Visualization and integration of quantum topological atoms by spatial discretization into finite elements

ARTICLE *in* JOURNAL OF COMPUTATIONAL CHEMISTRY · DECEMBER 2007

Impact Factor: 3.59 · DOI: 10.1002/jcc.20767 · Source: PubMed

2 AUTHORS, INCLUDING:

Paul L A Popelier
The University of Manchester
190 PUBLICATIONS **7,094** CITATIONS

Visualization and Integration of Quantum Topological Atoms by Spatial Discretization Into Finite Elements

M. RAFAT, P. L. A. POPELIER
Manchester Interdisciplinary Biocenter (MIB), The University of Manchester, 131 Princess Street, Manchester M1 7DN, Great Britain

Received 16 October 2006; Revised 21 March 2007; Accepted 2 April 2007
DOI 10.1002/jcc.20767
Published online 16 May 2007 in Wiley InterScience (www.interscience.wiley.com).

Abstract: We present a novel algorithm to integrate property densities over the volume of a quantum topological atom. Atoms are grown outward, starting from a sphere centered on the nucleus, by means of a finite element meshing algorithm. Bond critical points and ring critical points require special treatment. The overall philosophy as well as intricate features of this meshing algorithm are given, followed by details of the quadrature over the finite elements. An effort has been made to design a streamlined and compact algorithm, focusing on the core of challenges arising in tracing the electron density's gradient vector field. The current algorithm also generates a new type of pictures that can be a Graphical User Interface. Excellent integration errors, $L(\Omega)$, are obtained, even for atoms with (narrow) tails or sharp corners.

 J Comput Chem 28: 2602–2617, 2007

Key words: quantum chemical topology; atoms in molecules; volume discretization; quadrature; integration; meshing; finite elements

Introduction

Over the years quantum chemical topology (QCT)[1,2] has featured[3,4] as a means of obtaining chemical insight in areas as diverse as transition metal and main group chemistry, biochemistry, surface science, high resolution crystallography, and mineralogy, and involving topics such as quantitative structure activity relationships, van der Waals complexes, reaction mechanisms and radicals, to name a few. From an operational point of view, the key idea of QCT is the gradient vector field, a bundle of gradient paths (GPs) in three-dimensional (3D) space. A GP is a trajectory of steepest ascent in any 3D function such as the electron density ρ, its Laplacian $\nabla^2\rho$, or ELP.[5,6] The gradient vector field displays critical points (CP), that is, points where the gradient of the 3D function vanishes. There are four types of critical points in 3D space. The evaluation of property densities at the CPs and their connectivity via special GPs is instrumental in recovering the chemical insight that QCT offers. An exhaustive and clear classification[7] of all possible GPs that topologically connect two CPs in an arbitrary scalar 3D function was published before. What concerns us here is the integration over the volume of a topological atom, which is called the (atomic) basin. Such an atom is a bundle of GPs originating at infinity and terminating at a nucleus. Atoms are bounded by interatomic surfaces (IASs), which are a subset of GPs, again originating at infinity but now terminating at a so-called bond critical point (BCP). This is one of the two types of saddle CPs, found in between two topologically connected nuclei. The connection consists of two GPs, originating at the BCP and collectively called the bond path (BP). The second type of saddle point is a ring critical point (RCP). Integration of property densities over the often irregular volume of QCT atoms is notoriously difficult but necessary to obtain atomic properties.

Several computer codes accomplishing atomic integration with varying degrees of success and efficiency have been written. In the oldest program, called PROAIM, which is part of the AIMPAC suite of programs, the IASs are approximated by triangulation.[8,9] The integration process in PROAIM has been the basis of other algorithms, such as FASTINT[10] for example, which is an updated version of PROAIM. The programs TOPXD,[11] VALTOPO,[12] and WinXPRO,[13] which all compute the electronic density from experimental data, also use the PROAIM approach to approximate the IAS and integrate the atomic basin. In all these algorithms, the volume integration's quadrature scheme is based on a 3D Gauss—Legendre grid of spherical coordinates. The integration algorithm of AIM2000 does not represent the IAS explicitly. Instead, it performs an integration in so-called natural coordinates[14,15] introducing a specific Jacobian. Part of the program PROMOLDEN developed by Pendás and Luaña[16] is based on the same integration platform as AIM2000. The numerical methods of the algorithms described so far suffer from reduced accuracy in the presence of compli-

Correspondence to: P. Popelier; e-mail: pla@manchester.ac.uk
Contract/grant sponsor: Leverhulme Trust

cated topologies. For these cases, a bisection method[17] is used to find the boundary of the atom, in combination with for instance a Gauss–Legendre grid. The program InteGriTy[18] also uses a bisection method to find the IAS, the integration being performed by a Romberg procedure.[19] A unique feature of this program though is the calculation of ρ: It is computed from a tricubic Lagrange interpolation of a 3D grid in the atomic basin. The OCTREE algorithm,[20] which has been implemented in a local version of MORPHY, uses a recursive cube division to delineate the shape of an atomic basin. It is a very robust algorithm but its huge computational cost practically restricts it to very complicated topologies, as they occur in the Laplacian of ρ. The program TopMod[21] has been designed for the topology of the electron localization function[22] (ELF), using a molecular grid to perform the integration. Finally, we mention the algorithm of MORPHY98*, which is based on an analytical fit of the IASs with Chebyshev polynomials.[23]

In this article, we explain in detail a completely new integration algorithm based on the spatial discretisation (or meshing) of topological atoms. For this purpose, we divide the atom into a number of simply shaped regions called *finite elements*, which we constructed both in 2D (triangles, quadrilaterals) and in 3D (hexahedral or brick elements and prism† elements). We discuss the philosophy and principles behind the algorithm, and its modular structure. It will be shown that RCPs need extra care and a special treatment. We present some results of the 2D integration over IASs (to obtain surface properties), the 3D integration over atomic volumes (to obtain atomic properties), and the 6D integration between two atoms (to obtain atomic interaction energies).

Description of the Algorithm

General Philosophy

To perceive the overall picture, it is instructive to highlight the core design behind the algorithm. The main idea is to grow the basin outwardly, starting from an initial sphere that is centered on the nucleus and completely within the given topological atom. The GPs, traced in the reverse direction (i.e. decreasing electron density), guide this growth, which is interrupted at given values of the electron density. As the GPs unfold the topological shape of the atom, the growing space is represented by finite elements, gradually filling space. As such, layers of finite elements are constructed, each element giving rise to the next one, guided by the gradient vector field. This process is smooth as long as the GPs have marginal curvatures and stay aligned in broad bundles. If GPs start following different directions, a previously convergent bundle may start falling apart into strands each pursuing increasingly divergent directions. In that case, finite elements break up into smaller ones, endowing the meshing with enhanced flexibility in accommodating the topological tearing process.

One might naively expect that the growth procedure described above automatically leads to a basin that will approach its bounding IASs arbitrarily closely. In other words, one would expect that the outermost edge of the growing basin would coincide with the IASs. This assumption proved to be wrong. Hence, the IASs had to be traced independently and connected to the basin's growth edge. Secondly, it turned out to be necessary to "announce" the presence of BCPs early on in the growing procedure. The GPs surrounding a given BP appear as a funnel piercing into the atom, leaving an imprint on the initial sphere from which the growing procedure is started. The RCP is another topological object whose presence needs to be known at the outset of the growing process. In particular, a ring plane, if present of course, leaves an imprint on the initial sphere resembling a mouth. This "mouth" indicates that "space is torn" into a region above the ring plane and another region below it. Again, to guarantee a successful termination of the growing protocol, this information needs to be known early on. Note also that the full set of isodensities is stored from the start such that one knows the level at which connection will occur.

Overall, the design of a compact and robust algorithm suffered from a perpetual struggle between topology and geometry, the latter being represented by finite elements and the former by the gradient vector field. Obtaining topological information is much more expensive than obtaining geometrical information, which is embedded in lines and quadratic curves. Topological information can only be drawn from the compute intensive evaluation of GPs. On the other hand, geometrical constructions are computationally cheap but describe the topological reality correctly only at a local level. Topology is global and is ultimately in charge. Geometry is local and attempting to capture the topological reality. The art of designing a robust and efficient algorithm is making topology and geometry match. We now discuss the details of the meshing of the basin and then the details of the integration itself (i.e. quadrature).

Meshing of the Atomic Basin

The starting point of the algorithm is the β-sphere, an object coined by Biegler–König in the first ever integration algorithm. It serves as the initial object from which the meshing proceeds. To construct this sphere, we evenly space typically about 500 points on its surface. These points are connected and organized via a 3D Delauney triangulation,[24] creating a triangulated sphere, as shown in Figure 1. To obtain a more realistic description of the spherical surface, the triangles are allowed to be curved rather than planar. Second-order (quadratic) polynomials can adequately represent the curvature of the edges of the triangles provided midpoints are added to every edge. We explain in the section on 3D integration of the Appendix why it is important to have the best approximation of the β-sphere by using quadratic (i.e. curved) triangles.

Once the β-sphere is set, we "grow" the object by tracing the GPs in the reverse direction (i.e. outward) starting from the points on the β-sphere. The GPs are stopped at envelopes of predetermined constant ρ, each electron density value corresponding to a different *growth level*. The β-sphere can be regarded as a

*MORPHY98 is a program written by P. L. A. Popelier with a contribution from RGA Bone, UMIST, Manchester, England, EU (1998).

†Strictly speaking, a prism is a polyhedron made of an n-side polygonal base, a translated copy, and n faces joining the corresponding sides. Here, we relax the condition that two faces must be parallel and still call this object a prism.

