Topological analysis of functions on arbitrary grids: Applications to quantum chemistry

Michael J. Hutcheon and Andrew M. Teale
School of chemistry, University of Nottingham, University Park, Nottingham NG7 2RD, UK
(Dated: June 14, 2022)

Algorithms are presented for performing topological analysis of functions defined on grids of points. By connecting these points according to a Delaunay triangulation, a neighbourhood graph is constructed that allows the topological analysis to be recast as a problem in graph theory. This allows for the treatment of arbitrary grids, including those employed in standard density functional theory (DFT) calculations. The flexibility of the approach is demonstrated for various applications involving analysis of the charge and magnetically induced current densities in molecules, where features of the neighbourhood graph are found to correspond to chemically relevant topographical properties, such as Bader charges. These properties converge using an order of magnitude fewer grid points than previous approaches, whilst exhibiting an appealing $O(N \log(N))$ scaling of the computational cost. The issue of grid bias is discussed in the context of graph based algorithms and strategies for avoiding this bias are presented. Python implementations of the algorithms are provided.

I. INTRODUCTION

First-principles quantum mechanical calculations have been widely successful in describing chemical processes at a fundamental level. However, the interpretability of these calculations is still an ongoing subject of debate. How does one move between the electrons and nuclei of first-principles calculations to the more intuitive building blocks of chemistry, such as atoms, bonds, lone-pairs etc.? Significant progress has been made in incorporating chemical ideas into first-principles calculations. However, it is also important to be able to move in the other direction - to be able to incorporate chemical ideas into first-principles calculations. Ideally, one would be able to set up a feedback loop whereby chemically-intuitive quantities can be calculated from first-principles and fed back into the calculation to improve the results. This work investigates one route to achieving this for density functional theory (DFT) calculations, by providing a method to calculate topological properties of functions defined on a real-space integration grid. This is achieved by the construction of a neighbourhood graph over the DFT gridpoints and it is demonstrated that intrinsic properties of the graph, such as maximal spanning trees and strongly-connected subgraphs, correspond to chemically-relevant properties. Having such topological information available on a per-gridpoint basis allows for its direct incorporation into DFT calculations.

II. TOPOLOGICAL ANALYSIS ON ARBITRARY GRIDS

A. Terminology

It is important to clarify that in what follows, and in the field of QCT more broadly, the term “topology” is used in a looser sense (with some exceptions - see §17) than in the branch of mathematics bearing the same name. We use the term in it's broader sense as pertaining to properties of a geometric object (in our case, a quantum-mechanical function) that are preserved under continuous deformations (in our case, small deformations of a molecule). In this work, the topological properties of interest will be properties of the topography of the quantum-mechanical function of interest. For example, maxima, minima and saddle points are topographical features, but their existence and connectivity are topological properties. These topological properties are insensitive to the level of theory used to describe a molecule (e.g. Hartree-Fock or DFT). However, in contrast to stricter definitions of conservation in mathematics, topological properties in QCT are typically not conserved through
chemical processes, such as bond breaking or formation - a fact which underpins their usefulness in identifying and classifying such phenomena.

B. Grids

In numerical studies, it is common to represent a function \( f : \mathbb{R}^N \rightarrow \mathbb{R} \) by its values defined on a grid \( G \) of points in \( \mathbb{R}^N \):

\[
F = \{ f(x) : x \in G \subset \mathbb{R}^N \}. \tag{1}
\]

If the grid is constructed in a suitable fashion, it is possible to preserve information about the function in the neighbourhood of a particular point. For example, if \( G \) is a uniform grid with spacing \( s \),

\[
G = \{(n_1s, n_2s, \ldots, nNs) : n_i \in \mathbb{Z} \}, \tag{2}
\]

then we can define the neighbours of a particular grid point straightforwardly as

\[
N(r = (n_1s, n_2s, \ldots, nNs)) = \{(m_1s, m_2s, \ldots, mNs) : |m_i - n_i| \in \{0, 1\} \}. \tag{3}
\]

We can even go on to approximate the derivatives of \( f \) using, for example, finite differences,

\[
\frac{\partial f}{\partial x_1} (n_1s, \ldots) \approx s^{-1}[f((n_1 + 1)s, \ldots) - f(n_1s, \ldots)] \tag{4}
\]

assuming the spacing \( s \) is small enough to resolve variations in \( f \) accurately.

Despite the simplicity of a uniform grid, it is common to generate \( G \) in a less trivial fashion to reduce the storage requirements and the computational cost of operations on \( F \). For example, in order to make routine DFT calculations feasible, typical grids used to perform real space integration become less dense further from atomic nuclei, where the electron density is lower and quantum-mechanical functions vary more slowly. In this work, we will investigate how to recover topological properties of functions defined on such non-uniform grids.

