

# Piecewise polynomial Reconstruction of Scalar Fields from Simplified Morse-Smale Complexes

Léo Allemand-Giorgis, Georges-Pierre Bonneau, Stefanie Hahmann

**Abstract** Morse-Smale complexes have been proposed to visualize topological features of scalar fields defined on manifold domains. Herein, three main problems have been addressed in the past: (a) efficient computation of the initial combinatorial structure connecting the critical points; (b) simplification of these combinatorial structures; (c) reconstruction of a scalar field in accordance to the simplified Morse-Smale complex. The present paper faces the third problem by proposing a novel approach for computing a scalar field coherent with a given simplified MS complex that privileges the use of piecewise polynomial functions. Based on techniques borrowed from shape preserving design in Computer Aided Geometric Design, our method constructs the surface cell by cell using piecewise polynomial curves and surfaces. The benefit and limitations of using polynomials for reconstruction surfaces from topological data are presented in this paper.

## 1 Introduction

Morse-Smale (MS) complexes are combinatorial structures encoding topological features of smooth scalar fields by connecting their critical points. The discrete Morse theory [11] has been used to define MS complexes for scalar data defined on meshes.

Because of their ability to characterize the topology of level sets, MS complexes have found many applications in visualization [16]. Each cell in a MS complex is a set of integral lines of a discrete gradient flow (the equivalent of streamlines of a smooth gradient field) connecting a critical point with some value to another critical point with a higher value. Along an integral line, the data is monotonic, strictly increasing. Inside a 2d-cell, there are no critical points, and the level sets have the topology of an open interval. Previous works have their focus on the efficient computation of MS complexes [25, 26], on the simplification and hierarchical represen-

---

Université Grenoble Alpes, CNRS, LJK, INRIA, Grenoble, France e-mail: name@inria.fr

tation of MS complexes [5, 3] or on the extraction of features using MS complexes [16]. The simplification of a MS complex is a combinatorial operation consisting in deleting two adjacent 0d-cells in the complex. All adjacent cells to the deleted nodes affected by the simplification are either deleted or unified. Since the simplification is a purely combinatorial operation, for visualization purposes it is thus often required to compute new scalar data that correspond to the simplified complex. The scalar field reconstruction problem is the topic of the present work. The main requirements on the scalar field reconstructed from a simplified MS complex are the following

- interpolation of the scalar field values at the critical points
- bijection between the nodes of the MS complex and the critical points
- bijection between arcs of the MS complex and 1-cells.

This problem has been solved in the past by applying to each 1d- and 2d-cell a discrete laplacian smoothing in the original mesh with the addition of constraints in order to ensure the monotonicity of data inside the cells [3, 30]. In contrary to previous mesh-based approaches, the present work investigates the use of smooth surface representations.

In the fields of Computer Aided Geometric Design and Approximation Theory *shape preserving methods* seek to approximate a scalar or geometric function while preserving some "shape properties" such as the sign, the monotonicity or the convexity of the given data. Most of these methods produce piecewise polynomial Bézier and B-spline functions, parametric curves and surfaces.

In the present work we propose to reconstruct scalar fields from simplified MS complexes in 2D, based on shape-preserving methods. Instead of using discrete laplacian smoothing on the original mesh, our approach is to use piecewise polynomial curves and surfaces. Specifically, we first approximate data along 1d-cells by computing monotonic B-splines [19], and we then show how to fill-in 2d-cells with monotonic triangular Bézier patches based on our previous work on monotonic interpolation of gridded data [2]. The final resulting scalar field is a piecewise polynomial approximation of the original data that is coherent with the simplified MS complex. We present different approaches for parameterizing and computing these piecewise polynomial surfaces.

## 2 Related Work

Topology-based / Computational topology methods have become very effective for analysis, visualization and simplification of features in scientific data sets and geometric shape models. Most of these methods make use of results from Morse theory which dates back to the 19<sup>th</sup> century. It became useful in scientific visualization and computer graphics with a first efficient algorithm by Edelsbrunner et. al [5] for computing the MS complex for piecewise linear scalar fields. The MS complex is a topological representation of a scalar function which decomposes the domain of a function into regions having uniform gradient flow. It consists of the critical

points (minima, maxima, saddles) and the separatrices connecting them. Efficient algorithms to compute the MS complex for 2D and 3D scalar fields can be found in [25, 17, 18, 15, 14].

Topological simplification is used for denoising, feature discrimination and smoothing of scalar fields. Algorithms apply, inter alia, the notion of topological persistence [4] for iteratively cancelling pairs of critical points [6, 3] in MS complexes.

Closely related to topology-driven simplification algorithms are methods that reconstruct a function obeying the simplified topological data. A setup similar to ours are [3, 30] in the sense that the MS complex is simplified and a corresponding scalar function is computed by fully constraining the simplified complex. In [3] a MS complex is iteratively simplified by cancelling pairs of critical points ordered by persistence resulting in a hierarchical representation. The discrete embedding of boundary and the interior of the cells is smoothly reconstructed cell by cell using constrained Laplacian optimization. In [30] a constrained bi-Laplacian optimization is applied globally on the entire domain. Bi-Laplacians increase the order of continuity from  $C^0$  to  $C^1$  but lose the maximum principle, i.e. lose the important monotonicity property of the resulting function inside the MS cells. Inequality constraints are thus added to enforce the scalar field to be monotonic inside each cell. In [20] only the position and value of extremas are constrained to compute smooth shapes by minimizing a non-linear functional similar to [30]. They further provide an explicit control over the topology of a scalar field. As an extension of [30, 20] Günther et al. [12] present a method for reconstructing iteratively a smooth scalar field from a set of selected extrema by combining non-linear optimization based on a monotonicity graph with topological simplification. Both methods [20, 12] however do not constrain the entire MS complex, but use it to satisfy the monotonicity constraints in the optimization.

However, a big challenge of these numerical reconstruction approaches is the stability in the optimization process. This may create additional critical points in the output preventing it from strictly conforming to the input constraints. Additionally, the overall optimization process might be computationally expensive resulting in extensive running times. Tierny et al. [29, 28] propose an interactive combinatorial algorithm to edit a scalar field on a surface with a user-prescribed extrema that does not build on the computation of a topological structure such as MS complex or contour tree.

The major differences to our work are twofold: first, we provide an explicit polynomial representation of the surface patches and the cell boundaries.