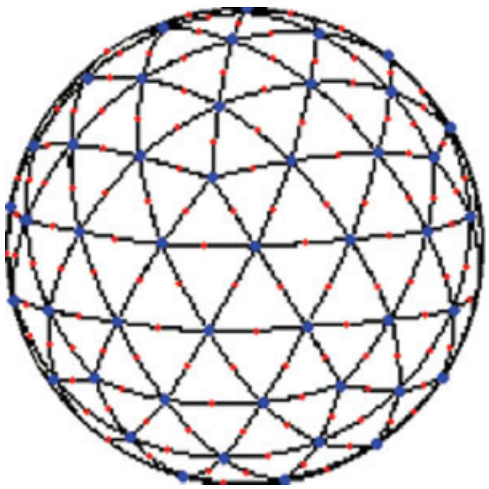


Figure 1. Triangulation of the β-sphere using quadratic triangles. The blue points represent the vertices of the triangles, the red points represent the midpoint of the edges. The midpoints are used to define curved triangles so that they fit almost exactly the β-sphere.

zeroth growth level. Typically, there are between twenty and thirty growth levels. The isodensity values are computed by successively dividing the first electron density (maximum value over the β-sphere) by a user-defined factor (typically 1.3). This factor increases during the growth process. The connections between the new points created at the first growth level are preserved in the triangles on the β-sphere. Hence, each triangle on the β-sphere has a corresponding triangle in the first growth level, grown from its vertices. The 3D space between these two corresponding triangles constitutes a prism that connects the β-sphere to the first growth level. The total collection of prisms, one for each β-sphere triangle, completely fills the space between the β-sphere and the first growth level (Fig. 2).

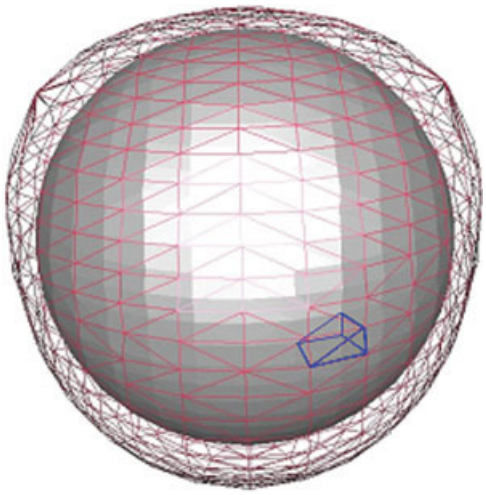


Figure 2. Triangulation of the β-sphere (in solid gray) and the first growth level (in red wireframe). The triangles of these two levels are connected with prisms. One such prism is displayed in blue. Note that the triangles of the β-sphere are quadratic.

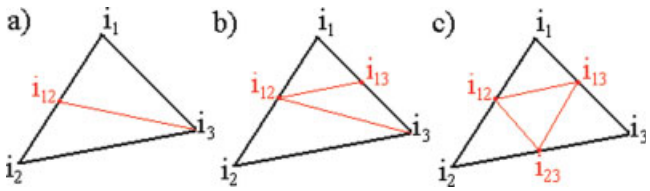


Figure 3. Division of a triangle (i_1, i_2, i_3) in three possible cases: (a) edge (i_1, i_2) is too long, hence the midpoint i_{12} is added, which creates two new triangles: (i_1, i_{12}, i_3) and (i_{12}, i_2, i_3); (b) two edges, (i_1, i_2) and (i_1, i_3), are too long, hence the midpoints i_{12} and i_{13} are added, which creates three new triangles: (i_1, i_{12}, i_{13}), (i_{12}, i_2, i_3), and (i_{23}, i_{12}, i_3); (c) all three edges are too long, hence the midpoints i_{12}, i_{23}, i_{13} are added, which create four new triangles: (i_1, i_{12}, i_{13}), (i_{12}, i_2, i_{13}), (i_{23}, i_{13}, i_3), and (i_{12}, i_{23}, i_{13}).

The process of tracing the points to the next level, creating the triangles on this new level and connecting the two growth levels with prisms, is repeated until we reach the final growth level. The atomic basin is therefore filled with prisms that can be integrated to obtain the atomic properties.

The GPs traced from the vertices of a triangle can have divergent trajectories. Such trajectories lead to stretched prisms, which can be difficult to integrate. In order to avoid this problem the algorithm divides a triangle. When one of its edges becomes longer than a predetermined (user-defined) value, a point is added in the middle of this edge. The division of the face can lead to 2, 3, or 4 triangles depending on the number of midpoints needed, as detailed in Figure 3.

The treatment of IASs is similar to the previous work on rendering of IASs[25] and is illustrated in Figure 4. The starting point of the IAS is a small triangulated circle centered (in green) at the BCP and lying in the plane perpendicular to the BP. The points on this circle are part of the IAS, and GPs starting from these points generate the complete IAS. The main difference with the previously published[25] IAS-tracing algorithm is that the current algorithm connects the circle around the BCP to the appropriate growth level. This level is that with an electron density value closest to that at the BCP. Note that if an atom is bounded by several IASs, the corresponding BCPs will be connected to the triangulated object at differing growth levels. The reader is referred to the Figure caption for the details of this connection process. Another difference is that, over each IAS, the growth of the original triangulated circle around the BCP forms quadrilaterals, not triangles as in ref. 25.

The resulting object is now more complicated, since quadrilaterals* appear on its surface rather than just triangles (Fig. 4). These quadrilaterals are the faces of the prisms added to connect the two triangulated circles (the BCP and its replica, blue and green, respectively, in Fig. 4). When the points of the quadrilaterals are traced, they will generate hexahedral (or brick) elements in the same way that the triangles generated prisms. The quadrilateral whose four vertices are marked in pink in Figure 4

*A quadrilateral is a polygon with four sides and four vertices. Sometimes the term quadrangle is used for etymological symmetry with triangle.

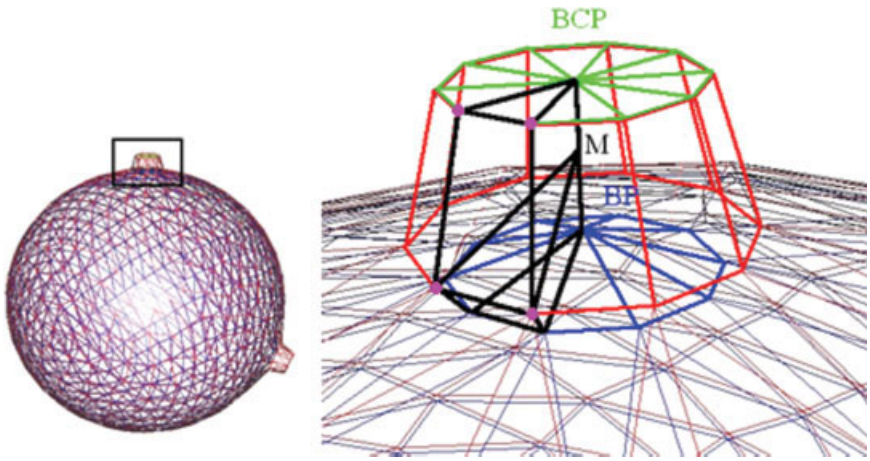


Figure 4. (Left) Two growth levels of the oxygen atom in water. The red growth level is the one with an electron density closest to that of the BCP during the growing process. The blue growth level is the one with an electron density second closest to that of the BCP. (Right) A magnification of the arrangement inside the black window of the left picture. The algorithm connects the red level to the triangulated circle around the BCP (in green). The point BP is the intersection between the bond path and the blue growth level. The point M is the middle of the segment (BP, BCP). Using point M, we construct two finite elements (prisms25 in black), linking each face of the blue triangulated circle to the corresponding face of the green triangulated circle. In this way, the BCP is incorporated into the red growth level and the space between the two growth levels is filled with prisms. A consequence of this method is the introduction of quadrilaterals28 in the growing process, which hitherto only used triangles. The pink points mark a quadrilateral discussed in the text.

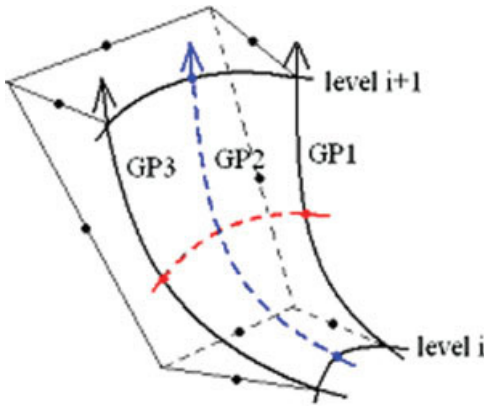


Figure 5. Positioning of midpoints in a (quadratic) hexahedral (or brick) element. The element is delineated by the quadratic quadrilaterals associated with growth level i and $i + 1$. The four curved edges in solid black form a quadratic quadrilateral, which is a patch on the IAS. The upper quadrilateral is obtained by tracing the GPs from the lower quadrilateral at level i. The GPs called GP1 and GP3 (black curved arrows) connect the points of the curved edge at level i and at level $i + 1$. There are two types of midpoint (red and blue) for the curved quadrilateral. Between two growth levels i and $i + 1$, the paths GP1 and GP3 are stopped at an intermediary isodensity value yielding the red type. GP2 provides the blue midpoints at level i and $i + 1$. For all other edges of the brick element, which are straight lines, the midpoints are put in the middle of the edge (black point).

will grow after the connection process (between the IAS and the red level). This growth generates a new quadrilateral at the next isodensity level. A brick element will be formed by the space between the two quadrilaterals. Note that the upper face of the

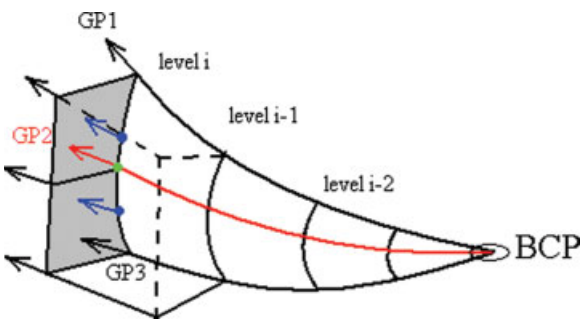


Figure 6. Division of a quadrilateral (marked in gray) at growth level i. The division is triggered by the large distance between the gradient paths GP1 and GP3. As gradient path GP2 is traced, starting from the initial circle centered at the BCP, it provides the midpoints between GP1 and GP3 at any growth level (e.g. i, $i - 1$ and $i - 2$). At level i, the quadrilateral is divided in two smaller quadrilaterals. The intersection of GP2 and level i (marked in green) constitutes the position of the new vertex, which is shared by the two smaller quadrilaterals. Two new GPs (in blue, only partially traced) are being traced in order to provide midpoints for the new quadrilaterals at level $i + 1$ and beyond.

brick element is part of the IAS itself. Finally, we note that midpoints in hexahedral elements also imply tracing an extra GP.

