


TopoMS: Comprehensive Topological Exploration for Molecular and Condensed-Matter Systems

Harsh Bhatia ^{*,[a]} Attila G. Gyulassy,^[b] Vincenzo Lordi,^[c] John E. Pask,^[d] Valerio Pascucci,^[b] and Peer-Timo Bremer^[a,b]

We introduce *TopoMS*, a computational tool enabling detailed topological analysis of molecular and condensed-matter systems, including the computation of atomic volumes and charges through the quantum theory of atoms in molecules, as well as the complete molecular graph. With roots in techniques from computational topology, and using a shared-memory parallel approach, *TopoMS* provides scalable, numerically robust, and topologically consistent analysis. *TopoMS* can be used as a command-line tool or with a GUI (graphical user interface), where the latter also enables an interactive exploration of the molecular graph. This paper presents algorithmic details of *TopoMS* and

compares it with state-of-the-art tools: Bader charge analysis v1.0 (Arnaldsson et al., 01/11/17) and molecular graph extraction using Critic2 (Otero-de-la-Roza et al., *Comput. Phys. Commun.* 2014, 185, 1007). *TopoMS* not only combines the functionality of these individual codes but also demonstrates up to 4× performance gain on a standard laptop, faster convergence to fine-grid solution, robustness against lattice bias, and topological consistency. *TopoMS* is released publicly under BSD License. © 2018 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.25181

Introduction

An important aspect of exploring the physical and chemical properties of complex molecular and condensed-matter systems is to understand the charge transfer between atoms and identify the presence of ionic charges and bonding structures. However, atomic charges in molecules are not directly observable through experimentation or simulation. Instead, the density of electronic charge can be calculated through quantum mechanical theory, which can then be used to compute atomic charges and other related properties.

This research direction, called the *Quantum Theory of Atoms in Molecules* (QTAIM),^[1–3] suggests that it is possible to understand intra- and intermolecular interactions based on the topology of the electron charge density. According to the QTAIM, the topological features of the electron charge density field, i.e., its critical points, basins, and ascending and descending manifolds, have physical meaning and can be used to partition the space into *topological basins**—regions in space associated with individual atoms. Each such atomic basin typically contains a single charge density maximum and is separated from other basins by *interatomic- or zero-flux surfaces*—surfaces on which the normal component of the gradient of the electron charge density is zero. QTAIM analysis[†] enables computing atomic properties such as atomic volume and atomic charge by integrating within atomic basins, and facilitates downstream analysis by describing atomic interactions, especially chemical bonding. Since it relies only on the electron charge density, the QTAIM has proven to be a versatile and general framework for exploring molecular and condensed-matter systems, and can be used to explore the

data obtained through various sources, e.g., quantum mechanical calculations and X-ray crystallography.

Performing QTAIM analysis, however, is not a simple task and poses many computational challenges. Several tools with increasing accuracy and robustness have been developed.^[4–14] While these tools have been successfully utilized over many years, they do not take advantage of parallel computing architectures, and therefore can be prohibitively slow when applied to large-scale data. Furthermore, these tools generally focus on computing atomic volumes and corresponding charges only, even though the underlying framework, topological analysis, can provide richer information about the data. For example, the so-called *molecular graph*^[15] describes how the atoms (maxima of the electron charge density corresponding to atom centers) are connected, and is a subgraph of the

[a] H. Bhatia, P.-T. Bremer
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, USA
E-mail: hbhatia@llnl.gov

[b] A. G. Gyulassy, V. Pascucci, P.-T. Bremer
Scientific Computing and Imaging Institute, The University of Utah, Salt Lake City, UT, USA

[c] V. Lordi
Materials Science Division, Lawrence Livermore National Laboratory, Livermore, CA, USA

[d] J. E. Pask
Physics Division, Lawrence Livermore National Laboratory, Livermore, CA, USA

Contract grant sponsor: U.S. Department of Energy Office of Science, Advanced Scientific Computing Research (ASCR), SciDAC program; Contract grant sponsor: U.S. Department of Energy Office of Science, Basic Energy Sciences (BES), SciDAC program; Contract grant sponsor: The National Science Foundation (NSF); Contract grant number: 1314896

© 2018 Wiley Periodicals, Inc.

*Also known as *Bader volumes* in the literature.

†Also known as *Bader analysis* in the literature.

complete *topological graph*, which describes the connectivity between all critical points in the field.

By providing an abstraction that encodes the structural properties of the underlying function, a general topological framework can enable direct reasoning using the features of interest. Detailed topological analysis of scalar functions, in general, has proved useful for a variety of other scientific applications, such as combustion science,^[16,17] turbulent mixing of fluids,^[18] and cosmology.^[19] In the context of chemistry, topological techniques have primarily been applied to perform bond detection.^[20,21] Furthermore, a more general research field, the *Quantum Chemical Topology* (QCT),^[22] aims at understanding the topology of other relevant fields, such as electron localization function and molecular electrostatic potential.

Although there exist tools, such as Critic^[23] and Critic2,^[24] for detailed topological analysis to support the QTAIM and the QCT, they have significant limitations. For instance, although Critic2 offers versatile functionality, it can generate topologically inconsistent results, typically due to the presence of noise and discretization artifacts.^[24]

In this work, we present *TopoMS*, a scalable, numerically robust, and topologically consistent computational tool to perform the two key tasks aforementioned: (1) analysis of atomic basins, i.e., the computation of atomic volumes and associated charges, and (2) the extraction of detailed molecular graphs. To accomplish this, *TopoMS* computes the *Morse–Smale complex* (MSC), a topological construct that captures the overall shape of a function by studying its gradient behavior, and utilizes it for analysis. *TopoMS* stands for **topological analysis of molecular systems using Morse–Smale complexes**. While exploratory MSC techniques have previously been applied to the analysis of chemical systems,^[20,21] *TopoMS* is the first-of-its-kind software package that explicitly maps these concepts to QTAIM analysis, and leverages their advantages in software. In particular, the focus of *TopoMS* and the contributions of this paper are listed below.

- **Scalability.** *TopoMS* uses a shared-memory parallel approach for the computation of topological basins and the complete topology of the given function. Our embarrassingly parallel algorithm scales with the available number of cores, and shows up to 4× performance gain on a standard laptop computer.
- **Topological consistency and numerical robustness.** By using a combination of highly accurate numerical integration and robust discrete representations, *TopoMS* provides numerically robust results and guarantees a topologically consistent molecular graph. We demonstrate fast convergence to fine-grid results and stability with respect to the mesh orientation, i.e., low lattice bias.
- **Interactive denoising.** *TopoMS* provides an interactive interface to explore the impact of noise and discretization artifacts on the resulting molecular graph. The topology of the data can be simplified by systematically removing noisy features in order of their importance.
- **Simple and open-source API.** *TopoMS* provides a simple-to-use API, and can be used as both a command-line

tool or with an associated GUI (graphical user interface) alongside a viewer. This paper describes the architecture of *TopoMS*, which is released open source under BSD license on <https://github.com/LLNL/TopoMS>.

Although the scope of this paper and the first version of *TopoMS* introduced here is limited to a detailed analysis of electron charge density, our software framework allows expanding the scope to encompass other chemical fields, i.e., the computation of QCT, which is planned for the future.

Fundamentals

The definitions used in the QTAIM^[1–3,15] find strong parallels in Morse theory,^[25] a well-studied branch of scalar-field topology. Indeed, when properly accounting for the singularities at atom locations, the QTAIM's "stable configurations" induced by the electronic charge density field satisfy the nondegeneracy requirements that define a Morse function.^[25, Chapter 1] Our principal motivation for recasting the QTAIM in terms of Morse theory is that many efficient and robust algorithms have been introduced for the computation and manipulation of the topology of the latter,^[26–30] enabling fast, accurate, and self-consistent topological analysis of electron charge density fields. In particular, by first computing the Morse–Smale complex,^[26] a topological structure that encodes the gradient flow behavior of a function, it becomes straightforward to extract nuclear positions, atomic interaction lines, interatomic surfaces, and basins of attractors directly from the charge density, which can then be mapped consistently to the meta-information about the system under investigation. This section first discusses the relevant background in topological analysis of scalar fields, and then connects it to the QTAIM.

Topology of scalar fields

Given a smooth function in three-dimensional (3D) space $f: \mathbb{M} \rightarrow \mathbb{R}$, a point \mathbf{p}_c is called a *critical point* if the gradient of f at \mathbf{p}_c is zero, i.e.,

$$\nabla f(\mathbf{p}_c) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)_{\mathbf{p}_c} = \mathbf{0}.$$

A critical point is nondegenerate if the Hessian, i.e., the matrix of second partial derivatives at the point, is invertible. Equivalently, a critical point is nondegenerate if the rank ω (number of nonzero eigenvalues) of the Hessian matrix equals 3 (in 3D). A nondegenerate critical point \mathbf{p}_c of f can be classified based on its signature σ , which is the algebraic sum of the signs of the eigenvalues of the Hessian.^[25] The number of negative eigenvalues is also called the *index* of the critical point.[‡] Table 1 shows the four types of critical points that can exist for functions in 3D.

A function f is a *Morse function* if all its critical points are nondegenerate; any smooth function f can be infinitesimally

[‡]The reader may encounter an alternate definition of index as the number of *positive* eigenvalues in some literature.^[31]

Table 1. Four types of critical points exist for 3D scalar functions, each having a specific meaning in the QTAIM. Critical points are classified based on their index, the number of negative eigenvalues of the Hessian matrix, or equivalently, their rank ω and signature σ .

(ω, σ)	Index	Type of critical point	Name in the QTAIM
(3, -3)	3	Local maximum	Nuclear critical point (NCP)
(3, -1)	2	2-saddle	Bond critical point (BCP)
(3, +1)	1	1-saddle	Ring critical point (RCP)
(3, +3)	0	Local minimum	Cage critical point (CCP)

perturbed into a Morse function. 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 \mathbf{p} is the solution to

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

with initial value $L(0)=\mathbf{p}$. Each integral line has an origin and destination at critical points of f corresponding to the limits as t respectively approaches $-\infty$ and ∞ . *Ascending* and *descending* manifolds are obtained as clusters of integral lines having a common origin and destination, respectively. The descending manifolds of f form a cell complex that partitions \mathbb{M} ; this partition is called the *Morse complex*. Similarly, the ascending manifolds also partition \mathbb{M} in a cell complex. In a d -dimensional domain, an index- i critical point is the destination for an i -dimensional descending manifold and the origin of a $(d-i)$ -dimensional ascending manifold. A Morse function f is a *Morse–Smale function* if ascending and descending manifolds of its critical points intersect only transversally. This intersection forms a cell complex known as the *Morse–Smale complex*, whose 1-skeleton is formed by *nodes* at critical points of f , and *arcs*, the 1-manifold integral lines connecting nodes that differ in index by one.

A fundamental result in topology is the Poincaré–Hopf theorem, which connects the topology of a given domain with the space of possible vector functions on that domain. In the context of Morse theory, an equivalent result states that the alternating sum of critical points by index, also called the *Morse sum*, equals the *Euler characteristic* $\chi(\mathbb{M})$ of the underlying domain \mathbb{M} , i.e.,

$$\chi(\mathbb{M})=\sum_i (-1)^i c_i, \quad (2)$$

where c_i is the number of critical points of index i . Since the Euler characteristic is an invariant of \mathbb{M} and does not depend on f , the strong implication of this result on the topology of f is that there cannot exist a physically consistent f that does not satisfy this property. Therefore, any unwarranted critical points, e.g., due to noise, must always exist as pairs of critical points of consecutive indices: a local maximum and a 2-saddle, a 2-saddle and a 1-saddle, or a 1-saddle and a local minimum. Morse theory also defines a systematic way of canceling these pairs of critical points such that the result described above remains valid.^[31] Nonperiodic domains, such as often used to simulate isolated molecular systems, are homeomorphic (topologically

equivalent) to a 3-ball (3D filled sphere), with $\chi=+1$. However, domains periodic in all three directions, which are often used to represent condensed-matter systems, are homeomorphic to a 3-torus, for which $\chi=0$. To be topologically consistent, an analysis must, at least, respect these invariants.