C. Graphs over grids

Determining the neighbours of a given grid point, as was done in Eq. (3) for a uniform grid, is a necessary prerequisite to performing a topographical analysis. Even simple topographical objects, such as local maxima and minima, are defined with reference to the behaviour of the function when moving to “nearby” points. It is possible to encode the necessary information about the neighbours of a given grid point in the edges of a neighbourhood graph \( N \) with nodes given by the points in \( G \), and edges connecting each node \( x \in G \) to a set of neighbours \( N(x) \subset G \setminus x \). In this section, the construction of such graphs is investigated.

1. Choice of graph construction

There are many ways to construct a neighbourhood graph \( N \) for an arbitrary set of points \( G \) (a few are shown in Fig. 1). In practice, \( G \) will be limited to a finite region of \( \mathbb{R}^N \) and we will not be primarily concerned with the boundary of points forming the convex hull \( H(G) \), but only the bulk \( B(G) = G \setminus H(G) \). The goal when choosing a construction is to most closely preserve the topography of \( f \) (and topology thereof) when moving from its representation in \( \mathbb{R}^N \) to its representation on \( G \). This leads to enforcing the following requirements for \( N \):

1. connected: \( N \) should be connected (any node can be reached from any other node via a path along edges).

2. undirected: \( y \in N(x) \implies x \in N(y) \).

3. basis-preserving: Given \( x \in B(G) \), the vectors \( \{ y - x : y \in N(x) \} \) must form a basis of \( \mathbb{R}^N \).

4. move-preserving: Given a point \( x \in B(G) \) and an arbitrary direction \( \delta \in \mathbb{R}^N \), a move can be made to a neighbour \( y \in N(x) \) so that the projection of the move onto \( \delta \) is positive. In short, \( \forall \delta \in \mathbb{R}^N, x \in B(G) \exists y \in N(x) : \delta \cdot (y - x) > 0 \).

Condition 3 ensures the existence of an approximate gradient \( g(x) \approx \nabla f(x) \) on the graph via finite differences, by minimizing the residual norm \( \sum_{y \in N(x)} |y|_2^2 \) of a first-order Taylor expansion (see Fig. 2) for an example:

\[
f(y \in N(x)) = f(x) + (y - x)^T g(x) + \epsilon_y \tag{5}
\]

leading to

\[
g(x) = M^{-1}b \tag{6}
\]

where

\[
M = \sum_{y \in N(x)} (y - x)(y - x)^T \tag{7}
\]

\[
b = \sum_{y \in N(x)} |f(y) - f(x)|(y - x) \tag{8}
\]

Which would not have a unique solution (\( M \) would be singular) if \( \{ y - x : y \in N(x) \} \) did not form a basis. Condition 3 is necessary, but not sufficient, for condition 4, which ensures that the gradient can be followed as well as approximated on the graph.

These conditions serve to narrow down the choice of graph construction. For example, given the goal of defining a neighbourhood, it might be tempting to use the set of \( n \) nearest neighbours of each point \( N^{(n)}(x) \) to define the \( n \)-nearest-neighbour graph:

\[
N_n(x) = \{ y \in G : y \in N^{(n)}(x) \text{ or } x \in N^{(n)}(y) \} \tag{9}
\]

where the reverse condition \( x \in N^{(n)}(y) \) has been included to ensure that the graph is undirected. Examples
FIG. 1. Three possible neighbourhood graphs for the grid $G$. Graph $N_2$ (red) is generated by connecting each gridpoint to its two nearest neighbours (note that the requirement of an undirected graph leads to more than two neighbours for some points). Graph $N_3$ (green) is generated by connecting each gridpoint to its three nearest neighbours. These nearest neighbour graphs can lead to disconnected regions (as circled for $N_2$) and nodes in the bulk that are not move-preserving (marked with black crosses). Graph $N_D$ (blue) is a Delaunay triangulation and exhibits no such issues.

of nearest neighbour graphs $N_2$ and $N_3$ are shown in Fig. 1 for a 2D grid, where we can see they suffer from several shortcomings. In particular, they are not necessarily connected or move-preserving which leads to the introduction of fictitious local maxima and local minima as can be seen in Fig. 3.

2. The Delaunay triangulation

A sensible choice of graph to overcome the issues with nearest-neighbour graphs is a triangulation. A triangulation of a grid $G$ is a set of simplices ($N$-dimensional analogues of triangles) that tile the convex hull $H(G)$ (see, for example, $N_D$ in Fig. 1). Any triangulation immediately satisfies the requirements given in section II C 1 and possesses high-quality numerical gradients, even for the pathological case of a random grid, as can be seen in Fig. 2. The specific case of a Delaunay triangulation is particularly suitable as it minimizes the size of the largest open ball $\{b(x; d) = \{y \in \mathbb{R}^N : |y - x| < d\}\}$ which bounds a simplex [19] and thus avoids large simplices corresponding to large neighbourhoods.