A brief digression explains how the blue triangulated circle in Figure 4 was obtained. Prior to the triangulation of the β-sphere, we trace the BP from the BCP to the nucleus and stop just at the β-sphere, thereby retrieving the intersection between the BP and the β-sphere. We create a replica of the circle around the BCP (not to be confused with the blue circle in Fig. 4) centered at this intersection point. The points of the replica are fully part of the triangulation of the β-sphere and treated in exactly the same way as other points during the growth process. This replica circle reappears at every growth level as the appropriate GPs are traced. The growth process eventually leads to the blue (triangulated) circle shown in Figure 4.

We note that, during the growth process, the GPs that encompass the BP like a funnel can possibly diverge. For each growth level where such a divergence is detected, a midpoint is added to the corresponding circle replica in the region where the GP gap emerges. It is vital to also add each extra midpoint to the circle centered at the BCP (green in Fig. 4). Without this addition, the connection described in Figure 4 would fail.

To represent the IAS more accurately, we use *quadratic* hexahedral elements. By adding a midpoint on the edges of the hexahedral element, the algorithm is able to bend the side of the element so that it closely follows the curvature of the IAS. There are two types of midpoints used for the curvature on each dimension of the IAS. Figure 5 demonstrates the addition of the two types of midpoint (in blue and red).

When the paths of the quadrilaterals are traced, they can diverge because of gaps emerging in the IAS. These gaps can be due to high ellipticity,[25,26] for example, when GPs bundle up in strands and avoid traversing certain regions of space. In that case, the affected quadrilateral (marked in grey, Fig. 6) is divided in two, using the midpoint already present to describe the curvature of one of the quadrilateral's edges. Figure 6 shows a schematic example of such a division. The paths GP1, GP2, and GP3 are traced from the initial circle around the BCP. Gradient paths of the type GP2 provide the midpoints that express the curvature of the IAS between GP1 and GP3. At growth level i, the distance between GP1 and GP2 is deemed to be too large, which triggers the division of the grey quadrilateral (Fig. 6) into two smaller quadrilaterals. Thus, the midpoint generated by GP2 at level i (Fig. 6, green point) marks the division of the quadrilateral into two smaller ones. Two new GPs are added (only partially traced, in blue, Fig. 6) to generate the midpoints that will define the curvature of these new quadrilaterals.

Treatment of Ring Critical Points

It should be clear from the Meshing of the Atomic Basin section that saddle points, such as BCPs, introduce complications in the growth process and therefore need careful attention. Ring critical points are the other type of saddle point that needs a special treatment. We believe that it is important for the robustness and maintenance of the overarching final meshing algorithm that it reuses the same conceptual ideas for the RCP as for the BCP. This similarity expresses itself in two aspects: the preparation of the β-sphere and the control of gaps appearing in topological surfaces. The ultimate origin of this similarity is the existence of dual relationships between topological objects. For example, the RCP is the dual of the BCP because if all minima are replaced by maxima (in each of the eigenvector directions), and *vice versa,* then a RCP becomes a BCP. We now discuss in turn the two aspects mentioned above.

Firstly, before the growth process starts, the β-sphere needs to be prepared for the presence and eventual addition of the RCP and the *ring line* (RL)[*]. For that purpose, the *ring plane* (RP)[†] is traced. Secondly, possible gaps in the RP can be overcome by the same algorithm developed for the triangulation of IASs that may contain gaps.[25] The RP is a separatrix analogous to the IAS. Hence, it is possible to construct an RP in the same way as an IAS, with minimal modification. Note that the triangulation of an RP also provides a robust method to establish the complete list of nuclei forming a ring. This method is a valuable alternative to a more vulnerable method used in MORPHY98.

Once the RP is traced, we seek its intersection with the β-sphere. This intersection consists of a set of triangles. The GPs passing through the vertices of these triangles will eventually diverge in the vicinity of the RCP. When traced in the reverse direction, one bundle of GPs bends above the RP and moves to infinity, while another bundle bends below the RP and moves to infinity. Figure 7 shows an example of a β-sphere that has its intersection with the RP removed, leaving an imprint resembling to a "mouth."

When the β-sphere is grown and after the BCPs have been added, two quadrilaterals will naturally emerge and become part of this mouth. These two quadrilaterals (Fig. 8c) link the triangles around the BP that intersect the RP on one hand and the triangles of the initial circle centered at the BCP on the other hand. Figure 8a shows the stage at which the growth level closest to the electron density of the RCP is reached. At that stage, the algorithm is ready to connect the RCP to the intersection triangles and two quadrilaterals of the mouth by means of simplices[‡] and prisms. Two points on either side of the RL are then created and linked to the growing atom by means of simplices and prisms. This is shown schematically in Figure 8b. These two points (marked in pink) are the starting points of the RL. They will ensure that the RL is part of the triangulated basin from this stage onward. Finally, from Figure 8c, it is clear that the mouth is flanked at each side by a quadrilateral. This figure

[*]The RL consists of a couple of GPs that originate at infinity and terminate at the RCP. An RL is topologically dual to a BP, in the sense that the roles of maxima and minima are reversed.

[†]The RP consists of GPs that originate at the RCP and terminate at the nuclei that constitute the ring. An RP is topologically dual to an IAS.

[‡]A simplex or n-simplex is an n-dimensional analog of a triangle. A simplex is the convex hull of a set of $(n+1)$ affinely independent points in some Euclidean space of dimension n or higher. For example, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron.

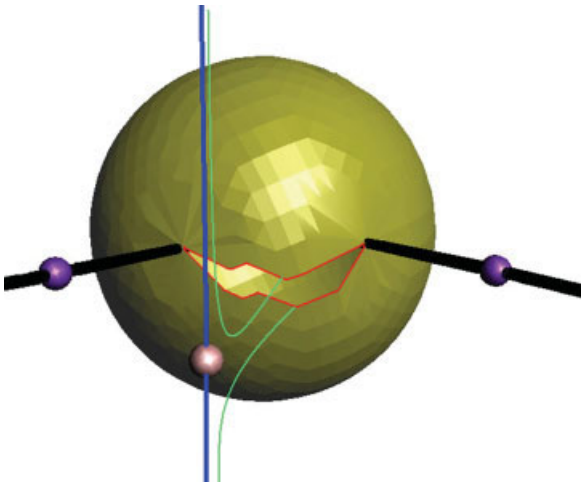


Figure 7. Triangulated β-sphere (in yellow) of which the intersection with a ring plane (i.e. the "mouth," see text) has been removed. The borders of the mouth are highlighted in red. The purple spheres represent the bond critical points, the pink one the ring critical point. The bond paths are shown in black and the ring line in blue. Two schematic gradient paths originating at infinity and attracted to two points of the mouth are drawn in green. These paths show the divergence of the gradient vector field in the vicinity of the RCP.

also shows how the two quadrilaterals are connected to the RL by means of prisms.

The actions and decisions explained in the last two sections are summarized in a flowchart given in the Appendix.

Integration

The use of finite elements enables a fast, robust, and flexible integration of any 2D or 3D object. In this section, we explain how to integrate over the finite elements introduced by the meshing of the basin.

The 2D integration of a point property $f(\mathbf{r})$ over a surface S is defined in eq. (1). In practice, S is the IAS and a surface (envelope) of constant ρ or outer isodensity surface,

$$F(S) = \int_S d\sigma \; f(\mathbf{r}) \tag{1}$$

where $d\sigma$ is an element of the surface S. Currently there are no other algorithms that can integrate over an isodensity surface. The area of this surface can be used for the calculation of the polar

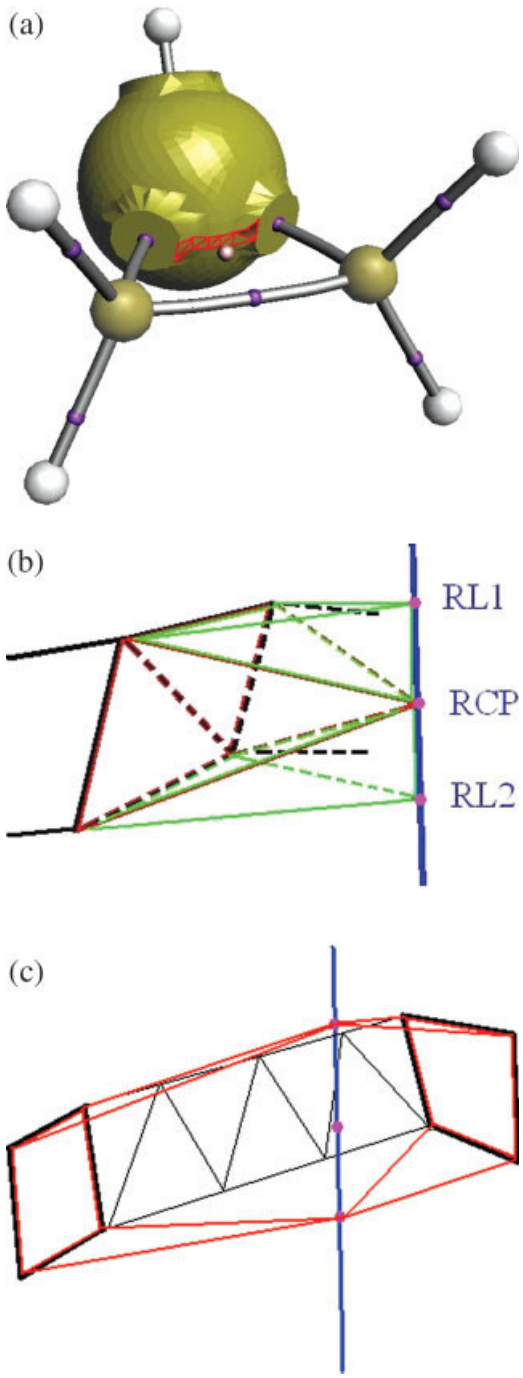


Figure 8. (a) Triangulated atomic basin (in yellow) of a carbon in cyclopropane at the growth level with an electronic density value closest to that of the ring critical point. The intersection of the triangulated object and the ring plane (i.e. the 'mouth', see text) is highlighted in red. Bond critical points are marked as purple spheres, the ring critical point as a pink sphere. The ring line and the bond paths are in blue and grey, respectively. The carbon nuclei and hydrogen nuclei are shown as large yellow and white spheres, respectively; (b) Schematic connection between the mouth's triangles (in black) and the ring line (in blue). The ring critical point (RCP) and the two initial points of the ring line (RL1 and RL2) are shown as pink dots. Each triangle of the mouth is connected to the RCP by means of a simplex (in red) and one side of this simplex is then connected to an initial point on the ring line with another simplex (in green); (c) Schematic connection between the mouth's quadrilaterals (in black) and the ring line (in blue). The ring critical point (middle) and the two initial points (up and bottom) of the ring line are shown as pink dots. Each quadrilateral is connected to the points of the ring line with a prism (in red).

surface area (PSA), a popular tool in drug design as it can predict the absorption of a molecule in the blood stream.[27] The equivalent 3D integration over an atomic basin Ω is defined in eq. (2),

$$F(\Omega) = \int_{\Omega} d\mathbf{r}\ f(\mathbf{r}) \tag{2}$$

In the current algorithm, we integrate over each element separately, and obtain the global properties $F(S)$ and $F(\Omega)$ as a sum over these element integrations.

