Shared-Memory Parallel Computation of Morse-Smale Complexes with Improved Accuracy

Attila Gyulassy, Peer-Timo Bremer, and Valerio Pascucci *Member, IEEE*

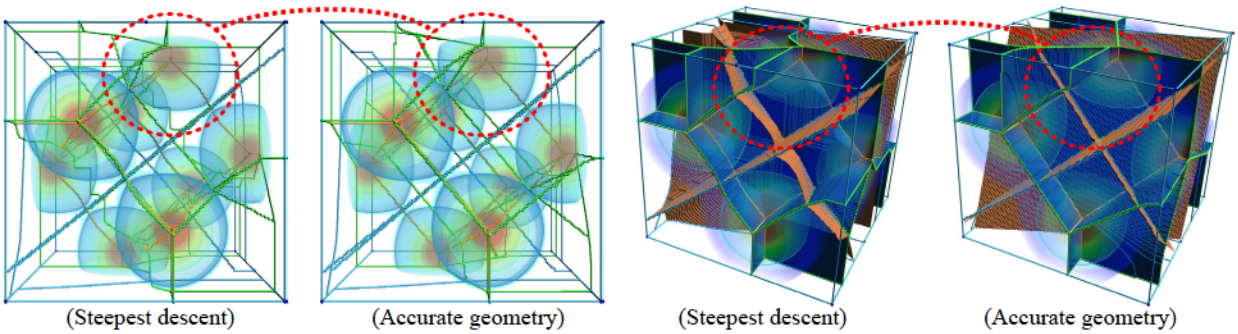


Fig. 1. A new approach computes Morse-Smale complexes in parallel with accurate geometry by combining numerical and discrete approaches. Traditional reconstructions based on steepest descent unnaturally align with the grid orientation. This example shows a scalar field formed by sum of eight Gaussians whose centers are arranged in a cube rotated with respect to the underlying grid. The arcs and surfaces of the complex should always be perpendicular to isosurfaces - a property much better preserved by our new approach. Better geometric reconstructions allow for better quantitative analyses, whether the feature based approach is used to measure distances, surface areas, or volumes, or if the features are used as a scaffolding for further analysis, such as conditional sampling.

Abstract— Topological techniques have proven to be a powerful tool in the analysis and visualization of large-scale scientific data. In particular, the Morse-Smale complex and its various components provide a rich framework for robust feature definition and computation. Consequently, there now exist a number of approaches to compute Morse-Smale complexes for large-scale data in parallel. However, existing techniques are based on discrete concepts which produce the correct topological structure but are known to introduce grid artifacts in the resulting geometry. Here, we present a new approach that combines parallel streamline computation with combinatorial methods to construct a high-quality discrete Morse-Smale complex. In addition to being invariant to the orientation of the underlying grid, this algorithm allows users to selectively build a subset of features using high-quality geometry. In particular, a user may specifically select which ascending/descending manifolds are reconstructed with improved accuracy, focusing computational effort where it matters for subsequent analysis. This approach computes Morse-Smale complexes for larger data than previously feasible with significant speedups. We demonstrate and validate our approach using several examples from a variety of different scientific domains, and evaluate the performance of our method.

Index Terms—Morse complex, Parallel Computation, Topology, Accurate Geometry

1 Introduction

The continuous growth in available computing resources and experimental capabilities has resulted in an unprecedented increase in the resolution of scientific data. However, this has come at the cost of having to analyze and visualize ever larger data sets. Not only does this create computational challenges, but interpreting the results also becomes increasingly difficult. In particular, as the analysis step becomes more costly and the outputs larger, exploring different parameter settings or different feature definitions becomes more challenging. In this context, topological techniques are particularly attractive, as they provide a language to robustly describe a wide range of phenomena in a flexible manner. More specifically, topological representations provide an intermediate structure that scientists can interactively query and explore to understand the impact of parameter changes or to develop new hypotheses.

Morse-Smale (MS) complexes are particularly useful in this setting, as they capture a wide range of features based on the gradient flow of a scalar function, including ridge- and valley-like structures, separating surfaces, basins, and "mountains". They have been used in a wide variety of scientific domains including computing burning regions in combustion experiments [6], counting bubbles in mixing fluids [25], analyzing the core structure of porous materials [17] or understanding lithium diffusion pathways [19]. Most existing approaches to compute MS complexes rely on discrete Morse theory [16, 32] which discretizes the gradient flow to the elements of a mesh. In particular, all existing parallel algorithms are based on discrete concepts [21, 34]. While these algorithms are guaranteed to produce the correct topological structure, they produce feature geometries that are unnaturally aligned with the mesh, and can misrepresent the true orientation of the features. This bias can skew subsequent analysis, highlighting the need for algorithms that result in an accurate geometric embedding. Unfortunately, the only approach to produce accurate geometry for all ascending/descending manifolds of volumetric domains [15] relies on a serial flood fill operation which severely limits the size of data that can be analyzed.

In this paper, we present a new approach for computing MS complexes with improved accuracy for volumetric scalar-valued data on shared-memory multi-core systems. It first builds a combinatorial representation based on discrete Morse theory. Subsequently, it then iteratively modifies the discrete gradient to align with numerically traced

- *Attila Gyulassy is with the SCI Institute, University of Utah. E-mail: jediati@sci.utah.edu.*
- *Peer-Timo Bremer is with Lawrence Livermore National Lab. E-mail: bremer5@llnl.gov.*
- *Valerio Pascucci is with the SCI Institute, University of Utah. E-mail: pascucci@sci.utah.edu*

Manuscript received 31 Mar. 2018; accepted 1 Aug. 2018.
Date of publication 16 Aug. 2018; date of current version 21 Oct. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2864848

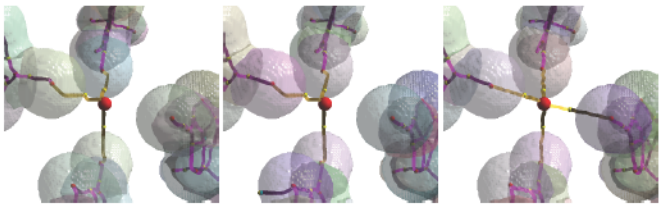


Fig. 2. The geometric artifacts of steepest descent construction (left) adds error to bond length calculations between lithium and oxygen atoms, which can lead to erroneous omission of bonds from a reconstructed molecular graph. The conforming algorithm [18] improves the 2- and 3-manifolds (middle), but not the 1-manifolds, leading to similar issues. The approach proposed in this paper (right) improves the accuracy of the geometric reconstruction, leading to more stable analysis.

features, while remaining in the confines of a combinatorially consistent structure. This allows a user, for the first time, to select which features defined using the MS complex they wish to extract with geometric accuracy, and the algorithm performs the necessary computation. Not only is the approach the fastest to date for constructing high-quality discretizations, but it also selectively applies the computational effort to only the structures needed for a particular use case. We evaluate our results using various feature-driven tasks with datasets taken from a variety of scientific domains. We report a 70x speedup with respect to previous approaches using data up to 10x larger than previously reported.

A Motivating Example: The geometric embedding of reconstructed topological features can play a significant role in subsequent analysis. This paper focuses on a new method for improving the accuracy of the geometric embedding of *all* ascending/descending manifolds of the MS complex. In particular, we extend the previous approach [18] to include accurate 1-manifolds and to improve the identification of 2- and 3-manifolds. These structures play a significant role in many application domains, for example in quantum chemistry. According to the quantum theory of atoms in molecules [2], the ascending 1-manifolds that connect 2-saddles with maxima form the bond paths between atoms. Bond paths can curve, indicating strain on a system, and the length of the paths can be used to determine whether or not a bond exists. Figure 2 shows an example of lithium in an electrolyte, where the bond path length calculation of the lithium-ethylene carbonate bond is overestimated by 11% by steepest descent approaches. This error can lead a scientist to erroneously omit a bond from the molecular graph of a system. This use case is further evaluated in Section 5.3.

2 Related Work

Topology-base techniques have recently been gaining traction as a standard for analysis and visualization of scientific data. While topological descriptions of features have existed since the 1800s [8, 27], recent advances describing their robust computation have accelerated adoption. The ability to define a feature robustly within a topological framework has enabled sophisticated analyses. For instance, Laney et al. [25] use Morse complexes on triangulated isosurfaces to identify and measure the formation of bubbles in Rayleigh-Taylor instabilities. Gyulassy et al. [17] used the 1-skeleton of the MS complex to identify the core structure of porous media and quantify loss of porosity. Bremer et al. [6] identified burning regions in turbulent combustion as cells of the Morse complex of temperature restricted to an isosurface of mixture fraction. Kasten et al. [22] used Morse cells of acceleration magnitude to segment vortical structures. Sousbie [35] computed the filamentary structure of the density value of cosmology simulations using the 1-skeleton of the MS complex, and characterized voids and surfaces using the ascending 2- and 3-manifolds.

