\int_{U_j} \|\psi_i(x)\| \, dx = \int_{U_j - x_O} \left\| \left(A_j^* - A_i^*\right) y \right\| \, dy. \tag{4.13}$$

Let L be the linear subspace parallel to $\operatorname{aff}(S_i^* \cap S_j^*)$. Recall that $\dim L = \dim S_i^* \cap S_j^* := d_{i,j}$. By Claim 5 in Appendix A, $L \subseteq \mathcal{N}(\left(A_j^* - A_i^*\right))$. Let $v_1, \ldots, v_{d_{i,j}}$ be an orthonormal basis for L. Let $v_{d_{i,j}+1}, \ldots, v_d$ be an orthonormal basis for $L^\perp$. Then $P = \begin{bmatrix} v_1 & v_2 & \cdots & v_d \end{bmatrix}$ is an orthogonal matrix. Rewrite (4.13) under the change of coordinates $z = P^{\mathrm{T}} y$,

$$= \int_{P^{\mathrm{T}}(U_j - x_O)} \left\| \left(A_j^* - A_i^*\right) P z \right\| \, dz. \tag{4.14}$$

Since the integrand is a nonnegative definite function, we may bound (4.14) by increasing the volume over which the integrand is integrated. By Claim 4 in Appendix A, there exists $\kappa_{i,j}$ such that for all $x \in U_j$, $\delta\big(x, \operatorname{aff}(S_i^* \cap S_j^*)\big) < \kappa_{i,j}\epsilon$. Equivalently $\delta(y, L) < \kappa_{i,j}\epsilon$ for any $y \in U_j - x_O$, from which it follows that the projection

of y onto $L^\perp$ must have magnitude less than $\kappa_{i,j}\epsilon$. Moreover, by the definition of $\bar{r}$, the projection of y onto L must have magnitude less than $\bar{r}$. It follows that $P^{\mathrm{T}}(U_j - x_O) \subseteq [-\bar{r}, \bar{r}]^{d_{i,j}} \times B_{\bar{d}_{i,j}}(\kappa_{i,j}\epsilon)$, where $\bar{d}_{i,j} = d - d_{i,j}$ and $B_{\bar{d}_{i,j}}(\kappa_{i,j}\epsilon)$ is the $\bar{d}_{i,j}$-dimensional ball of radius $\kappa_{i,j}\epsilon$. Then (4.14) is bounded by

$$\leq \int_{[-\bar{r},\bar{r}]^{d_{i,j}} \times B_{\bar{d}_{i,j}}(\kappa_{i,j}\epsilon)} \left\| \left(A_j^* - A_i^*\right) Pz \right\| \, dz. \tag{4.15}$$

The first $d_{i,j}$ columns of $\left(A_j^* - A_i^*\right) P$ are zero, since the first $d_{i,j}$ columns of P are in the nullspace of $\left(A_j^* - A_i^*\right)$. Thus, the integrand in (4.15) has no dependence on $z_1, z_2, \ldots, z_{d_{i,j}}$, so we can integrate through for $z_1, \ldots, z_{d_{i,j}}$, giving

$$= (2\bar{r})^{d_{i,j}} \int_{B_{\bar{d}_{i,j}}(\kappa_{i,j}\epsilon)} \left\| \left(A_j^* - A_i^*\right) P \begin{bmatrix} I \\ 0 \end{bmatrix} \bar{z}_1 \right\| \, dz_{d_{i,i}+1} \ldots dz_d, \tag{4.16}$$

where ${\bar{z}_1}^{\mathrm{T}} = \begin{bmatrix} z_{d_{i,j}+1} & \cdots & z_d \end{bmatrix}^{\mathrm{T}}$. Since the first $d_{i,j}$ columns of $\left(A_j^* - A_i^*\right) P$ are zero and P is orthogonal, it follows that $\left\| \left(A_j^* - A_i^*\right) P \begin{bmatrix} I & 0 \end{bmatrix}^{\mathrm{T}} \right\| \leq \left\| A_j^* - A_i^* \right\|$. Thus, (4.16) can be further bound by

$$\leq (2\bar{r})^{d_{i,j}} \left\| A_j^* - A_i^* \right\| \int_{B_{\bar{d}_{i,j}}(\kappa_{i,j}\epsilon)} \left\| \bar{z}_1 \right\| \, d\bar{z}_1. \tag{4.17}$$

From calculus (see, for example, [37]) it can be shown that[5]

$$\int_{B_k(\epsilon)} \|w\| \, dw = \frac{k}{k+1} \frac{\pi^{k/2}}{\Gamma(k/2)} \epsilon^{k+1}.$$

Applying this with (4.17) to (4.12), and then simplifying using the fact that $\epsilon^m \leq \epsilon^2$ for $m \geq 2$ since $\epsilon < \epsilon_c \leq 1$, gives

$$\|S_{xy,i}\| \leq \sqrt{1+\bar{r}^2}\, l_i \max_{\substack{j \text{ s.t. } j \neq i, \\ S_j^* \in \mathrm{St} S_i^*}} \left((2\bar{r})^{d_{i,j}} \left\| A_j^* - A_i^* \right\| {\kappa_{i,j}}^{1+\bar{d}_{i,j}} \frac{\bar{d}_{i,j}}{\bar{d}_{i,j}+1} \frac{\pi^{(\bar{d}_{i,j})/2}}{\Gamma((\bar{d}_{i,j})/2)} \right) \epsilon^2,$$

where $l_i = \sum_{S_j^* \in \mathrm{St} S_i^*} 1$. Then

$$\begin{aligned} \left\| \begin{bmatrix} \hat{A}_i^{\mathrm{T}} - A_i^{*\mathrm{T}} \\ \hat{b}_i^{\mathrm{T}} - b_i^{*\mathrm{T}} \end{bmatrix} \right\| &\leq \left\| {S_{xx,i}}^{-1} \right\| \|S_{xy,i}\| \\ &< c_{1,i}\epsilon^2, \end{aligned}$$

where

$$c_{1,i} := \left\| M_i^{*-1} \right\| \sqrt{1+\bar{r}^2}\, l_i \max_{\substack{j \text{ s.t. } j \neq i, \\ S_j^* \in \mathrm{St} S_i^*}} \left((2\bar{r})^{d_{i,j}} \left\| A_j^* - A_i^* \right\| {\kappa_{i,j}}^{1+\bar{d}_{i,j}} \frac{\bar{d}_{i,j}}{\bar{d}_{i,j}+1} \frac{\pi^{(\bar{d}_{i,j})/2}}{\Gamma((\bar{d}_{i,j})/2)} \right) \tag{4.18}$$