The integration of properties requires a specific quadrature scheme and shape function for each type of element. The quadrature scheme defines the position of the quadrature points with local coordinates. We denote these local coordinates by (p, s) in 2D (used for triangles and quadrilaterals, symbols explained in the Appendix) and (p, s, t) in 3D (used for prisms and brick elements). Traditionally, these "local coordinates" are referred to as "natural coordinates" in the finite element literature. However, we keep using the term "local coordinates" because "natural coordinates" is already reserved to refer to the global coordinates describing the atomic basin (i.e. path length and angular variables θ and φ, see Introduction section).

The shape functions $\{N_i(p,s)\}$ in 2D or $\{N_i(p,s,t)\}$ in 3D are a collection of functions[28] defined at all vertices and midpoints (if needed), used to interpolate a property defined on all vertices. Equations (3) and (4) present these formulae for 2D and 3D elements, respectively,

$$f(x,y,z) = \sum_i N_i(p,s) F_i(x_i, y_i, z_i) \quad \text{in 2D} \tag{3}$$

$$f(x,y,z) = \sum_i N_i(p,s,t) F_i(x_i, y_i, z_i) \quad \text{in 3D} \tag{4}$$

where (p, s, t) are the local coordinates, N_i is the shape function associated with the vertex i, $F_i(x_i, y_i, z_i)$ is a property defined at vertex i, and $f(x, y, z)$ is the value of that same property at the (x, y, z) position corresponding to the local coordinates (p, s, t) for this particular element.

There is an important special case of eqs. (3) and (4). If F is taken to be the position of a vertex in the global axis system, then f is the global position of this point defined by local coordinates (p, s, t). Thus, the shape functions enable conversion of local coordinates to global ones via the position of the vertices and midpoints. Equations (3) and (4) then become,

$$\mathbf{r}(x,y,z) = \sum_i N_i(p,s) \mathbf{A}_i(x_i, y_i, z_i) \quad \text{in 2D} \tag{5}$$

$$\mathbf{r}(x,y,z) = \sum_i N_i(p,s,t) \mathbf{A}_i(x_i, y_i, z_i) \quad \text{in 3D} \tag{6}$$

where $\mathbf{A}_i$ is the position vector of the ith node.

To each local quadrature point (p, s, t) (also called node) corresponds a weight w_{pst}. These weights are linked to the integrals defined in eqs. (1) and (2) with the determinant of the Jacobian J_{2D} (in 2D) and J_{3D} (in 3D). Equation (7) shows the use of local quadrature and weight to integrate $f(\mathbf{r})$ over one 2D element S_m,

$$F(S_m) = \int_{S_m} dp\ ds\ f\big(\mathbf{r}(p,s)\big) \det J_{2D} \tag{7}$$

Replacing the integral with a numerical integration leads to eq. (8),

$$F(S_m) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} f\big(\mathbf{r}_{ij}\big) \det J_{2D}\big(p_i, s_j\big) w_{ij} \tag{8}$$

where n_i and n_j are the number of quadrature points for the coordinates p and s, w_{ij} and $J_{2D}(p_i, s_j)$ are the weight and Jacobian associated to the position (p_i, s_j).

Similarly, eq. (9) shows the integration over one 3D element Ω_m

$$F(\Omega_m) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} f\big(\mathbf{r}_{ijk}\big) \det J_{3D}\big(p_i, s_j, t_k\big) w_{ijk} \tag{9}$$

where n_i, n_j, and n_k are the number of quadrature points for the coordinates p, s, and t, w_{ijk} and $J_{3D}(p_i, s_j, t_k)$ are the weight and Jacobian associated to the position (p_i, s_j, t_k).

From the derivatives of the shape functions, we are able to compute the determinant of the Jacobian. First, we calculate the derivatives of the position vector $\mathbf{r}$ with respect to p, s, and t. For instance, the derivative of $\mathbf{r}$ with respect to p is shown in eq. (10), which is derived from eq. (6), and $\mathbf{A}_i$ is a constant with respect to p.

$$\frac{\partial \mathbf{r}(x,y,z)}{\partial p} = \sum_i \frac{\partial N_i(p,s,t)}{\partial p} \mathbf{A}_i(x_i, y_i, z_i) \tag{10}$$

For two-dimensional elements, the determinant of the Jacobian is then equal to the dot product between the two derivatives of $\mathbf{r}$:

$$\det J_{2D}(x,y,z) = \frac{\partial \mathbf{r}(x,y,z)}{\partial p} \cdot \frac{\partial \mathbf{r}(x,y,z)}{\partial s} \tag{11}$$

For three-dimensional elements, the determinant of the Jacobian is equal to the triple product between the three derivatives of $\mathbf{r}$:

$$\det J_{3D}(x,y,z) = \left[\frac{\partial \mathbf{r}(x,y,z)}{\partial p} \times \frac{\partial \mathbf{r}(x,y,z)}{\partial s}\right] \cdot \frac{\partial \mathbf{r}(x,y,z)}{\partial t} \tag{12}$$

Note that, depending on the orientation of vertices in the finite element, the determinant can be negative. One can avoid this orientation problem by multiplying det J by a correction coefficient (of -1 if needed) or by using the absolute value of the determinant. We chose the correction factor rather than the absolute value.

The quadrature scheme and the shape functions of a finite element will enable us to integrate this element. The integration therefore always follows the same scheme:

- Definition of the local quadrature points.
- Use of the shape functions to compute the global position of the point and the Jacobian at this point.
- Evaluation of the properties (to be integrated) at the global position.

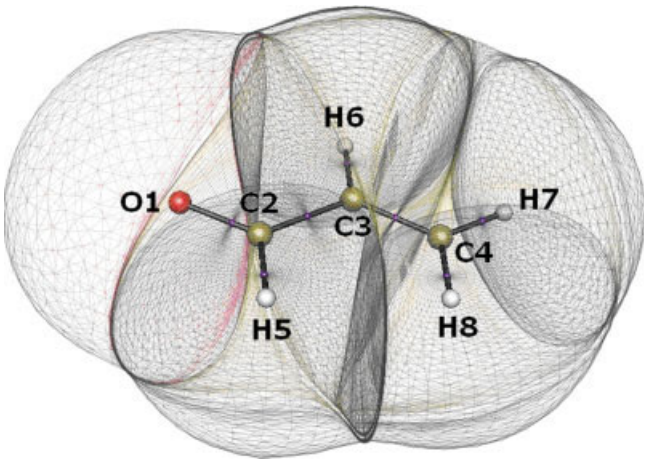


Figure 9. Meshing of the atomic basins in acrolein. Only the boundaries of the atoms (i.e. the outer growth level) are displayed. The red triangles are part of the oxygen basin, the yellow triangles are part of the carbon basins, and the gray triangles are part of the hydrogen basins. The IASs are represented in gray. The growth is stopped at $p = 10^{-3}$ au, which is the value of the outer envelope of constant electron density.

In the Appendix, we list the different quadrature and shape functions for the particular elements used in the algorithm. The number of quadrature points is decided as a function of ρ evaluated at the vertices. Finite elements in a region of low ρ can be integrated with less quadrature points without significant loss of accuracy. Details are given in the Appendix.

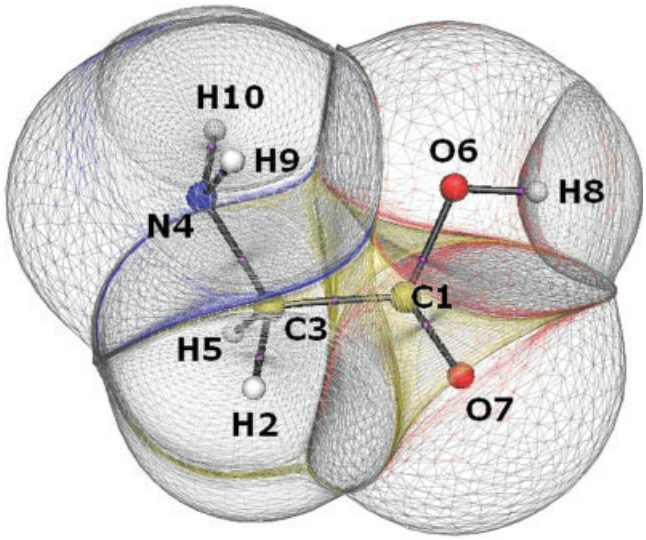


Figure 10. Meshing of the atomic basins in glycine. Only the boundaries of the atoms (i.e. the outer growth level) are displayed. The red triangles are part of the oxygen basin, the yellow triangles are part of the carbon basins, the gray triangles are part of the hydrogen basins, and the blue triangles are part of the nitrogen (blue sphere) basin. The IASs are represented in grey. The growth is stopped at $p = 10^{-3}$ au, which is the value of the outer envelope of constant electron density.

Results and Discussion

We illustrate the algorithm's performance on a four simple molecules: water, acrolein, glycine, and cyclopropane. The program GAUSSIAN98[29] generated the geometry-optimized wave function at B3LYP/6-311+G(2d,p)//HF/6-31G* for glycine, B3LYP/6-311+G(2d,p)//B3LYP/6-311+G(2d,p) for acrolein, HF/6-31G*//HF/6-31G* for water, and HF/6-31G//HF/6-31G for cyclopropane.