With the recent release of analysis tools based on Morse theory, such as the generally applicable Topology Toolkit (TTK) [36], and TopoMS [4] for the quantum theory of atoms in molecules [1–3], the examples of wide-scale adoption are expected to increase substantially. However, neither of these general tools is well suited for large data which is becoming increasingly common.

To address this challenge new parallel algorithms for topological analysis have been developed. In the distributed setting, Landge et al. [24] describe an approach to compute merge trees *in-situ*, Morozov and Weber [29] describe an approach to perform analysis with a distributed contour tree. Shared memory approaches have also been introduced, with Carr et al. [7] using a pointer-jumping technique to parallelize contour tree computation, and Gueunet et al. [14] using rage-space partitioning to find parallelism in the computation of a contour tree.

The first practical algorithm to compute Morse complexes of two dimensional, piecewise linear functions was introduced by Edelsbrunner [11], notable in that it provided the first adaptation of the continuous theory to practical sampled functions. While piecewise linear approaches [5] proved useful for the analysis of two-dimensional data, adaptation of this theory to volumetric data [10] proved prohibitively complicated.

A key innovation enabling computation of Morse complexes for volumetric data has been the introduction of discrete Morse theory by Forman [13]. Rather than directly computing consistent gradient features, this allowed the burden to shift to computation of a much simpler discretized version of the gradient field. Several approaches compute a discrete gradient field in a greedy manner: they assign an arrow locally in the direction of steepest descent. The approaches introduced by Lewiner [26], Gyulassy et al. [16], Reininghaus et al. [31, 32] all perform this assignment in serial algorithms. Gyulassy et al. [21] introduced a distributed parallel version of this algorithm, however this approach produced artifacts on block boundaries. Faster, embarrassingly parallel approaches were introduced by Robins et al. [33] and Shivashankar et al. [34]. However each of these techniques employs a kernel which locally picks discrete gradient arrows in the direction of steepest descent. Gyulassy et al. [15] showed that such local optimization produces an artificial bias in features extracted from the discrete gradient, and simply rotating the underlying grid with respect to the function caused large-scale changes not only in the geometric embedding of features, but also the connectivity of the MS complex.

Improved accuracy: Matching a computed MS complex to an underlying interpolating function has been a sought after by many approaches: Bremer et al. [5] produced exact complexes for two-dimensional triangulated linear interpolants by splitting triangles to keep 1-manifolds separated; and Norgard and Bremer [30] computed exact complexes for two-dimensional bilinear interpolants. Chattopahdhyay et al. [9] furthermore demonstrated that for continuous planar functions the exact MS complex can be computed. Nevertheless, the complexities introduced by a third dimension have prevented extensions of these approaches to volumetric data.

Instead, discrete Morse theoretic approaches remain the only practical option for general computation of consistent topological structures for sampled volumetric data. In this setting, the particular choices of discrete gradient arrows completely determines both the topology and geometry of computed features. Gyulassy et al. [15] showed that by carefully integrating the probabilities of the eventual origin and destinations of discrete gradient paths given a randomized gradient arrow selection, choices could be made to avoid crossing manifold boundaries, and thus recover an "accurate" MS complex with respect to an underlying continuous gradient field. However, the key algorithmic element making this approach tractable relied on solving a recurrence relation, a fundamentally serial approach that cannot be scaled to large data, both due to long computation times and the memory requirements of storing the recurrence relation.

Our proposed approach adopts the conforming discrete gradient algorithm presented by Gyulassy et al. [18]. In this work, the authors presented a technique that allowed gradient arrow assignment to be guided by a user-supplied labeling, while maintaining topological consistency. Through naive numeric integration, they obtained origination and termination maps for integral lines, using those maps to reconstruct accurate manifolds. Critically, however, this approach did not produce accurate embeddings of ascending/descending 1-manifolds for volumetric data, as the algorithm reverted to steepest descent constrained to the boundaries between 3-manifolds. Furthermore, the authors observed challenges in obtaining high-quality origination/termination maps from straightforward numeric integration. In this paper we present both an approach to improve and accelerate the integration for accurate 2- and 3-manifolds as well as a new approach to improve the accuracy of 1-manifolds.

3 Background

We briefly review key concepts from Morse theory, and draw parallels in discrete Morse theory to motivate our approach. This section restates some definitions from recent works on computing MS complexes [4, 15, 18].

3.1 Morse Functions and the MS Complex

We present some common definitions from Morse theory, and refer the reader to introductory books for more detail [28]. A scalar function $f : \mathbb{M} \to \mathbb{R}$ defined over a compact d-manifold $\mathbb{M}$ is a *Morse* function if all its *critical points* (points where the gradient vanishes, $\nabla f = 0$) are non-degenerate and have distinct values. A critical point is non-degenerate if its Hessian is non-singular. For Morse functions, the neighborhood of a critical point p takes on a quadratic form, and can be written as $f_p = \pm x_1^2 \pm x_2^2 \cdots \pm x_d^2$, where the number of minus signs in this equation defines the *index* of criticality. For instance, for volumetric functions, minima are index-0, 1-saddles are index-1, 2-saddles are index-2, and maxima are index-3.

An integral line in f is a path in $\mathbb{M}$ whose tangent vector agrees with the gradient of f at each point along the path. The integral line passing through a point p is the solution to

$$\frac{\partial}{\partial t} L(t) = \nabla f(L(t)), \forall t \in \mathbb{R} \tag{1}$$

with initial value $L(0) = p$. The limit points as $t \to \pm\infty$ are critical points of f, with the limit at $-\infty$ called the *origin*, and ∞ the *destination* of the integral line. More intuitively, an integral line is a streamline in the gradient vector field, whose endpoints are critical points of f. *Ascending* and *descending* manifolds are obtained as clusters of integral lines having common origin, and destination, respectively.

The descending manifolds of f form a cell complex that partitions $\mathbb{M}$; this partition is traditionally called the *Morse* complex. Similarly, the ascending manifolds also partition $\mathbb{M}$ in a cell complex. A Morse function f is a *Morse-Smale function* if ascending and descending manifolds of its critical points only intersect transversally. An index-i critical point has an i-dimensional descending manifold and a $(d-i)$-dimensional ascending manifold. For instance, for volumetric domains, a maximum (index-3 critical point) has descending 3-manifold, and ascending 0-manifold, a 2-saddle (index-2) has a descending 2-manifold and ascending 1-manifold, etc.

3.2 Discrete Morse Theory

Discrete Morse theory provides a parallel formulation of Morse theory in the context of meshes. Rather than approximate continuous functions with interpolation over elements, discrete Morse theory describes criticality, integral lines, and ascending/descending manifolds directly in terms of a discrete gradient vector field and a discrete flow operator. We present only concepts relevant to this work, and refer the reader to Forman's introductory work for more detail [13]. A mesh representation K of a d-dimensional domain $\mathbb{M}$ is formed by cells from dimension 0 (vertices) to d (d-cells). For regular complexes, such as triangulations and regular grids, an i-cell α has *faces* that are cells with lower dimension on its boundary, and *co-faces*, cells for which α is on their boundary. For instance, for volumetric regular grid, a hexahedron (3-cell), has 26 faces formed by 6 quads (2-cells), 12 edges (1-cells), and 8 vertices (0-cells). Faces and co-faces that differ in dimension by one are called *facets*, and *co-facets*, respectively.

A *discrete vector* is formed by a pair a cells $\langle \alpha^{(i)}, \beta^{(i+1)} \rangle$ where α is a facet of β. By convention, the lower-dimensional cell in a discrete vector forms the *tail* of an *arrow*, and the higher-dimensional cell forms the *head*. A *discrete vector field* V is a set of discrete vectors where each cell of K appears in at most one vector. Cells not appearing in any gradient vector are *critical cells* with index of criticality equal to the dimension of the cell. The combinatorial analogue to integration in a continuous vector field (equation (1)) is given by a discrete flow operator Φ that moves between cells of K using discrete vectors and the facet relation between cells. Formally, for an i-cell α,

$$\Phi(\alpha) = \partial^{i-1} v(\alpha), \tag{2}$$

where we use $v(\beta)$ to denote mapping the tail of a discrete vector to its head, i.e., $v(\alpha) = \beta$ if $\langle \alpha, \beta \rangle \in V$, and $\emptyset$ otherwise, and ∂^{i-1} maps a cell to its facets. Repeatedly applying the flow operator "advects" a cell in the discrete vector field. A *V-path* is a sequence of cells

$$\alpha_0^{(i)}, \beta_0^{(i+1)}, \alpha_1^{(i)}, \beta_1^{(i+1)}, \alpha_2^{(i)}, \ldots, \beta_r^{(i+1)}, \alpha_{r+1}^{(i)}$$

such that for each $j = 0,\ldots,$ r, $\langle \alpha_j^{(i)}, \beta_j^{(i+1)} \rangle$ is a gradient arrow, and $\alpha_{j+1}^{(i)} \in \Phi(\alpha_j^{(i)})$, i.e., advecting from $\alpha_j^{(i)}$ using the discrete flow operator Φ yields $\alpha_{j+1}^{(i)}$. A discrete vector field is the *discrete gradient field* of a discrete Morse function if the flow operator Φ does not produce any V-paths containing loops. Just as with integral lines, V-paths originate/terminate at critical cells, and the ascending/descending manifolds of the *discrete Morse-Smale* complex are formed by cells in V-paths originating and terminating at critical cells.