This explicit representation has two advantages: it allows for an arbitrarily dense sampling of the resulting function. It further allows for exact evaluation and computation of differential properties of the scalar function for post-processing applications. . Second, our resulting surface exactly conforms to the input MS complex, meaning that no undesired additional critical points occur as it may be the case with numerical optimization methods.

Our method does not depend on the density/size of the discretization of the input function.

### 3 Background and Overview

In this section, the required theoretical background on MS complexes is given, for scalar fields defined in a planar domain. First, Section 3.1 presents a summary of Morse theory, then Section 3.2 explains how MS complexes can be simplified. For further background about Morse theory, the reader is referred to [22, 27].

#### 3.1 Morse theory

Let  $f : D \rightarrow \mathbb{R}$  be a differentiable function, where  $D$  is a bounded open subset of  $\mathbb{R}^2$ . A point  $p \in D$  is called a non-degenerated critical point of  $f$  if  $\nabla f(p) = 0$  and the determinant of the Hessian of  $f$  in  $p$  is non zero.  $f$  is called a Morse function if its critical points are non degenerated. There are three kinds of critical points depending on the sign of the two eigenvalues of the Hessian matrix. The number of negative eigenvalues is called the index of the critical point. If the index is zero then  $p$  is a minimum, if the index is one, then  $p$  is a saddle point and if all eigenvalues are negative, then  $p$  is a maximum.

A  $C^1$  curve  $l : \mathbb{R} \rightarrow D$  is called an integral line of  $f$  if  $(\forall t \in \mathbb{R}) \frac{\partial}{\partial t} l(t) = \nabla f(l(t))$ . It is a path which follows the gradient of the function, so the values of  $f$  along this line parameterized by  $t$  are strictly increasing. The limit of  $l$  when  $t \rightarrow -\infty$  (resp.  $t \rightarrow +\infty$ ) is called the origin (resp. destination) of  $l$ . The origin and the destination of an integral line are critical points of  $f$ , and  $index(\text{destination}) > index(\text{origin})$ . The union of a critical point  $\{p\}$  with the set of all points of integral lines which share the origin  $p$  is called the ascending manifold of  $p$ . Analogously, the union of a critical point  $\{q\}$  with the set of all points of integral lines which have the same destination  $q$  is the descending manifold of  $q$ . The set of all ascending manifolds and the set of all descending manifolds form two distinct partitions of  $D$ .

The MS complex is the partition of the domain  $D$  using the sets of integral lines sharing the same origin and destination. In other words, each set in this partition is the intersection of an ascending and a descending manifold. It has the structure of a CW-complex, with cells of three types, as illustrated in Figure 1a :

- 0-cells : these cells are critical points of the function,
- 1-cells : these cells are specific integral lines called separatrices,
- 2-cells : these cells are always bounded by a cycle of four 1-cells. Two of these 1-cells link a minimum with two saddles, and the two other 1-cells link the two saddles with a maximum. There is no critical point in the interior of a 2-cell.

In the present paper, we do not work with differentiable functions, but with discrete scalar fields sampling a differentiable function at the vertices of a triangulation  $D$ . We therefore rely on Forman's discrete Morse Theory [11] which generalizes MS complexes using a combinatorial gradient vector field. In this setting, integral lines and cells of the MS complex are union of domain simplices (e.g. vertices, edges and triangles).

### 3.2 Simplification

To explain the simplification of MS complexes, we use the canonical mapping from a MS complex to an undirected graph which maps 0-cells to nodes and 1-cells to edges. The 2-cells then correspond to cycles in the graph. The simplification of the MS complex is obtained by sequentially removing pairs of adjacent nodes in the graph [5, 13]. The edges of least persistence are removed first [7, 23]. Let  $\{p, q\}$  be an edge in the graph such that  $index(q) = index(p) + 1$ . Removing the two nodes in the graph implies the removal of all edges incident to them, thereby creating a cycle of length strictly larger than four. New edges are then inserted between each node previously adjacent to  $p$  with the same index value than  $q$  except  $q$  and each node previously adjacent to  $q$  with the same index value than  $p$  except  $p$ . Each of these new edges replace three removed edges. These new edge insertions ensure that all new cycles are of length four and connect one minimum, one saddle, one maximum and a second saddle. This process is illustrated in Figure 1a and 1b with  $index(p) = 1$ , i.e. when  $p$  is a saddle point and  $q$  is a maximum. In this case a new edge is inserted between each saddle point that was adjacent to  $q$  and distinct from  $p$ , and the maximum that was adjacent to  $p$  and distinct from  $q$ .

The previous process is purely combinatorial. It builds a new simplified topology from the topology of the MS complex, but does not produce a new embedding. A trivial embedding is obtained by associating to each new edge the union of the three 1-cells it replaces, and to each new cycle of union of previous 2-cells. But this trivial embedding does not correspond to a MS complex since there are critical points along the new 1-cells, as illustrated in Figure 1b. Therefore a new function has to be reconstructed from the original function and from the simplified topology such that the topology of the MS complex of this reconstructed function matches the simplified topology, as illustrated in Figure 1c. The function reconstruction is the subject of the present paper, as explained in the next section.

### 3.3 Overview

Our method takes as input an initial discrete scalar field  $f : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ , where  $D$  is a planar triangulation. The scalar field values  $f$ , also called height values, are given at the vertices of  $D$ . We also take as input the persistence threshold, so that in the simplified MS complex, the value of least persistence should be larger than the input threshold. In practice the persistence threshold is specified as a percentage of the maximum persistence. 0% mean no simplification and 100% means maximum simplification.

We first use the recent parallel algorithm by Shivashankar et al. [25] to compute the discrete MS complex of  $f$  and to simplify its topology, as explained in Section 3.2. Our method is also working with output from other algorithms [6, 3].