Meshing of the Basin

Figures 9 and 10 show the results of the meshing of acrolein and glycine with the current algorithm. To simplify the picture, the inside structure of the meshing is not displayed, just the outer growth level is displayed.

Two-Dimensional Integration

An IAS is a zero-flux surface because the gradient of the density, $\nabla\rho(\mathbf{r})$, never intersects this surface. The latter statement is formally expressed by

$$\nabla\rho(\mathbf{r}).\mathbf{n} = 0, \quad \text{which is valid for all points on the IAS} \qquad (13)$$

where $\mathbf{n}$ is the normal vector to the IAS. In practice, this condition is not perfectly met and hence the IAS introduces an error in the integration of the atomic properties. The flux through a finite area P is defined as a surface integral,

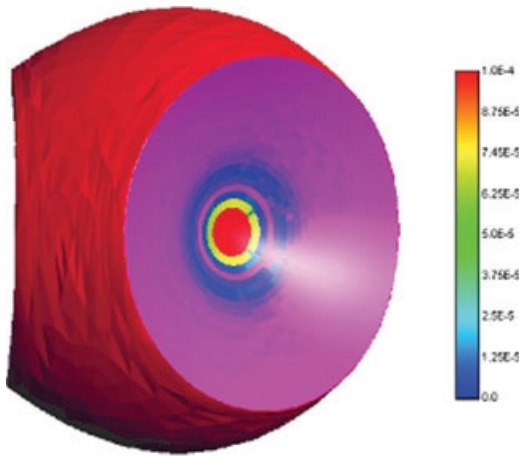


Figure 11. Ratio (in au) of flux over area of the 2D finite elements constituting the boundaries of the oxygen atom in water. Purple marks the lowest ratio. The atom is terminated at the isodensity value $\rho = 10^{-3}$ au.

Table 1. Integration of an IAS of Oxygen in Water via the Program AIM2000 and the Current Algorithm.

	AIM2000 ($\rho = 10^{-3}$ au)	Current algorithm ($\rho = 10^{-3}$ au)	Current algorithm ($\rho = 10^{-7}$ au)
IAS			
Flux	n/a	2.17×10^{-4}	-4.34×10^{-4}
Area	20.98	20.47	174.47
Population	0.7408	0.7607	1.5331
L[a]	−0.1398	−0.1281	−0.2773
Kinetic energy[b]	0.4144	0.4360	0.8650
Isodensity envelope			
Flux	n/a	-7.26×10^{-2}	-1.30×10^{-4}
Area	n/a	23.74	266.10

All properties are in atomic units.
[a] $L(\mathrm{IAS}) = -\frac{1}{4}\int_{\mathrm{IAS}} dS \nabla^2\rho(\mathbf{r})$.
[b] $E_{\mathrm{kin}}(\mathrm{IAS}) = -\frac{1}{2}\int_{\mathrm{IAS}} d\mathbf{r}\nabla^2\rho_1(\mathbf{r},\mathbf{r}')|_{\mathbf{r}=\mathbf{r}'} = \int_{\mathrm{IAS}} d\mathbf{r}K(\mathbf{r})$.

$$\mathrm{Flux}(P) = \int_P dS\ \nabla\rho \cdot n \tag{14}$$

The ability to visualize the flux of $\nabla\rho(\mathbf{r})$ through a surface is a unique feature of the current algorithm. Figure 11 shows this flux for each 2D finite element, normalized by the area of the element, for the boundaries of oxygen in water. From this picture, it is clear that the (total) flux through the IAS is largely due to the flux of the finite elements in the vicinity of the BCP.

Table 1 compares the integration results of the present algorithm with that of the program AIM2000 for the IAS and isodensity surface of oxygen in water. By default, AIM2000 terminates the surface at $\rho = 10^{-3}$ au. Because AIM2000 uses natural coordinates, it cannot compute the flux $\nabla\rho(\mathbf{r})$ through the surface. Table 1 shows that there is a reasonable agreement between the two methods.

It is appropriate to discuss, in this subsection, an important 3D quantity for it is linked to the flux. This quantity is denoted by $L(\Omega)$ and defined as,

$$L(\Omega) = -\frac{1}{4}\int_\Omega d\mathbf{r}\nabla^2\ \rho\mathbf{r} \tag{15}$$

has been used extensively in OCT to monitor the accuracy of the numerical integration of atomic basins.[30,31] The flux of $\nabla\rho(\mathbf{r})$ through the boundaries of the atom is directly linked to $L(\Omega)$ by means of the divergence theorem (or Green's theorem in space). When applied to $\nabla\rho(\mathbf{r})$, this theorem says that the sum of the flux over the surfaces bounding a given volume is equal to the volume integral of the Laplacian of $\rho(\mathbf{r})$, or specifically for a topological atom,

$$L(\Omega) = -\frac{1}{4}(\mathrm{Flux}(\mathrm{IAS}) + \mathrm{Flux}(\mathrm{Isodensity\ Envelope})) \tag{16}$$

The difference between the total flux of $\nabla\rho(\mathbf{r})$ and the 3D integration of $\nabla^2\rho(\mathbf{r})$ results from the error in the integration of the finite elements. Table 2 shows the difference between these two quantities. There are three factors that can influence the value of $L(\Omega)$: the approximate description of the IAS, the isodensity cutoff, and the integration process itself. Thus, even if IASs are perfectly represented as zero-flux surfaces and the algorithm is able to integrate without intrinsic error, $L(\Omega)$ would still have a nonvanishing value due to the fact that the atoms do not extend to infinity. This error will of course not appear if the atom is completely bounded by IASs, as in condensed matter. On a typical PC surface integrations take about 130 CPU seconds for hydrogens in acrolein or glycine, while the nonhydrogen atoms can take up to 450–600 CPU seconds. These timings have not been optimized at this stage of development, which is why we give only an idea of the upper limit. Again, the main strength of this algorithm is that it can generate tunably accurate integrations.

Three-Dimensional Integration

Table 3 compares the 3D integration over all atoms in water, acrolein, and glycine, by the current algorithm, the MORPHY01 algorithm with analytical IASs, and the natural coordinate algorithm of AIM2000. The precision of the integration can be measured by three different indicators: firstly, the value of $L(\Omega)$, which has to be as close to zero as possible; secondly, the total population of the molecule is compared to the total number of electrons; and finally the sum of the atomic kinetic energy is compared to the total Hartree–Fock energy. The grid used in MORPHY for water and acrolein is: $(n_r, n_\theta, n_\varphi) = (90, 40, 60)$ inside the β-sphere and $(n_r, n_\theta, n_\varphi) = (120, 60, 90)$ for the remaining part of the basin. All the atoms were terminated at $\rho = 10^{-7}$ au.

For water, the integration over the finite elements proves to be less accurate and more costly than that of MORPHY. Note that, from AIM2000's demo version, it was impossible to make a meaningful comparison between the number of quadrature points it used *versus* the number used by the current algorithm and that of MORPHY, AIM2000 counts the *total* number of points it evaluates the wave function at, whereas the current algorithm and MORPHY counted the actual quadrature points. All three accuracy indicators show that MORPHY is the most precise method. The IAS in water has a simple shape and can be efficiently fitted by the Chebyshev polynomials. For AIM2000, $L(\Omega)$ and the difference in population are of the same order of magnitude as MORPHY's and superior to the current algorithm. However, the integration of the kinetic energy by AIM2000 yields the largest error of the three methods.

Table 2. Difference Between the Total Flux of $\nabla\rho$(r) Through the Boundaries of Oxygen in Water and the 3D Integration of $\nabla^2\rho$(r) Over Its Volume (i.e. Atomic Basin), Calculated by the Current Algorithm.

	$\rho = 10^{-3}$ au	$\rho = 10^{-7}$ au
Total flux	−0.072392	-5.637×10^{-4}
Integration of $\nabla^2\rho(\mathbf{r})$	−0.072397	-5.042×10^{-4}
Difference	-4.93×10^{-6}	5.95×10^{-5}

Table 3. Integration Results for Atoms in Water, Acrolein, and Glycine.