[5]For even k, $\Gamma(k/2) = (k/2)!$. For odd k, let $k' = \frac{1}{2}(k-1)$; then $\Gamma(k/2) = \Gamma(\frac{1}{2} + k') = \sqrt{\pi} \frac{(2k'+2)!}{(k+1)! 4^{k'+1}}$.

and $\bar{d}_{i,j} = d - d_{i,j}$, $l_i = \sum_{S_j^* \in \mathrm{St} S_i^*} 1$, and $\bar{r} = r_1 + 2\epsilon_c$. ∎

Proof of Lemma 4.3. Let the constant Δ_0 from the statement of the lemma be given by

$$\Delta_0 := \min \left\{ \frac{1}{N} \sum_{j=1}^{N} \left\| A_j^* \right\| \quad , \quad \left(\frac{12}{N} \left\| H^{*-1} \right\| \sum_{j=1}^{N} \left\| A_j^* \right\| \right)^{-1} \right\}, \tag{4.19}$$

where H^* is given by (4.20).

Solving $p' = \arg\min_x \mathrm{var}\ L(x)$ is equivalent to solving (3.1) with $\lambda = 0$. As with (3.1), a closed form expression for p' can be found by "completing the square." Specifically, $p' = H^{-1}h$,

$$H := \left(\frac{1}{N} \sum_{j=1}^{N} \hat{A}_j{}^{\mathrm{T}} \hat{A}_j \right) - \overline{\hat{A}}^{\mathrm{T}} \overline{\hat{A}},$$
$$h := \left(\frac{1}{N} \sum_{j=1}^{N} \hat{A}_j{}^{\mathrm{T}} (\hat{q}_j - \hat{A}_j p^*) \right) - \overline{\hat{A}}^{\mathrm{T}} \overline{\hat{b}},$$

where $\overline{\hat{A}} = \frac{1}{N} \sum_{j=1}^{N} \hat{A}_j$ and $\overline{\hat{b}} = \frac{1}{N} \sum_{j=1}^{N} (\hat{q}_j - \hat{A}_j p^*)$. Let $\tilde{A}_j = \hat{A}_j - A_j^*$ and $\tilde{q}_j = \hat{q}_j - q^*$. Then $H = H^* + \tilde{H}$ and $h = h^* + \tilde{h}$, with

$$H^* := \left(\frac{1}{N} \sum_{j=1}^{N} A_j^{*\mathrm{T}} A_j^* \right) - \overline{A^*}^{\mathrm{T}} \overline{A^*}, \tag{4.20}$$
$$h^* := \left(\frac{1}{N} \sum_{j=1}^{N} A_j^{*\mathrm{T}} (q^* - A_j^* p^*) \right) - \overline{A^*}^{\mathrm{T}},$$

$$\tilde{H} = \frac{1}{N} \left[\sum_{j=1}^{N} (A_j^* + \tilde{A}_j)^{\mathrm{T}} \tilde{A}_j + \sum_{j=1}^{N} \tilde{A}_j{}^{\mathrm{T}} A_j^* \right] - \overline{A^*}^{\mathrm{T}} \overline{\tilde{A}} - \overline{\tilde{A}}^{\mathrm{T}} \overline{A^*} - \overline{\tilde{A}}^{\mathrm{T}} \overline{\tilde{A}},$$
$$\tilde{h} = \frac{1}{N} \left[\sum_{j=1}^{N} (A_j^* + \tilde{A}_j)^{\mathrm{T}} (\tilde{q}_j - \tilde{A}_j p^*) + \sum_{j=1}^{N} \tilde{A}_j{}^{\mathrm{T}} (q^* - A_j^* p^*) \right] - \overline{A^*}^{\mathrm{T}} \overline{\tilde{b}} - \overline{\tilde{A}}^{\mathrm{T}} \overline{b^*} - \overline{\tilde{A}}^{\mathrm{T}} \overline{\tilde{b}},$$

where

$$\overline{A^*} = \frac{1}{N} \sum_{j=1}^{N} A_j^*, \qquad \overline{b^*} = \frac{1}{N} \sum_{j=1}^{N} (q^* - A_j^* p^*),$$
$$\overline{\tilde{A}} = \frac{1}{N} \sum_{j=1}^{N} \tilde{A}_j, \qquad \overline{\tilde{b}} = \frac{1}{N} \sum_{j=1}^{N} (\tilde{q}_j - \tilde{A}_j p^*).$$

Notice that H^* and h^* depend only on A_j^*, p^*, and q^*. Moreover, since all functions in L^* go through (p^*, q^*), it must be that $p^* = \arg\min_x \mathrm{var}\ L^*(x)$, and thus $p^* =$

$H^{*-1}h^*$. Rewriting $p' = H^{-1}h$, gives $(H^* + \tilde{H})(p^* + (p' - p^*)) = h^* + \tilde{h}$. Applying $H^*p^* = h^*$ and solving for $p' - p^*$ yields

$$p' - p^* = \left(H^* + \tilde{H}\right)^{-1} \left(\tilde{h} - \tilde{H}p^*\right).$$

From the hypothesis it follows that $\|\tilde{A}_j\| < \Delta$ and $\|\tilde{q}_j\| < \Delta$. Applying these bounds and the properties of norms, it follows after some computation that

$$\left\|\tilde{H}\right\| \leq 2\Delta^2 + \frac{4\Delta}{N} \sum_{j=1}^{N} \left\|A_j^*\right\|,$$

$$\left\|\tilde{h}\right\| \leq 2\left(1 + \|p^*\|\right)\Delta^2 + \frac{2\Delta}{N} \sum_{j=1}^{N} \left[\left\|A_j^*\right\|\left(1 + \|p^*\|\right) + \|q^*\| + \left\|A_j^*\right\|\|p^*\|\right].$$

Since $\Delta < \Delta_0$ and by definition $\Delta_0 \leq \frac{1}{N}\sum_{j=1}^{N}\left\|A_j^*\right\|$, the above bounds may be further simplified to

$$\left\|\tilde{H}\right\| < \left(\frac{6}{N}\sum_{j=1}^{N}\left\|A_j^*\right\|\right)\Delta, \tag{4.21}$$

$$\left\|\tilde{h}\right\| < \left(\frac{2}{N}\sum_{j=1}^{N}\left[2\left\|A_j^*\right\|\left(1 + \|p^*\|\right) + \|q^*\| + \left\|A_j^*\right\|\|p^*\|\right]\right)\Delta. \tag{4.22}$$