The choice of Delaunay triangulation is related to the nearest-neighbour interpolation of the function:

$$f(x) \approx f_{NN}(x) = f(x \rightarrow G)$$

where $x \rightarrow G$ is the nearest neighbour of $x$ in $G$:

$$x \rightarrow G = \arg \min_{y \in G} |y - x|.$$  \hspace{1cm} (11)

Given a grid point $y \in G$, the region $\{x \in \mathbb{R}^N : x \rightarrow G = y\}$ where $f_{NN}(x) = f(y)$ is known as the Voronoi cell

FIG. 2. Analytic ($\nabla f$, top) and numerical ($g$, middle - calculated via Eq. 10) gradients for $f(x) = \exp(-|x - a|^2) + \exp(-|x - b|^2)$ ($a$ = red dot, $b$ = blue dot) with neighbours given by a Delaunay triangulation (light grey graph) of a set of random points (black dots). A histogram (bottom, log scale) of normalized dot products between analytic and numerical gradients.
of y. The neighbourhood graph obtained via a Delaunay triangulation is equivalent to connecting points with corresponding Voronoi cells that are adjacent in $\mathbb{R}^N$ [20].

D. Maxima families and basins of attraction

Along with a suitable definition for neighbourhoods, it is important to be able to identify regions of interest in $G$. In particular, given a function $f : \mathbb{R}^N \rightarrow \mathbb{R}$, it is essential to be able to identify connected subsets of $\mathbb{R}^N$ for which $f$ is locally maximal. These include pointlike maxima of $f$ (e.g. the point $x = 0$ for $f = -|x|$), but also spatially-extended maxima (e.g. the shell at $|x| = 1$ of $f(x) = -(|x| - 1)^2$). Such a subset (and its analogue on $G$) will be referred to as a \textit{maxima family} $M$ and the set of maxima families of $f$ as $\mathcal{M}(f)$. Then, for a given family $M \in \mathcal{M}(f)$, $f(x) \geq f(x + \delta) \quad \forall \ x \in M$ for any infinitesimal perturbation $\delta \in \mathbb{R}^N \rightarrow 0$. The concept of maxima families also permits the definition of \textit{basins of attraction} of $f$. Given a starting point $x \in \mathbb{R}^N$, we can define a \textit{point} of attraction $A(x)$ via repeated application of a steepest-ascent step

$$S_\delta(x) = \arg \max_{y \in b(x, \delta)} \frac{f(y) - f(x)}{|y - x|} \quad (12)$$

as

$$A(x) = \lim_{N \rightarrow \infty} \lim_{\delta \rightarrow 0} S_{\delta}^N(x) \quad (13)$$

where the open circle $\circ$ in $S_{\delta}^N$ denotes $N$ repeated applications of $S_\delta$ to $x$ (not taking the $N^{th}$ power). A basin of attraction of $f$ is then the region of $\mathbb{R}^N$ for which all steepest-ascent paths lead to a particular maxima family:

$$B(M \in \mathcal{M}) = \{ x \in \mathbb{R}^N : A(x) \in M \} \quad (14)$$

The concept of a steepest ascent path generalizes straightforwardly to a graph and so one might also expect basins of attraction to generalize straightforwardly. However, in general, the maxima of $f$ will not lie exactly on the grid $G$. This means that the set of points on the graph that are best suited to represent a particular maxima family will not all have exactly the same function values and maxima families can only be approximately defined. In the present work, the definition is based upon an expansion around the \textit{local maxima} of the graph $M_L(G) = \{ x \in G : f(y) < f(x) \ \forall \ y \in N(x) \}$ which are typically the closest points on $G$ to maxima families of $f$. In order to construct the basins of attraction, two objects must be constructed; $A : G \rightarrow M_L(G)$ which maps a point to the local maximum whose basin of attraction it resides in (in analogy to the point of attraction $A(x) \in \mathbb{R}^N$) and the families of local maxima $\mathcal{M}(f, G)$ (in analogy to the maxima families $\mathcal{M}(f)$ on $\mathbb{R}^N$). The basins of attraction for the maxima families are then

$$B(M \in \mathcal{M}(f, G)) = \{ x \in G : A(x) \in M \} \quad (15)$$

in analogy with Eq. [14].

The algorithm to determine $A$ is based on that of Henkelman et. al. [21], but applied to a graph rather than to a uniform grid. A schematic is shown in Fig. 4 and the steps are detailed below:

1. \textit{Initialize:} Let $D(A)$ be the domain of $A : G \rightarrow M_L(G)$ (i.e. the set of points assigned to a local maximum). Initially, $D(A) = \emptyset$.

2. \textit{Check termination:} If the set of unassigned points $G \setminus D(A)$ is empty, then $A : G \rightarrow M_L(G)$ is complete on $G$ (all points have been assigned) and the algorithm terminates.