As explained in Section 3.2, the simplification of the MS Complex is a purely combinatorial process which removes critical points and produces a simplified

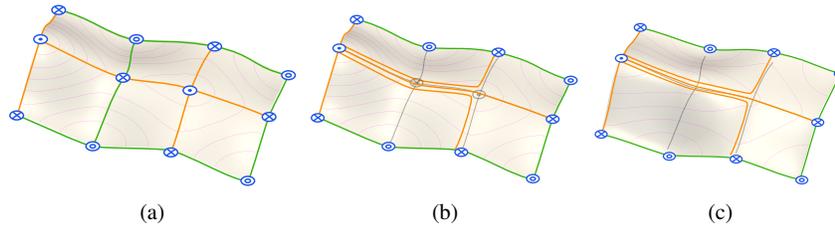


Fig. 1: (a) a scalar function on a rectangular domain with its MS complex. It has six 2-cells and seventeen 1-cells. The lines of steepest ascent (resp. descent) from the saddle points are shown in orange (resp. green) (b) a pair of adjacent critical points in gray is removed. As a consequence a simplified topology is produced with four new 2-cells and three new 1-cells. (c) a function is reconstructed which MS complex matches the simplified topology (see different isoline pattern).

topology connecting the remaining critical points. And the trivial embedding of this simplified topology consists in 1-cells and 2-cells inside which the original scalar field has critical points, as illustrated in Fig. 1b. Therefore we need to reconstruct a new scalar field whose MS complex matches exactly the simplified topology, i.e. it should be strictly monotonic along the simplified 1-cells, and should contain no critical points inside the simplified 2-cells.

In order to reconstruct such a scalar field we proceed in two steps. In the first step we traverse each 1-cell and compute a smooth uniform quadratic spline which is  $C^1$ -continuous, strictly monotonic and approximates the original scalar field values. This is described in Section 4. In the second step we reconstruct a scalar field inside each 2-cell, which interpolates the four monotonic uniform quadratic splines along its boundaries and contains no critical points, as explained in Section 5. To compute this scalar field, we first parameterize the domain of the 2-cell into the unit square, and then compute a monotonic piecewise cubic triangular Bézier surface defined in the unit square, which interpolates the height values of the uniform quadratic splines along the edges of the square.

As a consequence our reconstructed scalar field is defined by piecewise polynomial functions along the 1-cells and inside the 2-cells.

#### 4 Monotonic Smoothing of the 1-cells

The first part of our method consists in reconstructing smooth and strictly monotone curves along the new 1-cells. Before explaining the details we must point out two facts. First, the 1-cells of the simplified MS complex in the domain are known to have a jagged shape prohibiting a smooth surface reconstruction [3, 30]. Second, it happens often that two 1-cells, having the same maximum as destination

but distinct saddles as origin, merged before the maximum. In the other direction, two 1-cells, having the same minimum as origin and distinct saddles as destination, may share the same geometry before getting disconnected. Such merging or disconnection occur at special vertices called *junction points*. Figure 3-right shows two junction points between a minimum and three distinct saddles. These junction points segment the corresponding 1-cells into several pieces. We first compute new scalar field values at the junction points, so that subsequent processing may be done independently on each piece of 1-cells. Then for each piece we compute a piecewise quadratic uniform B-spline that is smooth, strictly monotonic and approximates the input scalar field values.

**1. Scalar field value at junction points.** If we follow a simplified 1-cell with junction points, the sequence of scalar field values at these junction points is not strictly monotone. Assigning iteratively a new value by linear interpolation of the critical end points as proposed in [30] may run in conflicts in case a junction point is shared by more than two 1-cells. We therefore propose to compute the junction point values globally by solving the following linear programming problem:

$$\begin{aligned} \sum |\tilde{f}(\mathbf{p}_j) - f(\mathbf{p}_j)| &\rightarrow \min \quad \text{subject to} \\ \max\_lower(\mathbf{p}_j) &\leq \tilde{f}(\mathbf{p}_j) \\ \tilde{f}(\mathbf{p}_j) &\leq \min\_larger(\mathbf{p}_j) \end{aligned}$$

where  $f$  is the initial scalar field,  $\tilde{f}$  the unknown value and  $\mathbf{p}_j$  the junction points in  $D$ .  $\max\_lower(p_j)$  (resp.  $\min\_larger(p_j)$ ) denotes the maximum (resp. minimum) height value of all junctions and critical points adjacent to  $p_j$  and are required to be lower (resp. larger) than  $\tilde{f}(p_j)$ .

The idea is to compute a new function value  $\tilde{f}(\mathbf{p}_j)$  as close as possible to  $f(\mathbf{p}_j)$  for all junction points, while satisfying the monotonicity requirements along all 1-cells.

**2. Monotone reconstruction of the scalar field along 1-cells.** We now compute a monotone piecewise polynomial function along each part of the 1-cells by applying a monotone B-spline smoothing [19] between two adjacent points (junction or critical) of the MS complex.

Let  $(\mathbf{x}_i)_{i=0\dots n}$  be the  $n+1$  vertices of a part of a 1-cell in the domain  $D$  between two junction or critical points  $x_0$  and  $x_n$ . Let  $(t_i)_{i=0\dots n}$  be the associated normalized arc length parameters, so that  $t_0 = 0 < t_1 < \dots < t_n = 1$ . New scalar field values have already been computed in  $\mathbf{x}_0$  and  $\mathbf{x}_n$ , and we have  $\tilde{f}(\mathbf{x}_0) < \tilde{f}(\mathbf{x}_n)$ .

We now compute a quadratic uniform B-spline function  $g_\alpha : [0, 1] \rightarrow \mathbf{R}$  that is strictly increasing, and approximates the input scalar field values. Let  $g_\alpha(t) = \sum_{i=0}^{k+1} \alpha_i N_i^2(t)$ , where  $t \in [0, 1]$ ,  $\alpha_i \in \mathbf{R}$  are the control points,  $0 = u_{-2} = u_{-1} = u_0 < \dots < u_k = u_{k+1} = u_{k+2} = 1$  the uniform knots and  $N_i^2$  the B-spline basis functions of degree 2 defined on the knot sequence  $\{u_j\}_{j=-2, \dots, k+2}$ . We set  $\alpha_0 = \tilde{f}(\mathbf{x}_0)$  and  $\alpha_{k+1} = \tilde{f}(\mathbf{x}_n)$  to ensure endpoint interpolation and compute the remaining coefficients  $\alpha_i$ ,  $i = 1, \dots, k$  as follows:

$$\sum_{i=1}^{n-1} |f(\mathbf{x}_i) - g_\alpha(t_i)| \rightarrow \min \quad \text{subject to} \quad (1)$$

$$g'_\alpha(u_0) = 0 \quad (2)$$

$$g'_\alpha(u_k) = 0 \quad (3)$$

$$g'_\alpha(u_i) \geq 0, \quad i = 1, \dots, k-1 \quad (4)$$

Condition (1) ensures approximation. (2) and (3) ensure zero gradients at the extremities and finally (4) ensures monotonicity. Figure 2 shows two results with the same input data but different numbers  $k$  of control-points.

