Promising insights in parallel grid-based algorithms for quantum chemical topology

Hilaire Chevreau, Julien Pilmé

Promising Insights in Parallel Grid-based Algorithms for Quantum Chemical Topology

Hilaire Chevreau [1]*and Julien Pilmé [1] *

[1] Sorbonne Université, CNRS, Laboratoire de Chimie Théorique CC 137 – 4, place Jussieu F. 75252 PARIS CEDEX 05 – France.
* Correspondence: chevreau@lct.jussieu.fr, julien.pilme@sorbonne-universite.fr

Abstract: Some straightforward improvements designed to make more efficient and fast the grid-based quantum chemical topology are presented. The strategy focuses on both the evaluation of the scalar function over three-dimensional discrete grids and the algorithms aimed to follow and integrate gradient trajectories over the basin volumes together. Beyond the density analysis, we show that the scheme is quite suitable for the electron localization function (ELF) and its complex topology. With the speeding-up of the parellelized process used to generate 3d-grids, the proposed scheme makes several orders of magnitude faster than the computing grid-based method previously implemented in our laboratory [*Computers & Chemistry* 1999, 23, (6), 597-604]. The efficiency of our implementation was also compared with well-known grid-based algorithms designed to assign the grid points to basins. The performances, speed vs. accuracy have been discussed on the basis of results obtained from selected illustrative examples.

Keywords: QCT; QTAIM; ELF; Based-Grid Method;

1. Introduction

The quantum chemical topology (QCT) has gained popularity as a bonding analysis method in molecules or solids since the seminal papers published about the quantum theory of atoms-in-molecules (QTAIM) and the topological analysis of the electron localization function (ELF). [1-5] In a QCT analysis, a partitioning of the molecular space into volumes (so-called basins) is carried out according to the theory of dynamical gradient systems through an assignment of a scalar function of the molecular space. More precisely, the process splits the space into regions associated to maxima of the function by studying the gradient trajectories (or integral lines). Note that the term of topology is used in its broad sense related to the existence of the stationnary points of the scalar field and their connectivity insured by the gradient trajectories. The most important QCT algorithm is the one that performs this assignment to basins localized around the maxima of the considered scalar function. Beforehand, in a grid-based approach, the function values have to be computed at each grid point. The basin volumes are separated by surfaces (so-called separatrices) according to a algorithm typically combining an analytical and a numerical methodology. In the QTAIM framework, the function considered is the one electron density and these basins are associated with each of the interacting atoms in the molecule. In contrast, the ELF topology exhibits non-atomic valence basins (bonding and non-bonding regions) in addition to core basins surrounding nuclei with atomic number $Z > 2$. The valence basins are characterized by the number of core basins with which they share a common boundary. This number is called the synaptic order [6] and are usually defined as following : monosynaptic basins (labeled V(A)) usually correspond to lone pair regions, while disynaptic and polysynaptic basins (labeled V(A, B, C, ..)) characterize the covalent bonds.

Since the pionner works of Bader and its coworkers until today, large efforts have been made to propound a large variety of efficient algorithms designed to the QTAIM analysis whether for molecular or condensed-matter systems.[7-21] Some works using an appropriate vectorization and parallelization[21] have been proposed, even for a GPU (graphics procession unit) implementation.[22] The X-ray crystallography community has also proposed some innovative methods.[23,24]

Although it is not always feasible to apply grid-based algorithms to large systems where a very large computational cost can be required, the grid-based methods were employed from the early days.[9,13,15,17,25-27]. The key of such strategy is to prevent some numerical difficulties during the grid points assigment with an easy implementation. Thus, the grid-based methods have evoked a renewed interest for uniform as well as for non-uniform grids.[8,11,21,28]. Since that time, much effort has been devoted to QTAIM developments, a rather small number of works were devoted to the ELF topological analysis which is more complex than QTAIM since typically non-atomic basins need to be characterized and possible degenerated attractors can be also found. [15,27-29]

In this article, we focus on an improved implementation of the off-grid-based algorithm which historically is the choice of both the QTAIM and ELF implementation in the own TopMod09 package. Our choice is to highlight the topological analysis of the ELF function because of its complex topology. We have especially focus the improvements of the methodology on the two following points :

(i) Efficiency to make 3d-grid in order to decrease the computational effort without comprimising the accuracy of the integrations processing.

(ii) Vectorization and parallelization of algorithms using OpenMP technology in order to follow the steepest ascent gradient paths of the scalar function.

The diagram flows of used grid-based algorithms and the new developpements are presented in sections 3 and 4. The overall organization of our programs is built in the same spirit as the TopMod09 package.[27] The QCT grid method is organized according to the three following steps :

1. Evaluation of the density function over a rectangular parallelepipedic uniform grid of dimension $N = N_x \times N_y \times N_z$ where each grid point has a constant spacing with its neighbours.

2. Assignment of each grid point to basins.

3. Integration of the electron charge density over the basins, and calculation of related integrated properties, the most commonly being the basin populations.

The last section contains some comparative tests speed vs. accuracy of our implementation with well known algorithms used for the QTAIM analysis, the Bader Charge near-grid algorithm of the Henkelman's group[13,25,26] and the Keith's promega algorithm implemented in the AIMALL package[12,30] applied to two selected molecular systems. Some tests have been also performed from the popular Multiwfn [31]. Note that the Yu and Trinkle weighted-grid algorithm[9] was also implemented in the code Bader charge analysis of Henkelman's group. Theses latter hinted programs are typical examples of well-maintained codes which perform the QCT analysis from gaussian (slater) molecular wavefunction data. These data are gathered in a wfn file or grid cube file format. The wfn format, intially introduced in the Bader's program AIMPAC, is currently supported by numerous quantum chemistry softwares, such as Gaussian, GAMESS-US, Q-Chem or NWChem.