	Current algorithm					Morphy				AIM2000			
	Quadrature	$L(\Omega)$	Population	Interpolation	E_{kin}	Quadrature	$L(\Omega)$	Population	E_{kin}	Quadrature	$L(\Omega)$	Population	E_{kin}
H_2O													
O1	1067355	7.9 E −04	9.04359	9.01648	−75.1676	415000	5.7 E −05	9.04305	−75.1667	8816752	1.8 E −04	9.04312	−75.1674
H2	730989	−3.1 E −04	0.47816	0.47892	−0.4089	415000	8.2 E −06	0.47846	−0.4093	1521232	7.0 E −06	0.47848	−0.4101
H3	730989	−3.1 E −04	0.47816	0.47890	−0.4089	415000	8.2 E −06	0.47846	−0.4093	1558078	7.0 E −06	0.47848	−0.4101
Total	*2529333*	*1.6 E −04*	*9.99991*	*9.97430*	*−75.9854*	*1245000*	*7.3 E −05*	*9.99998*	*−75.9854*	*11896062*	*1.9 E −04*	*10.00007*	*−75.9876*
Exact			10.00000	10.00000	−75.9854			10.00000	−75.9854			10.00000	−75.9854
Diff.			0.00009	0.02570	−0.1864			0.00002	0.0682			0.00007	5.8710
Acrolein													
O1	858633	2.0 E −04	9.07865	9.11178	−75.9176	864000	1.8 E −05	9.07850	−75.9173				
C2	1076743	1.1 E −05	5.02708	5.06744	−37.4759	864000	4.0 E −03	5.02418	−37.4770				
C3	1252263	5.1 E −05	6.04090	6.06244	−38.1190	864000	−1.3 E −03	6.04181	−38.1186				
C4	1178925	2.2 E −04	6.06665	6.08868	−38.0747	864000	1.7 E −04	6.06626	−38.0745				
H5	661893	−3.7 E −05	0.97561	0.98797	−0.6097	864000	6.5 E −06	0.97569	−0.6098				
H6	646365	−8.6 E −05	0.92859	0.93978	−0.5922	864000	6.3 E −06	0.92867	−0.5923				
H7	630357	−8.2 E −05	0.93693	0.94828	−0.5954	864000	7.8 E −06	0.93701	−0.5955				
H8	651311	−5.5 E −05	0.94568	0.95761	−0.5969	864000	5.8 E −06	0.94578	−0.5970				
Total	*6956490*	*2.1 E −04*	*30.00009*	*30.16398*	*−191.9815*	*6912000*	*2.9 E −03*	*29.99791*	*−191.9819*				
Exact			30.00000	30.00000	−191.9814			30.00000	−191.9814				
Diff.			0.00009	0.16398	0.2924			0.00209	1.3047				
Glycine													
C1	1561214	1.8 E −04	5.68773	5.70149	−37.8734	1647000	4.6 E −04	5.68686	−37.8731				
H2	652372	−1.6 E −04	0.94414	0.95445	−0.6102	1647000	5.5 E −06	0.94433	−0.6103				
C3	1261106	−3.0 E −04	4.43871	4.47300	−37.1129	1647000	1.4 E −04	4.43919	−37.1141				
N4	1158570	9.8 E −04	7.93945	7.97058	−55.0854	1647000	4.0 E −04	7.93846	−55.0845				
H5	653468	−1.6 E −04	0.94411	0.95440	−0.6101	1647000	5.5 E −06	0.94430	−0.6103				
O6	1062204	4.8 E −04	9.12493	9.15846	−75.8989	1647000	2.5 E −04	9.12436	−75.8983				
O7	852402	2.6 E −04	9.18201	9.21650	−75.9556	1647000	2.0 E −05	9.18167	−75.9549				
H8	518396	−2.2 E −04	0.41206	0.41897	−0.3740	1647000	6.1 E −06	0.41217	−0.3742				
H9	577700	−4.9 E −04	0.66336	0.67194	−0.5070	1647000	5.3 E −06	0.66369	−0.5075				
H10	580812	−4.9 E −04	0.66334	0.67189	−0.5070	1647000	5.3 E −06	0.66367	−0.5075				
Total	*8878244*	*8.3 E −05*	*39.99985*	*40.19167*	*−284.5345*	*16470000*	*1.3 E −03*	*39.99870*	*−284.5348*				
Exact			40.00000	40.00000	−284.5346			40.00000	−284.5346				
Diff.			0.00015	0.19167	0.1889			0.00130	0.5667				

n_{quad} is the number of quadrature points, "Population" is the integration of the electron density (au), and "E_{kin}" is the kinetic energy (au).
The "Exact" E_{kin} refers to the total HF energy of the molecule, and "Diff." is the difference between the total kinetic energy and the Hartree–Fock energy in kJ/mol.
The "Interpolation" column is the integration of the electron density using the shape functions to interpolate its value inside the finite elements.
The atoms are bounded at $\rho = 10^{-7}$ au.

Table 4. Integration Results for Atoms in Cyclopropane.

	n_{quad}	Population	$L(\Omega)$	E_{kin}
C1	1615156	5.99725	−2.28 E −04	−37.745
C2	1574945	5.99734	−2.05 E −04	−37.745
C3	1594470	5.99736	−2.00 E −04	−37.746
H4	1119131	1.00154	1.91 E −04	−0.627
H5	1117807	1.00154	1.91 E −04	−0.627
H6	1127851	1.00154	1.91 E −04	−0.627
H7	1124735	1.00154	1.91 E −04	−0.627
H8	1121415	1.00154	1.91 E −04	−0.627
H9	1120795	1.00154	1.91 E −04	−0.627
Total	11516305	24.00117	5.12 E −04	−116.998
Exact				−117.008
Diff. (kJ/mol)				27.329

n_{quad} is the number of quadrature points, "Population" is the integration of the electron density (au), and "E_{kin}" is the kinetic energy (au).
The "Exact" E_{kin} refers to the total HF energy of the molecule, and "Diff." is the difference between the total kinetic energy and the Hartree–Fock energy in kJ/mol.
The atoms are bounded at $\rho = 10^{-7}$ au.

Acrolein presents a more complicated topology. In particular, the *trigonal carbons are notoriously difficult to integrate* with high accuracy. The demo version of AIM2000 cannot process molecules with more than a few atoms. The current algorithm is able to integrate the carbons with great precision but MORPHY is still more accurate in the integration of the hydrogens and oxygen. For MORPHY, the poor integration of the carbons is reflected in the total population and kinetic energy, whereas remarkably for the current algorithm, these properties are of similar accuracy in acrolein and water. The total number of quadrature points is similar for the two methods but MORPHY uses the same number of points for all atoms, while in the current algorithm, the carbons are denser in quadrature points than the other atoms.

For glycine, we used the quadrature grid $(n_r, n_\theta, n_\varphi) = (120, 90, 140)$ outside the β-sphere. This finer grid improves the accuracy of the total population and total kinetic energy of the molecule by a factor 2. Although the current algorithm uses about half the number of quadrature points, it is still more precise than MORPHY in the total population by an order of magnitude and in the total energy by a factor 3. Nevertheless the $L(\Omega)$ values of individual atoms are always larger than those generated by MORPHY, except for C1.

In the "interpolation" column, the electron density is approximated using the shape functions defined in eq. (4) by replacing the property F_i by the electron density at the position of the vertices. Using this method, the integration of the atom is very fast but executed with low accuracy. The precision of the interpolation is linked to the size of the finite element; so, one can expect that with a finer grid this precision will improve. The downside is that the computational time required for the construction of the basin will also increase. If this method is to be used in the future, a balance should be struck between the time needed for the meshing of the basin and the accuracy of the interpolation (the time needed for the integration being negligible).

Finally, Table 4 lists the results for cycloproane, the molecule illustrating the performance of the algorithm in the case of a ring critical point. In spite of excellent $L(\Omega)$ values, there is still an excess of 0.001E in the population, which is surely due to the integration of the carbon atoms. Still, for most practical purposes, this accuracy is sufficient.

Inspection of Table 4 makes clear that the population of the carbons (and their $L(\Omega)$ values) are not perfectly symmetric. Unfortunately, at this stage of development, symmetry has not been looked into since the priority was to apply the algorithm on biomolecules, where symmetry is low or rare altogether. Of course, one could build in a point group detector and save CPU time by integrating only the minimum number of atoms required.

Although the main purpose of this article is to introduce the finite element approach into QCT and to demonstrate its capability to deliver very accurate integrations, we now give an idea of CPU timings. On a typical PC, the 3D integration of oxygen (O1) in acrolein took just over 3500 CPU seconds. When optimized, that is, after adjustment of various parameters, this figure could be brought down to 570 CPU seconds. Unoptimized CPU timings for C3 in glycine for example took 1400 CPU seconds. We expect that in a later stage of development these processing times will be further reduced.

Six-Dimensional Integration

We implemented the 6D integration of the Coulomb interaction energy between two atoms Ω_A and Ω_B, as defined in eq. (17),

$$E_{Coul}^{AB} = \int_{\Omega_A} d\mathbf{r}_A \int_{\Omega_B} d\mathbf{r}_B \frac{\rho_{tot}(\mathbf{r}_A)\rho_{tot}(\mathbf{r}_B)}{|\mathbf{R} + \mathbf{r}_B - \mathbf{r}_A|} \quad (17)$$

where $\mathbf{R}$ is the internuclear vector, $\mathbf{r}_A$ and $\mathbf{r}_B$ describe the respectively atoms' volume, and *total* charge density ρ_{tot} (which includes both the electron density and the nuclear density). Table 5 shows the results for acrolein. The main difference between MORPHY and the current algorithm lies in the interaction between the carbons and the oxygen. The accuracy of the two methods can only be judged with the error of the Hartree–Fock energy decomposition scheme.[32] However, the 6D integration of the exchange energy has not yet been implemented in the current algorithm. Note that the simplex algorithm is only used to obtain the integration points and weights. All pairs of grid points interact to give an exact 6D integration.

Table 5. Electrostatic Interaction Energy (kJ/mol) Between Selected Atoms in Acrolein, Obtained by 6D Integration via the Current Algorithm and via the Program MORPHY01.

	Current algorithm	MORPHY01	Difference
O1–C2	−1877.97	−1876.50	1.47
O1–C3	−4.48	−3.22	1.25
C3–C4	162.79	163.18	0.39
H6–H7	5.03	4.94	0.09
H7–H8	10.88	10.74	0.13

Conclusion

We explained in detail the features of a novel algorithm visualizing and integrating topological atoms, including those that border a ring critical point. Atoms are grown from their β-sphere, which needs to be modified by the advance awareness of the presence of bond and/or ring critical points. The growing atom then conquers the space of its basin by filling it with finite elements (prisms, bricks, simplices). Although the algorithm is complex in its full detailed implementation, it is built on clear and minimal principles. The dual relationship between topological objects is exploited. We presented some test cases for the 2D integration of the atomic boundaries, the 3D integration of the basin, and the 6D integration of the Coulomb interaction energy. Very accurate integration of trigonal carbons is now possible. Although the algorithm has been tested on a fair number of atoms, future fine-tuning will aim at making the method faster. We also believe that the current algorithm has already proven sufficient robustness. From the point of view of finite element methods, the most difficult part is the meshing of the objects. Since this is now accomplished, the exciting prospect of making further connections with existing (engineering and fluid dynamics) software is feasible.

Appendix

Quadratures

2D Integration

The first type of finite elements is linear triangles, which are used for both the rendering of the outer isodensity envelope and the initial circle around the BCP. The second type of elements is the quadratic quadrilateral used for the rendering of the IAS.

Linear Triangles. There are three different quadrature schemes:[33] 7 nodes (shown in Fig. A1), 3 nodes, or 1 node. Note that the number of nodes is always odd. For a linear triangle defined by three points i_1, i_2, and i_3, there are three shape functions corresponding to the three vertices: $N_1(p, s) = p$; $N_2(p, s) = s$; and $N_3(p, s) = 1\text{-}p\text{-}s$.