As the reconstructed surface, see next section, will be composed of cubic triangular Bézier patches, we first convert the quadratic B-spline function  $g_\alpha$  into quadratic Bézier curves by repeated knot insertion and then apply a degree elevation to obtain  $k$  cubic Bézier curves. See [9] for more details about Bézier and B-spline curves. Note that the parameter  $k$  represents the number of curve segments along the spline curve. The higher  $k$  the more flexible is the curve and the better it is able to approximate a point set. In contrary, the smaller  $k$  is the smoother is the resulting curve. In our experiments we choose either  $k = 4$  or  $k = 7$ , see discussion in Section 6.

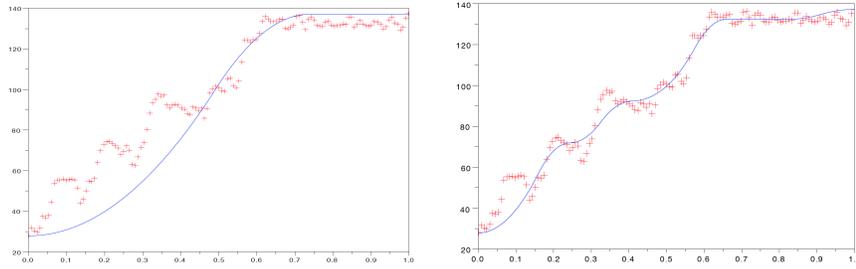


Fig. 2: Height values (red dots) are approximated by a *monotonic* quadratic B-spline function (blue curve). Left: Approximation with  $k = 4$  curve segments. Right: Approximation with  $k = 12$  curve segments.

**3. Smoothing of 1-cells in the domain.** As we aim to reconstruct the simplified MS complex with piecewise polynomial functions, we further smooth the jagged 1-cells in the domain by approximating in a least square sense each piece of 1-cell by a quadratic uniform parametric B-spline curve  $\mathbf{c}(t) = \sum_{i=0}^{k+1} \mathbf{c}_i N_i^2(t) \subset \mathbf{R}^2$ ,  $t \in [0, 1]$  with the same number of segments  $k$  and knot vector as for the monotone height function  $g_\alpha$ . Endpoints interpolation (critical points or junction points  $\mathbf{p}_i, \mathbf{p}_j$ ) is enforced by setting  $\mathbf{c}(0) = \mathbf{p}_i$  and  $\mathbf{c}(1) = \mathbf{p}_j$ . After knot insertion and degree elevation we get again  $k$  cubic Bézier curve segments. Figure 3 illustrates this smoothing for a real MS complex. The final monotone smooth height function can be written as  $(\mathbf{c}(t), g_\alpha(t))$ .

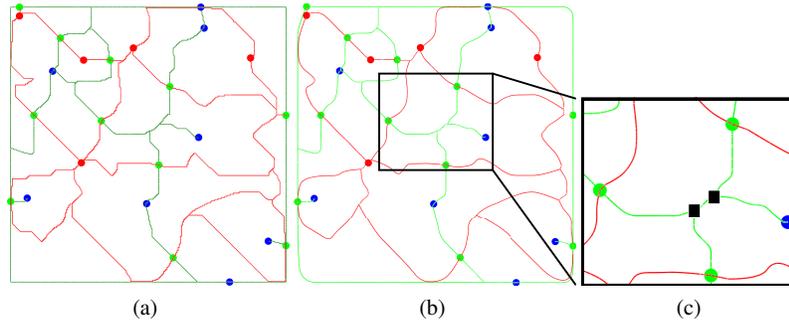


Fig. 3: Smoothing of the 1-cells in the domain. (a) original jagged 1-cells. (b) smoothing by uniform quadratic B-splines. (c) zoom into three 1-cells (green curves) descending from 3 distinct saddles (green dots) to the same minimum (blue dot). The three 1-cells merge at two junction points (black dots).

## 5 Monotone interpolation inside the 2-cells

In the previous section we have explained how to compute smooth monotonic piecewise polynomial functions that approximate the input scalar field values on each 1-cell of the simplified MS complex.

In this section we go further and described our method for computing for each 2-cell a piecewise polynomial surface that interpolates its boundary 1-cells and has no interior critical points. The construction is done independently for each 2-cell. We begin by parameterizing the 2-cell onto the unit square  $[0, 1]^2$  (Section 5.1). Then a monotone surface defined in the unit square is computed (Section 5.2). Combining the monotone surface and the parameterization, a monotone reconstruction of the 2-cell is computed, which interpolates the 1-cells, has the 0-cells as critical points and no other critical points.

### 5.1 Parameterization of a 2-cell

A 2-cell is a closed connected subdomain  $\Omega \subset D \subset \mathbb{R}^2$  bounded by four pieces of 1-cells. We need to parameterize  $\Omega$  in the unit square before computing the monotonic interpolant in the next section. In other words, a diffeomorphism  $\Phi : \Omega \rightarrow [0, 1]^2$  needs to be defined. Such a diffeomorphism maps functions without critical points on the unit square to functions without critical points on  $\Omega$ . If  $F$  is a function without critical points on the unit square, then  $\tilde{f} = F \circ \Phi : \Omega \rightarrow \mathbb{R}$  is a function without critical points on  $\Omega$ . We compute the parameterization  $\Phi$  using the following procedure.

First, we sample points on  $\partial\Omega$  by evaluating all Bézier curves bounding  $\Omega$  at a fixed number  $c$  of parameter values. Usually, we set  $c = 5$ . Then, a conforming Delaunay triangulation is computed which has all boundary points as constraints. The number of inserted points in the interior of  $\Omega$  depends on the expected triangle quality. We set the *triangle quality* parameter so that the size of inner triangles is similar to the average sample point distance on the boundary. We use the CGAL [1] library to compute the triangulation.