The main advantage to using discrete gradient vector fields, is that once the discrete vectors are computed, finding critical points and the ascending/descending manifolds of the MS complex becomes simple, applying combinatorial search algorithms. For instance, Shivashankar [34] described a simple breadth-first-search from each critical cell to construct the 1-skeleton as well as the ascending/descending manifolds of the complex. Furthermore, the discrete gradient field naturally handles degeneracies that may occur in scalar data, such as flat regions, guaranteeing a consistent topological structure.

3.3 Conforming Discrete Gradient

Gyulassy et al. [18] have presented an algorithm to compute a discrete gradient vector field for sampled scalar data that also conforms to a user-supplied labeling. In particular, it ensured that no discrete gradient arrows are created that pair cells with different labels. The algorithm is embarrassingly parallel with respect to vertices, and higher dimensional cells are processed along with their highest vertex. The authors presented a straightforward scheme to compute origination/termination maps for each vertex/hexahedron of a regular grid, and label cells occurring on the boundaries between 3-manifold regions. In this paper, we greatly improve both the stability and speed of this approach, reducing the amount of work that is needed to compute the labels. Furthermore, we extend the approach with a new way of labeling cells belonging to numerically integrated 1-manifolds, enabling the conforming discrete gradient algorithm to produce discrete MS complexes with accurate manifolds of every dimension.

3.4 Simplification and Persistence

A Morse-Smale function can be simplified by canceling a pair of critical points connected by exactly one gradient path, corresponding to a local smoothing of the function [12]. The cancellation operation can be realized in a discrete gradient field by reversing the unique V-path between the pair of critical cells [13]. Alternatively, cancellations can be performed on the combinatorial structure of the Morse-Smale complex by removing a pair of nodes and reconnecting of the nodes and arcs around them. In both instances, repeated application of the cancellation operation can be used to remove small features, such as occurring from either low-amplitude noise in the data or artifacts introduced when discretizing a continuous function onto a grid. The *persistence* of a pair of canceled critical points is the absolute difference in function value between them.

4 MS Complexes with Improved Accuracy

Adopting a discretized approach makes combinatorial algorithms possible. Regular grids remain a standard means of representing phenomena, as the implicit neighborhood encoded by the array index of an element allows more memory for high-resolution data values, rather than spatial data structures. However, when a grid is used to discretize the gradient flow operator, in the style of Forman's discrete Morse theory, there are limitations to the resolution of features that can be extracted. While integral lines may become arbitrarily close and then separate in continuous functions, in the discrete setting, once V-paths merge, they remain united for the rest of the path. Given this fundamental limitation the challenge is then to find the "best" discrete gradient field for a given mesh to approximate a continuous gradient field. A simple answer

would be choosing discrete gradient arrows that minimize *globally* the distance between the V-path originating from each cell and the continuous integral line passing through its centroid. However, not only is this global solution difficult to compute it is likely not optimal for any particular feature definition. Instead, focusing effort only on those integral lines that matter to a specific feature class better utilizes the finite resolution and representational capability of a discrete gradient, with lower chance of discretization artifacts occurring. In practice, most analysis tasks using the MS complex only require specific manifolds *at some persistence simplification threshold*, alleviating the need to reconstruct the finest topological features with accurate geometry.

Challenges of consistent numeric integration: For sampled data it is well known that obtaining a "continuous" representation that is both valid and self-consistent is extremely difficult. Even seemingly simple tasks, such as identifying the location of critical points, become difficult. For instance, discontinuities in the numeric derivatives, floating point error, and degenerate (flat) regions in data can cause errors in the location and index of critical points found. Indeed, in general, numerically based methods seldom are able to identify critical points in a manner consistent with the Poincare-Hopf theorem, i.e. there *Morse sum* fails to match the Euler characteristic of the domain. In addition, numerically computed integral lines have been observed to cross or merge. For example, naive integration near unstable critical points caused crossing integral lines, resulting in interleaved origin/destination regions and corrupted boundary maps in [18]. While simulation codes and some analysis systems, such as Diderot [23], utilize higher order stencils and therefore have continuous first and second derivatives it is almost impossible to guarantee that critical points and integral lines are computed consistent with the theoretical requirements of a Morse function.

Overview of approach: Our new approach combines several distinct stages depending on what features are constructed with improved accuracy. All use cases first construct a steepest descent discrete gradient field using Robin's algorithm [33]. If any accurate manifolds are required, each cell of K is given a uniform label. If a user requires accurate 2- or 3- manifolds, volumetric regions are identified through numeric integration, recording whether or not each cell is on the boundary of regions in the labeling. If accurate 1-manifolds are required, numeric streamlines are computed from each 1- and 2-saddle, and digitized, recording in the labeling which cells are traversed by the numeric lines. Finally, the conforming gradient algorithm [18] is supplied with the label map, and the neighborhoods of cells with changed labels are re-processed.

4.1 Numeric Integration for Volumetric Decomposition

Recall that each ascending/descending 3-manifold of the MS complex is composed of a critical point combined with all the integral lines that originate/terminate at that point. The approach initially seems straightforward: for each vertex of the mesh, trace an integral line numerically in the positive and negative gradient direction, and label vertices with the identifier of the maximum/minimum where that gradient line originates/terminates. To avoid the problems mentioned in numerically finding these critical points, we instead create large targets - regions where, upon entry, an integral line can be safely clamped to the combinatorial critical point. Specifically, the *certain region* of a minimum consists of the vertices where *any possible* monotonically decreasing V-path originates at the minimum. Our approach first computes these certain regions for each minimum and maximum of the steepest descent gradient field to use as targets for numeric integration. This has the dual benefit of increasing the agreement between the numeric and combinatorial approaches and also to accelerate the numeric integration through early termination. Indeed, we extend these regions as much as possible, even performing persistence cancellations to merge, and thus grow, the regions associated with low-persistence extrema. Next, numeric integration is performed from each unlabeled vertex. Finally, the resulting labeling is processed to remove discrepancies between the numeric classification and the constraints of discrete gradient flow being restricted to the cells of mesh. The output of this overall approach is a label for each cell of the mesh whether it is interior to a 3-manifold region, or on a boundary. Figure 3 illustrates the steps in this process.

Find Persistent Extrema: A discrete gradient vector field is computed using Robins' steepest descent approach [33]. The critical cells are extracted from this discrete gradient. Note that the ascending/descending 3-manifold associated with two extrema that merge during persistence simplification is simply the union of the individual ascending/descending 3-manifolds. By finding which ascending/descending 3-manifolds will end up merged, we will be able to identify larger regions for later combinatorial termination of integral lines. Therefore, given the critical points from the discrete gradient, extremum graphs

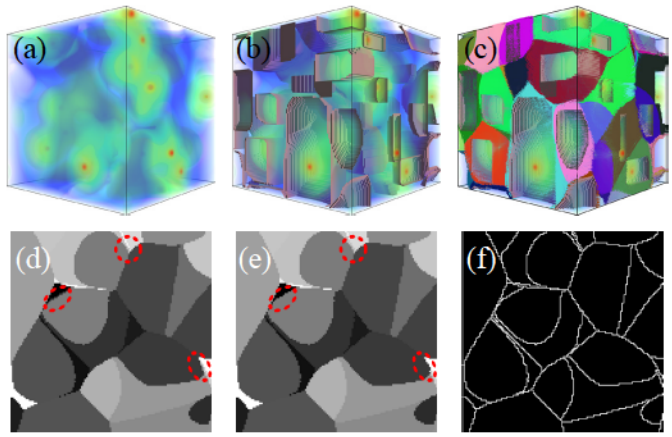


Fig. 3. We demonstrate creating a boundary label between descending 3-manifolds with the electron density of lithium in electrolyte scalar field, volume rendered in (a). All maxima are found and their certain regions computed, with their interface displayed as brown surfaces in (b). These act as targets for numeric integration; any integral line entering such a region can only end up at the combinatorial maximum that generates the surface. For all vertices that have not been labeled as part of a certain region, an integral line is traced in the gradient direction, until the streamline crosses into a certain region. In (c) we color each streamline according to its destination. Each vertex gets the label of its destination region (d). However, as streamlines can pass between voxels, these regions are not necessarily connected, as seen in the red circles. A connected component sweep is performed using a Union-Find algorithm to remove these regions (e). Finally, the cells on the boundaries of the regions are marked in a label map (f).