The quantum theory of atoms in molecules (QTAIM)

QTAIM analysis^[1–3,15] is a powerful and widely used tool to study chemical bonding, in particular, by analyzing the charges captured by atoms in molecules. The QTAIM utilizes topological ideas to provide mathematically rigorous and physically intuitive descriptions of atomic properties. By studying the gradient behavior of the electron charge density ρ , the QTAIM decomposes the space into regions associated with individual atoms. Such a decomposition is computed by considering the critical points of ρ . The QTAIM associates each type of critical point with an element of the chemical structure, as summarized in Table 1. At the positions of atomic nuclei, (local) maxima of ρ are found; such maxima are called *nuclear critical points* (NCPs). In the QTAIM with ideal point-nuclei, a nuclear maximum is not a true critical point, as $\nabla\rho$ is discontinuous there. However, in nature, nuclei are finite (though small) and so critical points exist at nuclear positions. Furthermore, in pseudopotential calculations, potentials are smooth at the nuclear positions and so critical points exist there. The gradient trajectories described in the QTAIM are equivalent to integral lines in scalar functions. Between two NCPs, there may exist a *bond critical point* (BCP) or 2-saddle if the corresponding atoms share electrons. The BCP, therefore, describes the “position” of the bond, and the gradient trajectories describe the *atomic interaction lines*, which correspond to *bond paths*^[15] when the forces on all atoms vanish. In terms of Morse theory, these atomic interaction lines are identified by the 2-saddle-maximum arcs of the Morse–Smale complex, or equivalently, the ascending 1-manifolds of 2-saddles. The basin of attraction associated with an NCP, i.e., the region whose gradient trajectories of ρ terminate at the NCP, defines the *atomic basin* occupied by the corresponding *topological atom*. These atomic basins are identified by descending 3-manifolds of the maxima of the Morse–Smale complex. Basins are separated by *interatomic surfaces*, which satisfy the zero-flux condition, i.e., $\nabla\rho(\mathbf{x}) \cdot \hat{n}(\mathbf{x})=0$, for every point \mathbf{x} on the surface, where $\hat{n}(\mathbf{x})$ is the unit normal to the surface at \mathbf{x} . These separation surfaces are equivalent to the descending 2-manifolds of 2-saddles of the Morse–Smale complex. A more detailed discussion of different types of critical points, gradient paths, and manifolds was provided by Malcolm and Popelier.^[32]

Once atomic basins are identified, various physical properties can be computed for each atom. For example, given the atomic basin Ω of a topological atom, the corresponding atomic volume V_Ω and atomic charge q_Ω ,[§] can be computed by integrating over the basin, i.e.,

[§]Also known as *Bader charge* in the literature.

$$V_{\Omega} = \int_{\Omega} d\mathbf{x},$$
$$q_{\Omega} = Z_{\Omega} - \int_{\Omega} \rho(\mathbf{x}) d\mathbf{x},$$

where $d\mathbf{x}$ is the volume element and Z_{Ω} is the charge of the corresponding nucleus.

Overview of combinatorial underpinnings

One of the fundamental techniques of computational science is to represent functions via discrete samples, e.g., on the vertices of a grid. While Morse theory (and the QTAIM) is well defined for continuous functions, discretization imposes challenges for subsequent analysis of the functions, as the interpolation used to reconstruct functions between sample points often biases analysis results. Challenges in direct numeric computation of Morse–Smale complexes include consistent identification of critical points and ensuring that integral lines do not cross separatrices. The approach taken in *TOPOMS* embraces the discrete world of the mesh representation of space, allowing for robust, combinatorial computation, ensuring consistency in the computed Morse–Smale complex that forms the basis for extracting features of the QTAIM. The adaptation of continuous Morse theory to meshes, called *discrete Morse theory*, was introduced by Forman,^[33] and has formed the basis for the most successful algorithms for computing Morse–Smale complexes for volumetric data.^[27–29] The motivation for using discrete Morse theory is that in computing an integral line (as in eq. (1)) and its destination, the limit as $t \rightarrow \infty$ reduces to a simple traversal of cells according to a discrete flow operator Φ . *TOPOMS* combines both numeric and combinatorial approaches to attain an unbiased, accurate decomposition of space while retaining consistency in the topological representation.

We provide a brief introduction to the terminology and theoretical background of discrete Morse theory that is used in *TOPOMS*. Let \mathcal{M} be a mesh representation of \mathbb{M} , and $f: \mathcal{V} \rightarrow \mathbb{R}$ be a scalar-valued function defined on \mathcal{V} , the set of vertices of

\mathcal{M} . For volumes represented as 3D regular grids, \mathcal{M} is composed of *cells* of dimension 0, 1, 2, and 3, called *vertices*, *edges*, *quadrilaterals*, and *hexahedra*, respectively. The *boundary* of a cell α , denoted $\partial\alpha$, is composed of the lower dimensional cells whose vertices form a proper subset of α . For cells $\alpha, \beta \in \mathcal{M}$, α is a *face* of β , denoted $\alpha < \beta$, if and only if α is on the boundary of β . In this case, β is a *co-face* of α . Furthermore, if $\dim(\alpha) = \dim(\beta) - 1$, we say α is a *facet* of β and β is a *co-facet* of α , and denote this $\alpha \prec \beta$. For example, a hexahedron in a 3D regular grid has six facets: the two quadrilaterals bounding the hexahedron in each axis direction, whereas the faces of a hexahedron include six quadrilaterals, twelve edges, and eight vertices. The *star* of a cell α , denoted $\text{St}(\alpha)$, is the set of co-faces of α in \mathcal{M} . The *lower star* of a vertex α , denoted $\text{St}^{-}(\alpha)$ is the subset of $\text{St}(\alpha)$ where for each $\beta \in \text{St}^{-}(\alpha)$, α is the vertex with highest value among faces of β . For consistent resolution in cases of equal values, we use the *simulation of simplicity*^[34] to assign a unique value f^* to each vertex v_i in \mathcal{V} . In particular, any given function f can be perturbed into an injective function $f^*: \mathcal{V} \rightarrow \mathbb{R}$, e.g., by using the memory location of the function values to break ties, i.e., $f^*(v_i) = f(v_i) + i\epsilon$, for $\epsilon > 0$, when $f(v_i) = f(v_j)$ for $v_i \neq v_j \in \mathcal{V}$.

A *vector* in the discrete sense is a pairing of cells $\langle \alpha, \beta \rangle$, where $\alpha \prec \beta$; we say that an arrow points from α to β , where α is the tail and β is the head of the arrow. The *direction* of the arrow relates the combinatorial notion of the pairing to the geometric interpretation of the flow, and is given by $B(\beta) - B(\alpha)$, where $B(\alpha)$ denotes the barycenter of a cell, i.e., the average coordinate location of its vertices. A *discrete vector field* V on \mathcal{M} is a collection of vectors $\langle \alpha, \beta \rangle$ of cells of \mathcal{M} such that each cell is in at most one vector of V . Cells that do not appear as the head or tail of a discrete vector in V are defined as *critical cells*, with the index of criticality equal to the dimension of the cell. For example, for 3D regular grids, critical vertices, edges, quads, and hexahedra are minima, 1-saddles, 2-saddles, and maxima, respectively. A discrete gradient field is illustrated in Figure 1. We can now define the flow operator Φ , which acts as the combinatorial equivalent of an integration

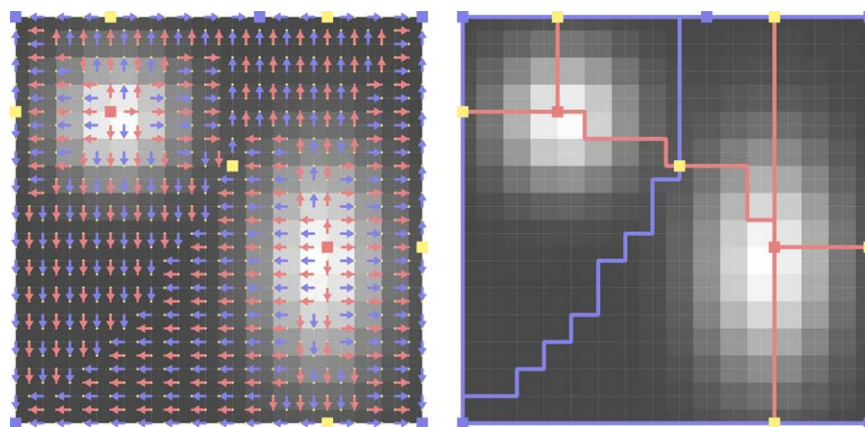


Figure 1. Discrete gradient field of a simple synthetic function. Left: Blue arrows illustrate the pairing of 0-cells (vertices) with 1-cells (edges), and red arrows show the pairing of 1-cells (edges) with 2-cells (faces). Right: Critical i -cells are shown as blue ($i = 0$), yellow ($i = 1$), and red ($i = 2$) squares, which are the minima, saddles, and maxima, respectively, of the underlying function. The discrete gradient field defines the Morse–Smale complex, with combinatorial separatrices (blue and red lines) connecting the critical cells. [Color figure can be viewed at wileyonlinelibrary.com]

step in the direction of the gradient in eq. (1). Formally, for an i -cell α ,

$$\Phi(\alpha) = \partial^{i-1} v(\alpha), \quad (3)$$

where we use $v(\alpha)$ 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. The combinatorial equivalent of an integral line (as in eq. (1)) is a V -path, a sequence of cells

$$\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2, \dots, \beta_r, \alpha_{r+1}$$

such that for each $j=0, \dots, r$, the vector $\langle \alpha_j, \beta_j \rangle \in V$, and α_j and α_{j+1} are both facets of β_j . Note that the discrete flow operator produces V -paths, i.e., $\alpha_i \in \Phi(\alpha_{i-1})$. A discrete vector field in which each nontrivial V -path has disjoint start and end points is a *discrete gradient field*, denoted G , of a discrete Morse function. Note that this condition implies that a discrete gradient field G does not contain any loops. The discrete gradient field G is the combinatorial analogue of ∇f ; thus, the discrete ascending and descending manifolds of a critical cell are given by the collections of V -paths starting and ending at that critical cell, respectively. By convention, discrete gradient vectors point in descending directions with respect to f^* , each V -path therefore having cells with monotonically decreasing vertices.

Related Work

The early approaches for QTAIM analysis were based on analytic formulations of electron charge density and corresponding topological elements, e.g., analytic representations of zero-flux surfaces and their application in computing properties of corresponding topological atoms. Such techniques^[3,6–10,12,35,36] were useful for small molecular systems, but had limited scalability. One example of contemporary and well-maintained software in this category is AIMAll,^[13] which performs the analysis starting from molecular wavefunction data.

With increasing computational and simulation capabilities, many recent advances have focused on improving grid-based methods,^[4,5,11,37–47] although initial ideas were proposed much earlier.^[48,49] The key advantage of such approaches is their reduced computational cost. Furthermore, since the grid-based data is readily available in density functional theory (DFT) applications, such techniques are particularly useful for real-space and planewave-based DFT calculations. Such techniques work with sampled data on regular (or adaptive^[43]) grids by detecting critical points and tracing gradient paths to determine atomic basins.