A parameter value  $(u, v) \in [0, 1]^2$  is computed for each vertex of the triangulation as follows. First, the boundary curves are mapped to the border of the unit square by associating the critical points (resp. junction points) with the corners of the square: the minimum is mapped to the parameter  $(0, 0)$ , the maximum to  $(1, 1)$  and the two saddle points to the remaining corners. In some cases  $\partial\Omega$  does not contain the minimum or the maximum of the 2-cell because the boundary 1-cells merge, as illustrated in the right image of Figure 3. In these cases, we map to the corner  $(0, 0)$  (resp.  $(1, 1)$ ) the junction point that is closest to the missing minimum (resp. maximum) along  $\partial\Omega$ . Between each corner along  $\partial\Omega$ , we assign the parameter that was used to sample the boundary Bézier curves. Then Floater's Mean Value Coordinates [10] algorithm is applied to compute a parameterization of the triangulation of  $\Omega$  in the unit-square, constrained by the parameter values on  $\partial\Omega$ . The parameterization process is illustrated in Figure 4.

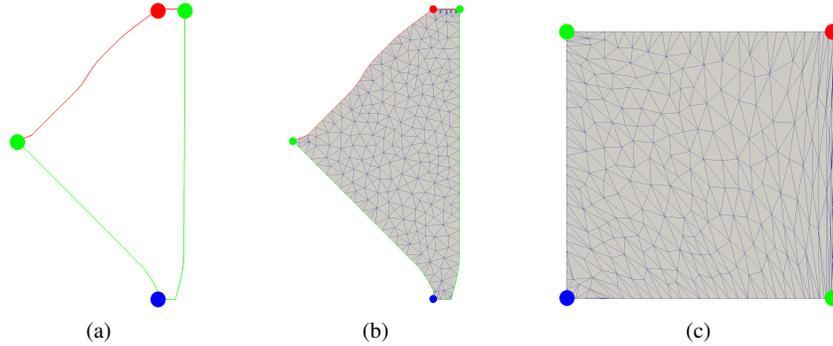


Fig. 4: (a) boundary of a 2-cell, the red point is a maximum, the blue point is a minimum and the two green points are saddles. (b) triangulation of the 2-cell (c) parameterization of the 2-cell

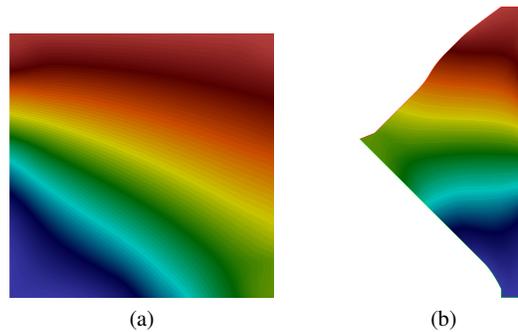


Fig. 5: (a) colormap of the piecewise cubic scalar field in the unit square. It has no critical points as can be seen from the isolines. (b) the same scalar field after reparameterization in the 2-cell domain.

## 5.2 Monotonic scalar field on the unit square

In Section 4 we have computed smooth 1-cells and new scalar field values strictly increasing along these 1-cells. In the previous section we have parameterized each 2-cell in the unit square. This parameterization maps the boundary 1-cells to the edges of the unit square. We now explain how to compute a scalar field on the unit square that interpolates the given increasing scalar field values on the edges of the domain and that has no critical point in the interior of the domain.

We apply techniques from [2], where an algorithm is developed for interpolating a grid of scalar values by a monotonic  $C^1$ -continuous piecewise cubic triangular Bézier surface. The interpolant of [2] is uniquely defined by interpolated scalar values and by two partial derivatives at the vertices of a uniform grid. Sufficient conditions on the interpolated values and the partial derivatives are derived to ensure that the interpolant has no interior critical points. More precisely, the interpolated values are supposed to be strictly increasing along the diagonals of the grid. Moreover if no partial derivative is known, then [2] provides an algorithm that computes partial derivative values leading to a monotonic interpolant.

The monotone interpolant [2] is applied as follows. If there is one or more junction points along one of the boundary 1-cells, the other 1-cells are subdivided<sup>1</sup>, so that the same number of cubic Bézier curves is used in the four 1-cells. We initialize a uniform grid that has in each direction a number of edges equals to the number of cubic Bézier curves. Along the boundaries of the grid, we prescribe the end-points of the cubic Bézier curves computed in Section 4 and the first derivative of these cubic Bézier curves at their end-points. To compute the scalar values to be interpolated at the interior points of the grid, a Laplace equation with the boundary values

<sup>1</sup> Subdivision of Bézier curves cuts them into several Bézier curves defining exactly the same curve [9]

as constraints is solved. In almost all cases the resulting values are strictly increasing along the grid diagonals. If not, we interpolate linearly the scalar values along the diagonal. Then, the interior partial derivatives are computed by using the results of [2]. In the end, we get a piecewise cubic triangular Bézier function that has no interior critical points, and that interpolates the strictly increasing 1-cells computed in Section 4. Such an interpolant is shown in Figure 5a. Its reparameterization in the 2-cell domain, as explained in Section 5.1, is shown in Figure 5b.

There is a particular case, called *pouch cells* which occurs when the two saddle points are merged. To handle this type of cells as well, we adapt our reconstruction process by only changing the parameterization step.

## 6 Results

We have implemented our algorithm and tested it on three types of functions. All the experiments were performed on a machine with an Intel Xeon W3520 CPU which provides 8 GB RAM. The MS complexes of the input function and its simplified versions were computed with [25] and preprocessed as explained in Section 3.3. We visualize the initial and reconstructed functions with color contour plots (linear color scale) augmented with a regularly sampled set of isolines. The MS complexes are visualized with the ascending (red curves) and descending 1-cells (green curves) together with the critical points, i.e. saddles (green dots), maxima (red dots) and minima (blue dots).

The first example is a hand-made scalar data set that has been created using Shepard’s scattered data interpolation method [24]. Given a set of  $N$  randomly chosen position and function values  $\{\mathbf{x}_i; f_i\}$ , the method computes a function  $F_\mu(\mathbf{x}) = \frac{\sum_{k=0}^N f_k \|\mathbf{x} - \mathbf{x}_k\|^{-\mu}}{\sum_{j=0}^N \|\mathbf{x} - \mathbf{x}_j\|^{-\mu}}$  such that  $F(\mathbf{x}_i) = f_i$  with zero-gradients at the input points when  $\mu > 1$ . Our so-called *Shepard* data set has then been built by taking randomly 50 data points and values and by evaluating  $F_2$  on a  $250 \times 250$  regular grid. The resulting function has at least these 50 extrema. A simplification of 25% is applied and the resulting MS complex reconstructed. Figure 6 shows the initial function (a) and its MS complex (b). A 25% simplification is applied (c). The middle and bottom line of Figure 6 show two different reconstructions of the same simplified MS complex. They differ in the number  $k$  of curve segments composing each piece of the 1-cells (see Section 4).