Also by definition $\Delta_0 \leq (\frac{12}{N}\|H^{*-1}\|\sum_{j=1}^{N}\|A_j^*\|)^{-1}$, so the bound in (4.21) can be simplified to $\|\tilde{H}\| < \frac{1}{2\|H^{*-1}\|}$, and thus $\|H^{*-1}\tilde{H}\| < \frac{1}{2}$. From [19], if $M \in \mathbb{R}^{n\times n}$ and $\|M\| < 1$, then $I - M$ is nonsingular and $\|(I-M)^{-1}\| \leq \frac{1}{1-\|M\|}$. Some computation using this fact and the bound on $\|H^{*-1}\tilde{H}\|$ provides

$$\left\|(H^* + \tilde{H})^{-1}\right\| < 2\left\|H^{*-1}\right\|.$$

So then

$$\begin{aligned}\|p' - p^*\| &\leq \left\|\left(H^* + \tilde{H}\right)^{-1}\right\| \left\|\tilde{h} - \tilde{H}p^*\right\| \\ &\leq 2\left\|H^{*-1}\right\|\left(\left\|\tilde{h}\right\| + \left\|\tilde{H}\right\|\|p^*\|\right) \\ &< c_2\Delta,\end{aligned}$$

where
$$c_2 := \frac{4\left\|H^{*-1}\right\|}{N}\sum_{j=1}^{N}\left(6\left\|A_j^*\right\|\|p^*\| + 2\left\|A_j^*\right\| + \|q^*\|\right), \tag{4.23}$$

which is the first part of the desired result. Applying this bound and the definition of q', we find after some computation that

$$\|q' - q^*\| < c_3\Delta,$$

$$\text{where} \quad c_3 := 1 + \frac{2c_2}{N} \sum_{j=1}^{N} \left\| A_j^* \right\| \tag{4.24}$$

which completes the desired result. □

Proof of Proposition 4.4. Let

$$\Delta_0^m = \min_{\substack{i \text{ s.t.} \\ p_i^* \in P^*}} \left\{ \frac{1}{N_i} \sum_{j=1}^{N_i} \left\| A_{i_j}^* \right\| \quad , \quad \left(\frac{12}{N_i} \left\| H_i^{*-1} \right\| \sum_{j=1}^{N_i} \left\| A_{i_j}^* \right\| \right)^{-1} \right\}, \tag{4.25}$$

where $S_{i_1}^*, \ldots, S_{i_{N_i}}^*$ are the N_i d-simplices in $\mathrm{St}\{p_i^*\}$, and

$$H_i^* = \left(\frac{1}{N_i} \sum_{j=1}^{N_i} A_{i_j}^{*\,\mathrm{T}} A_{i_j}^* \right) - \left(\frac{1}{N_i} \sum_{j=1}^{N_i} A_{i_j}^* \right)^{\mathrm{T}} \left(\frac{1}{N_i} \sum_{j=1}^{N_i} A_{i_j}^* \right).$$

We will examine the effect of a single iteration of the `minvar` algorithm on $p_i \in P$, $q_i \in Q$. The results will hold independently of i, giving the desired result.

The first stage of the `minvar` algorithm calculates the least squares projection $\Pi(f_{\mathcal{P}}^*|_{S_i})$ in each d-simplex S_i of the approximation (step 2 of the algorithm; since this proposition addresses the approximation version of the problem, there is no partitioning of data to be performed in step 1). Let $S_{i_1}, \ldots, S_{i_{N_i}}$ be the d-simplices in $\mathrm{St}\{p_i\}$. Since $f_{\mathcal{P}}^*$ and $f_{\mathcal{P}}$ are parameterized with the same abstract simplicial complex $\mathcal{S}^*$, $S_{i_1}^*, \ldots, S_{i_{N_i}}^*$ are the d-simplices in $\mathrm{St}\{p_i^*\}$. Let $f_{\mathcal{P}}^*|_{S_{i_j}^*}(x) = A_{i_j}^*(x - p_i^*) + q_i^*$ and $\Pi(f_{\mathcal{P}}^*|_{S_i}) = \hat{A}_{i_j}(x - p_i^*) + \hat{q}_{i_j}$. Since $\epsilon < \epsilon_d \leq \epsilon_c$, it follows from Lemma 4.2 that for each $j = 1, \ldots, N_i$,

$$\left\| \begin{bmatrix} \hat{A}_{i_j}^{\mathrm{T}} - A_{i_j}^{*\,\mathrm{T}} \\ \hat{q}_{i_j}^{\mathrm{T}} - q_i^{*\mathrm{T}} \end{bmatrix} \right\| < c_{1,i_j} \epsilon^2, \tag{4.26}$$

where c_{1,i_j} is given by (4.18). Let

$$c_1 = \max_{i=1,..,N} c_{1,i}, \tag{4.27}$$

where N is the total number of d-simplices in $\mathcal{T}(P, \mathcal{S}^*)$. Then for $j = 1 \ldots, N_i$,

$$\left\| \begin{bmatrix} \hat{A}_{i_j}^{\mathrm{T}} - A_{i_j}^{*\,\mathrm{T}} \\ \hat{b}_{i_j}^{\mathrm{T}} - b_{i_j}^{*\,\mathrm{T}} \end{bmatrix} \right\| < c_1 \epsilon^2. \tag{4.28}$$

Step 3 of `minvar` moves the knots taking $(p_i, q_i) \to (p_i', q_i')$. Since $\epsilon < \epsilon_d \leq \sqrt{\Delta_0^m / c_1}$ by hypothesis, it follows that $c_1 \epsilon^2 < \Delta_0^m$. Thus, (4.28) implies that the bound in (4.6) is satisfied, permitting application of Lemma 4.3, which gives

$$\begin{aligned} \|p_i' - p_i^*\| &< c_{2,i} c_1 \epsilon^2, \\ \|q_i' - q_i^*\| &< c_{3,i} c_1 \epsilon^2, \end{aligned}$$

where

$$c_{2,i} := \frac{4 \left\| H_i^{*-1} \right\|}{N_i} \sum_{j=1}^{N_i} \left(6 \left\| A_{i_j}^* \right\| \|p_i^*\| + 2 \left\| A_{i_j}^* \right\| + \|q_i^*\| \right), \tag{4.29}$$

$$c_{3,i} := 1 + \frac{2c_{2,i}}{N_i} \sum_{j=1}^{N_i} \left\| A_{i_j}^* \right\|. \tag{4.30}$$