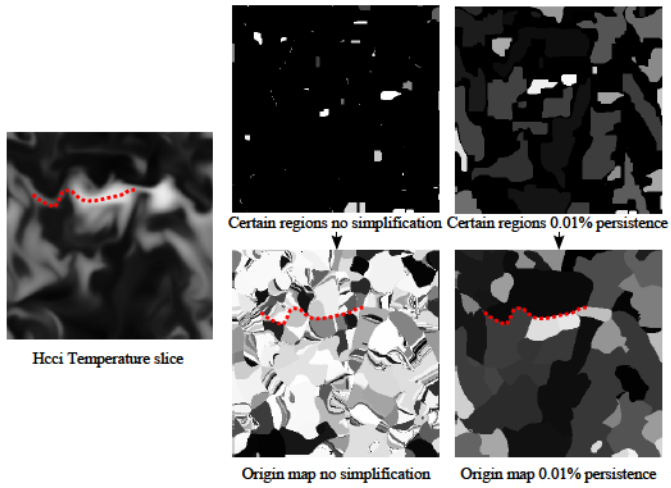


Fig. 4. We illustrate the effect of using pre-simplification of extremum graphs during the construction of certain regions. Black regions in the top images indicate vertices *not* part of certain regions. When extrema are merged, a larger portion of the domain can be identified as having all ascending/descending paths unambiguously terminating at an extremum. For a Homogeneous charge compression ignition (HCCI) simulation, the certain regions before (top middle) and after (top right) even a very small threshold simplification occupy vastly different extents, which are then mirrored in the origin maps that are computed through numeric integration (bottom). For extracting a ridge-like flame surface, the coarser topological scale is not only adequate, but potentially has fewer artifacts where the mesh resolutions was unable to accommodate the high density of varying integral line terminations.

are computed. The nodes of an extremum graph are formed by the extrema, and an arc exists between two nodes, if the 1-skeleton of the MS complex connects the extrema through a saddle. The smaller of the differences between the saddle and the two extrema are used as the cost of canceling the arc. The extremum graph is simplified up to a user provided threshold by successively merging the pair of nodes having the lowest cost arc, representing a cancellation of one extremum with a saddle. Our approach correctly identifies saddle-extremum persistence pairs, and furthermore conveys the information of which extrema are merged during simplification. The merging of nodes in the extremum graph is implemented with a straightforward Union-Find data structure, with lazy update to the arcs. Although serial, since this step only operates on critical points, it contributes a negligible time to the total running time. Figure 4 illustrates the benefits of performing this merging - the targets for early termination of numeric integration cover a significantly larger portion of the domain.

Certain regions: We define the certain region around a persistent minimum to be the set of vertices where all monotonically decreasing paths composed of vertices and edges terminate at the minimum, or at a minimum that was merged into the persistent minimum in the extremum graph. Each persistent minimum can be expanded to a certain region independently and in parallel through a simple priority-queue based region growing approach outlined in algorithm 1. Each vertex in a certain region is labeled as part of that minimum's ascending 3-manifold.

Algorithm 1 CertainRegion(V)

```
PriorityQueue pq = {}
R = {}
For v in V: pq.enqueue(v)
While(not pq.empty())
    u = pq.top_and_pop()
    if (all vertices w ∈ LowerNeighbors(u) are in R)
        pq.enqueue(UpperNeighbors(u))
        R.insert(u)
return R
```

The algorithm takes the set of minima V that have been merged into a single persistent minimum, and labels the associated region. Certain regions for maxima are identified identically, when considering $-f$ on the dual graph of K.

Numeric Integration: To identify which vertex should belong to which ascending 3-manifold, for every vertex not already labeled in a certain region, we numerically trace a streamline starting from the vertex location in the negative gradient direction. In our approach, we solve equation (1) using an adaptive stepsize Euler integrator, with trilinearly interpolated gradient vectors, computed using central differencing at the vertices. After every integration step, the point is checked to determine if it has entered a certain region. If it has, the streamline is stopped, and the original is point marked with the label of the certain region. The same process is performed for descending 3-manifold, integrating streamlines in the positive direction from the centers of hexahedra.

While this process can be done in an embarrassingly parallel manner, it is still computationally expensive. We accelerate the integration by labeling every vertex a streamline passes near as part of its destination's certain region. This path compression greatly reduces the amount of integration that needs to be done, however introduces incorrectly labeled vertices near boundaries. To solve this problem, a new streamline is computed from every vertex on the boundary of regions that does not terminate until it hits one of the original certain regions. As labels change in this process, the new boundary vertices are also re-integrated. In practice, using path-compression and re-integration more than doubled the speed of the labeling compared to not using path compression.

Topological cleaning: While it has been shown that a valid discrete gradient and Morse complex can be computed for any label restriction [18], in practice, producing a labeling that minimizes the number of spurious regions reduces the downstream work. In the case of numeric integration, an integral line may pass between vertices of another label to terminate at a region that is not reachable via a vertex-edge path, making the set of vertices belonging to a single region disconnected. These spurious regions will generate extra critical points, but typically with zero persistence, only existing in the discrete gradient due to the label itself. Rather than expending the work to produce boundaries for these regions, we perform a Union-Find to detect these regions and re-label them with the label of the neighbor they would simplify to in later simplification. Figure 3 (d,e) illustrates the result of this process.

Boundary labeling: The ultimate goal of the numeric integration is to discover which cells of K belong to boundaries between ascending 3-manifolds. A cell is labeled as interior if all its vertices share the same integrated origin label, and marked as boundary otherwise. For descending 3-manifold maps, a cell is considered interior if all coface hexahedra have the same destination label, and boundary otherwise. We use a label of 0 to indicate interior cells, 1 to indicated boundary between ascending 3-manifolds, 2 to indicate boundary between descending 3-manifolds, and 3 to indicate both.

4.2 Accurate 1-Manifolds

Given the steepest descent discrete gradient, we wish to ensure that the V-paths connecting 1-saddles to minima, and 2-saddles to maxima not exhibit serious artifacts due to compounding the local error induced by making steepest-descent decisions. Instead, given a continuous integral line, connecting a saddle to extremum, we wish to modify the discrete gradient such that the V-path deviates as little from the line as possible. At best, this must be an approximate solution, since there is no guarantee that the mesh has sufficient resolution to represent all saddle-extremum V-paths optimally. Furthermore, the discrepancy between interpolated values and interpolated gradients makes it possible for a numerically integrated path to be non-monotone. Finally, numerically computed integral lines can exhibit properties, such as crossing, that are undesirable. Given these challenges, our approach is pragmatic: compute approximate integral lines starting in the vicinity of where the continuous line would be expected, integrate until an extremum is reached, or the numeric integrator times out, fix the V-paths as best as possible, and at least ensure combinatorial validity.

Starting integral lines: In the continuous case, the location of a saddle point determines the starting point, and the eigenvectors of the Hessian determine the direction an integral line takes to reach an extremum. For a discrete 1-saddle, a gross approximation is to begin integration downwards starting from the two vertices adjacent to the edge. Figure 5 (c)(top) illustrates a case where this is not sufficient: integrating down from both vertices adjacent to the critical edge results in streamlines ending at the same minimum. This is due to the fact that the steepest descent construction only guarantees that a critical point of the right index appears in the lower star during a filtration - not that the geometric location within the lower star is optimal. Therefore, we start an integral line, not only from the immediately adjacent vertices, but also along vertices one step from the critical edge along V-paths terminating at that edge. The ascending integral lines from 2-saddles are computed in a similar manner, from the centers of hexahedra adjacent, and V-path connected to the critical quadrilateral.

Ending integral lines: We terminate an integral line when it enters the vicinity of a critical cell. For descending 1-saddle to minimum lines, if the numeric integration enters within a one-half grid cell from a combinatorially identified minimum, the integration terminates. Ascending 2-saddle maximum lines terminate when they enter a critical hexahedron. However, in practice, numeric integrators often stall, for instance in flat-bottomed valleys with steep sides, or in flat regions of the data. A maximum iteration limit is used to forcibly terminate numeric integration of streamlines that have stalled. Figure 5(a) shows the numeric streamlines that are computed for the 8-Gaussians example.