The role of  $k$  is twofold. On one hand it governs the smoothness and flexibility of the curves. The smaller  $k$  is, the smoother are the curves; but a higher value gives more flexibility to the curves and thus better approximates the height values. On the other hand, the number of Sibson patches is equal (or sometimes proportional) to  $k^2$  per 2-cell. The middle row shows a reconstruction with  $k = 4$ , the bottom row with  $k = 7$ . The first step of our algorithm is shown in (d) and (g), where the jagged 1-cells of the simplified MS complex are smoothed by parametric B-spline curves in the plane, and *monotone* B-spline height functions that best approximate the initial scalar field along the 1-cells. The final reconstructions are shown in (e,f) and (h,i).

Even though there is no much difference visible in the final reconstruction contour plots, it can be observed in the 1-cells reconstructions that the curves in the plane have less ondulations for  $k = 4$  in (d), but that the boundary 1-cells are less well approximated compared to  $k = 7$  in (g), see the green curves at the bottom left and top right corners.

The model statistics and computation times are listed in Table 1. As expected, reparameterization is the most costfull step. Triangulating the domain of each 2-cell followed by a parameterization takes 95% of the total computation time. It follows further that the influence of  $k$  on the computation time is neglectable, even though the number of Sibson patches is proportional to  $k^2$ .

In the second example is a real terrain data set of the *Mont Rainier*<sup>2</sup>. It consists of 140148 vertices defining a  $458 \times 306$  grid. This data set is more complex in the sense that the initial function is not smooth and has critical points that are not well distributed over the domain. Figure 7 shows the initial function (a), its MS complex (b) with 1931 critical points. After 2% simplification, the MS complex (c) has 69 critical points. Figure 7(d,e,f) show the reconstruction with the smoothed 1-cells and final function.

The last data set is sampled on a  $100 \times 100$  grid from the mathematical function  $a \cdot \sin(x/a) \cdot \sin(y/a)$  with  $a = 10$ . The function has 31 critical points in the domain  $[-49, 50]^2$ . Here we did not compute any simplification (0%), but we applied our algorithm to reconstruct a function from the initial MS complex, see Figure 8.

## 7 Conclusion

This paper experienced for the first time the use of geometric modeling techniques to MS complex reconstruction. The main contribution is the application of a shape preserving spline surface. In particular we approximate the jagged 1-cells on the complex with smooth B-splines. We introduce a  $C^1$  monotone Sibson spline interpolant and combine it with a reparameterization to reconstruct the 2-cells individually. The resulting surface obeys completely the given MS complex. The fact to provide an explicit closed form representation of the reconstructed function may be an advantage when follow-up evaluations of the function and any higher order derivatives are required. The independence of the present method of the mesh or grid on which the initial function is defined on may also be advantageous in case where the grid size is very large.

Beside these achievements we noticed several limitations of our method. First, the lack of existing  $C^1$  reparameterisation techniques makes our resulting surface being only piecewise  $C^0$  continuous at the end. Recent techniques in isogeometric analysis, where the problem of parameterizing a domain defined by boundary curves is also a central concern, enable to parameterize an arbitrary domain up to some restriction using  $C^1$  splines [8, 31, 21]. However, none of these techniques can

---

<sup>2</sup> <http://data.geocomm.com/catalog/>

	Model		MSC			1-cells		2-cells		
	#faces	#V	simplif. level	#CP	#2-cells	k	time I <sup>1</sup>	time II Delaunay +param.	time III geometry	# $\Delta$ Bézier patches
<i>Shepard</i>	124002	62500	25%	25	22	4	0,13	8,43	1,15	6672
			25%	25	22	7	0,13	8,80	0,39	19024
<i>MtRainier</i>	280296	140913	2%	69	53	7	0,31	9,64	2,16	43300
<i>Trigo</i>	19602	10000	0%	31	24	7	0,03	2,11	0,11	6240

<sup>1</sup> Time I is the computation time for smoothing and reconstruction of 1-cells.

Table 1: Measured performances and model statistics.

actually deal with arbitrary domains as they occur in MS complexes. A  $C^1$  parameterization method for arbitrary domains would improve the general smoothness of our results. We could then develop the Sibson interpolant further to reach globally  $C^1$  reconstructions.

Second, regarding computation times, the parameterization step seems to be the bottleneck of the presented method. As the reconstruction of the 2-cells is done independently for each 2-cell, the algorithm could be parallelized to improve this part of the method. We believe however, that it would be more efficient to overcome the parameterization step completely and to work directly on the domain. The triangular spline approximation approach in [21] could be a further avenue to explore in the futur.

**Acknowledgement:** This research was partially funded by the ERC advanced grant EXPRESSIVE (no. 291184).

## References

1. CGAL, Computational Geometry Algorithms Library. [Http://www.cgal.org](http://www.cgal.org)
2. Allemand-Giorgis, L., Bonneau, G.P., Hahmann, S., Vivodtzev, F.: Piecewise polynomial monotonic interpolation of 2D gridded data. In: J. Bennett, F. Vivodtzev, V. Pascucci (eds.) Topological and Statistical Methods for Complex Data, Mathematics and Visualization. Springer (2014)
3. Bremer, P.T., Edelsbrunner, H., Hamann, B., Pascucci, V.: A topological hierarchy for functions on triangulated surfaces. IEEE Transactions on Visualization and Computer Graphics **10**(4), 385–396 (2004)
4. Edelsbrunner, Letscher, Zomorodian: Topological persistence and simplification. Discrete & Computational Geometry **28**(4), 511–533 (2002)
5. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical morse complexes for piecewise linear 2-manifolds. In: Proceedings of the symposium on Computational geometry, SCG '01, pp. 70–79 (2001)
6. Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical morsesmale complexes for piecewise linear 2-manifolds. Discrete & Computational Geometry **30**(1), 87–107 (2003)
7. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, FOCS '00, pp. 454–. IEEE Computer Society, Washington, DC, USA (2000)