For each p_i, q_i we can compute such bounds. Let

$$c_4 := c_1 \max_i c_{2,i}, \tag{4.31}$$

$$c_5 := c_1 \max_i c_{3,i}. \tag{4.32}$$

Then, for all i, p_i' and q_i' satisfy

$$\|p_i' - p_i^*\| < c_4 \epsilon^2, \tag{4.33}$$

$$\|q_i' - q_i^*\| < c_5 \epsilon^2, \tag{4.34}$$

which is the desired result. □

Proof of Proposition 4.5. First we establish by induction that for $k \geq 1$,

$$\frac{1}{c_4} (c_4 \epsilon)^{2^k} < \epsilon_d, \tag{4.35}$$

$$\left\| p_i^{(k)} - p_i^* \right\| < \frac{1}{c_4} (c_4 \epsilon)^{2^k}, \tag{4.36}$$

$$\left\| q_i^{(k)} - q_i^* \right\| < \frac{c_5}{c_4} (c_4 \epsilon)^{2^k}. \tag{4.37}$$

For $k = 1$, $c_4 \epsilon^2 < \epsilon_d$ since $\epsilon < \epsilon_0 \leq \sqrt{\epsilon_d / c_4}$. For $k > 1$, $\frac{1}{c_4} (c_4 \epsilon)^{2^k} < \epsilon_d$ by the induction hypothesis. Moreover, $c_4 \epsilon < 1$ since $\epsilon < \epsilon_0 \leq \frac{1}{c_4}$. Then $\frac{1}{c_4} (c_4 \epsilon)^{2^{(k+1)}} = \left(\frac{1}{c_4} (c_4 \epsilon)^{2^k}\right) (c_4 \epsilon)^{2^k} < \epsilon_d$. This establishes (4.35). Since $\epsilon < \epsilon_0 \leq \epsilon_d$, it follows that for $k = 1$, after a single iteration of the `minvar` algorithm, (4.36) and (4.37) will hold by Proposition 4.4. For $k > 1$, for all i, $\|p_i^{(k)} - p_i^*\| < \frac{1}{c_4} (c_4 \epsilon)^{2^k}$ by the induction hypothesis. From (4.35), proven above, $\frac{1}{c_4} (c_4 \epsilon)^{2^k} < \epsilon_d$. Since for all i, $\|p_i^{(k)} - p_i^*\| < \epsilon_d$, it follows from Proposition 4.4 that

$$\left\| p_i^{(k+1)} - p_i^* \right\| < c_4 \left(\frac{1}{c_4} (c_4 \epsilon)^{2^k} \right)^2 = \frac{1}{c_4} (c_4 \epsilon)^{2^{(k+1)}},$$

$$\left\| q_i^{(k+1)} - q_i^* \right\| < c_5 \left(\frac{1}{c_4} (c_4 \epsilon)^{2^k} \right)^2 = \frac{c_5}{c_4} (c_4 \epsilon)^{2^{(k+1)}},$$

which establishes (4.36) and (4.37). Since $c_4 \epsilon < 1$, as argued above, it follows that (4.36) and (4.37) go to 0 as k goes to infinity. □

5. Numerical example. This section presents an example of the `minvar` algorithm's performance on a "test function," $f : [0,1]^2 \to \mathbb{R}^2$, given by

$$f(x) = \begin{bmatrix} \tanh \frac{5}{8} \left(2x_1 - 4{x_2}^4 + 3{x_2}^2 - 1 \right) \\ \tanh \frac{5}{8} \left(2{x_1}^2 - {x_1}^4 + 2x_2 - 1 \right) \end{bmatrix}, \tag{5.1}$$

which is invertible over the domain $D = [0,1]^2$. The implementation of `minvar` constructs an approximation to a discrete set of data and includes constrained motion of boundary vertices as well as data dependent retriangulation. Since the test function is neither piecewise linear nor directly available, Theorem 4.1 provides no performance guarantees, but good performance under these circumstances suggests `minvar`'s broader applicability. Two sets of numerical studies are presented. The first

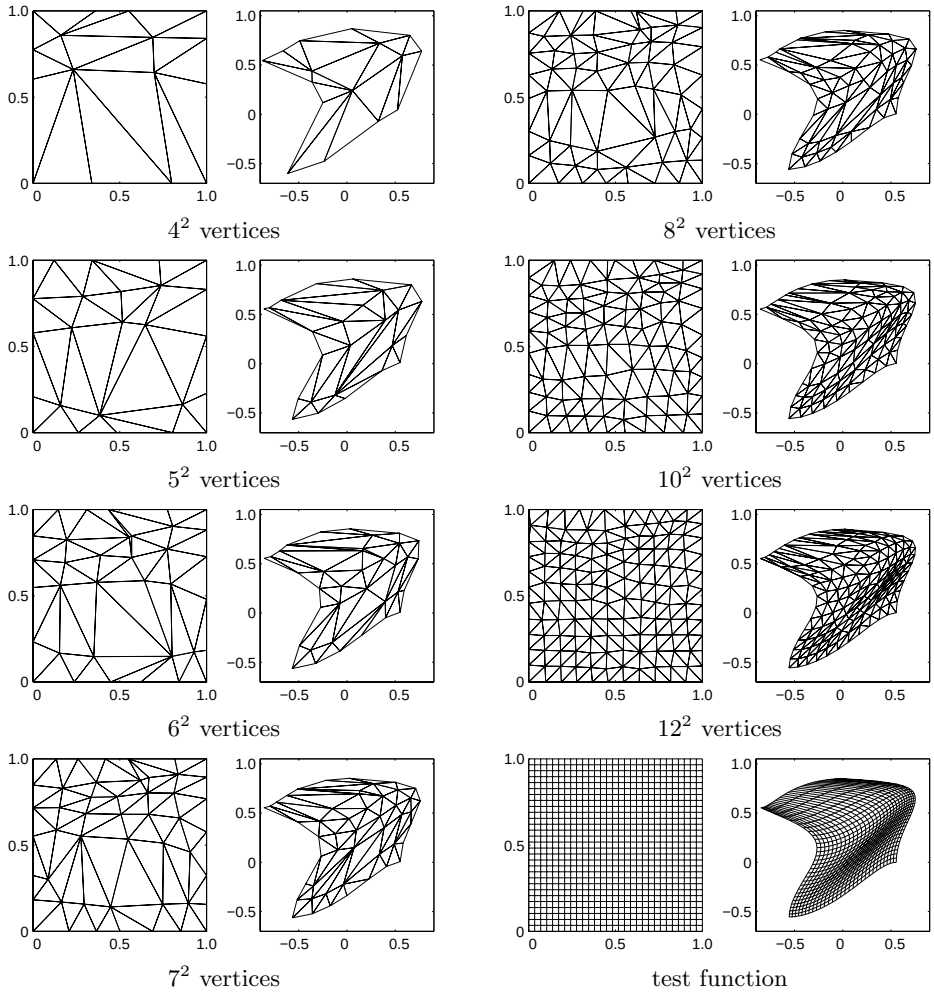

FIG. 5.1. *Visualizations of the test function (lower right) and a series of PL approximations to the test function. In each subfigure, the domain is displayed on the left and the range on the right. Note that the approximations are invertible, since there are no tangles in the range triangulations.*

examines the effects of varying the number of vertices in the PL approximation, and the second examines how data set size affects approximations with a fixed number of vertices.