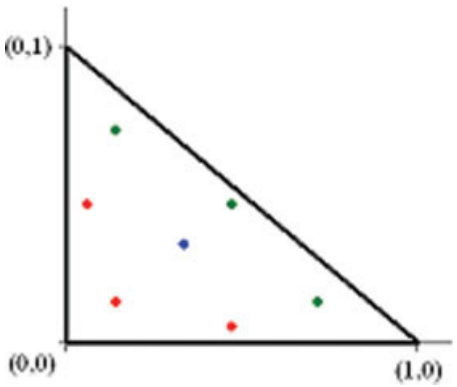
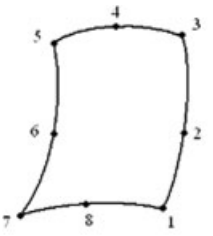


Figure A1. Spatial positions (colored points) of the quadrature points for a 7-node triangular role. The weight of the central node (in blue) is 9/40, the weights of the red points are equal to (6−15)/21, and the weights of the green points are equal to (6+15)/21.



$$N_1(p,s) = 1/4\ (1-p)\ (1-s)\ (-p-s-1)$$
$$N_3(p,s) = 1/4\ (1+p)\ (1-s)\ (p-s-1)$$
$$N_5(p,s) = 1/4\ (1+p)\ (1+s)\ (p+s-1)$$
$$N_7(p,s) = 1/4\ (1-p)\ (1+s)\ (-p+s-1)$$
$$N_2(p,s) = 1/2\ (1-p^2)\ (1-s)$$
$$N_4(p,s) = 1/2\ (1+p)\ (1-s^2)$$
$$N_6(p,s) = 1/2\ (1-p^2)\ (1+s)$$
$$N_8(p,s) = 1/2\ (1-p)\ (1-s^2)$$

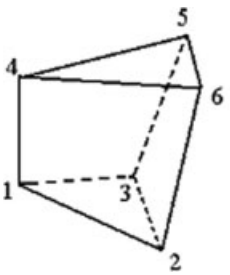
Figure A2. Shape functions corresponding to the vertices and midpoints in a quadratic quadrilateral.

Quadratic Quadrilaterals. The quadrature rules for quadrilaterals are easy to generate: they are simply the result of a product rule using two one-dimensional Gauss–Legendre quadrature schemes. They are also more flexible than their triangular counterpart because we can use any number of quadrature points. We chose to use three different rules depending on the electronic density at the vertices: 10 × 10, 5 × 5, and 2 × 2 nodes. Because of the quadratic nature of the quadrilaterals, the shape functions are more complicated.[34] They are listed in Figure A2.

3D Integration

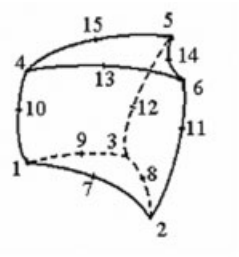
We now discuss the 3D integration of the atomic basin. First of all, we decided not to integrate the β-sphere with finite elements. In earlier versions of the algorithm, the β-sphere was triangulated with linear triangles. In this case, it is easy to fill it with simplices by connecting each triangle of the faceted sphere to the nucleus. However, tests demanded a huge amount of quadrature points for the integration of the simplices. Hence, we decided to define the β-sphere as a spherical object and use a 3D spherical Gauss–Legendre integration, which has proven to be very efficient in other programs.[23,35] However, this choice adds some complexity to the use of finite element. We have to use *quadratic* prisms to connect the β-sphere to the first growth level. Since the triangles in all the growth levels are defined as linear, the connection between these growth levels is achieved with linear prisms. The link between the prisms and the IAS is completed with quadratic brick elements.

Linear Prism. The quadrature points and their weights can be obtained from a product rule between a 2D triangular (with n_{tri}



$$N_1(p,s,t) = 1/2\ s\ (1-t)$$
$$N_2(p,s,t) = 1/2\ p\ (1-t)$$
$$N_3(p,s,t) = 1/2\ (1-p-s)\ (1-t)$$
$$N_4(p,s,t) = 1/2\ s\ (1+t)$$
$$N_5(p,s,t) = 1/2\ p\ (1+t)$$
$$N_6(p,s,t) = 1/2\ (1-p-s)\ (1+t)$$

Figure A3. Shape functions corresponding to the vertices of a linear prism.



$$N_i(p,s,t) = 1/2\, L_i\, [(1-t)(2L_i-1) - (1-t^2)] \quad i=1,2,3$$
$$N_i(p,s,t) = 1/2\, L_j\, [(1+t)(2L_j-1) - (1-t^2)] \quad i=4,5,6 \text{ and } j=i-3$$
$$N_7(p,s,t) = 2\, L_1\, L_2\, (1-t)$$
$$N_8(p,s,t) = 2\, L_2\, L_3\, (1-t)$$
$$N_9(p,s,t) = 2\, L_1\, L_3\, (1-t)$$
$$N_{10}(p,s,t) = L_1\, (1-t)\, (1+t)$$
$$N_{11}(p,s,t) = L_2\, (1-t)\, (1+t)$$
$$N_{12}(p,s,t) = L_3\, (1-t)\, (1+t)$$
$$N_{13}(p,s,t) = 2\, L_1\, L_2\, (1+t)$$
$$N_{14}(p,s,t) = 2\, L_2\, L_3\, (1+t)$$
$$N_{15}(p,s,t) = 2\, L_1\, L_3\, (1+t)$$

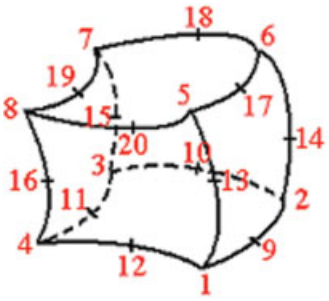
Figure A4. Shape functions corresponding to the vertices and midpoints of a quadratic prism with $L_1 = 1- p - s$, $L_2 = p$, $L_3 = s$.

points) and a 1D Gauss–Legendre (with n_h points) quadrature scheme. As a consequence, there are $n_{tri} \times n_h$ points in the prism, which we chose to be 7×10, 5×5, 1×2 points. The shape functions of a linear prism are detailed in Figure A3.

Quadratic Prism. The curved nature of the quadratic prism only affects its shape functions. The position and weight of the quadrature points are identical to the ones used for the linear prism. The shape functions are detailed in Figure A4.

Quadratic Brick Elements. The positions and weights of the quadrature points for a brick element can easily be obtained by a product rule between 3 one-dimensional Gauss–Legendre quadrature schemes. There are two different sets of points used by the algorithm: $3 \times 3 \times 5$ and $1 \times 1 \times 2$ points.

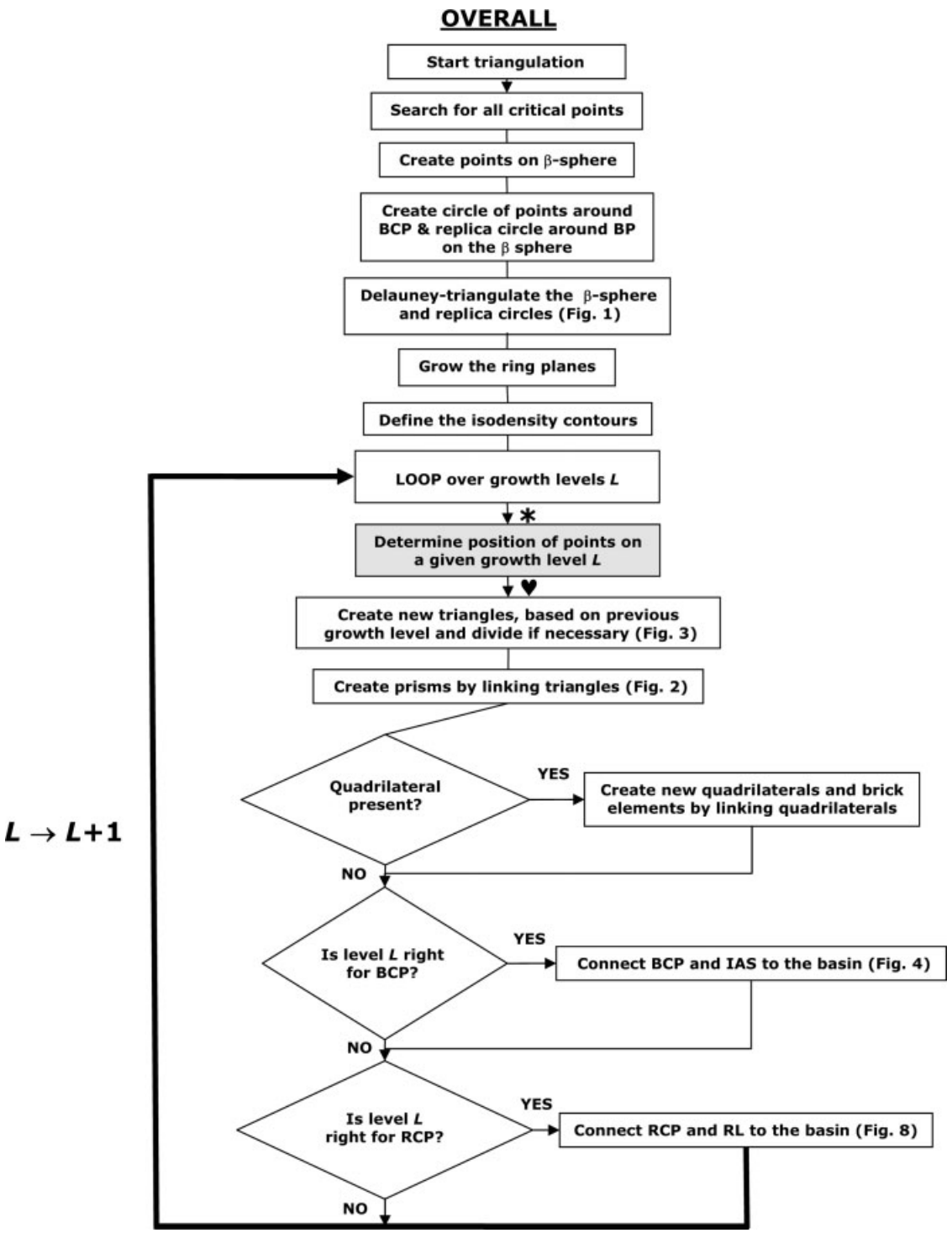
From the way we constructed the prisms and brick elements, the points of the bases are supposed to be on the same isodensity growth level. As a consequence, the density and (and other properties linked to the density) will be fairly constant on this base. This is the reason why we use less quadrature points on these dimensions compared to the height of the prisms and brick elements. The shape functions are detailed in Figure A5.