3. \textit{New path:} Identify an unassigned point $x \in G \setminus D(A)$ and start a steepest ascent path $P = \{ x \}$.

4. \textit{Reached maxima:} If $x \in M_L(G)$ then let $A(p) = x \ \forall \ p \in P$ and return to step 2.

5. \textit{Steepest step:} Identify $y \in N(x)$ that maximizes $\frac{|f(y) - f(x)|}{|y - x|}$ and add it to $P$.

6. \textit{Shortcut:} If $y$ is assigned ($y \in D(A)$) then let $A(p) = A(y) \ \forall \ p \in P$ and return to step 2.

7. \textit{Iterate:} Let $x = y$ and return to step 3.

As noted in [22], step 3 of this algorithm allows rediscovery of previous steepest ascent paths and significantly improves runtime performance.

Once we have constructed the map $A : G \rightarrow M_L(G)$ associating points to local maxima, we turn our attention to the algorithm to cluster local maxima into families $\mathcal{M}(f, G)$. This clustering is based upon a measure of deviation $d(x, y) : M_L(G) \times G \rightarrow \mathbb{R}$ that increases as $y$ moves away from the maxima family containing the local maximum $x$. In the present work, the following measure is used:

$$d(x, y) = \frac{|f(y) - f(x)|}{\max(f(y), f(x)) - \min(f(z) : z \in G)} \quad (16)$$
FIG. 4. A schematic showing how steepest ascent paths (arrows) on a graph allow us to reconstruct the two regions of attraction (the set of red and blue dots respectively) for two separate maxima of the same function.

FIG. 5. Regions with a deviation $d(x_1, x)$ of less than $t = 0.25$ for two maxima $x_1$ and $x_2$ of a function $f(x)$. Note that the region for the smaller maxima is smaller, thanks to the scale-independence of the deviation measure (Eq. 16).

This is essentially the fractional change in the function value due to moving from $x$ to $y$, and therefore $d(x, y) \in [0, 1]$ independently of the scale of the function. Once $d(x, y)$ has been defined, a tolerance $t$ can be chosen such that $d(x, y) < t$ defines a stationary region around each maximum (see Fig. 5) and apply the following algorithm to cluster local maxima into families. The algorithm begins by constructing a flood fill around each local maxima according to the tolerance and ends by merging overlapping flood fills into connected families (see Fig. 6):

1. **Initialize**: Let $i = 0$ and $F_0 = \emptyset$ be an empty flood fill.

2. **New maxima**: Identify a local maximum that is not yet in a flood fill $x \in M_L(G) \setminus \bigcup F_i$ and let $F_i = \{x\}$. If no such maxima exist, go to step 4.

3. **Identify shell**: Identify the shell $S$ of neighbours surrounding $F_i$ as $S = \bigcup_{y \in F_i} N(y) \setminus F_i$. Identify the points in the shell that are still within tolerance of the initial maximum $T = \{y \in S : d(x, y) < t\}$.

4. **Expand**: Expand $F_i$ to include points in $T$: $F_i \rightarrow F_i \cup T$. Return to step 3.

5. **Merge floods**: If any two flood fills overlap ($\exists i \neq j : F_i \cap F_j \neq \emptyset$) then merge into a single flood fill: $F_{\min(i,j)} \rightarrow F_i \cup F_j$, $F_{\max(i,j)} \rightarrow \emptyset$. Repeat this process until no flood fills overlap.

6. **Assign families**: Group maxima into families according to the merged flood to which they belong $M(f, G) = \{\{F_i \cap M_L(G)\} \forall i : F_i \neq \emptyset\}$.

**E. Basins of attraction: Example applications**

Example applications presented in the rest of this work were carried out using quantities from a Hartree-Fock calculation with a cc-pVDZ basis set. For topology analysis, the quantity of interest is then evaluated on a DFT grid generated using an LMG radial grid (with a threshold of $10^{-10}$), a Lebedev angular grid (with degree between 15 and 25 depending on the radius) and by pruning points with a weight of less than $10^{-12}$. This results in a relatively coarse DFT grid ($\sim 10^4$ points per atom), with the hope of replicating the worst-case scenario that would be encountered in real-world applications. Hartree-Fock was used rather than DFT so that the dependence of the topology analysis on the grid could be investigated independently of DFT grid integration.
1. Bader regions

An object of central importance in quantum chemistry is the electron density \( \rho(r) : \mathbb{R}^3 \to \mathbb{R}^+ \). Bader demonstrated a correspondence between the basins of attraction of the electron charge density and atoms in molecules [15]. Specifically, each basin of attraction contains exactly one atom in a molecular system, allowing one to uniquely assign the electronic charge present on each atom as the integral of the charge density over its basin of attraction. This leads to the bader charges, here defined in \( \mathbb{R}^3 \) as

\[
q(M \in \mathcal{M}(\rho)) = \int_{B(M)} \rho(r) d^3r
\]

and on \( G \) as

\[
q(M \in \mathcal{M}(\rho, G)) = \sum_{x \in M} \rho(x) w(x)
\]

where \( w(x) \) are grid integration weights (typically generated along with the grid itself [18], but which could be taken as the volume of the Voronoi cell of \( x \)). The basins of attraction for the electron density of a benzene molecule are shown in Fig. 7. Note how the grid points become more dense towards the nuclei. The convergence of Bader charges as a function of grid size will be discussed in detail in Sec. II H 3.