2. Computational details.

Geometry optimizations have been performed at the hybrid density functional B3LYP level with the Gaussian 2009 software.[32] The standard all-electron basis set 6-31+G(d,p) was employed for all atoms. To illustrate the applicability of our implementation, we present a set of results using a standard uniform 3d-grid for both electron density and ELF. The grid step varying between 0.075 bohr and 0.1 bohr was used. This range is known as a standard quality step and already gives fine grids, tailored to semi-numerical based partitions algorithms. The grid-based corrected algorithms have been implemented within an OpenMP technology parallelisation scheme where a shared-memory parallel model is used. The grid-based parallelization scheme and its implementation used in this work is similar to that described by Rodriguez et al.[8] Briefly, it relies on the fact that the assignement for a set of grid points remains independent of the assignement of any other set. Thus, sets of grid points to be automatically assigned to a basin are distributed to several processors using the OpenMP technology.

The implementation was carried out in the TopChem2 program package, written in Fortran 90 compiled by the open-source GNU Fortran compiler.[15] TopChem2 was optimized for GNU Fortran compilers and capable of using all available CPU cores, which can significantly reduce the computational time. A MPI implementation, termed TopModMPI, is also currently under developments.

3. Theory and Computational Strategy

3.1. The 3d- grid acceleration process

A scalar function $\mathbb{R}^3 \rightarrow \mathbb{R}$ can be numerically represented by its values distributed on a suitable uniform grid of N points with a constant spacing between the point which preserves the function properties in the neighborhood of a given point. The complexity of grid-based algorithms to assign points to basin volumes stronlgy depends on the total number of grid points (grid size) N but also the number of involved primitives functions M. The number of operations is roughly proportional to N x M x M. The challenge is always then to decrease as much as possible, the computational effort in the innermost loops. Among the pionnering works, the Silvi's group has proposed to take advantage of the factorization of the Cartesian Gaussian functions in external short loops which has enabled to decrease the heavy CPU-consuming.[27] Another improvement was to parallelise the loops within an OpenMP or MPI environments. However, the computational effort remains rather expensive and strongly depends to the total number of primitives M. Prior to a detailed examination of our method, we first recall how the electron density and ELF are typically related to M and are computed at a given grid point **r**.

Using the variational principle under the only constraints of the single or double occupation for each spatial orbital $\varphi_i(\mathbf{r})$ (Hartree-Fock or Kohn-Sham formalism), we recall that the electron density is computed at each grid point as:

$$\rho(\mathbf{r}) = \sum_{i}^{occ} n_i \, \varphi_i^*(\boldsymbol{r}) \, \varphi_i(\mathbf{r}) = \sum_{\mu=1}^{M} \sum_{\upsilon=\mu}^{M} P_{\mu\upsilon} \, \chi_\mu(\mathbf{r}) \, \chi_\upsilon(\mathbf{r}) \qquad (1)$$

where $\varphi_i(\mathbf{r})$ are expanded using M atom-centered gaussian basis functions, $\chi_\mu(\mathbf{r})$(GTO), and $P_{\mu\nu}$ are elements of the total one-electron density matrix defined as follows:

$$P_{\mu\upsilon} = \sum_{i}^{occ} n_i c_{i\mu} (c_{i\upsilon})^*$$

n_i is the occupation of the $\varphi_i(\mathbf{r})$ orbital and $c_{i\mu}$ and $c_{i\nu}$ are the real expansion coefficients. ELF was expressed in the framework of the DFT theory by Savin et al and rationalized in terms of the local excess kinetic energy due to the Pauli repulsion. ELF can be computed at each grid point as follows:

$$\eta(\boldsymbol{r}) = \left[1 + \left(\frac{T_s(\boldsymbol{r}) - T_w(\boldsymbol{r})}{\frac{3}{10}(3\pi^2)^{2/3}\rho(\boldsymbol{r})^{5/3}}\right)^2\right]^{-1} \quad (2)$$

Where,

$$T_s(\boldsymbol{r}) = \frac{1}{2}\sum_{\mu=1}^{M}\sum_{\upsilon=1}^{M} P_{\mu\upsilon}\left(\frac{\partial\chi_\mu(\boldsymbol{r})}{\partial x}\frac{\partial\chi_\upsilon(\boldsymbol{r})}{\partial x} + \frac{\partial\chi_\mu(\boldsymbol{r})}{\partial y}\frac{\partial\chi_\upsilon(\boldsymbol{r})}{\partial y} + \frac{\partial\chi_\mu(\boldsymbol{r})}{\partial z}\frac{\partial\chi_\upsilon(\boldsymbol{r})}{\partial z}\right)$$

is the kinetic energy density. $T_w(\boldsymbol{r}) = \frac{1}{8}\frac{|\nabla\rho(\boldsymbol{r})|^2}{\rho(\boldsymbol{r})}$ is the von Weizsäcker kinetic energy density. Note that from a rigorous point of view, this expression of ELF is only valid for closed-shell systems described by a single determinant.

The quantum chemistry programs often employ cartesian atomic orbitals in the algorithms for computational reasons.[33] The gaussian molecular wavefunction data are typically provided as a set of unnormalized cartesian primitives, χ_μ. A normalized cartesian GTO χ_μ^N can be easily defined from its unnormalized form as :

$$\langle\chi_\mu^N|\chi_\mu^N\rangle = \left(\frac{2\alpha}{\pi}\right)^{3/2}\left[\frac{(4\,\alpha)^{i+j+k}}{(2\,i-1)!!\;(2\,j-1)!!\,(2\,k-1)!!}\right]\langle\chi_\mu|\chi_\mu\rangle = 1$$

where the GTO is defined as $\chi_\mu = (x - X_\mu)^i\,(y - Y_\mu)^j(z - Z_\mu)^k\,e^{-\alpha\,|\boldsymbol{r}-\boldsymbol{R}_\mu|^2}$ with (x,y,z) is the relative cartesian position of the electron to the atom-centered origin of the GTO $\boldsymbol{R}_\mu = (X_\mu, Y_\mu, Z_\mu)$. Now, we look for the minimal radius r_μ of a β-sphere centered at the $\boldsymbol{R}_\mu$ location. As a reminder, a β-sphere is expressed as a spherical volume that is enclosed within the basin volume at whose attractor it is centered.[34] By analogy, the β-sphere with a radius r_μ is here defined as the lowest spherical volume at whose the primitive χ_μ^N it is centered such as:

$$\int_0^{r_\mu} |\chi_\mu^N|^2\; d\boldsymbol{r} > 1 - \varepsilon \quad (3)$$

Where ε is an arbitrary threshold. Practically, a value of 0.001 has been used in this work. As shown by both equations (1) and (2), the computational effort depends of the number of primitives locally involved at a given grid point for the computing of density functions $\varrho(\mathbf{r})$ or $\eta(\mathbf{r})$. We notice that the contribution of a given primitive varies according to the distance between the grid point and the location of its atom center. For the distances upper than r_μ, the contribution of the primitive becomes negligible and can be smoothly negleted from the calculation of one-density functions. The efficiency and the robustness of this straightforward method is briefly illustrated in Table 1. Table 1 shows that more often than not a large number of primitives can be disregarded in order to compute the density functions at most of grid points. For example, that is the case of the benzene molecule where we found that only a maximum of 50 primitives are needed at 80% of grid points. Some tests, speed vs. accuracy, of our implementation are discussed below in the *Results and Discussion* section.

Number of primitives kept per grid point	*0-50*	*50-100*	*100-150*	*> 150*
C_6H_6 (M=234 total primitives)	*78.8%*	*18.9%*	*2.3%*	*0.0%*
CG complex (M=678 total primitives)	*44.6%*	*33.6%*	*15.8%*	*6.0%*

Table 1. Purcentage of grid points where the number of retained primitives is lower than the total number of primitives. These numbers corresponds to a grid built for the benzene molecule and for

the guanine (G) –cytosine (C) complex optimized at the B3LYP/6-31+G(d,p) level of theory. An intermediate grid step of 0.1 bohr was used.

3.2. The numerical grid-based strategy for the basin analysis

3.2.1. ***The improved off-grid algorithm of the Silvi's group***. Within quantum chemical topology, basins with any shapes need to be determined in the molecular space. In the grid-based methods, the integration of trajectories is carried-out without any explicit build of the zero-flux surfaces (separatrices) and the algorithm looks for the assignment of a given grid point belong the steepest gradient paths leading to the corresponding attractor. Interestingly, the process can be easily vectorized and parallelized. [8] In this paper, we use a parallellized and revisited version of the historically grid-based Silvi's algorithm initially implemented in the TopMod09 package.[27] This algorithm will termed as off-grid algorithm throughout the paper.

The algorithm starts with an initial part designed to strongly reduce the length of the followed trajectories and also the number of gradients calculations. Most of grid points near to attractors or points so far from the atoms can be assigned using the nearest neighbors points if it encountered a previously assigned grid point (wandering point). The trajectory integration is then making this approach scales linearly with the system size. Therefater, the off-grid gradient ascent needs only to be performed around separatices (border points). Here, we improve the convergence using a well-known adaptative gradient ascent (AdaGrad) widely documented in the litterature.[35] The scheme 1 details the implemented gradient ascent. The computational cost also has been reduced because the assigment of the current point (x, y, z) is only evaluated after a small empirical jump (termed *Dcomp* and holded to 0.4 bohr) from the starting position (x_0, y_0, z_0). Each assigment of the new point (x, y, z) along the trajectory is then evaluated by looking at the nearest neighbors. If all of these latter are already assigned to a same basin, the initial grid point (x_0, y_0, z_0) is then assigned to this basin.

Input *: starting from an unassigned grid point (i_0, j_0, k_0)*
Output *: attractor code, grid assignement code (integer) of grid point (i_0, j_0, k_0)*

Compute cartesian coordinates (x_0,y_0,z_0) of the starting grid point (i_0, j_0, k_0)
step ← *0.1 bohr* # initial step
(x, y, z) ← *(x_0,y_0, z_0)*
ascent ← *True*
While ascent :
    compute gradient g_k components of the scalar function
    (x ,y ,z) ← *(x, y ,z)* + $\frac{step \cdot g_k}{\sqrt{e + \Sigma_k g_k^2}}$
    # update *(x ,y ,z)* coordinates along the path. The g_k^2 are the squared gradient components and e is a smoothing term (10^{-8}) that avoids division by zero
    D ← *distance(x_0, y_0, z_0, x, y ,z)*
    # distance between the current point (x,y,z) and (x_0, y_0, z_0)
    If D > Dcomp :
        (i, j, k) ← *near-grid check (x, y, z)*
        if rho_attr(i ,j, k) > 0 : # 0 means an unassigned point
            attr ← *rho_attr(i, j, k)*
            ascent ← *False*

Scheme 1. Flow diagram of the off-grid gradient ascent part : *ascent rho silvi (i_0, j_0, k_0, attractor code)*. Example for QTAIM.

As mentioned at the beginning of this section, the length of the followed gradient path can be strongly reduced when the grid points are firstly considered in the decreasing order of the scalar function (electron density or ELF), since the trajectory crosses an already attributed point. Now, consider the set of all neighboring grid points (ii, jj, kk) from a given grid point (i, j, k) having a value of the density or ELF higher than the fisrt one at (i, j, k), the ascent path will follow necessarily though or near one of its (ii, jj, ,kk) grid points. Therefore, if all of these neighbors are already assigned to the same maxima, then we are assured that the point (i, j, k) can be directly assigned without considering any gradient ascent process, focusing the computational effort on grid points close to the basin boundaries. This improvement avoid many gradient calculations. The overall scheme including the gradient ascent part can be summarized in the following flow, which is the one implemented :

```
Compute electron density ρ(i, j, k) on the grid point (i, j, k)
L ← heap sort ρ(i, j, k) in decreasing order
for all points (i, j, k) in L :
    consider neighbors (ii, jj, kk) with ρ(ii, jj, kk) > ρ(i, j, k)
    if all are attributed to the same attractor then
        rho attr(i ,j ,k) ← attractor code
    else
        ascent rho silvi (i ,j ,k, attractor code) # off-grid gradient ascent part
        rho attr(i,j,k) ← attractor code
    end
end
```