8. Farin, G., Hansford, D.: Discrete coons patches. *Computer Aided Geometric Design* **16**(7), 691 – 700 (1999)
9. Farin, G.E.: *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Code*, 4th edn. Academic Press, Inc., Orlando, FL, USA (1996)
10. Floater, M.S.: Mean value coordinates. *Comput. Aided Geom. Des.* **20**(1), 19–27 (2003)
11. Forman, R.: A user’s guide to discrete morse theory. In: *Proc. of the 2001 Internat. Conf. on Formal Power Series and Algebraic Combinatorics*, A special volume of *Advances in Applied Mathematics*, p. 48 (2001)
12. Günther, D., Jacobson, A., Reininghaus, J., Seidel, H.P., Sorkine-Hornung, O., Weinkauff, T.: Fast and memory-efficient topological denoising of 2D and 3D scalar fields. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Scientific Visualization / Information Visualization 2014)* **20**(12), to appear (2014)
13. Gyulassy, A.: *Combinatorial construction of morse-smale complexes for data analysis and visualization*. Ph.D. thesis, University of California, Davis (2008)
14. Gyulassy, A., Bremer, P., Pascucci, V.: Computing morse-smale complexes with accurate geometry. *Visualization and Computer Graphics*, *IEEE Transactions on* **18**(12), 2014–2022 (2012). DOI 10.1109/TVCG.2012.209
15. Gyulassy, A., Bremer, P.T., Hamann, B., Pascucci, V.: A practical approach to morse-smale complex computation: Scalability and generality. *Visualization and Computer Graphics*, *IEEE Transactions on* **14**(6), 1619–1626 (2008)
16. Gyulassy, A., Kotava, N., Kim, M., Hansen, C., Hagen, H., Pascucci, V.: Direct feature visualization using morse-smale complexes. *Visualization and Computer Graphics*, *IEEE Transactions on* **18**(9), 1549–1562 (2012)
17. Gyulassy, A., Natarajan, V., Pascucci, V., Bremer, P.T., Hamann, B.: A topological approach to simplification of three-dimensional scalar functions. *Visualization and Computer Graphics*, *IEEE Transactions on (SPECIAL ISSUE IEEE VISUALIZATION)* pp. 474–484 (2006)
18. Gyulassy, A., Natarajan, V., Pascucci, V., Hamann, B.: Efficient computation of morse-smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics* **13**(6), 1440–1447 (2007)
19. He, X., Shi, P.: Monotone b-spline smoothing. *Journal of the American Statistical Association* **93**, 643–650 (1996)
20. Jacobson, A., Weinkauff, T., Sorkine, O.: Smooth shape-aware functions with controlled extrema. *Comp. Graph. Forum* **31**(5), 1577–1586 (2012)
21. Jaxon, N., Qian, X.: Isogeometric analysis on triangulations. *Computer-Aided Design* **46**(0), 45 – 57 (2014). 2013 {SIAM} Conference on Geometric and Physical Modeling
22. Morse, M.: *The Calculus of Variations in the Large*, vol. 18. Colloquium Publication - AMS (1934)
23. Robins, V.: *Computational topology at multiple resolutions: foundations and applications to fractals and dynamics*. Ph.D. thesis, University of Colorado, Boulder, Boulder, CO, USA (2000). AAI9979393
24. Shepard, D.: ia two-dimensional interpolation function for irregularly spaced data. In: *Proceedings of the Symposium on Geometry Processing, Proceedings of the 1968 23rd ACM National Conference*, p. 517523. iACM (1968)
25. Shivashankar, N., M, S., Natarajan, V.: Parallel computation of 2d morse-smale complexes. *IEEE Transactions on Visualization and Computer Graphics* **18**(10), 1757–1770 (2012)
26. Shivashankar, N., Natarajan, V.: Parallel computation of 3d morse-smale complexes. *Computer Graphics Forum* **31**(3pt1), 965–974 (2012)
27. Smale, S.: On gradient dynamical systems. *Annals of Mathematics* **74**(1), pp. 199–206 (1961)
28. Tierny, J., Günther, D., Pascucci, V.: Optimal general simplification of scalar fields on surfaces. In: J. Bennett, F. Vivodtzev, V. Pascucci (eds.) *Topological and Statistical Methods for Complex Data, Mathematics and Visualization*, p. to appear. Springer International Publishing (2014)
29. Tierny, J., Pascucci, V.: Generalized topological simplification of scalar fields on surfaces. *Visualization and Computer Graphics*, *IEEE Transactions on* **18**(12), 2005–2013 (2012)

30. Weinkauff, T., Gingold, Y., Sorkine, O.: Topology-based smoothing of 2d scalar fields with  $c_1$ -continuity. In: Proceedings of the 12th Eurographics/IEEE-VGTC conference on Visualization, EuroVis'10, pp. 1221–1230 (2010)
31. Xu, G., Murrain, B., Duvigneau, R., Galligo, A.: Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Computer Methods in Applied Mechanics and Engineering* **200**(2324), 2021 – 2031 (2011)