2. Electron shells from \( \nabla^2 \rho \)

Bader charge analysis as carried out in section II E is insensitive to the treatment of maxima families. This is because, for molecular systems, the electron density \( \rho \) has no extended maxima, only distinct point-like maxima near to each nucleus. However, this is not true for the Laplacian of the electron density \( \nabla^2 \rho \). Indeed, the electronic shell structure of atoms leads to \( \nabla^2 \rho \) exhibiting alternating regions of charge concentration (\( \nabla^2 \rho < 0 \)) and charge depletion (\( \nabla^2 \rho > 0 \)) as one moves radially away from the nucleus [10]. This naturally leads to spatially-extended maximal shells of \( \nabla^2 \rho \) and derived quantities, as can be seen for a Neon atom in Fig. 8. The changes in the shell structure of the Laplacian upon bond formation will be discussed in Sec. II G 2.

3. Isosurfaces

Given a target function value \( f_{\text{iso}} \), an isosurface of \( f \) can be defined as \( \{ x \in \mathbb{R}^N : f(x) = f_{\text{iso}} \} \). Due to the delocalized nature of electrons in molecules, isosurfaces are commonly used in molecular visualization. The ability to identify families of maxima allows the topological analysis of such isosurfaces by defining an auxiliary function \( f_I(x) = \exp(-|f(x) - f_{\text{iso}}|) \) which will be maximal where \( f(x) = f_{\text{iso}} \). The maxima family (or families) where \( f_I(x) \approx 0 \) then serve as a suitable definition of isosurfaces. An example of this can be seen in Fig. 9 where non-covalent bonding in \( \text{H}_2 \) under a strong magnetic field (as explored in [23]) can be identified as the separation of the half-maximum-value isosurface of the electron density (where \( \rho(x) = \max(\rho)/2 \)) into two distinct maxima families.

F. Critical paths

A critical path is defined as a path linking two local maxima on \( N \) that maximizes the minimum value of \( f \) encountered (the critical value of that path). The equivalent of this path in \( \mathbb{R}^N \) necessarily passes through a first-order saddle point of \( f \) known as a critical point and, in analogy, the point of minimum \( f \) on a critical path in \( N \) is labelled as a critical point (see Fig. 10). Given a neighbourhood graph \( N \), edge weights are assigned as the average of the function values at the endpoints of the edge. It is then possible to find critical paths rapidly by noting that they are paths on the maximum spanning tree (MST) of \( N \), which is denoted as \( M(N) \) (the critical-path problem essentially becomes the widest path problem from graph theory). In fact, the critical path between two local maxima on \( N \) is the only path linking the maxima on \( M(N) \), thanks to the fact that \( M(N) \) is acyclic. The union of all critical paths is called the
FIG. 8. The atomic shells of a Ne atom, visualized by plotting the distinct maxima families of $|\nabla^2 \rho|$ on a DFT grid, identified by the algorithm given in section II D. The axes are in Bohr.

critical tree, which can be found rapidly and in its entirety by repeatedly pruning the maximum spanning tree according to the following algorithm (shown in Fig. 10):

1. Maximum spanning tree: Let $M$ be the maximum spanning tree of $N$, evaluated with edge weights given by the average of function values on the endpoints of each edge.

2. Identify leaf nodes: Let $C$ be the set of leaf nodes of $M(N)$ (nodes with only one neighbour) that are not local maxima. If there are no such nodes, terminate the algorithm.

3. Pruning: Remove the nodes $C$ from $M$. Return to step 2.

One can avoid searching the entire tree for leaf nodes at every iteration of step 2 by expanding from the previous set of pruned leaf nodes.