A particularly relevant grid-based technique was presented by Henkelman et al.^[5] This technique finds the atomic basins by integrating gradient-ascent trajectories from every grid point. Each trajectory was represented as a sequence of grid points, with the algorithm extending a trajectory in the direction of steepest ascent among the 26 neighbors of a point. To improve the computational efficiency, the trajectory integration could be stopped if it encountered a previously assigned grid point, making this approach scale linearly with system

size. Nevertheless, since the integration picked steepest-ascent trajectories, this algorithm introduced a bias that caused interatomic surfaces to artificially follow the orientation of the lattice. Sanville et al.^[40] first pointed out this limitation and presented a modified algorithm to remove this bias, where trajectories were not constrained to the grid, yet retained the linear scaling of the original approach. Tang et al.^[41] also improved the original algorithm^[5] to remove the lattice bias. In their approach, although the trajectories are constrained to grid points, a correction vector between the true trajectory and the discretized “on-grid” trajectory is calculated and propagated alongside. When the correction vector becomes large enough, the discrete trajectory is corrected. Nevertheless, the piecewise-constant interpretation of the gradient induces integration error that scales linearly with grid spacing, requiring a finely sampled grid, which limits the application to large and complex systems. Yu and Trinkle^[44] further advanced the state of the art by presenting a technique for subgrid accuracy in the computation of atomic volumes. They introduced a weighting scheme to represent the fraction of a *voxel* (a volume element in a 3D grid) to be associated with a surrounding grid point, and therefore, with a surrounding atomic basin. These techniques^[5,41,44] form the basis of one of the most widely used and publicly available tools for QTAIM analysis, which was published by Arnaldsson et al.^[4]

There also exist other approaches for grid-based analysis, such as the work of Rodríguez et al.,^[42,43] which utilize cellular grids, both fixed and adaptive, and compute the QTAIM properties without explicitly computing the zero-flux surfaces. This technique has been implemented in the commercial ADF Modeling Suite.^[14]

In the context of critical point computation, Rodríguez^[46] proposed a grid-based approach using the Newton-Raphson (NR)^[50] method for root finding. By applying filtering on starting points and appropriate vectorization and parallelization, a modified NR approach was implemented. Hernández-Esparza et al.^[47] provided a GPU (graphics processing unit) implementation of this approach for better scalability.

Topological analysis of several other related fields of interest, such as the electron localization function and molecular electrostatic potential, has also been explored: an overview was provided by Popelier.^[22] An example in the context of the QTAIM is the analysis of the Laplacian of the electron charge density.^[51,52] We shall focus here on a recent open-source software package, Critic2,^[23,24] developed by Otero-de-la-Roza et al., which can be used for a detailed exploration of such fields. In particular, Critic2 can be used not only to compute the charge inside atomic basins, but also to extract the complete molecular graph of the system, which contains all bond paths in the system. The molecular graph provides rich information about the system under investigation by capturing the bonding structures through automated analysis. Critic2 builds upon several published approaches, including some of the integration techniques discussed above, such as that given by Yu and Trinkle.^[44]

The tool presented in this paper, *TOPOMS*, offers two important features: QTAIM analysis of atomic volumes and charges, and the extraction of molecular graphs. In this context, we

consider Bader charge analysis v1.0^[4] and Critic2^[24] as representatives of the current state of the art for the two types of analysis, respectively. Section “TopoMS: Experiments and Results” will discuss the results of analysis using *TopoMS* in terms of qualitative, quantitative, and performance comparisons with these established tools.

Many algorithms have been proposed to compute discrete gradient vector fields, with variants for large-scale data,^[27] fast parallel computation,^[29,30] or accuracy with respect to some underlying continuous interpretation of the gradient.^[28] The most practical approaches rely on an embarrassingly parallel formulation, restricting discrete gradient vector assignment to the lower star of a vertex.^[29] Within the lower star, discrete gradient arrows are constructed in the direction of steepest descent. Gyulassy et al.^[28] showed that such local optimization leads to compounding errors in a *V*-path compared to a numerically traced integral line, leading to features heavily biased along the axis directions of the underlying computational grid. While the authors presented a solution, the algorithm required a serial traversal, and is too slow for practical use for QTAIM analysis. Instead, in *TopoMS*, we utilize an approach that allows fast parallel computation of discrete gradient fields that also conforms to a prior labeling computed through numeric integration.^[53]

TopoMS: Algorithmic Details

TopoMS is developed for the exploration of large-scale molecular and condensed-matter systems through topological analysis of relevant physical fields. In particular, *TopoMS* provides two types of analysis: (1) computation of the atomic basins as a volumetric decomposition of the domain with associated atomic charges, (2) the extraction of the complete molecular graph for a more general analysis of molecular fields, such as electron charge density or electrostatic potential. Thus, *TopoMS* can be used as a single comprehensive solution for topological analysis of such systems.

TopoMS has been designed keeping in mind three important considerations: (1) computational cost and scalability, (2) numerical robustness and consistency, and (3) flexible and easy-to-use API. Where possible, algorithms have been designed to utilize multicore architectures using shared-memory parallelism. Compounded effects of numerical errors in algorithms can often create inconsistent configurations, e.g., overlapping atomic basins. Special care has been taken to ensure consistency in the derived results. Finally, *TopoMS* has been designed in a modular fashion to allow adding new features as the software expands in the future, with the possibility of interfacing with standard simulation packages such as VASP.^[54]

We extract an accurate and consistent topological representation of features from the QTAIM by computing a discrete gradient vector field in a manner that matches numerically computed integral lines. First, we generate a map \mathcal{L} from vertices \mathcal{V} to maximal vertices by tracing integral lines numerically. For the case where only atomic basins are of interest, the computation returns this decomposition, matching atoms

to basins, and computing required statistics. Further computation of a molecular graph uses this map \mathcal{L} as a constraint in combinatorial construction of a discrete gradient vector field, ensuring that no *V*-paths cross the boundaries of atomic basins identified in the numeric integration. As a result, topological features can be extracted from the discrete gradient using simple combinatorial algorithms, with guarantees of consistency, while having the accuracy of the numerical approach.

The remainder of this section discusses implementation details for the numeric integration to create the map \mathcal{L} assigning each vertex of \mathcal{V} to a maximum of ρ (see Section “Computation of atomic basins”), and construction of the discrete gradient field and subsequent molecular graph extraction (see Section “Extraction of molecular graph”). Note that, by convention in discrete Morse theory, discrete gradient arrows point in the direction of descent, and critical points of index-*d* occur on *d*-cells of the mesh. However, the primary decomposition of space in the QTAIM is into atomic basins Ω , i.e., maxima of charge density and the integral lines that terminate there. To simplify the intuition of using discrete Morse theory for QTAIM analysis, we let *f* (and hence *f*^{*}) be the *negative* of the charge density, $f(x) = -\rho(x)$. In this way, vertices at local maxima in ρ occur at local minima in *f*, and the atomic basin Ω of an atom in ρ is equivalent to the vertices in *V*-paths terminating at the associated minimum in *f*, its discrete ascending manifold.

Computation of atomic basins

Atomic basins are the descending 3-manifolds of maxima in the charge density ρ , which can be computed as the limit as *t* approaches ∞ of eq. (1). Approximating this integral can be done with numeric integration, with values of $\nabla\rho$ evaluated between samples using some interpolation scheme. However, straightforward computation reveals several problems with this approach. For example, integral lines may terminate at nonmaximum critical points, and the interpolating function may create spurious critical points where integral lines terminate. Furthermore, evaluating an integral line from every vertex in the domain, tracing all the way to a maximum, is computationally prohibitive. Steps 1 and 2 given below discuss using combinatorial stopping criteria to both reduce the appearance of spurious maxima and decrease the number and length of integral lines that must be traced, and Step 3 presents the integration technique along with path compression optimizations to further reduce the computational cost.

Step 1: Avoiding computation in “vacuum” regions. Although topological atoms fill all space, i.e., every point in space belongs to a topological atom, regions far away from atomic centers typically contain zero or near-zero values of electronic charge density. Regions with extremely low charge values may be dominated by floating-point error, noise introduced by basis (e.g., Fourier series) truncation, and other such errors associated with the methodology used to produce the data. For practical purposes (e.g., numerical integration over an atom’s volume and visualization), Popelier^[55] implemented an envelope of constant density, typically set at 0.001 a.u.,^[2] to

define a practical boundary. Rodríguez et al.^[42] also used this idea of “screening” the data to filter small values, both to improve computational efficiency and avoid numerical noise. Similarly, we define “vacuum regions” as regions with charge density below a user-specified *vacuum threshold*.

Although all subsequent processing in our approach handles these regions without issue, we pre-screen the data to reduce computational cost. To eliminate low-valued regions in ρ , we correspondingly eliminate high-valued regions in f^* . More specifically, we define the *sublevel complex* with respect to the threshold f_t as the complex formed by the union of the lower stars of vertices having a value below the threshold, i.e., $\mathcal{V}_{f_t} = \{\alpha \in \mathcal{M} \mid \dim(\alpha) = 0 \text{ and } f^*(\alpha) < f_t\}$ and $\mathcal{M}_{f_t} = \bigcup_{\alpha \in \mathcal{V}_{f_t}} \text{St}^-(\alpha)$.

To eliminate regions below a threshold value of charge density, ρ_t , we restrict our mesh to the sublevel complex of $-f_t$ with respect to f^* . As any regular complex can form the space for discrete Morse functions, we can restrict all subsequent computation to \mathcal{M}_{-f_t} to accelerate computation. For example, integral lines must be traced only from vertices in \mathcal{M}_{-f_t} , and the discrete gradient computed only in that subset of the grid. Specifically, we filter such vertices out in a preprocessing step such that numerical gradients are not computed for them. The total volume of and total charge within the vacuum region are reported to the user at the end of the analysis.

A similar optimization has been implemented by Arnaldsson et al.^[44] For isolated systems with few atoms and a relatively large vacuum region, such an optimization can save substantial (~ 5 to $8\times$) computational effort.

Step 2: Identification of maxima and certain regions. Given $f^* = -\rho$ on a mesh \mathcal{M} , maxima of ρ (minima of f^*) are identified as vertices with an empty lower star with respect to f^* , i.e., vertex α is a maximum (in ρ) if and only if $\text{St}^-(\alpha) = \emptyset$ in f^* . For a regular grid, a maximum vertex has higher values (in ρ) than any of its six neighboring vertices.

The maxima identified in this manner are then used as destinations to terminate numerically integrated gradient lines. As numeric integration is relatively computationally expensive, we identify for each maximum, all the vertices in the grid where *any* strictly ascending path terminates at the maximum. Such regions are called *atomic certain regions*. Atomic certain regions can be computed independently and in parallel, through a simple priority-queue-based region-growing approach. A maximum is inserted into a priority queue ordered by decreasing ρ value. While the queue is not empty, the top element is popped, and inserted into the atomic certain region if and only if all its higher valued edge-connected neighbors are also in the same certain region. When a vertex is inserted into the certain region, all its lower valued neighbors are inserted into the priority queue. This process not only avoids having to compute integral lines from grid points in atomic certain regions, but also expands the regions where numerically computed streamlines can terminate, shortening the integral lines that must be computed from all other grid points.

A similar idea, “atomic trust spheres,” was used by Rodríguez et al.,^[42] who employed spherical Lebedev grids centered at atomic nuclei to find the largest radius in which all gradients point toward the corresponding nuclei, and considered them to be “certain regions” where no gradient paths need to be traced. This approach, however, requires creating a separate grid for each maximum, and a threshold angle is needed to define when a gradient is considered to be pointing “toward” the center. In comparison, *TopoMS* directly uses the input grid without any hard cutoffs.

Step 3: Numerical integration of gradient. For every vertex not part of a vacuum region or an atomic certain region, an integral line is traced from the vertex location using the gradient of f . *TopoMS* uses central differences to approximate the gradient, trilinear interpolation, and an adaptive Euler integrator to trace integral lines. In particular, we solve eq. (1) using straightforward advection and two-step error estimation,

$$\mathbf{p}_{n+1} = \mathbf{p}_n + t \nabla f(\mathbf{p}_n),$$

$$\mathbf{p}'_{n+1/2} = \mathbf{p}_n + \frac{t}{2} \nabla f(\mathbf{p}_n),$$

$$\mathbf{p}'_{n+1} = \mathbf{p}'_{n+1/2} + \frac{t}{2} \nabla f(\mathbf{p}'_{n+1/2}),$$

$$\epsilon \approx \|\mathbf{p}_{n+1} - \mathbf{p}'_{n+1}\|$$

doubling the stepsize t when the error estimate ϵ is below a user supplied threshold, and halving it and re-evaluating \mathbf{p}_{n+1} when it exceeds the threshold. We limit the stepsize in each integration step to obtain a maximum displacement of 1/2 the grid spacing. Gradient values at the vertices of a hexahedron are cached to accelerate re-evaluation of the trilinear gradient. Integration terminates when the point being advected enters an atomic certain region, i.e., when the advected point comes within half grid spacing of any vertex that is part of an atomic certain region. The vertex where the integration originated is labeled as part of the basin of the maximum corresponding to the certain region.