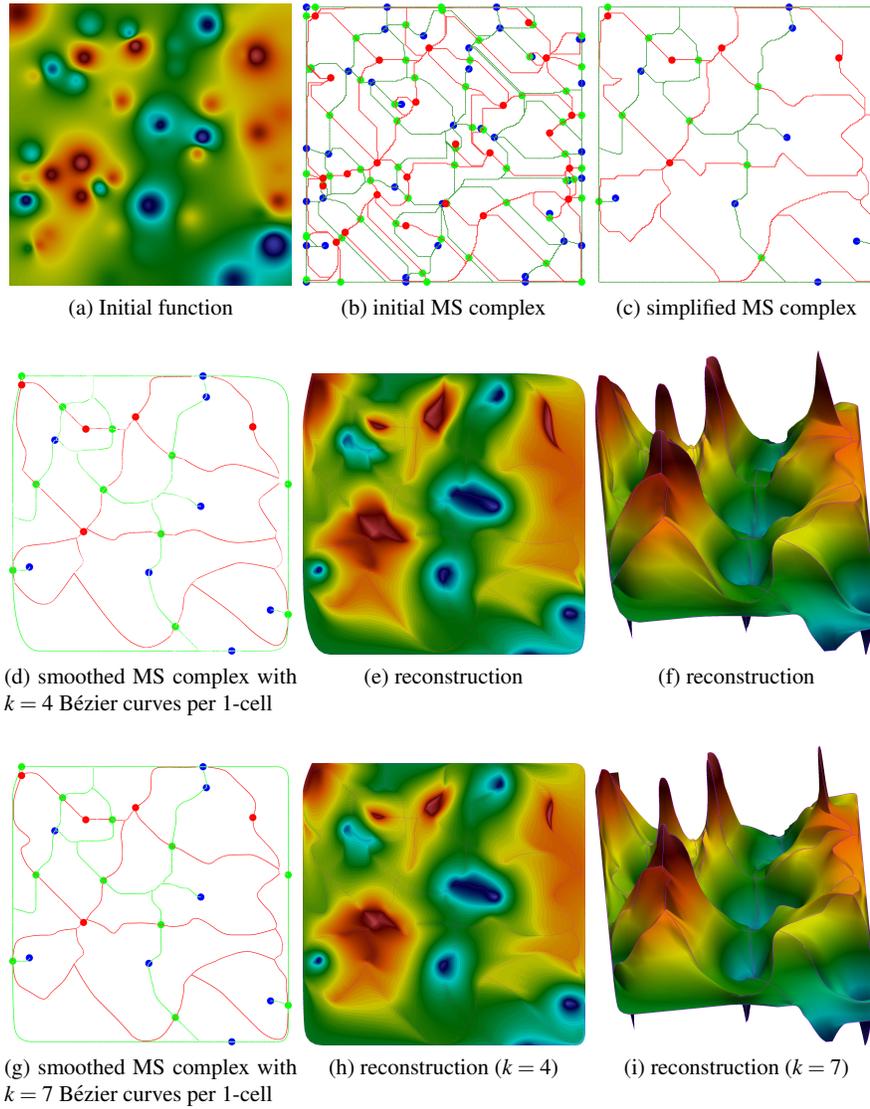


Fig. 6: Applying our algorithm on a synthetic example that has been generated using Shepard's scattered data interpolation. We tested our algorithm with two different values of  $k$  (number of Bézier curves per 1-cell, grid size of Bézier patches). In both cases our method reconstructs a piecewise smooth scalar field that exactly obeys the topological structure of the MS complex.

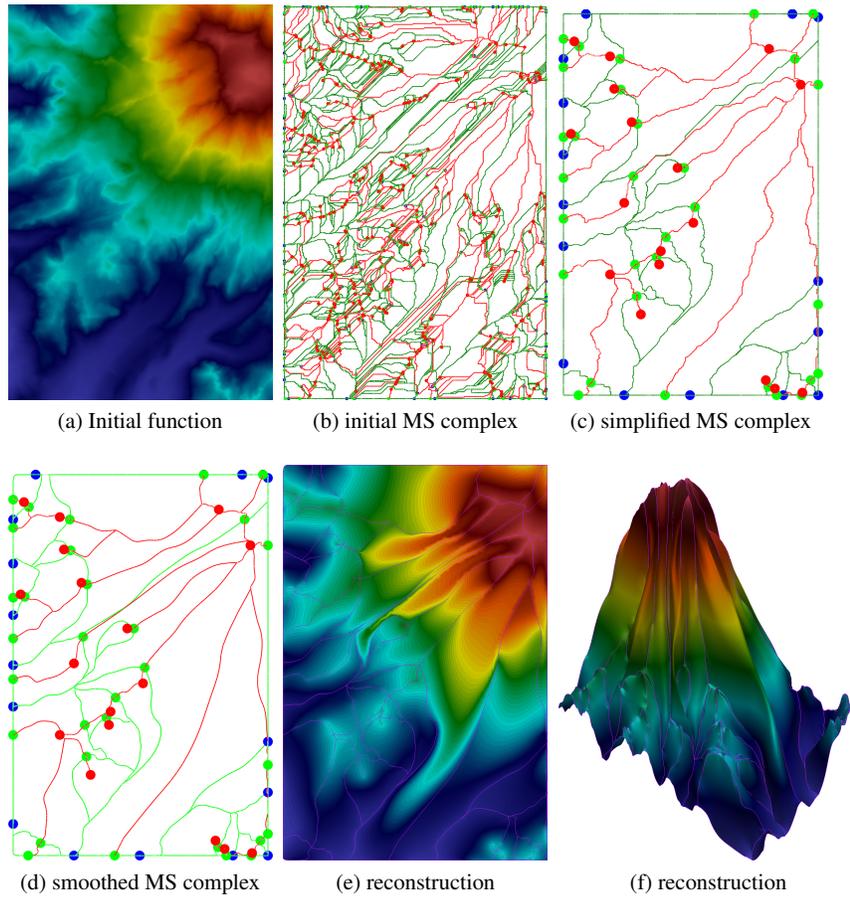


Fig. 7: The terrain data set of Mt Rainier (a) has 1931 critical points (b). The simplified MS complex with 69 critical points is reconstructed.

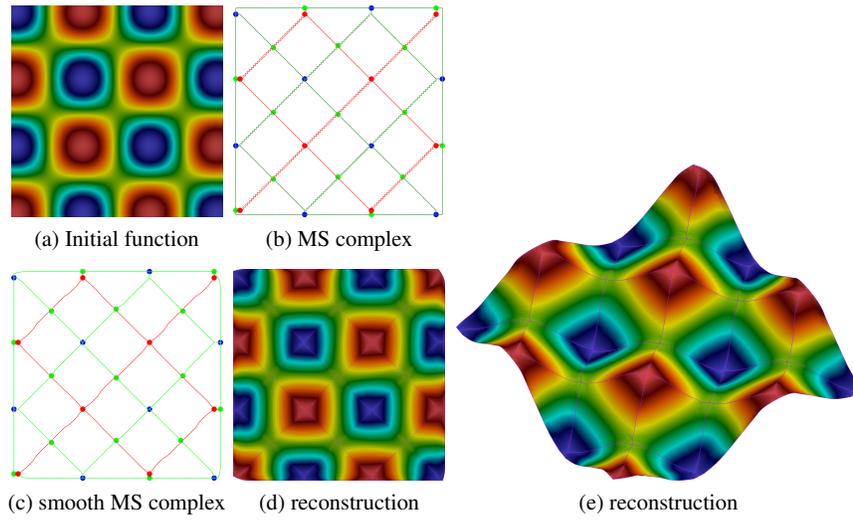


Fig. 8: Reconstruction of data sampled from  $\sin(x) \cdot \sin(y)$ . The original data set (a) has been reconstructed (d,e) without topological simplification.