The first set of experiments fits PL approximations of differing sizes to a single data set. The data set was generated by sampling the test function on an 80×80 uniform grid over the domain. For each $n = 2, \ldots, 13$, three different PL approximations with n^2 vertices were computed: (i) the least squares continuous PL approximation on a fixed uniform triangulation of the domain (referred to as "uniform LS"), (ii) `minvar`, initialized on a uniform triangulation of the domain (referred to as "`minvar`"), and (iii) the least squares continuous PL approximation on the final triangulation from `minvar` (referred to as "`minvar` LS"). Recall that when the domain triangulation is fixed, the least squares continuous PL approximation problem becomes linear-in-parameters and the solution can be computed directly [7, 35]. Figure 5.1 shows the test function and several exemplars of the `minvar` approximations. Table 5.1 and

TABLE 5.1
RMSE of approximations by uniform LS, `minvar`*, and* `minvar` *LS.*

Vertices	Uniform LS	`minvar`	`minvar` LS
2^2	1.79801e-01	1.79818e-01	1.79801e-01
3^2	9.85617e-02	4.56123e-02	4.48297e-02
4^2	4.45294e-02	1.92465e-02	1.89605e-02
5^2	2.47517e-02	1.38427e-02	1.36523e-02
6^2	1.55282e-02	7.35190e-03	7.25304e-03
7^2	1.05933e-02	5.40420e-03	5.34596e-03
8^2	7.63439e-03	4.63040e-03	4.56706e-03
9^2	5.84815e-03	3.38636e-03	3.34218e-03
10^2	4.53419e-03	2.96688e-03	2.92793e-03
11^2	3.71421e-03	2.50778e-03	2.47792e-03
12^2	2.99647e-03	2.34302e-03	2.32279e-03
13^2	2.49846e-03	1.86386e-03	1.84316e-03

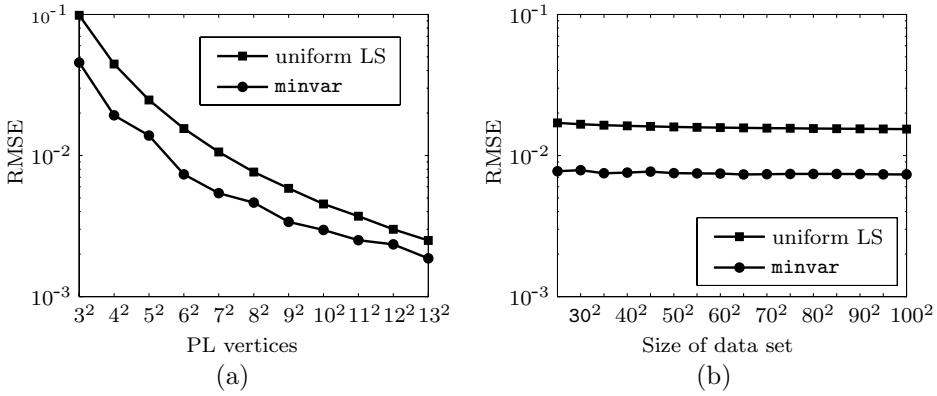

FIG. 5.2. *RMSE performance of* `minvar` *compared to a least squares continuous PL approximation on a uniform triangulation.* (a) *Vertices in approximation vs. RMSE for approximations to the* 80×80 *data set.* (b) *Density of training data set vs. RMSE on validation data for approximations with* 6^2 *vertices.*

Figure 5.2(a) show the root mean square error (RMSE) of the approximations as a function of the number of vertices. For 2^2 domain vertices, all of them are on the corners of the domain and must remain fixed, so `minvar` can change only the range vertices. Since `minvar` is not guaranteed to give the least squares continuous PL approximation for a given triangulation, it is not surprising that the uniform LS approximation's RMSE is slightly lower than `minvar`'s in this case. The RMSE difference between `minvar` and `minvar` LS approximations is less than 2%. Least squares could be applied as a post processing step to `minvar`, but since the differences are relatively small, this might not be necessary in application settings. From the triangulations of the `minvar` approximations in Figure 5.1, the domain triangulations move farther for lower numbers of vertices. As the number of vertices increases, the triangulations visually seem to deviate less from the initial uniform triangulation. The RMSE performance of `minvar` reflects this, giving the biggest reductions in RMSE as compared to uniform LS for triangulations with 3^2 to 6^2 vertices. Since this study uses initial conditions in which the vertices are on a uniform grid, approximations with large numbers of vertices may be getting caught in local minima near their initial condi-

tions. In this case, performance could be improved by refining converged less dense approximations to create initial conditions for the dense approximations [39].

In the second set of experiments, PL approximations with 6^2 vertices were trained using data sets of varying size. The PL approximations were chosen to have 6^2 vertices because, as mentioned above, this is in the region of sizes where the performance gains from `minvar` are greatest. The data sets were generated by sampling the test function on a uniform $n \times n$ grid, $n = 25, 30, 35, \ldots, 100$. The approximations were compared using a validation data set generated by evaluating the test function at 1000 points sampled from a uniform probability distribution over the domain. Figure 5.2(b) shows the validation set RMSE for `minvar` and uniform LS. With a 20×20 data set, `minvar` fails to run with a 6^2 vertex approximation, because as the vertices move, several simplices shrink to the point that they do not contain enough data to make the linear least squares approximation unique. Data sparsity is a serious issue in this type of local approximation.

From this example, `minvar` shows marked benefit when the approximation has relatively few vertices compared to the complexity of the test function. We expect that `minvar`'s performance on higher order (more vertices) approximations could be improved by seeding initial conditions based on lower order approximations. The algorithm produces a consistent approximation to variously sized data sets, so long as there is enough data for it to run.

6. Conclusion. Numerous applications require the simultaneous approximation of a function and its inverse from a set of discrete data. While there is a substantial literature on function approximation, very little of it addresses the constraint of invertibility. The inverse of a continuous PL function can be computed in closed form, which is ideal for applications requiring the approximation of a function and its inverse. In the PL literature, the partition is often fixed, in which case the minimum squared error approximation problem is linear-in-parameters. The problem becomes nonlinear-in-parameters when the domain partition is allowed to move.