As numeric integration is expensive, even when performed in parallel, we use path compression to initially reduce the number and length of integral lines that must be computed. Specifically, when integration of a line terminates, every vertex adjacent to the path of integration is assigned to the basin of the maximum. Furthermore, integration is cut short and the path labeled, not just when entering certain regions, but also in the neighborhood of any previously labeled vertex. In *TopoMS*, this label map is stored as a flat array in memory and used by parallel threads, where reads and writes to the array avoid synchronization between threads, instead using the atomicity of reading/writing integers to avoid memory corruption. While greatly accelerating an initial labeling of the vertices, this strategy clearly leads to incorrect labeling near the boundaries of basins, due both to parallel read-then-write errors, and also through the path compression itself. A second pass is performed to correct these labels, where every vertex on the boundary between basins is reintegrated from its start

to terminate only in a certain region. This process is repeated on subsequent boundary vertices, until all basin boundary vertices have been corrected.

Step 4: Matching atoms to maxima and basins. Steps 1–3 create a decomposition of the nonvacuum volume with respect to maxima, i.e., each grid vertex is mapped to a unique maximum, which is the destination of its ascent trajectory. When atom positions are known, such as the use-case for *TopoMS*, we match each topological maximum with its closest atom efficiently using a kd-tree,^[56] a multidimensional divide-and-conquer approach. Since maxima are computed directly from ρ , and are independent of the atomic positions (Step 2), any non-nuclear attractors^[42] present in the system are identified by construction, and separated from nuclear attractors (maxima found close to atomic positions in the systems). Finally, volumes and charges of topological atoms are computed using the vertices of the grid labeled as belonging to the associated topological maximum.

Extraction of molecular graph

The molecular graph described in the QTAIM is composed of the arcs of the Morse–Smale complex that connect maxima (NCPs) with 2-saddles (BCPs). While a numerical approach was useful for creating accurate basins, no numerical approach has proved successful in recovering a full topological decomposition that respects invariants such as the Euler characteristic (eq. (2)), and extracting unstable configurations such as 1-saddle–2-saddle connections. A full, consistent description is required for self-consistent analysis, especially when applying topological simplification to account for noise in the data. In *TopoMS*, we recover a molecular graph by building a discrete gradient vector field that conforms to the numerically integrated basins from Section “Computation of atomic basins” i.e., maxima that terminate integral lines in ρ correspond to minima in f^* , and the associated basins correspond to discrete ascending 3-manifolds. Given this conforming discrete gradient, a topological skeleton, basin boundaries, and a volumetric decomposition can all be extracted in a self-consistent manner by simple combinatorial traversal using the discrete flow operator Φ (eq. (3)).

Step 5: Computation of conforming discrete gradient. The map \mathcal{L} from vertices of \mathcal{M} to a set of maxima is extended to a map $\partial\mathcal{L}$ from all cells of \mathcal{M} to $\{0, 1\}$ by considering whether a cell α lies on the boundary between basins of \mathcal{L} . In particular,

$$\partial\mathcal{L}(\alpha) = \begin{cases} 0, & \text{if } |\{\mathcal{L}(v) \text{ s.t. } v \text{ is a vertex of } \alpha\}| = 1 \\ 1, & \text{otherwise.} \end{cases}$$

The boundary map $\partial\mathcal{L}$ is nonzero only for cells having vertices with different labels in \mathcal{L} . We use the algorithm proposed by Gyulassy et al.^[53] In the following discussion, we denote a cell that has been identified as critical by pairing it with itself, e.g., $\langle\alpha, \alpha\rangle$. Furthermore, a cell is *assigned* if and only if it has been identified as critical or paired in a discrete gradient vector. The

function $\#\text{UCF}(\gamma)$, (number of unassigned conforming facets) counts the number of facets $\beta \prec \gamma$ of a cell γ such that $\beta, \gamma \in \text{St}^-(\alpha)$, β has not been assigned, and $\partial\mathcal{L}(\beta) = \partial\mathcal{L}(\gamma)$.

The algorithm processes each vertex independently, first creating a vertex-edge vector in the direction of steepest descent (Lines 3–7), restricted to the set of edges sharing the same label in $\partial\mathcal{L}$ as the vertex. If no pairing for the vertex is possible, it is made critical (Line 5). Next, simple homotopy type expansions are performed in order of increasing dimension (Lines 8–13), again restricting possible candidates for pairing to those sharing the same label in $\partial\mathcal{L}$. For each dimension i , while there exist unassigned i -cells in the lower star of α , simple homotopy expansions of an unassigned i -cell with unassigned $i+1$ cells are attempted (Lines 10–11), marking the i -cell critical (Line 13) when such an expansion is not possible. The test to check if there exist unassigned i -cells in the lower star of α can be implemented by placing the i -cells in $\text{St}^-(\alpha)$ in a list, whose size is typically bounded by a small constant. The output is guaranteed to produce a discrete gradient vector field, since all pairings are restricted to the lower star of a vertex, and a homotopy expansion is performed only when all faces of the i -cell have previously been assigned, and the $i+1$ -cell has only one unassigned face. These two conditions along with the fact that every cell of the domain is either paired or marked critical ensure that all V -paths produced are monotonically decreasing and V is acyclic, and hence a discrete gradient vector field. Algorithm 1 can be applied to every vertex in the domain in an embarrassingly parallel manner, as the lower stars form a partition of \mathcal{M} .

Algorithm 1

```
ConformingGradient ( $\mathcal{M}, f, \partial\mathcal{L}$ )
1:  $V = \{\}$ 
2: for  $\alpha \in \mathcal{V}$  do
3:    $S = \{\beta \in \text{St}^-(\alpha) \mid \partial\mathcal{L}(\alpha) = \partial\mathcal{L}(\beta), \alpha \prec \beta\}$ 
4:   if  $S = \emptyset$  then
5:      $V = V \cup \langle\alpha, \alpha\rangle$ 
6:   else
7:      $V = V \cup \langle\alpha, \beta\rangle$  where  $\beta \in S$  is in direction of steepest descent
8:   for  $i \in [1, \dots, d]$  do
9:     while  $\exists$  unassigned  $\beta^i \in \text{St}^-(\alpha)$  do
10:      while  $S^{i+1} = \{\gamma^{i+1} \in \text{St}^-(\alpha) \mid \#\text{UCF}(\gamma) = 1\} \neq \emptyset$  do
11:         $V = V \cup \langle\beta^i, \gamma^{i+1}\rangle$ ,  $\beta^i$  is the unassigned conforming facet of  $\gamma^{i+1} \in S^{i+1}$ 
12:      if  $\exists$  unassigned  $\beta^i \in \text{St}^-(\alpha)$  then
13:         $V = V \cup \langle\beta^i, \beta^i\rangle$ ,  $\beta^i$  is unassigned
14: return  $V$ 
```

Step 6: Computation of the Morse–Smale complex (MSC). The discrete gradient field computed above can be used to compute a Morse–Smale complex: critical points occur at unpaired cells of the complex, and integral lines are formed by sequences of paired cells according to the discrete flow operator Φ . To extract the topological 1-skeleton of the discrete Morse–Smale complex, critical cells are recorded as nodes, and descending V -paths are traced according to Φ with the V -paths between critical cells recorded as arcs. This traversal is implemented as a depth-first search. Furthermore, the discrete

ascending and descending manifolds of critical cells are recovered by gathering the cells reachable through a breadth-first search using the discrete flow operator.

The data structures and specific algorithm used in *TopoMS* for computing and storing the topological 1-skeleton (nodes and arcs) of the complex are described by Gyulassy et al.^[27,57] This representation allows for interactive exploration and simplification of the structure, as well as interactive reconstruction of the ascending and descending manifolds of each critical point.

Step 7: Computation of persistence hierarchy. A key result from Morse theory is that a pair of critical points attached by a single arc in the Morse–Smale complex can be canceled to obtain a successively coarser representation of a function.^[32] A critical point pair cancellation corresponds to a local smoothing of the function, and provides a mechanism to reason about the impact of local perturbation on the topological structure. For 3D Morse–Smale complexes, it is well understood^[27,58] that cancellations can be used to successively remove low-amplitude noise. The *persistence* of a pair of canceled critical points is the absolute difference in function value between them. For functions sampled onto a grid, sometimes the discretization itself creates spurious critical points; however, the hierarchy of topological reconstructions can be used to remove such small-scale artifacts. *TopoMS* implements the simplification strategy outlined by Gyulassy et al.^[27] to allow interactive exploration of topology at various topological scales. Critical points are canceled by marking the persistence at which a critical point pair and corresponding arcs are removed, and adding new arcs connecting the 1-skeleton at a coarser scale. As a result, obtaining the topological representation for any scale can be done interactively.

Step 8: Extraction of molecular graph geometry. The discrete Morse–Smale complex provides the topological connectivity for consistent and robust extraction of the molecular graphs that agrees with numerically computed basins. However, the arcs connecting critical points follow *V*-paths along a steepest-ascent trajectory and suffer from grid bias. More specifically, the discrete integral line from a 2-saddle to a maximum follows edge-connected vertices of the grid. To obtain a smooth representation of a 2-saddle–maximum path, an integral line is numerically traced from the first vertex of the path to within a half grid cell of the maximum, and then attached to the maximum. In the rare case where the numeric streamline terminates at a different maximum, perhaps due to numeric instability, the numeric path starts from the next vertex of the discrete 2-saddle–maximum path, and so on, until the numerically computed path agrees with the combinatorially computed topological one. These numerically computed paths are used to visualize the atomic interaction lines and the molecular graph.

TopoMS: Software Details

TopoMS v1.0 is released publicly under the BSD License, and can be downloaded from <https://github.com/LLNL/TopoMS>. *TopoMS* has been designed with a simple and extensible API in mind. The software is written in C++ and is designed in a modular fashion such that it can be used with a command-line interface

(cli) or a Qt-based graphical user interface (gui) alongside a viewer. Currently, *TopoMS* supports two input file formats: VASP CHGCAR/AECCAR and Cube; we plan to support other common file formats in the future. *TopoMS* can be executed through a simple configuration file that specifies required details about the input data, the type of analysis to be performed (QTAIM volumes and charges and/or molecular graph), and some user-defined parameters needed for the analysis.

We anticipate that users interested in QTAIM volumes and charges only, in most cases, will need to work only with cli, which outputs the decomposition as a list of atoms with their corresponding volumes and charges, as well as a volume labeling in the same format as the input file. In addition, for easy visualization using standard tools such as “Paraview”^[59] and VisIt,^[60] a binary VTK image file (*.vti) is also written. The visualizations in Figures 2, 3 and 5 were generated using Paraview.

Molecular graph analysis through cli is also possible, and *TopoMS* outputs the molecular graph at the chosen levels of simplification and filtering (through the configuration file) as a VTK polydata file (*.vtp), which can be visualized using Paraview and VisIt. However, the initial exploration phase may require interacting with the complete topology, e.g., to choose an appropriate simplification, for which *TopoMS* can be used in the gui mode. Figures 10 and 11 were generated using *TopoMS* directly.

Dependencies. *TopoMS* is cross-platform and uses shared-memory architecture. It can be installed on any machine with a standard C++ compiler and support for OpenMP.^[61] The gui mode requires Qt^[62] to support the user interface widgets, and QGLViewer^[63] to support the visual interface. The user may optionally choose to install the Visualization Tool Kit (VTK)^[64] to enable corresponding functionality.

TopoMS: Experiments and Results

In this section, we demonstrate the scalability, numerical robustness, and topological consistency of *TopoMS*. First, we discuss our experiments and results for QTAIM volumes and charges, and compare them to the software of Arnaldsson et al.^[4] as a representative of the current state of the art. Since the “ground truth” results, i.e., exact volumes and charges of topological atoms, cannot be known, we instead focus on robustness against lattice bias and convergence to the fine-grid limit to evaluate the two techniques. For a complete topological analysis and extraction of molecular graph, we compare *TopoMS* against Critic2.^[24] In addition to performance scalability, we highlight the topological consistency in our results as well as the functionality to interactively remove topological noise. The experiments presented in this section were performed on a MacBook Pro with a 2.8 GHz Intel i7 processor with 4 cores and 16 GB memory.