$$N_1(p,s,t) = 1/8\, (1-s)(1-t)(1-p)(-s-t-p-2)$$
$$N_2(p,s,t) = 1/8\, (1+s)(1-t)(1-p)(s-t-p-2)$$
$$N_3(p,s,t) = 1/8\, (1+s)(1+t)(1-p)(s+t-p-2)$$
$$N_4(p,s,t) = 1/8\, (1-s)(1+t)(1-p)(-s+t-p-2)$$
$$N_5(p,s,t) = 1/8\, (1-s)(1-t)(1+p)(-s-t+p-2)$$
$$N_6(p,s,t) = 1/8\, (1+s)(1-t)(1+p)(s-t+p-2)$$
$$N_7(p,s,t) = 1/8\, (1+s)(1+t)(1+p)(s+t+p-2)$$
$$N_8(p,s,t) = 1/8\, (1-s)(1+t)(1+p)(-s+t+p-2)$$
$$N_9(p,s,t) = 1/4\, (1-s^2)(1-t)(1-p)$$
$$N_{10}(p,s,t) = 1/4\, (1+s)(1-t^2)(1-p)$$
$$N_{11}(p,s,t) = 1/4\, (1-s^2)(1+t)(1-p)$$
$$N_{12}(p,s,t) = 1/4\, (1-s)(1-t^2)(1-p)$$
$$N_{13}(p,s,t) = 1/4\, (1-s)(1-t)(1-p^2)$$
$$N_{14}(p,s,t) = 1/4\, (1+s)(1-t)(1-p^2)$$
$$N_{15}(p,s,t) = 1/4\, (1+s)(1+t)(1-p^2)$$
$$N_{16}(p,s,t) = 1/4\, (1-s)(1+t)(1-p^2)$$
$$N_{17}(p,s,t) = 1/4\, (1-s^2)(1-t)(1+p)$$
$$N_{18}(p,s,t) = 1/4\, (1+s)(1-t^2)(1+p)$$
$$N_{19}(p,s,t) = 1/4\, (1-s^2)(1+t)(1+p)$$
$$N_{20}(p,s,t) = 1/4\, (1-s)(1-t^2)(1+p)$$

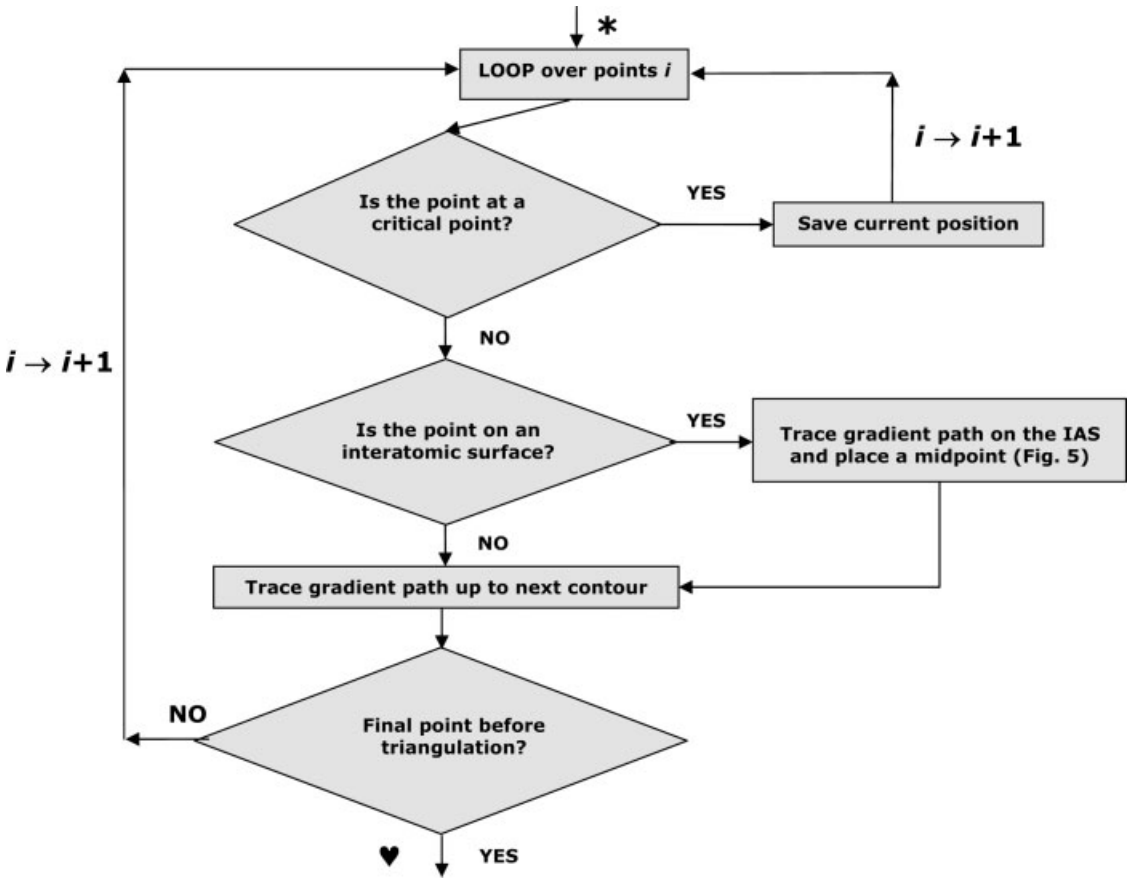
Figure A5. Shape functions corresponding to the vertices and midpoints of a quadratic brick element.

Flowcharts



The gray box in the flowchart above is amplified in the flowchart that follows. The symbols * and ♥ mark the connection between the flowcharts.

Determine Position of Points at a Given Growth Level



References

1. Bader, R. F. W. Atoms in Molecules; Clarendon Press: Oxford, 1990.
2. Popelier, P. L. A. Atoms in Molecules. An Introduction; Pearson: London, Great Britain, 2000.
3. Popelier, P. L. A.; Smith, P. J. In Chemical Modelling: Applications and Theory, Vol. 2, Royal Society of Chemistry Specialist Periodical Report; Hinchliffe, A., Ed.; Royal Society of Chemistry: Cambridge, UK, 2002; pp. 391–448, Ch. 8.
4. Popelier, P. L. A.; Aicken, F. M.; O'Brien, S. E. In Chemical Modelling: Applications and Theory, Vol. 1, Royal Society of Chemistry Specialist Periodical Report; Hinchliffe, A., Ed.; Royal Society of Chemistry: Cambridge, UK, 2000; pp. 143–198, Ch. 3.
5. Becke, A. D.; Edgecombe, K. E. J Chem Phys 1990, 92, 5397.
6. Silvi, B.; Savin, A. Nature (London) 1994, 371, 683.
7. Malcolm, N. O. J.; Popelier, P. L. A. J Comput Chem 2002, 24, 437.
8. Biegler-Koenig, F. W.; Nguyen-Dang, T. T.; Tal, Y.; Bader, R. F. W.; Duke, A. J. J Phys B 1981, 14, 2739.
9. Biegler-Koenig, F. W.; Bader, R. F. W.; Tang, T. H. J Comput Chem 1981, 3, 317.
10. Breneman, C. M.; Rhem, M. Program FASTINT, Rensselaer Polytechnic, New York, USA, 1992.
11. Volkov, A.; Gatti, C.; Abramov, Y.; Coppens, P. Acta Crystallogr A 2000, 56, 252.
12. Bianchi, R.; Forni, A. J App Cryst 2005, 38, 232.
13. Stash, A.; Tsirelson, V. J Appl Cryst 2002, 35, 371.
14. Biegler-Koenig, F. W.; Schonbohm, J.; Bayles, D. J Comput Chem 2001, 22, 545.
15. Biegler-Koenig, F. W.; Schonbohm, J. J Comput Chem 2002, 23, 1489.
16. Pendás, A. M.; Luaña, V. Program PROMOLDEN, University of Oviedo, 2002. Available at http://web.uniovi.es/qcg/d-DensEl/.
17. Popelier, P. L. A. Comput Phys Commun 1998, 93, 212.
18. Katan, C.; Rabiller, P.; Lecomte, C.; Guezo, M.; Oisona, V.; Souhassou, M. J Appl Cryst 2003, 36, 65.
19. Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. Numerical Recipes; Cambridge University Press: Cambridge, UK, 1994.
20. Malcolm, N. O. J.; Popelier, P. L. A. J Comput Chem 2002, 24, 1276.
21. Noury, S.; Krokidis, X.; Fuster, F.; Silvi, B. Comput Chem 1999, 23, 597.
22. Savin, A.; Nesper, R.; Wengert, S.; Fassler, T. F. Angew Chem Int Ed Engl 1997, 36, 1808.
23. Popelier, P. L. A. Mol Phys 1996, 87, 1169.
24. Joe, B. Adv Eng Software 1991, 13, 325.
25. Rafat, M.; Devereux, M.; Popelier, P. L. A. J Mol Graph Model 2005, 24, 111.
26. Popelier, P. L. A. Comput Phys Commun 1998, 108, 180.
27. Clark, D. E.; Pickett, S. D. Drug Disc Today 2000, 5, 49.

28. Zienkiewicz, O. C. Finite Element Method; McGraw Hill; London, 1977.
29. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery, Jr J. A.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudirn, K. N.; Strain, M. C.; Farkas, O.; Tomasi, J.; Barone, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Baboul, A. G.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Andres, J. L.; Gonzalez, C.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A. Gaussian 98, Revision A7. Gaussian, Inc, Pittsburgh PA, USA, 1998.
30. Popelier, P. L. A. Mol Phys 1996, 87, 1169.
31. Aicken, F. M.; Popelier, P. L. A. Can J Chem 2000, 78, 415.
32. Rafat, M.; Popelier, P. L. A. J Comp Chem 2007, 28, 292.
33. Stroud, A. H. Approximate Calculation of Multiple integrals; Prentice-Hall Englewood Cliffs, New Jersey, USA, 1971.
34. Gartling, D. K.; Hogan, R. E. COYOTE—A Finite Element Computer Program for Nonlinear Heat Conduction Problems, Part I: Theoretical Background; Sandia National Laboratories: Albuquerque, New Mexico, USA, 1994.
35. Biegler-Koenig, F. W. J Comput Chem 2000, 21, 1040.