The `minvar` algorithm is a novel method for computing continuous PL approximations to data. Rather than using gradient descent on the parameters, `minvar` takes advantage of the structure of PL functions, iteratively moving the vertices of the approximation based on local least squares fits. The `minvar` algorithm is proven to converge locally in the special case when the data generating function is itself PL and available directly rather than through discrete data. While this result seems very natural, complexity in the proof arises from the interaction of the domain triangulations of the data generating function and its approximation. Indeed, many difficulties in constructing PL approximations, such as triangulation tangles, arise primarily from the combinatorial properties of PL functions. For general approximation problems, this added complexity may cause PL approximation to appear less attractive than other nonlinear-in-parameters approximation techniques, but for an important subset of applications the PL function's closed form invertibility makes the combinatorial complexity cost effective.

The present work can be extended in several directions. The analysis here addresses the approximation rather than the estimation version of the problem. A formal connection between the approximation and estimation versions could be constructed in the appropriate statistical framework. Similarly, there is no study of the effects of noise on the convergence properties of `minvar`. Since data from the data generating function are used only for computing the least squares approximations, and least squares has good noise properties, the authors expect that the `minvar` al-

gorithm will also have good noise properties. The present analytical work considers only PL data generating functions. A desirable extension would be to show that the algorithm converges to the locally best PL approximation, given that one exists. For scalar functions there are generalized convexity conditions that characterize existence and uniqueness of (possibly discontinuous) PL approximations [4, 18, 26], but the authors are unaware of similar results for dimensions greater than one or with continuous piecewise approximations. Numerical experience suggests that `minvar` does have good convergence properties on other types of functions. Practical application of `minvar` raises a number of challenging issues such as constrained motion of vertices on the boundary of the domain, methods to avoid and correct triangulation tangling, and retriangulation or adaptation of the combinatorial parameters of the PL approximation. Due to space limitations, these topics will be addressed in detail in a subsequent paper on the use of `minvar` in engineering applications.

Appendix A. Geometric properties of simplices and triangulations. Proving properties of the `minvar` algorithms requires some insight into the underlying geometric structures upon which PL functions are constructed. This appendix presents a number of geometry facts used in the proof of convergence. Proofs of these facts are available in [20].

A.1. Barycentric coordinates as distances. It is often convenient to represent points in $\mathbb{R}^d$ using barycentric coordinates with respect to the vertices of some d-simplex. Let $p_1, \ldots, p_{d+1}$ be affinely independent points in $\mathbb{R}^d$. Let $x \in \mathbb{R}^d$ and let $\bar{\alpha} = \begin{bmatrix}\alpha_1 & \cdots & \alpha_{d+1}\end{bmatrix}^{\mathrm{T}}$ be such that $x = \sum_{i=1}^{d+1} \alpha_i p_i$ and $\sum_{i=1}^{d+1} \alpha_i = 1$. $\bar{\alpha}$ are called the *barycentric coordinates* of x with respect to $p_1, \ldots, p_{d+1}$. If $\bar{\alpha} \in \Delta_{d+1} := \{\bar{\alpha} \in \mathbb{R}^{d+1} | \alpha_i \geq 0, \sum_{i=1}^{d+1} \alpha_i = 1\}$, then $x \in \mathrm{conv}(p_1, \ldots, p_{d+1})$, and Δ_d is called the standard d-simplex.

The distance between a point $x \in \mathbb{R}^d$ and a nonempty set $A \subseteq \mathbb{R}^d$ is well defined [41] and written as $\delta(x, A) := \inf_{z \in A} \|x - z\|$. Let $H_i = \mathrm{aff}(\{p_1, \ldots, p_{d+1}\} - \{p_i\})$, the hyperplane opposing p_i. Let (a_i, c_i) be an implicit representation for H_i, that is, $H_i = \left\{z \in \mathbb{R}^d | {a_i}^{\mathrm{T}} z + c_i = 0\right\}$. The distance of a point $x \in \mathbb{R}^d$ to H_i is given by

$$\delta(x, H_i) = \frac{\left|{a_i}^{\mathrm{T}} x + c_i\right|}{\|a_i\|}.$$

We define $\delta_s(x, H_i)$ as the *signed distance* of x from H_i. That is, $|\delta_s(x, H_i)| = \delta(x, H_i)$, and $\delta_s(x, H_i) > 0$ for x on the same side of H_i as p_i, and $\delta_s(x, H_i) < 0$ for x on the opposite side of H_i as p_i.

By the following claim, the barycentric coordinates of x can be interpreted as the scaled distances of x from hyperplanes $H_1, \ldots, H_{d+1}$.

CLAIM 1. *Let $x \in \mathbb{R}^d$. Let $\bar{\alpha} = \begin{bmatrix}\alpha_1 & \cdots & \alpha_{d+1}\end{bmatrix}^{\mathrm{T}}$ be the barycentric coordinates of x with respect to the affinely independent points $p_1, \ldots, p_{d+1} \in \mathbb{R}^d$. Then $\delta_s(x, H_i) = \alpha_i \delta(p_i, H_i)$.*

Similarly, the barycentric coordinates of x can be used to measure the distance to an affine subspace that is the intersection of two or more of $H_1, \ldots, H_{d+1}$.

CLAIM 2. *Let $x \in \mathbb{R}^d$. Let $\bar{\alpha} = \begin{bmatrix}\alpha_1 & \cdots & \alpha_{d+1}\end{bmatrix}^{\mathrm{T}}$ be barycentric coordinates of x with respect to the affinely independent points $p_1, \ldots, p_{d+1}$. The distance from x to the affine subspace $A = \mathrm{aff}(\{p_1, \ldots p_k\})$ is given by $\delta(x, A) = {\bar{\alpha}_s}^{\mathrm{T}} G \bar{\alpha}_s$, where $\bar{\alpha}_s = \begin{bmatrix}\alpha_{k+1} & \alpha_{k+2} & \cdots & \alpha_{d+1}\end{bmatrix}^{\mathrm{T}}$ and $G \in \mathbb{R}^{(d-k+1)\times(d-k+1)}$ is a positive definite matrix whose entries depend only on $p_1, \ldots, p_k$.*

The explicit form for G, which derives from the Gram determinants used in the proof, is provided in [20].

A.2. Properties of the dilation. The dilation arises in Lemma 4.2 in measuring how the approximation's mismatched triangulation affects the least squares affine approximations. The first claim establishes the general properties of the dilation, and the second claim establishes a relationship between incident d-simplices that is required for the proof of Lemma 4.2.