1. Recovering cyclic graphs (gap-filling)

The critical tree is inherently acyclic (as it is a subgraph of the maximum spanning tree) -- a direct consequence of the definition of a critical path. However, it is possible that there are multiple paths with very similar critical values between a given pair of local maxima. For example, the electronic charge density of a benzene molecule exhibits local maxima at the nuclei which can be linked together by traversing the aromatic ring either clockwise, or anticlockwise (see Fig. 11). Both of these routes have very similar critical values, but only one (that which has the slightly larger critical value within a finite precision computation) will be included in the critical tree. For the benzene system this means that whichever bond happens to have the lowest electron density will be excluded from the maximum spanning tree, and hence also from the critical tree. However, such bonds can be re-introduced by considering neighbouring basins of attraction, using the following gap-filling algorithm (this produces a critical network according to the definition of Bader [8]):

1. Initialize: Let $C$ be the critical tree (as determined via the above algorithm).

2. Iterate: Iterate over pairs of maxima $x, y \in M_L(G)$.

3. Check already linked: If the path between $x$ and $y$ on $C$ passes through only two basins of attraction,
FIG. 10. Schematic demonstrating how critical paths can be identified by pruning the maximum spanning tree of a graph according to the algorithm presented in section II F. Nodes that are pruned are labelled by the algorithm iteration number at which they are pruned.

FIG. 11. Two paths (red and blue arrows), with similar critical values, connecting nuclei A and B around the bonding network of a benzene molecule.

then $x$ and $y$ are already critically-linked in $C$ and we can continue to the next iteration (goto step 2).

4. Identify basins: Let $B_x$ ($B_y$) be the basin of attraction containing the point $x$ ($y$).

5. Check neighbouring: If the basins $B_x$ and $B_y$ are not adjacent (i.e. $B_x \cap \bigcup_{z \in B_y} N(z) = \emptyset$), then $x$ and $y$ are not critically linked. Continue to the next iteration (goto step 2).

6. Construct subgraph: Construct the subgraph of $G$ containing only nodes in the basins of attraction $B_x$ and $B_y$ as $G_{xy} = G \cap (B_x \cup B_y)$ and it’s boundary $B(G_{xy}) = \{z \in G_{xy} : \exists z_2 \in N(z) \text{ s.t. } z_2 \notin G_{xy}\}$ (where $N(z)$ are the neighbours of $z$ in $G$).

7. Construct MST: Construct the maximum spanning tree $M_{xy}$ of $G_{xy}$. Identify the path $P_{xy}$ linking $x$ and $y$ on $M_{xy}$.

8. Reject non-critical path: If, at any point, the path $P_{xy}$ touches the boundary (i.e $P_{xy} \cap B(G_{xy}) \neq \emptyset$), then $x$ and $y$ are not critically linked. Continue to the next iteration of the loop (step 2).

9. Fill gap: $P$ is a critical path in $G_{xy}$ linking $x$ and $y$ and the edges along $P$ should be added to $C$. Continue to the next iteration of the loop (step 2).

Step 8 identifies a non-critical path between neighbouring regions by noting that the critical point is pushed right to the edge of the shared border of the regions (see path $P_{BC}$ in Fig. 12). In order for two regions to be critically linked, the critical point must instead constitute a saddle point in the bulk of the shared boundary (as is the case for paths $P_{AC}$ and $P_{AB}$ in Fig. 12).

We could have generated all of our critical paths by following this gap-filling algorithm, by starting instead with an empty graph $C$. However, starting with the critical tree avoids having to construct the subgraph $G_{xy}$ for every pair $x, y$ and is thus more efficient.

2. Cleaving

By construction, the critical tree contains and connects all local maxima in the network. However, as we will see later, it is useful to be able to divide the critical tree into sub-trees within which $f(x)$ varies only weakly. This process is called cleaving and it proceeds as follows:

1. Initialize: Let $C$ be the critical tree of $f(x)$ on $G$ and $P(x, y)$ be the path between points $x$ and $y$ on $C$.

2. Get paths: Let $\mathcal{P}$ be the set of all critical paths in $C$ (that is, paths between local maxima that do not pass through other local maxima): $\mathcal{P} = \{P(x, y) \mid x, y \in M_L(G) : x \neq y, P(x, y) \cap M_L(G) = \{x, y\}\}$.

3. Set function scales: For each path, set a function scale as the maximum endpoint value $f_{\text{scale}}(P(x, y)) = \max(f(x), f(y))$. 

FIG. 12. A schematic of critical ($P_{AC}$ and $P_{AB}$, blue) and non-critical ($P_{BC}$, red) paths linking three maxima of a function on the plane (whose basins of attraction are separated by dashed lines). Note that the non-critical path between $B$ and $C$ touches the boundary of the union of their basins of attraction.
4. **Calculate deviations**: For each path $P(x, y)$, calculate a deviation

$$D(P(x, y)) = \max_{z \in P} s(z) - \min_{z \in P} s(z)$$  \hspace{1cm} (19)

where $s(z)$ is the function value, scaled so the maximum endpoint value is 1:

$$s(z) = \frac{(f(z) - f_{\text{min}})}{(f_{\text{scale}}(P) - f_{\text{min}})}$$  \hspace{1cm} (20)

and $f_{\text{min}} = \min\{f(a) : a \in G\}$ is the global minimum function value.