Scheme 2. Overall flow diagram of the Silvi's algorithm: example for QTAIM

3.2.2 The on-grid and the corrected near-grid algorithms of the Henkelmam's group. A relevant grid-based method was proposed by Henkelman and its coworkers.[13,25]. In this approach, the ascent on-grid projected gradient works by integrating gradient-ascent trajectories from every grid point. Thereafter, Tang et al [26] have improved the original algorithm with a near-grid method which reduce the on-grid lattice bias, a correction vector between the off-grid trajectory and the on-grid trajectory being propagated. The process finds all basins with a very fast computational time. However, this algorithm rather depends on the grid spacing and it requires, in principle, a finely sampled grid which could limit, in principle, the application to large systems but our acceleration process described above enabled a such applications. It was implemented is several package owing to its ease of implementation and relatively good performances.[15,31]. In the next section, its performances are compared with those of the Silvi's algorihtm. Note that the Yu and Trinkle algorithm [9] was implemented in the code Bader charge analysis[25] The method introduced a subgrid accuracy in the computation of QTAIM volumes. They implemented a weighting scheme to handle the fraction of an elementary grid volume to be connected with a neighboring grid point. The Henkelman and Tang algorithms will respectively termed as on-grid and near-grid throughout the paper.

3.2.3 **Mixing the initial part of the Silvi's algorithm and the near-grid algorithm.** The near-grid algorithm of Tang [26] is very fast owing to the use of on-grid projected gradients. However, the process typically requires a fine grid spacing for a full relevant connection between near grid points. While noting that the initial part of the Silvi's algorithm is very efficiency and fast, we can suggest to combine this latter process with the near-grid algorithm namely we consider that the wandering points can be firstly attributed without considering any gradient path, focusing the computational effort on border grid points close to the basin boundaries. This algorithm will be termed as the mixed-near-grid algorithm throughout our paper. Practically, this *mixed-near-grid* lead us

to only replace in Scheme 2, the line '*ascent rho silvi (i ,j ,k, attractor code)'* by '*ascent near-grid (i ,j, k, attractor code)'*. We expect the basins populations might be slightly different with respect to the near-grid algorithm while maintening a rather fast assignment of grid points to basins. These grid-based methods detailed above have been evaluated and their time efficiency as well as their robustness. All these results are presented in the next section.

4. Results and Discussion

4.1 The 3d- grid acceleration process

Figures 1 and 2 depict the comparative performances to make the 3d-grid without or with the accelerated process applied to two selected systems, namely the benzene (18 atoms) and the guanine (G) –cytosine (C) complex (30 atoms). The electron density and ELF values have been distributed on an uniform grid of N points with an intermediate spacing of 0.1 bohr between the point. Here, N = 5229015 for the benzene and N = 10515021 for CG complex. As shown in Figures 1 and 2, an obvious reduction of the computational effort is clearly observed owing to the introduction of our acceleration process for both QTAIM and ELF. For example on Figure 2, the accelerated process (green curves) using a single core approximatively makes four times faster the build of the entire grid than without it (red curves). The average efficiency of the combined acceleration process and a modest parallelization (16 processors) becomes approximatively 60 times faster than the single core calculation without the acceleration. For example, the calculation ELF grid approximatively requires 1800 seconds using a single core (red curve) whereas this latter calculation only requires 30 seconds using the acceleration process combined with 16 CPU cores. Note that TopMod09 (the blue square on Figures 1 and 2), only implemented for a single core calculation, remains very efficient in this context. Although the calculation of the 3d-grid remains a rate-limiting step of a QCT analysis, the reduction of the computational cost obtained with our acceleration process makes this possible for applications to enough large systems where large grids need to be produced.

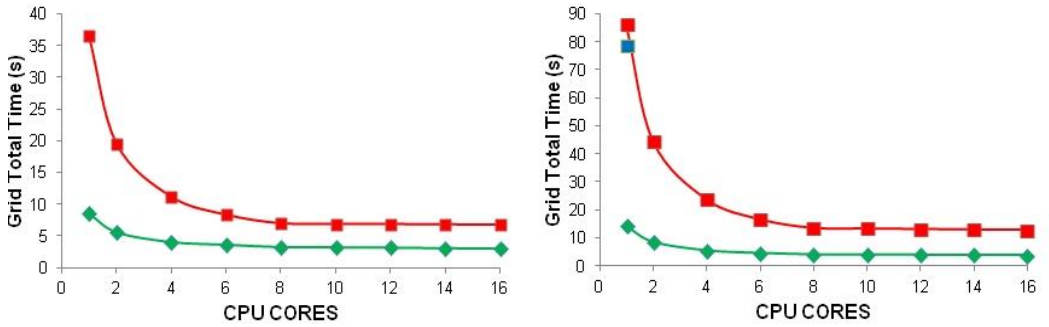

Figure 1. Total time, in s, elapsed for the 3d-grid computing of the electron density (left) and ELF (right) applied to the benzene molecule computed at the B3LYP/6-31+G(d,p) level of theory. In this case, the β-radius r_μ was compeled to be lower than 7.5 bohr. Color Code: **Green**: using grid acceleration process; **Red**: without the acceleration process**. Blue** : TopMod09 program (single core). The used grid step is 0.1 bohr.