CLAIM 3. *Let $S \subseteq \mathbb{R}^d$ be a d-simplex with vertices $p_1, \ldots, p_{d+1}$. Let*

$$\epsilon_{\min} = -\left(\sum_{i=1}^{d+1} 1/\delta(p_i, H_i) \right)^{-1}, \tag{A.1}$$

where H_i is the opposing hyperplane to p_i, as defined above. $\mathcal{D}(S, \epsilon_{\min})$ is a single point. For $\epsilon > \epsilon_{\min}$, $\mathcal{D}(S, \epsilon)$ is a d-simplex, with faces parallel to and translated distance $|\epsilon|$ away from the faces of S. For $\epsilon_{\min} \leq \epsilon \leq 0$, $\mathcal{D}(S, \epsilon) \subseteq S$, while for $\epsilon \geq 0$, $S \subseteq \mathcal{D}(S, \epsilon)$.

CLAIM 4. *Let $S_a, S_b \subseteq \mathbb{R}^d$ be incident d-simplices, that is, $S_a \cap S_b = s_{ab}$, where $s_{ab} \leq S_a, S_b$ is a $(k-1)$-simplex, $1 \leq k < d-1$. Let $\operatorname{vert} S_a = \{p_1, \ldots, p_{d+1}\}$, $\operatorname{vert} S_b = \{p_1, \ldots, p_k, q_{k+1}, \ldots, q_{d+1}\}$, and $\operatorname{vert} s_{ab} = \{p_1, \ldots, p_k\}$. Let $A = \operatorname{aff} s_{ab}$. Then $\exists \kappa_{a,b} > 0$ such that for all $\epsilon > 0$, if $x \in \mathcal{D}(S_a, \epsilon) \cap S_b$, then $\delta(x, A) < \kappa_{a,b}\epsilon$.*

A.3. Properties of PL functions. Claims pertaining to the parameterization, continuity, and invertibility of PL functions are provided below.

CLAIM 5. *Let $f_{\mathcal{P}}$ be a continuous PL function parameterized by $(P, Q, \mathcal{S})$. Let $S_i, S_j \in \mathcal{T}(P, \mathcal{S})$ be such that $S_i \cap S_j \neq \emptyset$. Let $S_i \cap S_j$ be a $(k-1)$-simplex, so then S_i and S_j share k vertices in common. Let $\operatorname{vert} S_i = \{p_{i_1}, \ldots, p_{i_{d+1}}\}$, $\operatorname{vert} S_j = \{p_{i_1}, \ldots, p_{i_k}, p_{j_{k+1}}, \ldots, p_{j_{d+1}}\}$, and $\operatorname{vert} S_i \cap S_j = \{p_{i_1}, \ldots, p_{i_k}\}$. Then,*

1. *$f_{\mathcal{P}}$ in S_i and S_j is given by*

$$f_{\mathcal{P}}\big|_{S_i}(x) = U_i V_i^{-1} \begin{bmatrix} x^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}}, \qquad f_{\mathcal{P}}\big|_{S_j}(x) = U_j V_j^{-1} \begin{bmatrix} x^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}},$$

$$U_i = \begin{bmatrix} q_{i_1} & \cdots & q_{i_{d+1}} \end{bmatrix}, \qquad U_j = \begin{bmatrix} q_{i_1} & \cdots & q_{i_k} & q_{j_{k+1}} & \cdots & q_{j_{d+1}} \end{bmatrix},$$

$$V_i = \begin{bmatrix} p_{i_1} & \cdots & p_{i_{d+1}} \\ 1 & & 1 \end{bmatrix}, \qquad V_j = \begin{bmatrix} p_{i_1} & \cdots & p_{i_k} & p_{j_{k+1}} & \cdots & p_{j_{d+1}} \\ 1 & & 1 & 1 & & 1 \end{bmatrix}.$$

Then $U_i V_i^{-1} \begin{bmatrix} x^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} = U_j V_j^{-1} \begin{bmatrix} x^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}}$ for $x \in \operatorname{aff}(S_i \cap S_j)$.

2. *In nonhomogeneous form, let $f_{\mathcal{P}}\big|_{S_i}(x) = A_i x + b_i$ and $f_{\mathcal{P}}\big|_{S_j}(x) = A_j x + b_j$. Let L be the linear subspace parallel to $\operatorname{aff}(S_i \cap S_j)$. Then $L \subseteq \mathcal{N}(A_i - A_j)$.*

CLAIM 6 (continuity in vertices). *Consider two continuous PL functions, $f^*_{\mathcal{P}}$ parameterized by $\mathcal{P}^* = (P^*, Q^*, \mathcal{S}^*)$ and $f_{\mathcal{P}}$ parameterized by $(P, Q, \mathcal{S}^*)$, such that $|\mathcal{T}(P, \mathcal{S}^*)| = |\mathcal{T}(P^*, \mathcal{S}^*)|$. Let $c > 0$. There exists $c' = c'(\mathcal{P}^*, c)$ such that, for $0 < \epsilon < r_3$, where*

$$r_3 = \min_{S^* \in \mathcal{T}(P^*, \mathcal{S}^*)} \sup \left\{ \epsilon \,\middle|\, \mathcal{D}(S^*, \epsilon) \subseteq |\operatorname{Cl} \operatorname{St} S^*| \right\}. \tag{A.2}$$

If $\|p_i - p_i^*\| < \epsilon$ and $\|q_i - q_i^*\| < c\epsilon$ for all i, then

$$\|f_{\mathcal{P}} - f^*_{\mathcal{P}}\|_\infty < c'\epsilon.$$

That is, a continuous PL function is continuous in its vertices.

The explicit form for c' is provided in [20].

CLAIM 7. *Let $f_{\mathcal{P}}$ be a PL function parameterized by $\mathcal{P} = (P, Q, \mathcal{S})$. If $\mathcal{T}(Q, \mathcal{S})$ is also a triangulation, then the PL function is invertible on its range, and the inverse, ${f_{\mathcal{P}}}^{-1}$, is parameterized by $\mathcal{P}^{-1} = (Q, P, \mathcal{S})$.*

REFERENCES

[1] C. G. ATKESON, A. W. MOORE, AND S. SCHAAL, *Locally weighted learning*, Artificial Intelligence Rev., 11 (1997), pp. 11–73.

[2] M. J. BAINES, *Algorithms for optimal discontinuous piecewise linear and constant L_2 fits to continuous functions with adjustable nodes in one and two dimensions*, Math. Comp., 62 (1994), pp. 645–669.