5. **Cluster paths**: Cluster the paths into a flat set $P_{\text{flat}} = \{P \in \mathcal{P} : D(P) < t_{\text{flat}}\}$ where the function value changes by a small amount (according to some tolerance $t_{\text{flat}}$) along the path. Consider the rest of the paths to be non-flat $P_{\text{non-flat}} = \mathcal{P} \setminus P_{\text{flat}}$.

   In the present work, a kernel density estimate is used to inform the choice of cluster tolerance $t_{\text{flat}}$.

6. **Cleave non-flat paths**: Remove the edges of each non-flat path from $C$.

In an alternative scheme, one might use the subgraphs of the cleaved critical tree to define the maxima families when identifying basins of attraction. However, the flood fill technique introduced in section II D is more robust in practice (as the flood fills are more densely connected over surface-like maxima than a tree).

**G. Critical paths: Example applications**

1. **Bond paths**

   In Bader analysis, paths on the critical network are called bond paths, and provide a unique (although not necessarily optimal) definition of molecular bonds. The bond paths for benzene, evaluated using the algorithm given in section II F, are shown in Fig. 13. All bonds are recovered (one of which via the gap filling algorithm given in section II D), leading to the familiar hexagonal benzene bonding network.

2. **Valence charge concentration and depletion graphs**

   Charge concentration ($\nabla^2 \rho < 0$), or depletion ($\nabla^2 \rho > 0$), is most relevant to chemistry when it occurs in the valence region of an atom in a molecule. In particular, it has been noted that the maxima of valence charge concentration (depletion) correlate with the active regions for electrophilic (nucleophilic) attack. Critical networks spanning these maxima form the *valence shell charge concentration/depletion (VSCC/D)* graphs.

   Such graphs can be easily examined by constructing the critical network of $-\nabla^2 \rho$ (charge concentration) or $\nabla^2 \rho$ (charge depletion). An example for the VSCC graph of a water molecule is shown in Fig. 14 (top, c.f Fig. 3 of [10]). This VSCC graph can clearly be seen to connect the lone pairs either side of the oxygen atom. This behaviour is reflected in the critical network of the 90% ELF isosurface (Fig. 14 middle), where the lone pairs can be very clearly seen as lobes aligned along the perpendicular direction to the bonds. Such charge concentration arises from distortions in the valence shell of the oxygen atom due to the hydrogen atoms, which can be seen by looking at the maxima families of $|\nabla^2 \rho(r)|$ (Fig. 14 bottom, valence shells are shown in blue, c.f the shells of Ne in Fig. 3) - note that core shells (pink, for example) retain their spherical nature.

3. **Stagnation graphs**

   Applying a magnetic field to a molecule induces a current density vector field $\mathbf{J}(x) : \mathbb{R}^3 \to \mathbb{R}^3$. The *stagnation graph* of $\mathbf{J}$ is the subset of $\mathbb{R}^3$ where $|\mathbf{J}(x)| = 0$ and in general consists of isolated stagnation points and extended stagnation lines.

   These stagnation graphs form a compact representation of the topology of the vector field and have significance in ring-current models and NMR spectra. The stagnation graph can be obtained as the critical network of $-|\mathbf{J}|$, as can be seen for a $C_2H_2$ molecule in Fig. 15. This stagnation graph exhibits the same features as a more detailed analysis at signifi-
FIG. 14. Topographical analysis for various properties of the water molecule. The molecular geometry is shown as a dotted line.

FIG. 15. The stagnation graph of C₂H₂, visualised as the cleaved critical tree of −|J|. The axes are in Bohr.

FIG. 16. Convergence properties of the Bader charges of the oxygen basin in a water molecule for different grids and graph ascent methods. Datapoints are shown as crosses connected by solid lines and the region within one standard deviation of an exponential fit is shaded for each series. For the DFT grids, this standard deviation is too small to be resolvable, and the fit line is shown instead. Uniform grids are scaled by decreasing the grid spacing, DFT grids are scaled by reducing the LMG tolerance and increasing the Lebedev degree simultaneously.

H. Performance

1. Convergence

Thanks to the favorable properties of the Delaunay triangulation (see Sec. II C 1), topographical properties (such as Bader charges) converge quickly with grid size. This can be seen in Fig. 16 where convergence is achieved for DFT grids well before 1 million grid points. This is in contrast to simple uniform-grid methods, where tens of millions of points are often required [30]. DFT grids converge particularly quickly as they are designed for rapid convergence of integral quantities, but even the uniform grids shown in Fig. 16 perform well thanks to their connectivity with a triangulation, rather than a simple grid (see also Fig. 17). Topological properties, such as the number of maxima and saddle points etc., converge almost instantly.
FIG. 17. The divergence of steepest on-graph paths from the true gradient $\nabla f$ for different grids. At each step of the steepest-ascent path the gradient is followed as closely as possible on the graph, but small errors at each step accumulate and the paths eventually diverge. Note that the diagonal moves introduced into a uniform grid via triangulation help (more so in 3D), but do not remove the problem. The easiest way to see that this problem is scale independent is by considering the upper-left uniform grid case - the steepest ascent path will always be “upward” (the horizontal moves will never be taken), regardless of the grid spacing.