Discretizing numeric lines: An integral line computed through numeric integration consists of a sequence discrete point locations connected by straight line segments. To translate this back to the combinatorial discrete gradient setting, we employ a similar strategy to the boundary map used with 3-manifolds: we label each cell of the mesh that the line passes near as being part of an ascending/descending 1-manifold. More specifically, each segment of an ascending integral line is checked, labeling all hexahedra and quadrilaterals they pass through as part of an ascending 1-manifold. Descending lines are handled similarly, marking each vertex and edge if the corresponding hexes and quads in the dual mesh are crossed. The effect is to rasterize each streamline onto cells of the mesh. Figure 5(b) illustrates the set of cells

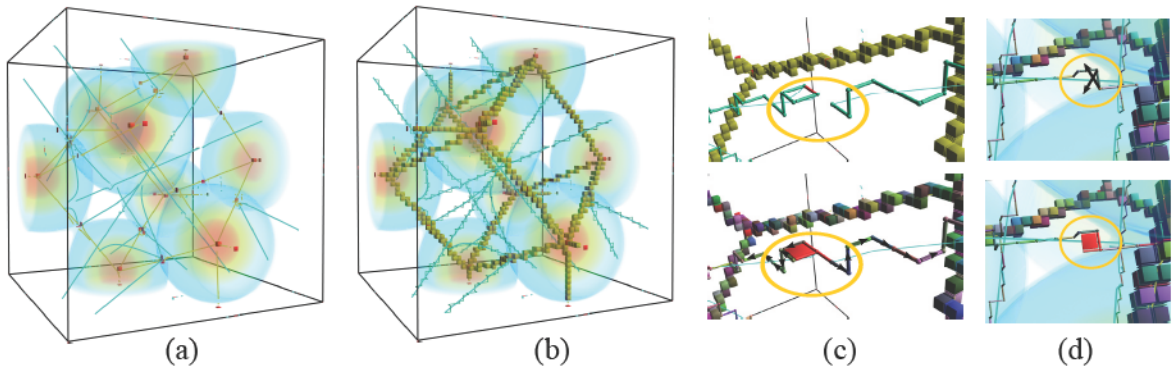


Fig. 5. Accurate 1-manifold construction is shown for a smaller 8-Gaussian dataset. Numeric streamlines are computed starting from cells in V-paths within 1 step from to saddles (a). These paths are rasterized, labeling the nearby cells of appropriate dimension (b). Inconsistencies between numerically computed lines and the combinatorial critical points (c), or insufficient mesh resolution (d) will cause new critical cells to appear after a conforming discrete gradient is computed.

marked by the numerically computed ascending and descending integral lines. During discretization, if a numeric path enters a previously labeled region, it is truncated to ensure that once vertex-edge and quad-hex V-paths merge, they will not split; no further information from the numeric integral line can be used after it merges with an existing path. This opportunistic labeling introduces an order-dependence that is non-deterministic with parallel execution. The first 1-manifold to write the label map is reconstructed optimally, while subsequent arcs may shift 1/2 cell every time their discretized path joins another.

Local Cancellations: In certain cases, the numerically computed integral line is discretized in a manner that the discrete path is no longer monotonic. Figure 5(d) illustrates such a case, where the labeled vertex-edge path can not follow the descending path, which results in the subsequent conforming discrete gradient algorithm to mark those cells as critical. We provide the option for a user to perform local cancellations up to a threshold, where only adjacent critical cells having the same label are considered. This local cancellation creates a discrete gradient arrow that allows the continuation of the V-path along the numerically integrated one.

4.3 Parallel Discrete Gradient Computation

So far, the parallel numeric integration resulted in labels on cells as to whether they belong ascending/descending 1-manifolds or the boundaries of ascending/descending 3-manifolds. We use these labels as in input to the conforming discrete gradient algorithm [18]. This algorithm assigns discrete gradient arrows in parallel, guaranteeing that both head and tail of each gradient arrow belong to the same label class. The effect of this constraint is that no discrete gradient arrow crosses between the ascending 3-manifolds, ensuring that their separating 2-manifolds, as computed in the discrete gradient, live on the boundary of the regions computed numerically. Furthermore, cells belonging to 1-manifolds are labeled such that they can only pair amongst themselves, guiding V-paths from saddles to extrema. Figure 5(c)(bottom) colors cells based on which lower star they belong to: similar colors can be paired in discrete gradient arrows. As the conforming gradient algorithm operates independently on the lower stars of vertices, it is sufficient to detect which vertices have cells in their lower star with non-zero label, and recompute the discrete gradient only on those vertices.

5 Results

We demonstrate the accuracy of our approach by comparing qualitatively and quantitatively to a known simple example. We illustrate the flexibility of our approach by allowing selective computation of manifolds with improved accuracy. Next, we demonstrate a practical use case in molecular analysis where the accuracy our approach is needed to reduce the error of a specific analysis task. Additional examples are shown motivating the need for selective reconstruction with improved accuracy. Finally, performance characteristics are reported.

5.1 Accurate Geometry of a Simple Example

We have chosen a simple example to validate our approach. The 8-Gaussians dataset is constructed by summing eight Gaussian functions centered at the eight corners of a cube with sigma equal to half the distance between corners. Figure 6 (a) shows a volume rendering of the dataset, and (b) shows the analytic MS complex for this function (omitting only arcs/nodes connecting to the domain boundary). This dataset is chosen to make validation simple as its analytic solution is known (to within a small epsilon): maxima should appear at the corners of the cube, 2-saddles at the mid-point of the edges, 1-saddles at the

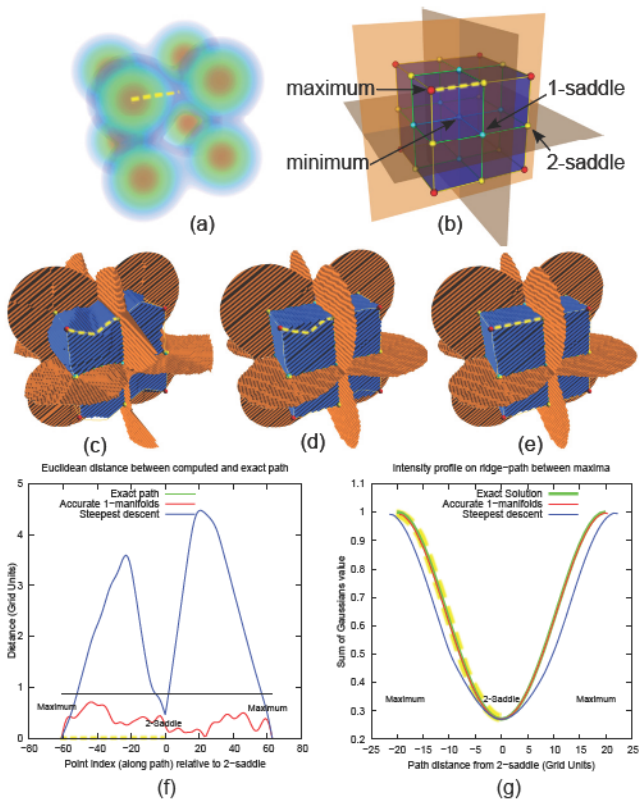


Fig. 6. The 8-Gaussians function (a) has an analytic MS complex (b) that evenly divides the cube. When the function is sampled onto a grid rotated with respect to the 8-Gaussians cube, the quality of reconstructed features is impacted by the choice of algorithm. The yellow dotted line identifies the same 1-manifold in all sub-images. All ascending/descending manifolds of the steepest descent discrete gradient construction [33] are biased by the underlying grid, and thus are deformed (c). Computing accurate 3-manifolds using the conforming algorithm [18] provides no guarantees for the quality of 1-manifolds (d). Our approach produces the reconstruction with accurate 1-, 2-, and 3-manifolds (e). The distance between the (yellow dotted) exact 1-manifold and computed ones is plotted (f), showing that our approach remains within a one-half grid cell spacing from the analytic path. Plotting the function value vs. distance along the path (g) further illustrates the quantitative reduction in error when using our approach.