QTAIM analysis

We evaluate the performance of *TopoMS* for the QTAIM analysis for six different datasets, as tabulated in Table 2. In all cases, *TopoMS* provides an approximate 2.5–3.9× speed-up.

Table 2. Performance comparison between *TOPOMS* and the software of Arnaldsson et al.^[4] for QTAIM analysis shows that *TOPOMS* takes about 2.5–3.9× less time. In general, the total time for such analysis depends upon the size of the grid, the number of atoms in the system, and the proportion of the vacuum region.

Name	Data			Time (s)		Gain
	# Atoms	Grid size	# pts.	Arnaldsson et al. ^[4]	<i>TOPOMS</i>	
Water	3	[201×201×201]	≈ 8120 K	1.51	0.597	2.5293
Ethylene-a	6	[140×140×140]	2744 K	0.95	0.263	3.6122
Ethylene-b	6	[140×140×140]	2744 K	0.97	0.390	2.4872
Benzene	12	[200×200×200]	8000 K	5.66	1.842	3.0728
NaCl crystal	8	[160×160×160]	4096 K	29.37	7.581	3.8742
Lithium in EC	638	[280×280×280]	21952 K	62.53	18.155	3.4442

Qualitative^[5] and quantitative evaluation on four of these datasets are given below.

Water. The first dataset is the total electron charge density (Cube input file with MP2 Total Density) of a single water (H₂O) molecule defined on a [201×201×201] regular grid. The water molecule is well understood; therefore, we use it as the first experiment to demonstrate the validity of the results of the QTAIM analysis results produced by *TOPOMS*. Figure 2 shows the topological basins corresponding to the atoms in the molecule, highlighting the concavity in the oxygen atom basin. Table 3 provides quantitative results and confirms that the hydrogen atoms lose charge to the oxygen atom. We note, in particular, the symmetry in the results, numerically confirming that the oxygen atom pulls equal charge from the two hydrogen atoms, deforming them by the same amount.[†]

To generate these results, charge density values below 10⁻³ e/Å³ were considered part of the vacuum to isolate the molecule and remove numerical noise. The same cutoff was used by the software of Arnaldsson et al.^[4] (by default), and produced numerical results matching the results shown in Table 3 up to three decimal places (numerical comparison not given for this dataset).

Ethylene. The second dataset is the valence electron charge density (VASP CHGCAR input file) of a single ethylene (C₂H₄) molecule defined on a [140×140×140] regular grid spanning 10 Å on each side. Figure 3 shows the topological basins corresponding to the six atoms in the molecule in different colors computed using *TOPOMS* and the software of Arnaldsson et al.^[4] a visual comparison indicates that the two tools produce almost identical decompositions, with minor discrepancies along the boundaries of the atomic basins. A vacuum threshold of 10⁻³ e/Å³ was used on the charge density values to isolate the molecule from the background region. As the per-atom quantitative comparison in Figure 4a shows, the two tools produce numerically comparable results. However, since the “true” charges contained in the atoms are not known, it is not possible to evaluate the accuracy of the results from either software. Instead, we focus on the stability of the results with respect to data discretization.

[†]For visualization purposes, the bounding surfaces of the atomic basins in Figures 2, 3, and 5 were smoothed, and otherwise are discrete in nature depending on the sampling mesh.

To test the stability of *TOPOMS* with respect to discretization artifacts, we use a second version of the same molecule, rotated differently with respect to the orientation of the mesh (denoted as ethylene-b as compared to ethylene-a discussed above). Comparing Figures 4a and 4b, one can notice a slight dependence of charge assignment on the orientation and position of the molecule with respect to the mesh. The importance of removing (or reducing) the lattice bias has been noted by several researchers.^[40,41] Figure 4c shows that *TOPOMS* produces smaller variability than that by the software of Arnaldsson et al.^[4] between the results for the two orientations.

Lithium salt in ethylene carbonate. Next, we consider the total electron charge density (VASP AECCAR0 + AECCAR2 input file) in a system containing a single molecule of a lithium salt, LiPF₆, in 63 molecules of ethylene carbonate, (CH₂O)₂CO. These 638 atoms are present in a periodic box approximately 19.283 Å on each side, and the charge density is sampled on a [280×280×280] regular grid. This condensed system is significantly

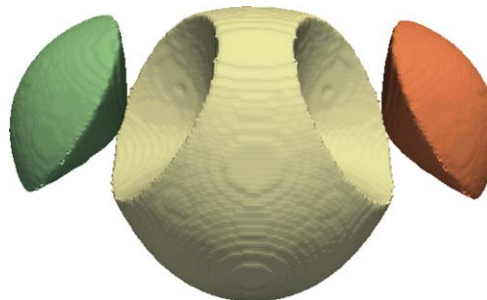


Figure 2. Topological atoms in a water molecule. The volumes corresponding to the two hydrogen atoms are shifted to highlight the concavity in the oxygen atom's basin. [Color figure can be viewed at wileyonlinelibrary.com]

Table 3. Charges and volumes of topological atoms in a water molecule. Charge density lower than 10⁻³ e/Å³ was considered to be vacuum to isolate the molecule and remove numerical noise. The net nonzero charge in the system shows the numerical artifacts of grid-based data.

Atom	QTAIM charge (e)	QTAIM volume (Å ³)
H-1	0.6325736	20.6241691
O	-1.1746194	154.1277976
H-2	0.6325736	20.6241691
Vacuum	-0.0946613	3311.8516406
Total	-0.0041335	3507.2277764

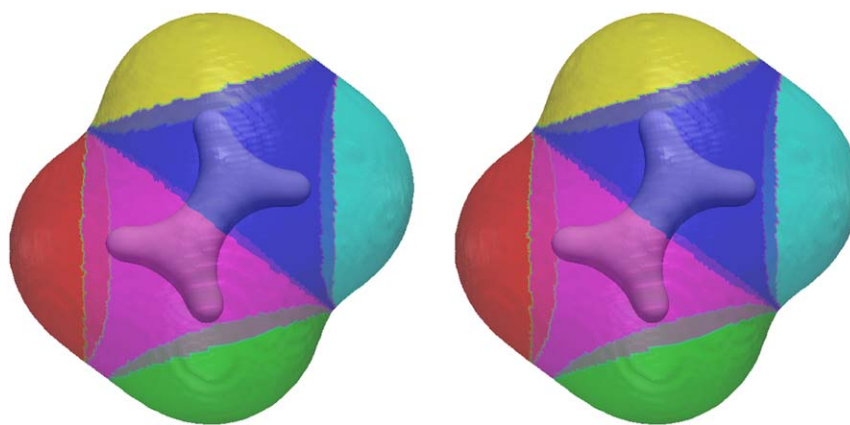


Figure 3. Topological atoms in ethylene molecule computed using the software of Arnaldsson et al.^[4] (left) and *TopoMS* (right). Different atoms are shown in different colors, overlaid on a closed isosurface highlighting the shape of the molecule. [Color figure can be viewed at wileyonlinelibrary.com]

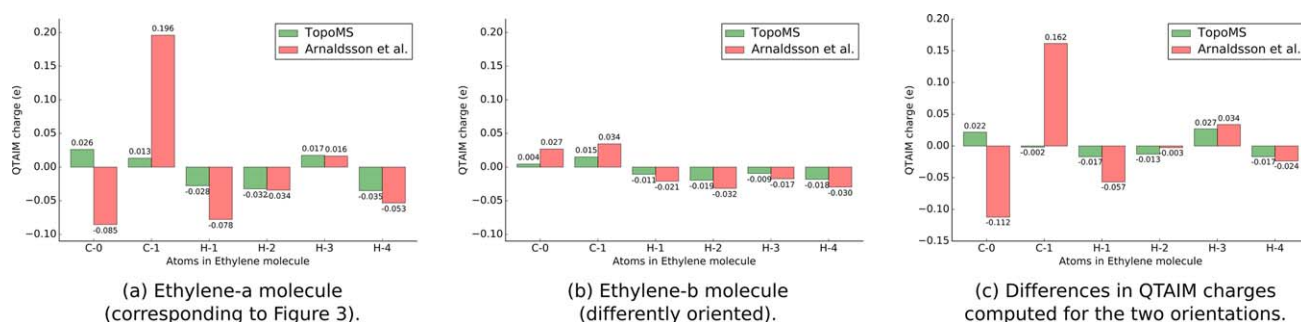


Figure 4. Numerical comparison of QTAIM charges for two differently aligned ethylene molecules is shown in (a) and (b). (c) Shows the difference in charge computed for each atom between the two orientations, with *TopoMS* showing the smaller variability. [Color figure can be viewed at wileyonlinelibrary.com]

more complex compared to the two isolated systems presented above. Figure 5 shows the topological atoms produced by *TopoMS* and the software of Arnaldsson et al.^[4] for a few atoms, which are visually indistinguishable.

Figure 6 presents the QTAIM charges computed for each atom of this system. Figure 6a shows the charge computed through *TopoMS* for different types of atoms. The result shows that the charge distribution remains quite consistent for

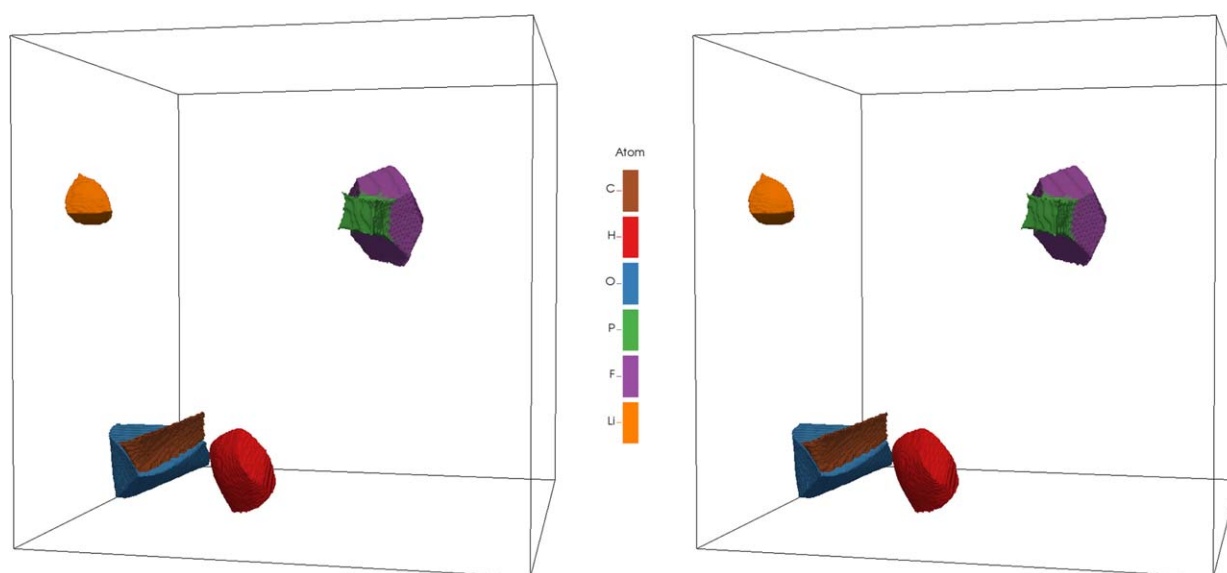


Figure 5. Topological atoms in the lithium salt in ethylene carbonate dataset computed using the software of Arnaldsson et al.^[4] (left) and *TopoMS* (right). To avoid clutter, only a single atom of each type is shown. (a) QTAIM charges computed by *TopoMS*. (b) Differences in the QTAIM charges computed by the two tools (*TopoMS*—Arnaldsson et al.^[4]). [Color figure can be viewed at wileyonlinelibrary.com]