[3] A. R. BARRON, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945.

[4] D. L. BARROW, C. K. CHUI, P. W. SMITH, AND J. D. WARD, *Unicity of best mean approximation by second order splines with variable knots*, Math. Comp., 32 (1978), pp. 1131–1143.

[5] K. Q. BROWN, *Voronoi diagrams from convex hulls*, Inform. Process. Lett., 9 (1979), pp. 223–228.

[6] E. W. CHENEY, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966.

[7] C. K. CHUI, *Multivariate Splines*, CBMS-NSF Regional Conf. Ser. in Appl. Math 54, SIAM, Philadelphia, 1988.

[8] N. J. COWAN, J. D. WEINGARTEN, AND D. E. KODITSCHEK, *Visual servoing via navigation functions*, IEEE Trans. Rob. Autom., 18 (2002), pp. 521–533.

[9] P. J. DAVIS, *Interpolation and Approximation*, Dover, New York, 1975.

[10] C. DE BOOR, K. HÖLLIG, AND S. RIEMENSCHNEIDER, *Box Splines*, Springer-Verlag, New York, 1993.

[11] N. DYN, D. LEVIN, AND S. RIPPA, *Algorithms for the construction of data dependent triangulations*, in Algorithms for Approximation, II, Chapman and Hall, London, 1990, pp. 185–192.

[12] N. DYN, D. LEVIN, AND S. RIPPA, *Data dependent triangulations for piecewise linear interpolation*, IMA J. Numer. Anal., 10 (1990), pp. 137–154.

[13] H. EDELSBRUNNER, *Geometry and Topology for Mesh Generation*, Cambridge University Press, Cambridge, UK, 2001.

[14] H. EDELSBRUNNER AND N. SHAH, *Incremental topological flipping works for regular triangulations*, Algorithmica, 15 (1996), pp. 223–241.

[15] S. FIORI, *Probability density function learning by unsupervised neurons*, Internat. J. Neural Systems, (2001), pp. 399–417.

[16] J. H. FRIEDMAN, *Multivariate adaptive regression splines*, Ann. Statist., 19 (1991), pp. 1–141.

[17] S. FURRY AND J. KAINZ, *Rapid algorithm development applied to engine management systems*, in Proceedings of the 1998 SAE International Congress & Exposition, Detroit, MI, 1998.

[18] R. C. GAYLE AND J. M. WOLFE, *Unicity in piecewise polynomial L_1-approximation via an algorithm*, Math. Comp., 65 (1996), pp. 647–660.

[19] G. E. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.

[20] R. E. GROFF, P. P. KHARGONEKAR, AND D. E. KODITSCHEK, *A Local Convergence Proof for the Minvar Algorithm for Computing Continuous Piecewise Linear Approximations*, Technical report CSE-TR-464-02, University of Michigan, Ann Arbor, MI, 2002.

[21] R. E. GROFF, P. P. KHARGONEKAR, D. E. KODITSCHEK, T. T. THIERET, AND L. MESTHA, *Modeling and control of color xerographic processes*, in Proceedings of the 38th IEEE Conference on Decision and Control, Vol. 2, Phoenix, AZ, 1999, pp. 1697–1702.

[22] R. E. GROFF, D. E. KODITSCHEK, AND P. P. KHARGONEKAR, *Piecewise linear homeomorphisms: The scalar case*, in Proceedings of the International Joint Conference on Neural Networks, Como, Italy, 2000.

[23] R. E. GROFF, D. E. KODITSCHEK, P. P. KHARGONEKAR, AND T. T. THIERET, *Representation of color space transformations for effective calibration and control*, in IS&T's NIP16: International Conference on Digital Printing Technology, Society for Imaging Science and Technology, Vancouver, BC, Canada, 2000, pp. 255–260.

[24] N. HAI, J. NI, AND J. YUAN, *Generalized model formulation technique for error synthesis and error compensation on machine tools*, in Proceedings of the NAMRX XXVI Conference, Society of Manufacturing Engineers, Atlanta, GA, 1998.

[25] J. B. KIOUSTELIDIS, *Optimal segmented approximations*, Computing, 24 (1980), pp. 1–8.

[26] J. B. Kioustelidis, *Optimal segmented polynomial L_s-approximations*, Computing, 26 (1981), pp. 239–246.
[27] D. E. Koditschek and E. Rimon, *Robot navigation functions on manifolds with boundary*, Adv. in Appl. Math., 11 (1990), pp. 412–442.
[28] C. L. Lawson, *Properties of n-dimensional triangulations*, Comput. Aided Geom. Design, 3 (1986), pp. 231–246.
[29] G. G. Lorentz, *Approximation of Functions*, Holt, Rhinehart and Winston, New York, 1966.
[30] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
[31] M. H. Raibert, *Legged Robots That Balance*, MIT Press, Cambridge, MA, 1986.
[32] E. Rimon and D. E. Koditschek, *The construction of analytic diffeomorphisms for exact robot navigation on star worlds*, Trans. Amer. Math. Soc., 327 (1991), pp. 71–115.
[33] E. Rimon and D. E. Koditschek, *Exact robot navigation using artificial potential fields*, IEEE Trans. Rob. Autom., 8 (1992), pp. 501–518.
[34] W. J. Schwind and D. E. Koditschek, *Approximating the stance map of a 2 dof monoped runner*, J. Nonlinear Sci., 10 (2000), pp. 533–568.
[35] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*, CRC Press, Boca Raton, FL, 1995.
[36] E. H. Spanier, *Algebraic Topology*, McGraw-Hill, New York, 1966.
[37] M. Spivak, *Calculus on Manifolds*, Westview Press, Boulder, CO, 1965.
[38] C. J. Stone, M. H. Hansen, C. Kooperberg, and Y. K. Truong, *Polynomial splines and their tensor products in extended linear modeling*, Ann. Statist., 25 (1997), pp. 1371–1470.
[39] Y. Tourigny and M. J. Baines, *Analysis of an algorithm for generating locally optimal meshes for L_2 approximation by discontinuous piecewise polynomials*, Math. Comp., 66 (1997), pp. 623–650.
[40] Y. Tourigny and F. Hülsemann, *A new moving mesh algorithm for the finite element solution of variational problems*, SIAM J. Numer. Anal., 35 (1998), pp. 1416–1438.
[41] R. Webster, *Convexity*, Oxford University Press, New York, 1994.
[42] J. C. Ziegert and P. Datseris, *Basic considerations for robot calibration*, International Journal of Robotics and Automation, 4 (1989), pp. 158–166.