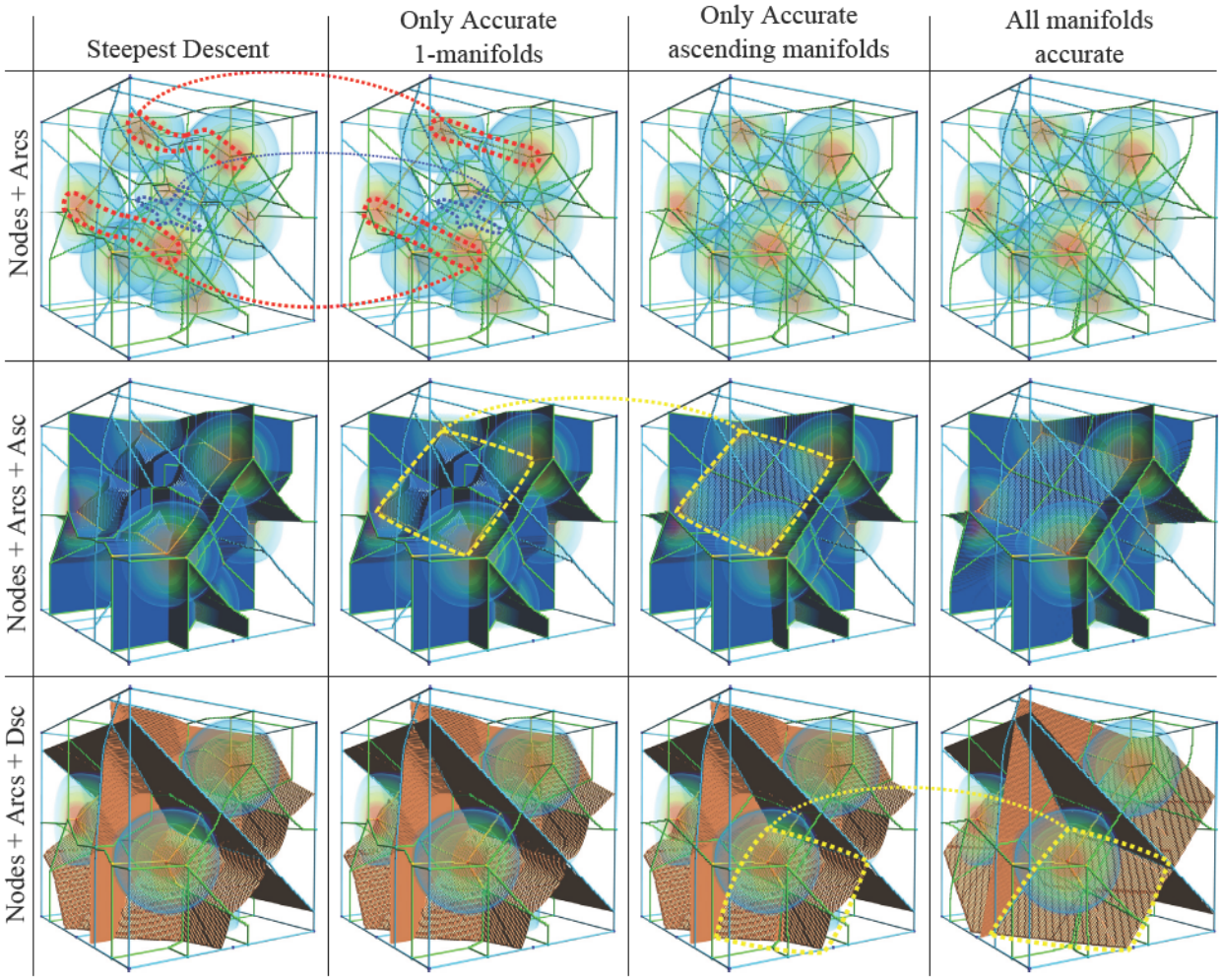


Fig. 7. Each column shows the components of the MS complex computed from a discrete gradient with selected accuracy. The rows show the geometric embedding of nodes and arcs (top), ascending 3-manifold boundaries (middle) and descending 3-manifold boundaries (bottom).

mid-point of quads, and a minimum in the exact center. Furthermore, ascending 1-manifolds should be straight connectors between maxima, descending 1-manifolds form straight lines from the center of the cube out through the middle of each quad, the ascending 3-manifold of the central minimum have the boundary in the shape of a cube, and the descending 3-manifolds have planar boundaries exactly dividing the cube into eight sub-cubes. The 8-Gaussians function is sampled onto a uniform grid rotated first around the Z axis, then around the Y axis. An isometric transformation of the cube corresponds to the same isometric transformation of the analytic MS complex. Figure 6 (c) shows that steepest descent gradient assignment biases 1-, 2-, and 3-manifolds in the direction of the underlying grid, deforming the reconstructed features. Figure 6 (d) illustrates the result of only computing accurate 2- and 3-manifolds, as done by Gyulassy et al. [18], where the highlighted 1-manifold displays the same artifact as the steepest descent approach. Figure 6 (e) shows the result of our new approach, with each reconstructed feature being at most one-half grid cell away from the analytic solution. Figure 6 (f) plots the deviation of the 1-manifolds compared to the exact solution. Note that under mesh refinement, the error associated with steepest descent does not decrease, in contrast with our improved accuracy approach. The differences in the 1-manifold reconstructions matter: figure 6 (g) shows that any analyses based on the geometric embedding of the arc, such as function value vs. length, will be impacted.

5.2 Flexibility of the Approach

We demonstrate the flexibility of our approach with the 8-Gaussians example in figure 7, demonstrating a user's ability to select which ascending/descending manifolds are constructed with improved accuracy. Focusing computational effort on only those features required by an analysis task accelerates the overall computation.

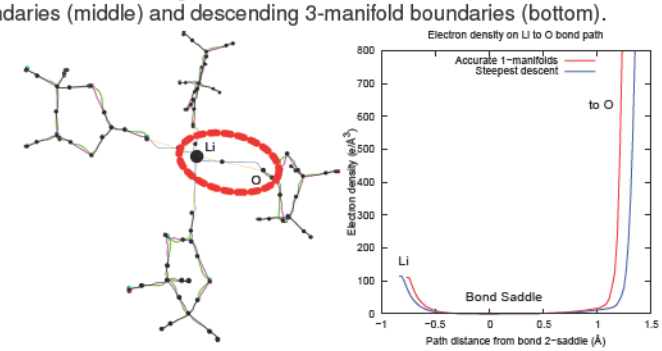


Fig. 8. The 2-saddle-maximum arcs of the MS complex form possible bond paths between atoms and molecules. The steepest descent paths (blue, green) are displayed over the accurate ones (yellow, magenta). The circled bond paths (left) are plotted (right) illustrating dramatic differences in their intensity profiles along the paths.

5.3 A Practical Use Case: Lithium Bonding

We mirror the analysis performed on the 8-Gaussians example in section 5.1 with the electron density field of a lithium in electrolyte simulation. Introduced by Bader's group, the quantum theory of atoms in molecules (QTAIM) [2] casts molecular features in topological terms. For example, an atomic basins corresponds to descending 3-manifolds in the electron density field, interatomic surface correspond to descending 2-manifolds from 2-saddles, and bond path correspond to ascending 1-manifolds from 2-saddles. We refer the reader to Bhatia et al. [4] for a more in-depth discussion. Bond paths, in particular are used in classifying bonds, and the curvature of the path can indicate strain on

a system. A common approach to determine whether two atoms are bonded is to compare the bond path length to known bonding radii for different types of atoms. The charge associated with the atom is computed as the integral of density over atomic basins. We extract the 2-saddle-maximum arcs between lithium and its surrounding ethylene carbonate molecules from the computed MS complexes, and overlay the reconstructions computed using steepest descent and our accurate 1-manifolds in figure 8 (a). Note that this displays the same pattern of artifacts as the 8-Gaussians example. Plotting the electron density as a function of distance along the path, one can observe an order-of-magnitude difference between the measured density for the same distance value, in figure 8 (b), e.g. at distance $= 1.2$. Furthermore, the total distance computed using steepest descent is overestimated by 11%. In this simulation, ethylene carbonate molecules form and break bonds with the lithium over time: for *any* length threshold applied by domain scientists to characterize the existence of the bond, the steepest descent approach will consistently underestimate the *duration* a bond is in existence, and the misclassification illustrated in figure 2 is guaranteed to occur.