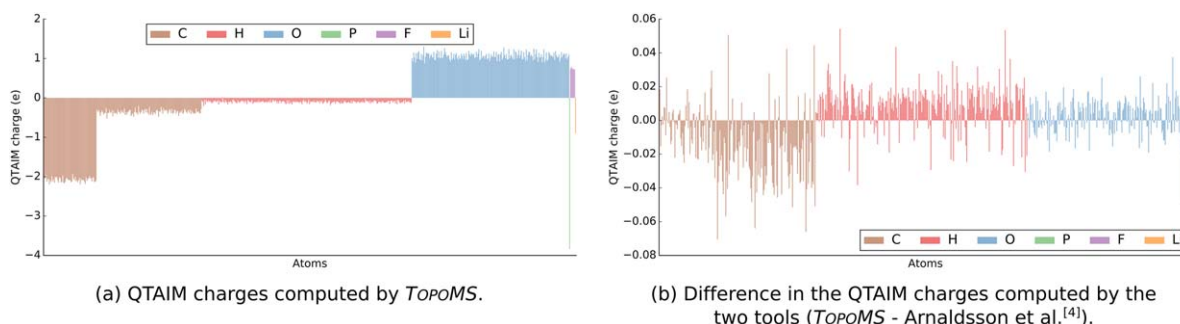


Figure 6. Quantitative evaluation of *TopoMS* shows physically anticipated values of QTAIM charge for each type of atom (a). The first third of the carbon atoms are carbonyl carbons, which are expected to contain less charge than the remaining ether carbons. The figure also shows (b) the differences between the results produced by *TopoMS* and the software of Arnaldsson et al.,^[4] except for the ether carbons and hydrogen atoms, i.e., C—H bonds, the differences between the two tools are relatively small. (a) Differences in QTAIM charges (*TopoMS*—Arnaldsson et al.) at various grid resolutions. (b) Errors in QTAIM charges with respect to corresponding fine-grid results. [Color figure can be viewed at wileyonlinelibrary.com]

different atoms of each type, confirming that the atoms in the EC molecules display similar behavior throughout the domain. Also note that the average QTAIM charge of carbonyl carbon atoms (first one-third of the brown bars) is about -2.08 electrons each, whereas that of the ether carbon atoms (remaining two-thirds) is about -0.36 electrons each, a result consistent with chemical intuition about the electropositivity of carbon in C=O polar bonds. Figure 6b plots the differences between the QTAIM charges computed using *TopoMS* and the software of Arnaldsson et al.^[4] For most atoms, the differences are small, roughly bounded within 0.02 electrons, except for the hydrogen atoms, which show the largest difference of up to 5% relative to the number of its valence electrons, and ether carbon atoms bounded by about 2% relative difference. In general, it appears that the results of the two tools mostly differ in assigning charges for the C—H bonds, where comparatively, *TopoMS* assigns more charge to H (and less to ether C).

Despite the existence of these noticeable differences, in the absence of ground truth, the accuracy of either of the tools is not possible to evaluate. Nevertheless, it can be verified (see Figure 7a) that the differences between the results of the two tools reduce substantially as the mesh resolution is improved, suggesting that both tools converge to a common (although unknown) limit. To evaluate the convergence of the two tools, we use the analysis results of a highly refined mesh ([1120×1120×1120]

compared to a practically suggested resolution [280×280×280] for this data), and study the convergence of the two tools with respect to their respective fine-grid solutions. As shown in Figure 7b, the mean, the maximum, and the standard deviation of the relative errors (with respect to the fine-grid solution) produced by the two tools show rapid convergence to zero. For resolutions lower than [210×210×210], both tools create comparable errors, suggesting that such coarse sampling is lossy, but as the resolution improves, *TopoMS* converges at a slightly faster rate.

Figure 8 shows plots of per-atom errors for different resolutions. From top (resolution 120) to bottom (resolution 560), the scale of error reduces by about 100 times from 3 electrons to about 0.03 electrons. As the analysis is moved to higher resolutions, one starts to notice that due to the overall reduction in the scale of errors, the errors in hydrogen charges appear more pronounced.

Furthermore, notice the positive and negative trends in the errors for ether carbon atoms (last two-thirds of the brown bars) and hydrogen atoms, respectively, as computed by the software of Arnaldsson et al. (left column). This behavior indicates that their approach systematically over- and underestimates these charges, respectively, with respect to their fine-grid solutions. This observation is consistent with the remark made earlier in the context of Figure 6b, and indicates that the results of *TopoMS* produced for the [280×280×280] grid could be considered

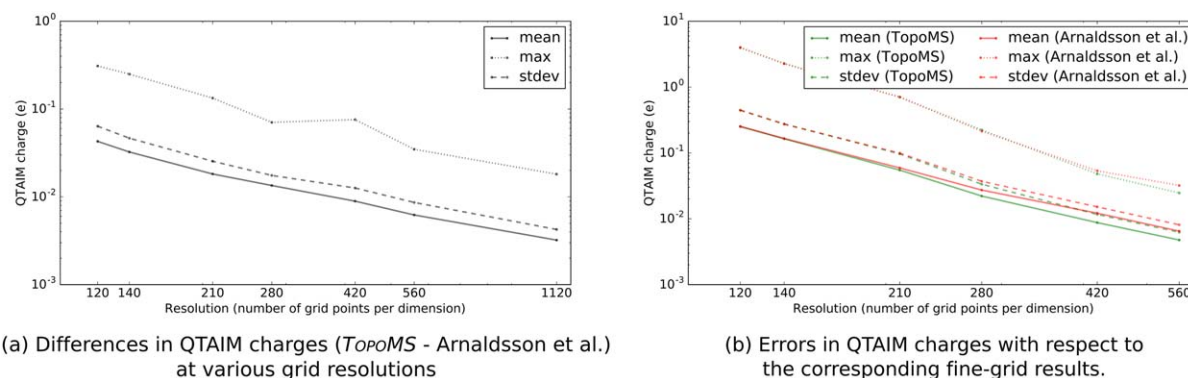


Figure 7. With increasing grid resolution, both tools are expected to produce increasingly accurate results, and ultimately converge to the “correct” solution. (a) Shows the differences between the results produced by the two tools, and confirms that the differences reduce for finer meshes. (b) Shows that both tools converge to their respective fine-grid resolution solutions with *TopoMS* showing faster convergence. [Color figure can be viewed at wileyonlinelibrary.com]

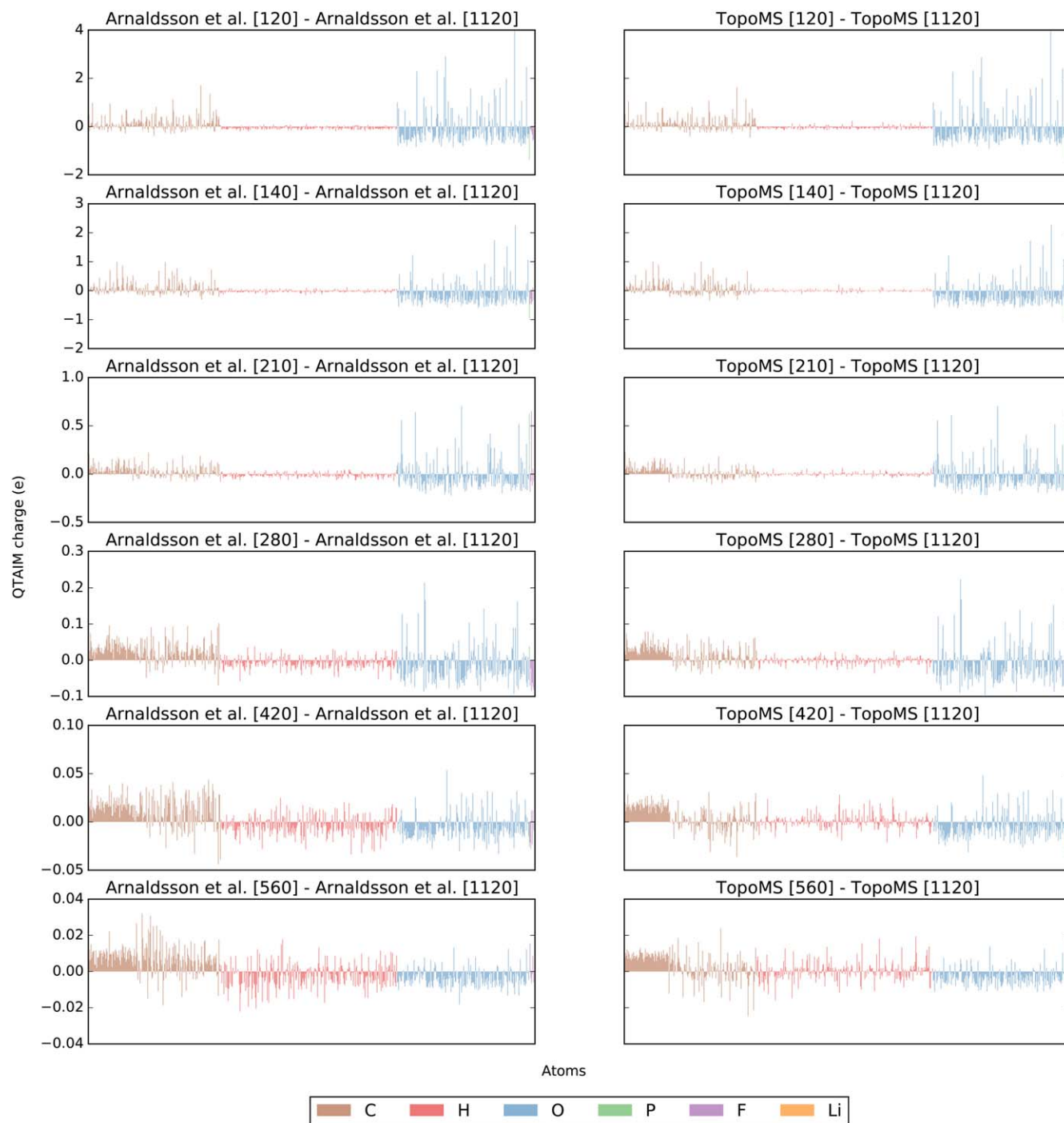


Figure 8. With increasing grid resolution, both tools are expected to produce increasingly accurate results, and ultimately converge to the “correct” solution. Different rows in the figure show the differences between the results for each atom (laid out on the horizontal axes) produced by the two tools with respect to their respective fine-grid solutions ($1120 \times 1120 \times 1120$). The plots confirm that the differences reduce for finer meshes; about $100\times$ reduction is obtained from top to bottom. The errors in hydrogen atoms become more pronounced as the overall scale of the errors becomes smaller. It is also noted that the results in the left column typically overestimate ether carbons (last two-thirds of the brown bars) and underestimate hydrogen atoms, suggesting that the software of Arnaldsson et al. assigns more charge to carbons in C–H bonds and less to hydrogen atoms. Other trends, such as overestimations in carbonyl carbon atoms and underestimations in oxygen atoms, are also observed; however, comparatively, *TopoMS* performs slightly better than the competing software for almost all resolutions. [Color figure can be viewed at wileyonlinelibrary.com]

more accurate. We also notice that both tools consistently overestimate the charge in carbonyl carbons (first one-third) whereas underestimation is observed in most of the oxygen atoms.

Finally, Table 4 shows the performance for the experiments performed for different resolutions, and that the performance gain of *TopoMS* scales with the grid size and remains consistent at about $2.5\text{--}3.5\times$.

Table 4. Scaling performance comparison between *TopoMS* and the software of Arnaldsson et al.^[4] for QTAIM analysis with increasing grid size shows *TopoMS* improving with grid sizes.

Data	Grid	# pts.	Time (s)		
			Arnaldsson et al. ^[4]	<i>TopoMS</i>	Gain
	[120×120×120]	1728 K	4.13	1.535	2.6906
	[140×140×140]	2744 K	6.59	2.292	2.8752
	[210×210×210]	9261 K	23.47	7.398	3.1725
	[280×280×280] ^[a]	21952 K	62.53	18.155	3.4442
	[420×420×420]	74088 K	209.46	60.402	3.4678
	[560×560×560]	175616 K	506.06	141.163	3.5849

[a] The results for the [280×280×280] mesh are repeated from Table 2.