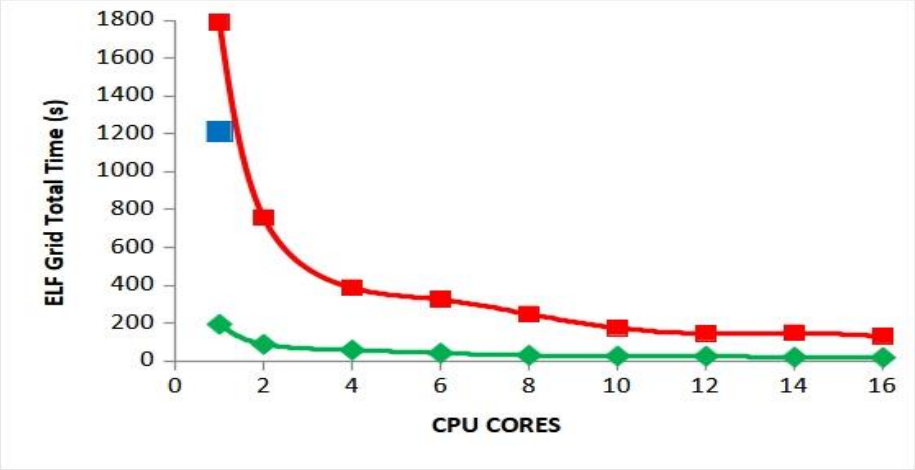

Figure 2. Total time, in s, elapsed for the 3d-grid computing of ELF applied to the guanine (G) – cytosine (C) complex computed at the B3LYP/6-31+G(d,p) level of theory. In this case, the β-radius r_μ was compeled to be lower than 7.5 bohr. Color Code: **Green**: using grid acceleration process; **Red**: without the acceleration process**. Blue** : TopMod09 program (single core). The used grid step is 0.1 bohr.

Figure 3 shows the computed ELF profiles for C_2H_2 along its molecular axis when all primitives are kept (black curve) and when some primitives are disregarded (red curve) according to the computational scheme used for the acceleration grid process. The both profiles being fully overlapped, this clearly illustrates the sustainability of the accuracy of the acceleration scheme.

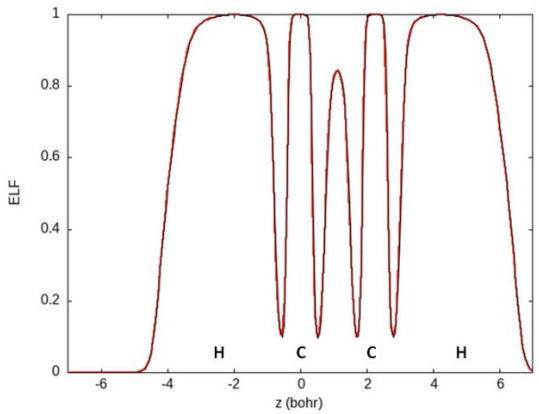

Figure 3. ELF profiles for the C_2H_2 molecule computed along the z axis (bohr) at the B3LYP/6-31+G(d,p) level of theory. Color code : red, using the radii acceleration process and black, without the acceleration process (black).

4.2 The basin analysis.

To illustrate the applicability of our analysis, we have evaluated the performance of the basin analysis using the grid-based algorithms discussed in this paper.

Figure 4 (left) reports the comparative parallel speeds of the assignment process as a function of the number of CPU-cores plotted for the three disctinct algorithms namely the off-grid, the near-grid and the mixed-near-grid. For this example, the selected ELF grid is the one already used in the previous section 4.1 for the CG complex where an

intermediate grid constant spacing of 0.1 bohr was used. It was well-known that the computational effort of theses algorithms scales approximatively linearly with the number of grid points (system size) with a fixed computational effort per grid point. [15,27] So, the goal of this study is rather to compare their relative speedup efficiencies using the grid acceleration process which has been used for make the 3d-grid. Figure 4 shows a clear split between the off-grid algorithm and the other two algorithms near-grid and mixed-near-grid. The gradient ascent part being a rate-limiting step, the off-grid algorithm remains logically slower than the two others. Nevertheless, when the number of processors increases, the speed of the off-grid algorithm becomes closer to others. For example, the ELF basin analysis for the CG complex takes 662 s using a single core and it takes only 39 s using 16 CPU cores (green curve on Figure 4). The similar analysis using the near-grid algorithm takes 25 s using a single core and it takes only 3 s using 16 CPU cores (red curve on Figure 4). We conclude that the computation cost of the off-grid analysis using 16 cores became approximatively comparable to the near-grid analysis using a single core. This is an important finding which makes possible the application of the off-grid basin analysis regardless of the on-grid methods commonly known as associated to a low computational time. Figure 5 displays the speedup quantity which can be evaluated by the calculation of the well-know efficiency defined as the ratio of the computational time associated to a single processor T_{1p} (serial execution) and the time T_{Np} associated to N_p processors. [36] As observed in Figure 5, the speedup scales lineraly with the number or CPU cores from one core to thirty cores, approximatively. Clearly, the reduction of the computational cost due to the twofold effectiveness both the accelerated grid process and the parallization scheme make possible the grid-based analysis to large systems where large grids need to be produced.

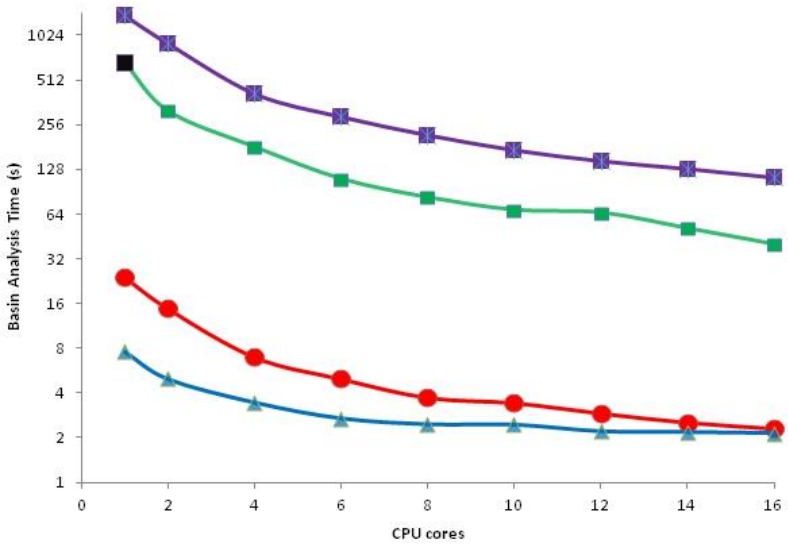

Figure 4. Log scale of the total time (seconds), elapsed for the ELF basin analysis for the guanine (G) –cytosine (C) complex computed at the B3LYP/6-31+G(d,p) level of theory. Color code : **Green**: off-grid algorithm with the acceleration process. **Purple**: off-grid algorithm without the acceleration process**, Red**: near-grid algorithm and **Blue**: mixed-near-grid algorithm. **Black square** : TopMod09 program (single core). The used grid step is 0.1 bohr.