5.4 Examples with Improved Accuracy

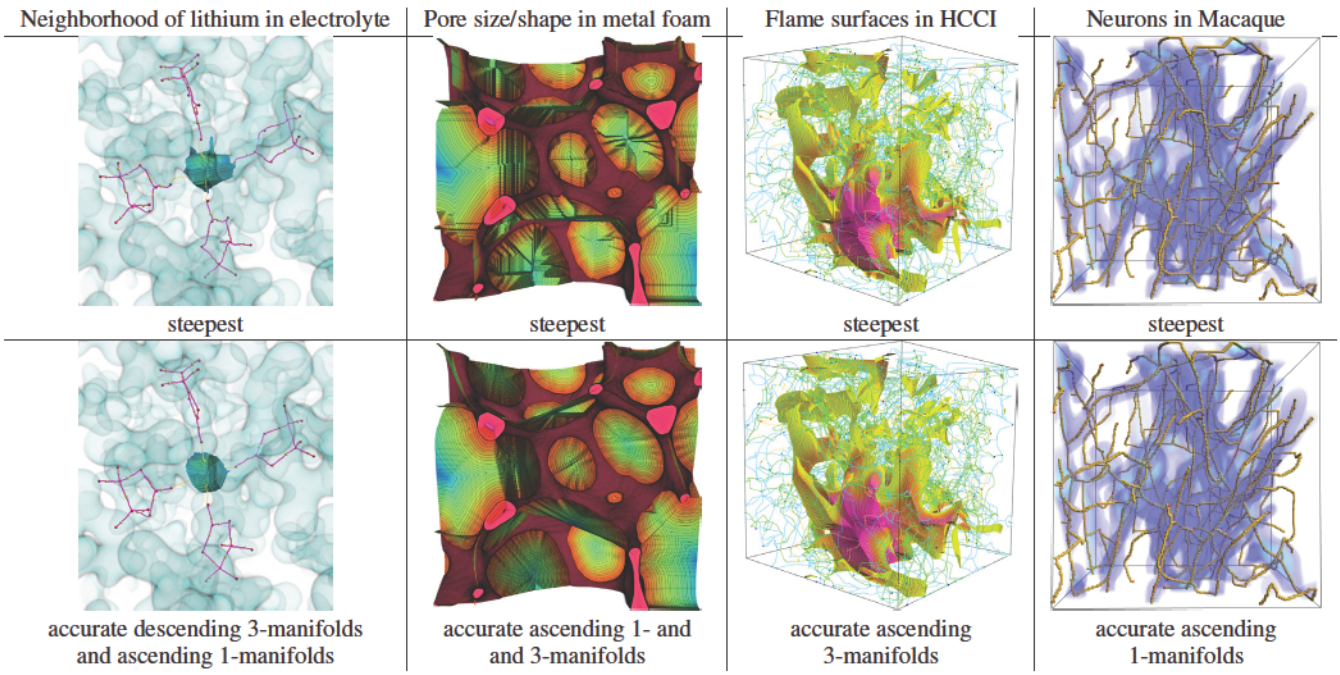
Several examples are included with varying accuracy requirements for features. The datasets and reconstructed features are shown in figure 9.

Lithium: For the lithium data from section 5.3, we use accurate descending 3-manifolds and accurate ascending 1-manifolds. Correct identification of atomic basins is needed in order to compute the charge associated with each atom. The purely steepest-descent approach identifies an atomic basin with clearly non-physical spatial extents, and is corrected by our approach.

Foam: The study of foams in material science has been driven by a need for lighter and stronger materials. Quantifying the size and shape distributions of pores can lead to insight guiding foam construction parameters. This dataset consists of a CT scan of a foam converted to a signed distance field from the interface between material and air. The pores are the non-material portions of the ascending 2-manifolds separating internal voids. Furthermore, the core structure is given by ascending 1-manifolds. For this data, we compute accurate ascending 1- and 3-manifolds. The quality of the reconstructed surfaces clearly impacts any measurements regarding size, shape, and orientation. Note the high running time for our accurate reconstruction of this dataset; a large void surrounding the foam causes a load imbalance when computing certain regions, with one thread taking the vast majority of work, and each numerically integrated streamline crosses tens, or hundreds of voxels before termination. Furthermore, numeric integration in distance fields is particularly slow, as lines stall on flat, steep-sided ridge- and valley-like structures.

HCCI: In combustion science, understanding the properties of a reaction localized to a flame front can reveal the impacts of turbulence on combustion efficiency. In this homogeneous charge compression ignition simulation, we compute the flame surface as ascending 2-manifolds. These surfaces are used to sample other fields, for instance scalar dissipation rate and heat release. Even slight deviations in the location of the surface can cause significant differences in the sampling, as these other fields tend to be aligned with the flame front as well.

Neuron: Automating the extraction of axons and dendrites remains a goal of neuroscience, as the quantity of data is a major challenge in building a connectome of the brain. The 2-saddle-maximum arcs of the MS complex have potential to help this process, as they encode all ridge-like structures. This dataset of selected neurons in Macaque, imaged with a 2-photon microscope, is first processed to reduce noise and smooth the signal before MS complex computation. While the differences between the steepest descent reconstruction and accurate ascending 1-manifolds is subtle, improving the accuracy of the 1-manifolds only doubles the run time with respect to steepest descent. Furthermore,



Excerpts from the full scale data are used for visualization of the resulting features.

	dimensions	# vertices	steepest (144 threads-fsm)	serial conv [15]	accurate (144 threads-fsm)	accurate mem(Gb)	speedup
lithium	280x280x280	21,952,000	1.2s	571s	17.8s	0.9	30x
hcci	560x560x560	175,616,000	9.5s	9,851s	128s	7.38	77x
neuron	1024x1024x1024	1,073,741,824	56s	–	114s	40.4	∞
foam	1055x1024x1024	1,106,247,680	63s	–	5,452s	45.5	∞

Fig. 9. The datasets used in our evaluation vary by size, topological complexity, and the class of feature for accurate reconstruction. The table reports the steepest descent [33] time, the time taken by the serial convergent algorithm, the time to compute the improved accurate discrete gradient, and the speedup over the serial convergent algorithm, where available.

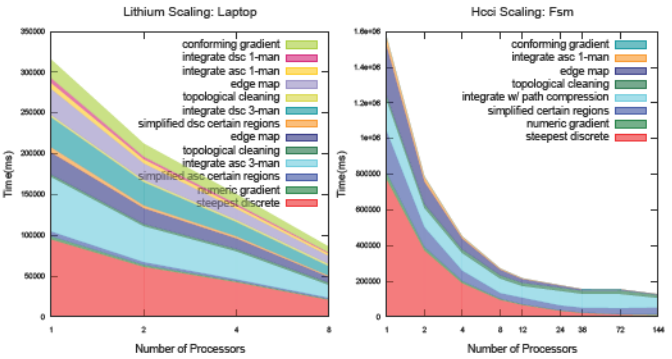


Fig. 10. Scaling for each component of the accurate workflow for the lithium on a 4-core laptop (left), and HCCI on an analysis server (right). Each plot shows, from bottom to top, the order and duration of each stage of the computation. Note that to illustrate the full timing characteristics, the lithium computation includes all improved accuracy components: both ascending and descending 1- and 3- manifolds. The HCCI computation includes only accurate ascending 1- and 3- manifolds.

any improvement in the reconstruction could have meaningful impacts in being able to trace lines automatically. For example, an algorithm deciding which lines to include in an axon may use geometric continuity as a criterion, which can be thrown off due to steepest-descent artifacts.

5.5 Performance Results

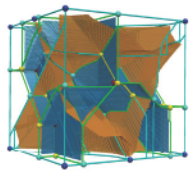
The parallel performance of our approach is evaluated on a commodity laptop and an analysis server.

Laptop: an off-the-shelf laptop with an Intel Core i7-3630QM processor with 4 cores running at 2.4 GHz, and 16 GB DRAM.

Fsm: is a 72-core (4-socket x 18 core) Intel Xeon E7-8890 v3 (Haswell-EX) running at 2.4 GHz, with 3 TB DRAM in a Brickland-EX NUMA server platform.

Figure 9 shows the run times for the improved feature computation for our selected examples. All reported run times in this section exclude the I/O to read the data and write the final discrete gradient vector field. Figure 10 shows the execution time for each component of our algorithm as the number of threads increases for two selected use cases. "Steepest discrete" computes the initial steepest descent discrete gradient using Robins' algorithm [33]. "Numeric gradient" reports the time compute and cache central difference gradients for each vertex of the mesh. "Simplified certain regions" records the time needed to create the simplified extremum graph and grow the certain regions to terminate integral lines. "Integrate asc/dsc 3-manifold" records the time taken to first compute numeric integral lines with path compression and then to re-integrate the boundaries. Although theoretically an embarrassingly parallel operation, this stage did not scale well and came to dominate run times at high core counts. Given the path compression, checking for terminal regions, and low-cost kernel to advect a line inside a voxel, it is likely that this operation is memory-bound, or suffers from enforcing cache-coherence. "Topological cleaning" refers to removing isolated connected components from the terminal map that are below resolution of the representative capacity of the discrete gradient field. "Edge map" refers to labeling each cell of the mesh based on whether it belongs to the boundary of the numerically computed termination maps. "Integrate asc/dsc 1-man" reports the time takes to gather saddles and integrate and digitize streamlines. Finally "conforming gradient" refers to the time needed to re-visit and re-compute the discrete gradient of vertices having cells in their lower star whose label is non-zero, i.e., having been identified as part of an accurate manifold. Figure 11 compares the runtimes for discrete gradient computation of the Topology ToolKit [36] to our implementation of steepest descent [33] and improved accuracy. Note that TTK results are similar in quality to steepest descent, e.g. the left column of figure 7.

The performance of our approach highly depends on both the topological complexity of the data as well as the features to be accurately extracted. We found that greatest obstacles for performance were long integral paths (slow integration), large flat regions (imbalanced certain region expansion), and steep-sided flat ridges and valleys (stalled integration). Given that critical points tend to be sparse in a dataset,



single thread 8-Gauassians at 150^3

	run time	mem(Mb)
TTK [36]	18.2s	695
Steepest	13.4s	71
Accurate	39.0s	155

Fig. 11. Comparison of single thread execution of of steepest descent and improved accuracy with TTK v0.9.6 [36] for the 8-Gaussians example. The features extracted by TTK (left) share similar biases as other steepest descent techniques. The run times are comparable, with differences due to cubical vs. triangulated meshes and choice of steepest descent algorithm. The large memory footprint of TTK is due to explicit storage of discrete gradient pairs, which could be improved in the future.

ascending and descending 1-manifolds can be computed with low overhead with respect to a purely steepest approach.