Extraction of molecular graph

We demonstrate the extraction of molecular graph on two datasets and compare our findings to the results of Critic2.^[24] In both cases, we focus not only on the performance gain, but more importantly, also on the topological consistency and ability to simplify numerical and topological noise.

Benzene. We explore the total charge density (VASP AEC-CAR0 + AEC-CAR2 input file) for an isolated benzene (C_6H_6) molecule, sampled on a [200×200×200] grid. Benzene is a symmetrical molecule, expected to demonstrate isotropic behavior about its center. As a result, it provides a predictable test case to evaluate the stability of the molecular graph extraction with respect to the underlying grid.

Figure 9 shows the molecular graph computed using Critic2. The atomic locations are highlighted by red and blue spheres, whereas orange dots represent the points along the gradient paths in the graph, as are output by Critic2. Although the general shape of the molecule can be observed even in the left image, the visualization suffers from clutter due to the presence of numerical and topological noise. Notice that there exist many

paths that connect the molecules across the periodic domain, all of which are artifacts of discretization and numerical noise; indeed, these are physically incorrect due to the nature of the molecule, and much of the observed noise could be discarded as vacuum to isolate the molecule. In their current form, these results offer significant challenges for analysis due to a lack of the notion of physical significance of these features. Due to the noninteractive nature of the tool, cleaning up this noise by using appropriate thresholds requires not only prior knowledge about the molecule, but also manual effort through trial and error. Figure 9 (right) was generated by discarding critical points with charge density value less than 10^{-6} . Note that whereas the resulting graph still appears to retain spurious paths across the periodic boundary, the bond paths between all pairs of adjacent carbon atoms have been lost. In fact, these physically important features are missing even from the original noisy result, without any simplification.

These results lead to the final concern about topological consistency, which is highly dependent upon noise. In the original case (left image: without noise simplification), Critic2 detects a total of 865 critical points (95 NCPs, 325 BCPs, 323 RCPs, and 122 CCPs); the resulting Morse sum is -29 , which not only produces incomplete topology,^[24] but also is fundamentally inconsistent as it violates the Poincaré–Hopf invariant. Upon simplification (right image), however, Critic2 produces topologically consistent result: 35 critical points (12 NCPs, 15 BCPs, 6 RCPs, and 2 CCPs), with a valid Morse sum of $+1$.

In comparison, irrespective of the data and any inherent noise, *TopoMS* always produces a topologically consistent graph, and it is guaranteed to compute a super-set of physically relevant critical points. In addition to the “true” critical points, a number of less persistent critical points may be identified, present in the data mostly as numerical and topological noise. Figure 10 shows the original, noisy molecular graph computed by *TopoMS*. Most of the noise is concentrated away

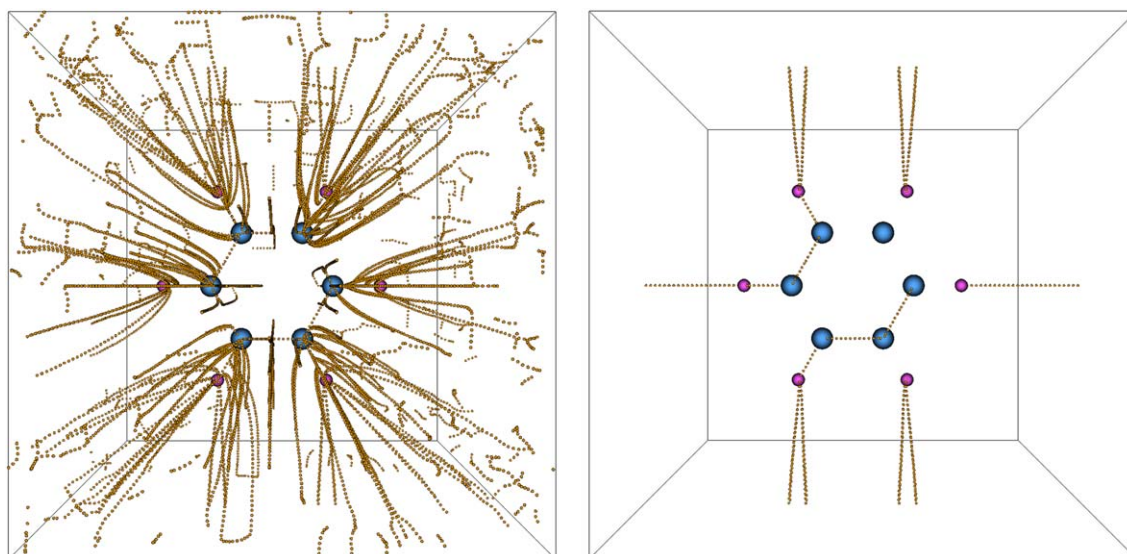


Figure 9. Molecular graph of the total electron charge density of the benzene molecule extracted using Critic2^[23] and visualized using Paraview.^[59] Upon simplification from the original noisy graph (left) to a simplified graph (right), many noisy features are removed. Nevertheless, three C–C bonds are not captured by Critic2, even in the noisy case. Carbon and hydrogen atoms are shown as blue and pink spheres, respectively; bond paths in the graph are rendered as sequences of orange spheres. [Color figure can be viewed at wileyonlinelibrary.com]

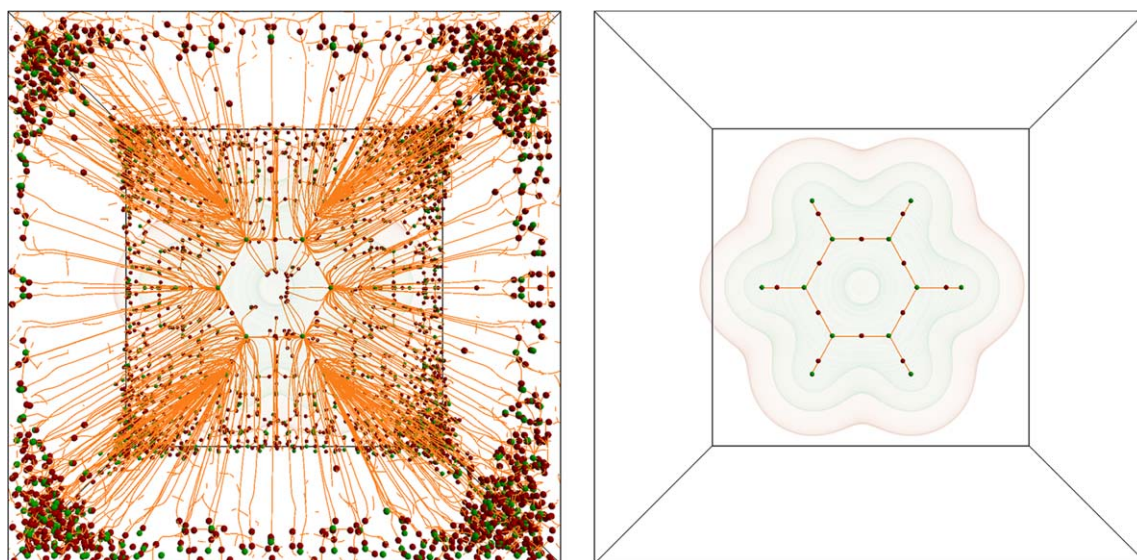


Figure 10. Molecular graph of the total electron charge density of the benzene molecule extracted and visualized using *TopoMS*. The original graph (left) contains many noisy critical points, especially toward the corners of the domain where the gradient is very small. However, interactive simplification and filtering of noisy features using *TopoMS* allows focusing only on the most important features (right) to capture its hexagonal shape and describe its bonding structure. NCPs (maxima) and BCPs (2-saddles) are shown as green and red spheres, respectively, and the connections between them are shown as orange lines. The topology is overlaid on the volume rendering of the density. [Color figure can be viewed at wileyonlinelibrary.com]

from the molecule where the function values are low, thus reducing the signal-to-noise ratio substantially. A total of 13418 (discrete) critical points (1255 NCPs, 5060 BCPs, 5454 RCPs, and 1649 CCPs) were identified in this case; despite the high number of noisy critical points, the results stay topologically consistent, with a valid Morse sum of 0. (Since no threshold is applied, this data is considered as a nonisolated system, whose Euler characteristic is 0.) The simplified molecular graph in Figure 10 clearly shows the hexagonal shape of the benzene molecule with oxygen atoms (not shown) present at the six corners, and outward arcs extended toward hydrogen

atoms (not shown), and captures the correct bonding behavior of the benzene molecule. All numerical noise was removed by interactively choosing appropriate levels of simplification and filtering through the gui version of *TopoMS*. Since each simplification step is performed in a topologically consistent manner (by canceling pairs of critical points of adjacent indices), by induction, every simplified molecular graph is topologically consistent.

To generate the complete and noisy graph, Critic2 took about 15 seconds compared to the 19.3 seconds taken by *TopoMS* to compute the topology of this dataset (for a large range of noise

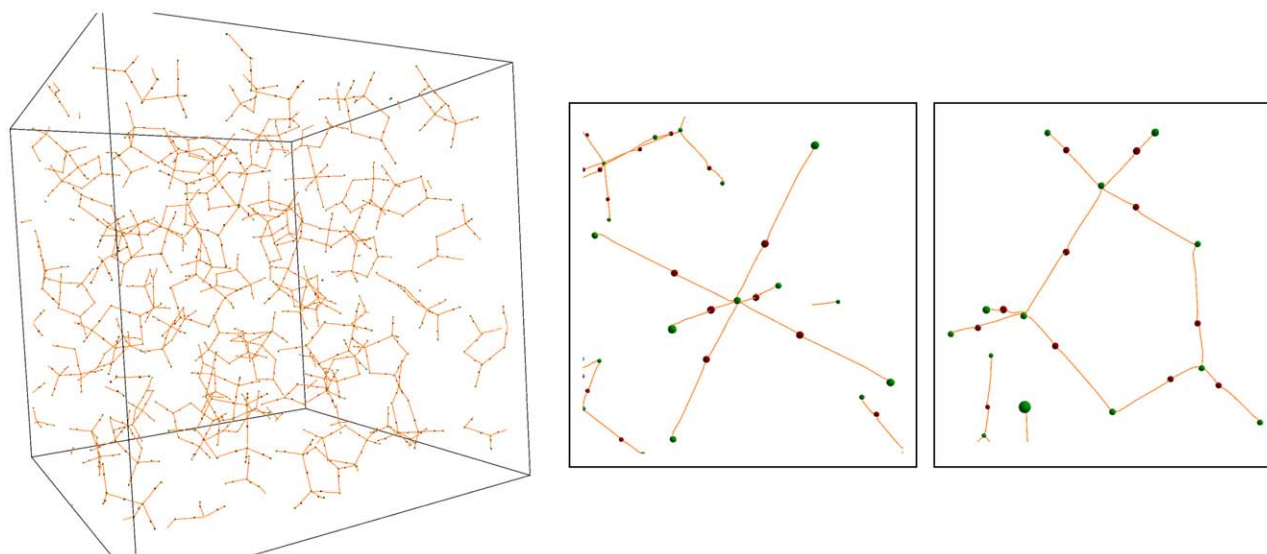


Figure 11. Molecular graph of the total electron charge density for the lithium salt in ethylene carbonate dataset. Appropriate simplification and filtering through the UI allows removing the numerical and topological noise and enables capturing important bonding structures, for example, the PF_6^- ion and the EC molecules, also shown as insets. NCP (maxima) and BCP (2-saddles) are shown as green and red spheres, respectively; the lines connecting them (bond paths), are shown in orange. [Color figure can be viewed at wileyonlinelibrary.com]

simplification), after which the user can interactively simplify the noise to a desired level. We note that to highlight the artifacts due to noise, this data was treated artificially as a nonisolated system (without a vacuum threshold). Therefore, the discrete representation in *TopoMS* identified a larger number of critical points in a degenerate region of near-zero gradient as compared to many fewer numerical critical points in Critic2. Such behavior leads to a more dense discrete representation, which takes longer to construct and even longer to simplify. Therefore, Critic2, which is entirely numerical in nature, performs better. When correctly run as an isolated system, *TopoMS* took 12.4 seconds only, compared to Critic2's 14 seconds.