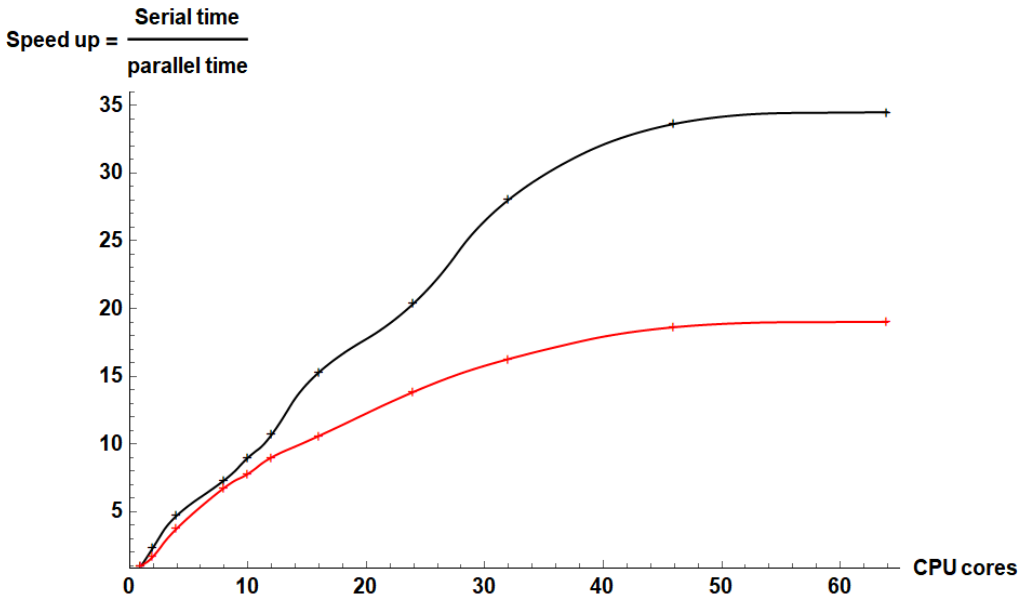

Figure 5. Parallel speedup elapsed for the ELF basin analysis applied to the GC complex computed at the B3LYP/6-31+G(d,p) level of theory. Color code : **Black:** off-grid algorithm **Red**: near-grid algorithm.

4.3 Numerical Integrated Properties.

To illustrate the applicability of our implementation and its accuracy, we discuss in this section the basins populations after the basin analysis for each of algorithm considered in this paper. The accuracy of integrated populations is of crucial interest because it often the preliminary step to compute more elaborate QCT descriptor such as the delocalization index[37] or more generally, the variance and covariance matrix elements [38] which explicitely depend on the populations. Note that no computational effort is needed in this work for the calculation of populations since the numerical integration of the electron density over the basin volume is plainly computed by summing over the grid points assigned to the volume. Tables 2 and 3 show the results obtained for QTAIM and ELF, respectively. The basin numbering is given with respect to the numbering given in Figure 6. A fine grid spacing (0.075 bohr) as well as an intermediate grid step (0.1 bohr) were been used. The grid spacing of 0.075 bohr was selected because populations obtained from a lower grid spacing than 0.075 bohr (typically 0.05 bohr) only differ in average with a negligible discrepancy lower than 0.001 e . The benzene and the CG complex have been selected as test systems. Indeed, the QCT analysis of C_6H_6 is expected to be consistent with the high symmetry of the molecule (D_{6h} group). For example, it means that the basin populations of carbon atom or ELF bonding bassin V(C, C) should be the same. In contrast, the CG complex doesn't show any symmetry and seems to be a reasonable choice as test of an enough large system.

QTAIM. The carbon atomic populations have been reported in Table 2. The results obtained with our off-grid implementation are in good agreement with the values obtained with AIMALL taken as reference for a fine grid spacing (0.075 bohr) as well as for an intermediate grid step (0.1 bohr). A standard deviation of 0.001 e is computed from the both off-grid algorithm and AIMALL populations. In contrast, the on-grid (near-grid and mixed-near-grid) algorithms display a clear symmetry breaking , a group of two

atoms highlighting a slight smallest value (< 5.97e instead of 6e) for an intermediate grid spacing is used. The origin of the effect can be understood as an intresic failure of on-grid algorithms for which the basin's assigment is mandatory restricted to the grid. Note that the results obtained with TopMod09 seems to break slightly the D_{6h} symmetry. Whatever the used algorithms, the QTAIM populations of the CG complex are, in average, rather consistent with all populations obtained from the AIMALL package.

ELF. The core and valence populations were also computed in the benzene molecule and the CG complex. The near-grid and mixed-near-grid algorithms have been compared satisfactorily to the related results provided by the off-grid populations taken as reference (see Table 2). Once again, the populations of selected core basins obtained from the on-grid assignment lightly differ from the off-grid populations. Moreover, all the tested algorithms depict a small difference between the V(C, C) populations of C_6H_6. The latter difference is of 0.04e in average for the on-grid implementation whereas this difference is only of 0.01e for the off-grid implementation. These findings remain unchanged whatever the used step grid of 0.1 bohr or of 0.075 bohr. In contrast, Table 4 shows the ELF populations obtained with a more large grid spacing of 0.175 bohr. For this latter case, the off-grid valence populations remain rather unchanged in comparison to the on-grid process where some large variations can be oberved. This finding is highlighted for the core basins where gradients are sharply modified around the nuclear regions. For example, a population of 2.03e was found for $C(O_8)$ basin using the off-grid implementation while a untenable value of 2.27e is found using on-grid algorithms, the reference value being of 2.12e obtained from the off-grid implementation using a fine grid step of 0.075 bohr. Once again, this is a no-surprising result because the real space integrations using uniform grids less dense near atomic nuclei and the on-grid algorithms appear as a less efficient. Overall, although the ELF topology remains enough stable even for an enough large grid step spacing, this confirms that an off-grid implementation remains more robust and should be prefered for any QCT study where the accuracy of descriptors is required.