Experimental parameters: Numeric integration was implemented with an adaptive Euler advector, with maximum error tolerance set to 0.01 grid units. We found that further reducing the error threshold did not yield appreciable differences in feature quality. Furthermore, a hard limit of 10,000 iterations was set to force termination of numeric integration, only rarely reached in practice in degenerate regions. For constructing accurate 3-manifolds, the pre-simplification threshold was set to 0.01% of maximum persistence for each dataset, reducing numeric integration time by up to 3.5x (for HCCI), while maintaining sufficient resolution of feature for all subsequent analysis.

Memory tradeoffs: Memory size was the most significant factor limiting which data could be computed on the laptop. For each vertex of the input mesh a high-water mark of 41 bytes are used: scalar value (4 bytes), cached numeric gradient (12 bytes), cached lower stars (8 bytes), discrete gradient (8 bytes), origin/termination map (4 bytes), boundary map (8 bytes), and internal markers (1 byte). Removing the cached numeric gradient saves 12 bytes/vertex, however invokes a 2-4x performance penalty for the numeric integration stages. For the lithium dataset, this corresponded to a 2.07x increase in run time for the end-to-end algorithm. Similarly, not storing lower stars in cache saves 8 bytes/vertex, invoking a 2.8x performance penalty for discrete gradient assignment stages. For the lithium data, this corresponded to a 1.4x increase in run time.

6 Conclusions/Future Work

We have introduced a new scalable approach for computing discrete gradient fields with accurate geometry on shared memory multi-core systems. It has enabled faster computation of MS complexes for larger data. Memory consumption and bandwidth has been a limiting factor in SMP implementation, and will ultimately necessitate distributed computation to solve the largest datasets. In this setting, we expect that our shared-memory approach can become the component of such a distributed computation that handles the on-node parallelism. A limitation of the approach is that it uses a 3-manifold labeling to extract accurate 2-manifolds; in the cases of strangulations, where a saddle is doubly connected to the same extremum, the 2-manifold of that saddle cannot be reconstructed accurately. We will investigate integrating this approach with tools for general exploration of MS complexes, such as TTK [36]. Finally, we will investigate alternative methods for performing the numeric integration to achieve better scalability.

Acknowledgments

This work is supported in part by NSF: CGV: Award:1314896, NSF:IIP Award :1602127 NSF:ACI:award 1649923, DOE/SciDAC DESC0007446, CCMSC DE-NA0002375, and PIPER: ER26142 DE-SC0010498. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. "This material is based upon work supported by the Department of Energy, National Nuclear Security Administration, under Award Number(s) DE-NA0002375." This work was also supported by the Department of Energy Office of Science, Office of Advanced Scientific Computing Research under the guidance of Dr. Lucy Nowell and Richard Carson.

REFERENCES

[1] R. F. W. Bader. Atoms in molecules. *Accounts of Chemical Research*, 18(1):9–15, 1985. doi: 10.1021/ar00109a003

[2] R. F. W. Bader. *Atoms in Molecules – A Quantum Theory*. Oxford University Press, 1994.

[3] R. F. W. Bader. A bond path: a universal indicator of bonded interactions. *The Journal of Physical Chemistry A*, 102(37):7314–7323, 1998. doi: 10.1021/jp981794v

[4] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P. Bremer. Topoms: Comprehensive topological exploration for molecular and condensedmatter systems. *Journal of Computational Chemistry*, 0(0), 2018. doi: 10.1002/jcc.25181

[5] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.

[6] P.-T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010.

[7] H. A. Carr, G. H. Weber, C. M. Sewell, and J. P. Ahrens. Parallel peak pruning for scalable smp contour tree computation. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 75–84, Oct 2016. doi: 10.1109/LDAV.2016.7874312

[8] A. Cayley. On contour and slope lines. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XVIII:264–268, 1859.

[9] A. Chattopadhyay, G. Vegter, and C. K. Yap. Certified computation of planar Morse-Smale complexes. *Journal of Symbolic Computation*, 78:3 – 40, 2017. Algorithms and Software for Computational Topology. doi: 10.1016/j.jsc.2016.03.006

[10] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proceedings of the 19th Symposium on Computational Geometry*, pp. 361–370, 2003.

[11] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry*, 30:87–107, 2003.

[12] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002. doi: 10.1007/s00454-002-2885-2

[13] R. Forman. A user's guide to discrete Morse theory. In *Séminare Lotharinen de Combinatore*, vol. 48, 2002.

[14] C. Gueunet, P. Fortin, and J. Jomier. Contour forests: Fast multi-threaded augmented contour trees. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 85–92, Oct 2016. doi: 10.1109/LDAV.2016.7874333

[15] A. Gyulassy, P. Bremer, and V. Pascucci. Computing Morse-Smale complexes with accurate geometry. *IEEE Transactions on Visualization and Computer Graphics*, 18:2014 – 2022, 12 2012.

[16] A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. A practical approach to Morse-Smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

[17] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E.Bringa, A. Higginbotham, and B. Hamann. Topologically clean distance fields. *IEEE Transactions on Computer Graphics and Visualization*, 13(6):1432–1439, 2007.

[18] A. Gyulassy, D. Günther, J. A. Levine, J. Tierny, and V. Pascucci. Conforming Morse–Smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2595–2603, Dec 2014. doi: 10.1109/TVCG.2014.2346434

[19] A. Gyulassy, A. Knoll, K. Lau, B. Wang, P. Bremer, M. Papka, L. Curtiss, and V. Pascucci. Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials. *IEEE Trans. Vis. Comp. Graph.*, PP(99):1–1, 2015.

[20] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3D scalar fields. *IEEE Transactions on Computer Graphics and Visualization*, 12(4):474–484, 2006.

[21] A. Gyulassy, T. Peterka, V. Pascucci, and R. Ross. Characterizing the parallel computation of Morse-Smale complexes. In *Proceedings of IPDPS '12*. Shanghai, China, 2012.

[22] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2080–2087, 2011.

[23] G. Kindlmann, C. Chiw, N. Seltzer, L. Samuels, and J. Reppy. Diderot: a domain-specific language for portable parallel scientific visualization and image analysis. *IEEE Transactions on Visualization and Computer Graphics (Proceedings VIS 2015)*, 22(1):867–876, Jan. 2016. doi: 10.1109/TVCG.2015.2467449

[24] A. G. Landge, V. Pascucci, A. Gyulassy, J. C. Bennett, H. Kolla, J. Chen, and P.-T. Bremer. In-situ feature extraction of large scale combustion simulations using segmented merge trees. In *High Performance Computing, Networking, Storage and Analysis, SC14: International Conference for*, pp. 1020–1031. IEEE, 2014.

[25] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions Visualization and Computer Graphics*, 12(5):1052–1060, 2006.

[26] T. Lewiner. Constructing discrete Morse functions. Master's thesis, Department of Mathematics, PUC-Rio, 2002.

[27] J. C. Maxwell. On hills and dales. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XL:421–427, 1870.

[28] J. Milnor. *Morse Theory*. Princeton University Press, 1963.

[29] D. Morozov and G. H. Weber. Distributed contour trees. In *Topological Methods in Data Analysis and Visualization*, 2014.

[30] G. Norgard and P.-T. Bremer. Robust computation of MorseSmale complexes of bilinear functions. *Computer Aided Geometric Design*, 30(6):577 – 587, 2013. Foundations of Topological Analysis. doi: 10.1016/j.cagd.2012.03.017

[31] J. Reininghaus and I. Hotz. Combinatorial 2d vector field topology extraction and simplification. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, Mathematics and Visualization, pp. 103–114. Springer, 2011.

[32] J. Reininghaus, C. Lowen, and I. Hotz. Fast combinatorial vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 17:1433–1443, 2011. doi: 10.1109/TVCG.2010.235

[33] V. Robins, P. Wood, and A. Sheppard. Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1646–1658, 2011.

[34] N. Shivashankar, M. Senthilnathan, and V. Natarajan. Parallel computation of 2d Morse-Smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, to appear, 2012.

[35] T. Sousbie. The persistent cosmic web and its filamentary structure - I. theory and implementation. *Monthly Notices of the Royal Astronomical Society*, 414(1):350–383, 6 2011.

[36] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology ToolKit. *IEEE Transactions on Visualization and Computer Graphics (Proc. of IEEE VIS)*, 2017. `https://topology-tool-kit.github.io/`.