2. Grid bias

As noted in Ref. [30] for uniform grids, topographical properties (such as Bader charges) can be affected by a grid bias, whereby a systematic error arises due to geometrical properties of the arrangement of grid points. This error is due to steepest-ascent paths on the graph diverging from the true gradient path of the function, and, remarkably, persists even in the infinite-grid-density limit. In particular, gradient paths which are nearly, but not quite, aligned with move directions on the graph will lead to gradually-diverging steepest-ascent paths, as can be seen in Fig. 17, leading to distorted region boundaries.

Ref. [30] provides a solution to the grid bias problem by allowing the trajectory of ascent paths to go “off-graph”.

3. Scaling

The rate-limiting step in performing topological analysis via a graph over grid points is the construction of the Delaunay triangulation which scales as $O(N \log(N))$ in the number of grid points $N$ [32]. This scaling is reflected in our calculation times (see Fig. 18). Note that we use a largely unoptimized python code, so the absolute time shown on the $y$ axis of Fig. 18 could be improved relatively easily if desired, but the scaling will remain $O(N \log(N))$. 

FIG. 18. Time scaling for Bader analysis of a water molecule as a function of grid size for both a uniform and a DFT grid.

For the graphs employed in this work, an analogous “off-graph” method can be straightforwardly implemented by allowing our ascent path to move freely in $\mathbb{R}^N$, following the nearest neighbour gradient $g_{NN}(x)$ given by

$$\nabla f \approx g_{NN}(x) = g(x \to G) \quad (21)$$

where $g(x)$ is the finite-difference gradient given by Eq. 6. The nearest-neighbour lookup $x \to G$ (see Eq. 11) can be implemented efficiently as a KD-tree [31]. In Fig. 18, it is clear that this technique corrects the grid bias for a DFT grid so that it agrees with the uniform result. The uniform grid shows significantly less grid bias in Fig. 16 due to the inclusion of diagonal moves by the Delaney triangulation (the DFT grid also has such diagonal moves, but they are less helpful as a significant radial bias remains). These diagonal moves were not present in previous uniform-grid-based approaches [30], which therefore exhibited significantly larger grid bias than the present method.
III. SUMMARY

A method has been presented to extract topographical and topological properties of a function defined on an arbitrary set of points in space. By connecting the points with a neighbourhood graph, well-defined and robust algorithms were developed that allow for identification of local and global maxima (both point-like and spatially-extended), saddle points, critical paths (and their critical points) and basins of attraction. By simple transformations of the function, one can also identify local and global minima, isosurfaces and stagnation graphs. Applications of the analysis were demonstrated for a few problems in quantum chemistry including Bader charge and bond analysis, identification of valence shells and their charge concentration, location of lone pairs via the electron localization function or the Laplacian of the electron density and identification of stagnation graphs of magnetically-induced currents. All of these investigations were carried out directly on a real-space integration grid used in density functional theory (DFT) calculations, allowing the results to be easily, efficiently and directly incorporated into DFT calculations. The analysis was found to scale as $O(N \log(N))$ where $N$ is the number of grid points and quantities of interest were found to converge rapidly with $N$, requiring an order of magnitude fewer gridpoints than existing methods. Topographical results calculated using such DFT grids were found to exhibit a significant “grid bias” when the algorithm was constrained to stay on the graph. The source of this bias was analysed and found to be removed by allowing off-graph moves.

ACKNOWLEDGEMENTS

We acknowledge financial support from the European Research Council under the European Union’s H2020 research and innovation programme/ERC Consolidator Grant topDFT (grant agreement no. 772259). We are grateful for access to the University of Nottingham High Performance Computing facility.
[24] G. R. Terrell and D. W. Scott, The Annals of Statistics 20, 1236 (1992).
[25] R. F. W. Bader, P. J. MacDougall, and C. D. H. Lau, Journal of the American Chemical Society 106, 1594 (1984).
[26] J. A. N. F. Gomes, The Journal of Chemical Physics 78, 4585 (1983).
[27] S. Pelloni, F. Faglioni, R. Zanasi, and P. Lazzeretti, Phys. Rev. A 74, 012506 (2006).
[28] I. Morao and F. P. Cossio, The Journal of Organic Chemistry 64, 1868 (1999).
[29] T. J. P. Irons, A. Garner, and A. M. Teale, Chemistry 3, 916 (2021).
[30] W. Tang, E. Sanville, and G. Henkelman, Journal of Physics: Condensed Matter 21, 084204 (2009).
[31] J. L. Bentley, Commun. ACM 18, 509–517 (1975).
[32] R. A. Dwyer, Algorithmica 2, 137 (1987).