Lithium salt in ethylene carbonate. We revisit this dataset for a detailed topological exploration, which produces the molecular graph shown in Figure 11. As is shown in the figure, the resulting molecular graph clearly captures the bonded atoms, e.g., in the EC molecule and PF_6^- ion. This detailed topological analysis provides a parameter-free approach to identify bonding structures without the need to impose hard thresholds for distance-based bond detection using pair-correlation functions. As a result, *TopoMS* provides a robust and scalable tool to determine atomic bonds using a well-founded and widely accepted mathematical framework. The computation of the complete topological graph through *TopoMS*, and its simplification and filtering to an appropriate level take about 55.4 seconds, including the time for QTAIM analysis. In comparison, Critic2^[24] requires about 223 seconds to process the data for a similar analysis.

Conclusion

QTAIM analysis is an important tool for exploring complex molecular and condensed-matter systems, particularly through understanding bonding and charge transfer. Although there exist publicly available tools for this task, the scope of such tools is limited. For example, the software of Arnaldsson et al.,^[4] which combines several sophisticated numerical algorithms to compute volumes and charges of topological atoms, has proved very successful in the task it is designed for. However, it does not provide the molecular graph. An open-source alternate to extract the molecular graph is Critic2,^[24] which offers attractive features for more general analytic purposes. Nevertheless, computation of detailed topological structures is a difficult task, presenting challenges in both performance and numerical stability. Such approaches, in general, are particularly affected by noise and degeneracies in the data.

Leveraging recent advances in computational topology, and in the computation of Morse–Smale complexes in particular, *TopoMS* provides a means to perform a comprehensive QTAIM analysis, including the extraction of the complete molecular graph. *TopoMS* combines the best of numerical integration algorithms with robust discrete representations to provide a scalable, numerically robust, and topologically consistent analysis. This paper demonstrates that *TopoMS* provides up to 4× performance gain relative to current state-of-the-art codes, and the ability to interactively explore the noise created by sampling artifacts and extract only the important and physically relevant features.

In the future, we plan to extend the functionality of *TopoMS* in several ways, including support for different file formats. We would also like to demonstrate interfacing *TopoMS* with VASP, to allow in situ analysis of large molecular and condensed-matter systems. An important direction will be to extend *TopoMS*'s API to include other relevant fields, such as electron localization function and molecular electrostatic potential, and address the more general field of Quantum Chemical Topology.^[22]

Acknowledgements

The authors would like to thank Mitchell Ong and Tuan Anh Pham for helping with generating data for experiments. The data for water and ethylene molecules was obtained online, courtesy of Graeme Henkelman. We also thank the groups of Henkelman et al. and Otera-de-la Roza et al. for making their codes, Bader charge analysis v1.0 and Critic2, respectively, publicly available. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344.

Keywords: atoms in molecules · molecular graph · topological analysis · Morse–Smale complex

How to cite this article: H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, P.-T. Bremer. *J. Comput. Chem.* **2018**, *39*, 936–952. DOI: 10.1002/jcc.25181

- [1] R. F. W. Bader, *Acc. Chem. Res.* **1985**, *18*, 9.
- [2] R. F. W. Bader, M. T. Carroll, J. R. Cheeseman, C. Chang, *J. Am. Chem. Soc.* **1987**, *109*, 7968.
- [3] R. F. W. Bader, *Atoms in Molecules—A Quantum Theory*; Oxford University Press: Oxford, U.K., **1994**.
- [4] A. Arnaldsson, W. Tang, S. Chill, W. Chai, G. Henkelman, *Bader Charge Analysis v1.0*, **2017** [cited 01/11/17]. URL <http://theory.cm.utexas.edu/bader/>
- [5] G. Henkelman, A. Arnaldsson, H. Jónsson, *Comput. Mater. Sci.* **2006**, *36*, 354.
- [6] P. L. A. Popelier, *Comput. Phys. Commun.* **1996**, *93*, 212.
- [7] B. B. Stefanov, J. Cioslowski, *J. Comput. Chem.* **1995**, *16*, 1394.
- [8] F. De Proft, C. Van Alsenoy, A. Peeters, W. Langenaeker, P. Geerlings, *J. Comput. Chem.* **2002**, *23*, 1198.
- [9] P. L. A. Popelier, *Theor. Chim. Acta* **1994**, *87*, 465.
- [10] P. L. A. Popelier, *Comput. Phys. Commun.* **1998**, *108*, 180.
- [11] F. Biegler-König, J. Schönbohm, D. Bayles, Aim2000, *Journal of Computational Chemistry* **2001**, *22* 545.
- [12] P. L. A. Popelier, *Theor. Chem. Acc.* **2001**, *105*, 393.
- [13] T. A. Keith, AIMAll (Version 17.01.25); TK Gristmill Software, **2017**. URL <http://aim.tkgristmill.com/>
- [14] Amsterdam Density Functional (ADF), URL <https://www.scm.com/product/adf/>
- [15] R. F. W. Bader, *J. Phys. Chem. A* **1998**, *102*, 7314.
- [16] P. T. Bremer, G. Weber, V. Pascucci, M. Day, J. Bell, *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 248.
- [17] P. T. Bremer, A. Gruber, J. C. Bennett, A. Gyulassy, H. Kolla, J. H. Chen, R. W. Grout, *Commun. Appl. Math. Comput. Sci.* **2016**, *11*, 37.
- [18] D. Laney, P. T. Bremer, A. Mascarenhas, P. Miller, V. Pascucci, *IEEE Trans. Vis. Comput. Graph.* **2006**, *12*, 1053.
- [19] T. Sousbie, *Monthly Notices R. Astron. Soc.* **2011**, *414*, 350.
- [20] A. Gyulassy, A. Knoll, K. C. Lau, B. Wang, P.-T. Bremer, M. E. Papka, L. A. Curtiss, V. Pascucci, In *Topological Methods in Data Analysis and Visualization IV*; H. Carr, C. Garth, T. Weinkauff, Eds.; Springer International Publishing, Gewerbestrasse 11, 6330 Springer, Cham: Switzerland, **2017**; pp. 135–149.
- [21] D. Günther, R. A. Boto, J. Contreras-García, J. P. Piquemal, J. Tierny, *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 2476.

- [22] P. L. A. Popelier, In *The Chemical Bond II: 100 Years Old and Getting Stronger*; D.M.P. Mingos, Ed.; Springer, Cham, **2016**, pp. 71–117.
- [23] A. Otero-de-la Roza, M. A. Blanco, A. M. Pendás, V. Luaña, *Comput. Phys. Commun.* **2009**, *180*, 157.
- [24] A. Otero-de-la Roza, E. R. Johnson, V. Luaña, *Comput. Phys. Commun.* **2014**, *185*, 1007.
- [25] Y. Matsumoto, An Introduction to Morse Theory (Translations of Mathematical Monographs, Vol. 208), **2002**. American Mathematical Society, Providence, Rhode Island, USA.
- [26] H. Edelsbrunner, J. Harer, A. Zomorodian, *Discrete Comput. Geometry* **2003**, *30*, 87.
- [27] A. Gyulassy, P. T. Bremer, B. Hamann, V. Pascucci, *IEEE Transactions on Visualization and Computer Graphics* **2008**, *14*, 1619.
- [28] A. Gyulassy, P. T. Bremer, V. Pascucci, *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 2014.
- [29] V. Robins, P. J. Wood, A. P. Sheppard, *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1646.
- [30] N. Shivashankar, S. M. V. Natarajan, *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1757.
- [31] H. Edelsbrunner, D. Letscher, A. Zomorodian, *Discrete Comput. Geometry* **2002**, *28*, 511.
- [32] N. O. J. Malcolm, P. L. A. Popelier, *J. Comput. Chem.* **2003**, *24*, 437.
- [33] R. Forman, *Sémin. Lotharingien Combinatoire* [electronic only] **2002**, *48*, B48c, 35.
- [34] H. Edelsbrunner, E. P. Mücke, *ACM Trans. Graph.* **1990**, *9*, 66.
- [35] J. Cioslowski, B. B. Stefanov, *Mol. Phys.* **1995**, *84*, 707.
- [36] P. L. A. Popelier, *Mol. Phys.* **1996**, *87*, 1169.
- [37] Y. Aray, J. Rodríguez, J. Rivero, *J. Phys. Chem. A* **1997**, *101*, 6976.
- [38] S. Noury, X. Krokidis, F. Fuster, B. Silvi, *Comput. Chem.* **1999**, *23*, 597.
- [39] C. Katan, P. Rabiller, C. Lecomte, M. Guezo, V. Oison, M. Souhassou, *J. Appl. Crystallogr.* **2003**, *36*, 65.
- [40] E. Sanville, S. D. Kenny, R. Smith, G. Henkelman, *J. Comput. Chem.* **2007**, *28*, 899.
- [41] W. Tang, E. Sanville, G. Henkelman, *J. Phys.: Condens. Matter* **2009**, *21*, 084204.
- [42] J. I. Rodríguez, A. M. Köster, P. W. Ayers, A. Santos-Valle, A. Vela, G. Merino, *J. Comput. Chem.* **2009**, *30*, 1082.
- [43] J. I. Rodríguez, R. F. Bader, P. W. Ayers, C. Michel, A. W. Götz, C. Bo, *Chem. Phys. Lett.* **2009**, *472*, 149.
- [44] M. Yu, D. R. Trinkle, *J. Chem. Phys.* **2011**, *134*, 064111.
- [45] L. Palatinus, S. J. Prathapa, S. van Smaalen, *J. Appl. Crystallogr.* **2012**, *45*, 575.
- [46] J. I. Rodríguez, *J. Comput. Chem.* **2013**, *34*, 681.
- [47] R. Hernández-Esparza, S.-M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, J.-M. Hernández-Pérez, R. Vargas, J. Garza, *J. Comput. Chem.* **2014**, *35*, 2272.
- [48] F. W. Biegler-König, T. T. Nguyen-Dang, Y. Tal, R. F. W. Bader, A. J. Duke, *J. Phys. B: Atom. Mol. Phys.* **1981**, *14*, 2739.
- [49] F. W. Biegler-König, R. F. W. Bader, T.-H. Tang, *J. Comput. Chem.* **1982**, *3*, 317.
- [50] W. H. Press, B. P. Flannery, S. A. Teukolsky, T. Vetterling, *Numerical Recipes in Fortran*; Cambridge University Press: Cambridge, UK, **1992**.
- [51] Y. Aray, J. Rodríguez, R. López-Boada, *J. Phys. Chem. A* **1997**, *101*, 2178.
- [52] Y. Aray, J. Rodríguez, D. Vega, *J. Phys. Chem. B* **2000**, *104*, 5225.
- [53] A. Gyulassy, D. Günther, J. A. Levine, J. Tierny, V. Pascucci, *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 2595.
- [54] VASP: Vienna Ab Initio Simulation Package, URL <https://www.vasp.at/>
- [55] P. L. A. Popelier, *Curr. Top. Med. Chem.* **2012**, *12*, 1924.
- [56] J. L. Bentley, *Commun. ACM* **1980**, *23*, 214.
- [57] A. Gyulassy, N. Kotava, M. Kim, C. D. Hansen, H. Hagen, V. Pascucci, *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 1549.
- [58] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, B. Hamann, *IEEE Trans. Comput. Graph. Vis.* **2006**, *12*, 474.
- [59] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*; Kitware, Inc.: USA, **2015**.
- [60] VisIt, URL <https://visit.llnl.gov>
- [61] OpenMP, URL <http://www.openmp.org/>
- [62] Qt, URL <https://www.qt.io/>
- [63] libQGLViewer, URL <http://libqglviewer.com/>
- [64] W. Schroeder, K. Martin, B. Lorensen, *The Visualization Toolkit*; Kitware, Inc.: USA, **2006**.

Received: 26 July 2017
Revised: 21 December 2017
Accepted: 13 January 2018
Published online on 23 March 2018