Basins	**off-grid[a]**	**near-grid[b]**	**mixed-near-grid off-grid and near-grid[c]**	**TopMod09 package *+**	**AIMALL package***
C_6H_6					
C_1	6.025 (6.022)	6.018 (6.019)	6.013 (6.001)	6.06 (6.09)	6.004
C_2	6.023 (6.010)	6.010 (5.964)	6.001 (5.969)	6.03 (6.03)	6.003
C_3	6.025 (6.022)	6.018 (6.019)	6.013 (6.001)	6.03 (6.07)	6.004
C_4	6.025 (6.022)	6.018 (6.019)	6.013 (6.001)	6.03 (6.01)	6.004
C_5	6.023 (6.010)	6.010 (5.964)	6.001 (5.969)	5.99 (5.98)	6.003
C_6	6.025 (6.022)	6.018 (6.019)	6.013 (6.001)	6.03 (6.03)	6.004
Mean C atoms	*6.024 (6.018)*	*6.015 (6.001)*	*6.009 (5.990)*	*6.03 (6.04)*	*6.003*
Standard deviation	*0.001 (0.006)*	*0.004 (0.029)*	*0.006 (0.016)*	*0.02 (0.04)*	*0.001*
C-G complex					
C_2	4.865 (4.861)	4.874 (4.866)	4.902 (4.864)	4.87 (4.87)	4.856
C_4	4.185 (4.171)	4.170 (4.150)	4.163 (4.205)	4.18 (4.19)	4.159
C_{10}	4.659 (4.661)	4.646 (4.642)	4.660 (4.667)	4.66 (4.66)	4.635
N_3	8.193 (8.203)	8.199 (8.227)	8.185 (8.217)	8.20 (8.19)	8.204
N_7	8.345 (8.348)	8.302 (8.295)	8.312 (8.274)	8.36 (8.39)	8.295
O_8	9.230 (9.212)	9.217 (9.195)	9.217 (9.194)	9.21 (9.21)	9.236

O_9	9.226 (9.210)	9.209 (9.196)	9.205 (9.182)	9.20 (9.20)	9.223
H_{20}	0.455 (0.456)	0.494 (0.487)	0.509 (0.495)	0.45 (0.43)	0.488
H_{21}	0.445 (0.445)	0.471 (0.456)	0.478 (0.464)	0.44 (0.42)	0.477
H_{25}	0.419 (0.419)	0.451 (0.456)	0.456 (0.463)	0.41 (0.39)	0.446
Mean	*5.002 (4.999)*	*5.003 (4.997)*	*5.008 (5.002)*	*4.99 (4.99)*	*5.002*
Standard deviation	*3.656 (3.653)*	*3.635 (3.636)*	*3.632 (3.626)*	*3.65 (3.66)*	*3.641*

Table 2. Comparative selected QTAIM populations (electrons) obtained from several algorithms and programs performed to the benzene and the guanine (G) –cytosine (C) complex computed at the B3LYP/6-31+G(d,p) level of theory. The basin numbering is given with respect to the numbering given in Figure 6. An used grid step of 0.075 bohr was used. In brackets, the grid step is 0.1 bohr. (a) off-grid algorithm (b) Original Henkelman's group algorithm performed with the bader charge analysis package (c) Mixed-near-grid off-grid and near-grid *Analytical integrations are performed. (+ Only two significant digits are provided by the program.

Basins	**off-grid[a]**	**near-grid[b]**	**mixed-near-grid off-grid and near-grid[c]**	**TopMod09 package*+**
C_6H_6				
V(C_1, C_2)	2.780 (2.772)	2.783 (2.766)	2.783 (2.766)	2.82 (2.81)
V(C_2, C_3)	2.780 (2.772)	2.783 (2.766)	2.783 (2.766)	2.72 (2.74)
V(C_3, C_4)	2.776 (2.787)	2.743 (2.736)	2.743 (2.737)	2.78 (2.78)
V(C_4, C_5)	2.780 (2.773)	2.783 (2.766)	2.783 (2.766)	2.72 (2.74)
V(C_5, C_6)	2.780 (2.773)	2.783 (2.766)	2.783 (2.766)	2.82 (2.81)
V(C_6, C_1)	2.776 (2.787)	2.743 (2.736)	2.743 (2.737)	2.78 (2.78)
C(C1)	2.090 (2.088)	2.110 (2.120)	2.109 (2.119)	2.09 (2.09)
C(C2)	2.090 (2.088)	2.110 (2.120)	2.109 (2.119)	2.10 (2.09)
C-G complex				
Core basins				
C(C_2)	2.087 (2.090)	2.107 (2.124)	2.107 (2.124)	2.09 (2.09)
C(O_8)	2.122 (2.130)	2.170 (2.194)	2.170 (2.194)	2.13 (2.12)
C(N_3)	2.109 (2.121)	2.142 (2.175)	2.142 (2.175)	2.11 (2.11)
Valence basins				
V(C_2, N_7)	2.256 (2.255)	2.251 (2.238)	2.251 (2.238)	2.26 (2.26)
V(C_2, N_3)	2.369 (2.369)	2.356 (2.363)	2.356 (2.363)	2.36 (2.36)
V(C_4, N_3)	2.339 (2.333)	2.314 (2.282)	2.314 (2.282)	2.34 (2.36)
V(C_4, O_8)	2.216 (2.225)	2.201 (2.209)	2.201 (2.208)	2.19 (2.20)
V(O_8)	2.851,2.707 (2.845 , 2.709)	2.853, 2.657 (2.832, 2.635)	2.853, 2.657 (2.832, 2.636)	2.85, 2.72 (2.86, 2.72)
V(O_9)	2.944, 2.617 (2.944, 2.622)	2.950, 2.571 (2.932, 2.579)	2.950, 2.571 (2.932, 2.579)	2.98, 2.60 (2.98, 2.60)
V(N_3)	3.031 (3.037)	3.034 (3.019)	3.035 (3.019)	3.05 (3.04)
V(H_{20}, N_{11})	2.110 (2.111)	2.086 (2.066)	2.086 (2.066)	2.08 (2.08)
V(H_{21}, N_{19})	2.000 (1.998)	2.016 (2.005)	2.016 (2.004)	2.00 (2.00)
V(C_{10}, O_9)	2.128 (2.121)	2.101 (2.056)	2.101 (2.056)	2.09 (2.10)

Table 3. Comparative selected ELF populations (electrons) obtained from several algorithms and programs performed to the benzene and the guanine (G) –cytosine (C) complex optimized at the B3LYP/6-31+G(d,p) level of theory. The basin numbering is given with respect to the numbering given in Figure 6. The populations have been obtained using a fine equidistant cube grid (step of 0.75 bohr), in brackets, a grid step of 0.1 bohr was used. a) off-grid algorithm (b) Original Henkelman's group algorithm (c) Mixed-near-grid off-grid and near-grid *Analytical integrations are performed. + Only two significants digits are provided by the program.

Basins	off-grid[(a)]	near-grid[(b)]	mixed-near-grid [(c)]	Reference[(d)]
$C(N_8)$	2.020	2.229	2.229	2.087
$C(O_8)$	2.030	2.274	2.274	2.122
$C(C_2)$	2.050	2.142	2.141	2.109
$V(C_2, N_7)$	2.262	2.179	2.179	2.256
$V(C_2, N_3)$	2.363	2.249	2.249	2.369
$V(C_4, N_3)$	2.330	2.262	2.262	2.339
$V(C_4, O_8)$	2.218	2.149	2.149	2.216
$V(O_8)$	2.859, 2.731	2.791, 2.571	2.791, 2.571	2.851, 2.707
$V(O_9)$	2.940, 2.654	2.882, 2.491	2.882, 2.491	2.944, 2.617
$V(N_3)$	3.060	2.981	2.981	3.031
$V(H_{20}, N_{11})$	2.109	2.002	2.002	2.110
$V(H_{21}, N_{19})$	1.998	1.945	1.945	2.000
$V(C_{10}, O_9)$	2.119	2.028	2.028	2.128

Table 4. Comparative selected ELF populations (electrons) obtained from several algorithms and programs performed for the guanine (G) –cytosine (C) complex optimized at the B3LYP/6-31+G(d,p) level of theory. Basins numbering with respect to the numbering given in Figure 6. A coarse grid step of 0.175 bohr was used. a) off-grid algorithm (b) near-grid algorithm (c) Mixed-near-grid (d) off-grid algorithm using a fine grid spacing of 0.075 bohr is taken as a reference.

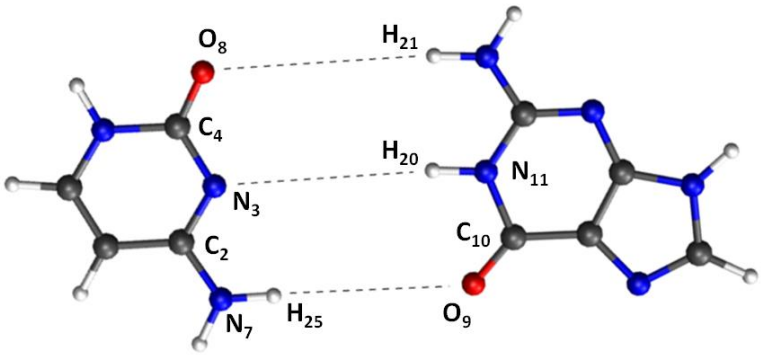

Figure 6. Guanine (G) –cytosine (C) complex optimized at the B3LYP/6-31+G(d,p) level of theory. The numbering of atoms correspond to that numbering reported in Tables 2, 3 and 4.

5. Conclusion.

In this paper, we have presented a robust strategy designed to make more efficient and fast both the calculation of the scalar function over a three-dimensional discrete grid and its assignment to basins. A comparative study of well-known grid-based algorithms designed to partition the 3d-grid into QCT basin volumes was also presented. The capabilities of grid-based numerical analysis to handle different elaborated one-electron density functions such the ELF topological analysis have been illustrated. In light of these investigations, the efficiency of the off-grid implementation and its accuracy were demonstrated. Taken into account the speeding-up of the parallelized process used to generate grids, the grid-based quantum chemical topology is expected to be well suited to molecular or solid systems where large grids need to be produced whatever the used one-density function, i.e. for the QTAIM analysis and beyond. Future works will focus on the development of new abilities of grid-based algorithms to allow the handle of typical adaptative grids where gradients of the scalar function are sharply modified around the nuclear regions and a fine grid spacing is required. Note that the QCT analysis on such non-uniform grids remain scarce.[11,28] A new MPI parallel implementation of the off-grid based algorithm is also currently under developments in our group.

Acknowledgment

We thank Pr. Paul Fleurat-Lessard for his constant support, stimulating discussions and for his critical reading of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data that support the findings of this study are available. Additionnal information can be requested from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. P. L. A. Popelier. In Applications of Topological Methods in Molecular Chemistry; Chauvin, R.; Lepetit, C.; Silvi, B.; Alikhani, E., Eds.; Springer International Publishing: Cham, **2016**, p 23.
2. R. F. W. Bader, *Chem. Rev.* **1991**, 91(5), 893.
3. M. Causá, A. Savin, B. Silvi, *Found. Chem.* **2014**, 16(1), 3.
4. B. Silvi, A. Savin, *Nature* **1994**, 371(6499), 683.
5. A. D. Becke, K. E. Edgecombe, *J. Chem. Phys.* **1990**, 92(9), 5397.
6. B. Silvi, *Journal of Molecular Structure* **2002**, 614(1-3), 3.
7. F. W. Biegler-könig, R. F. W. Bader, T.-H. Tang, *J. Comput. Chem.* **1982**, 3(3), 317.
8. J. I. Rodríguez, R. F. W. Bader, P. W. Ayers, C. Michel, A. W. Götz, C. Bo, *Chem. Phys. Lett.* **2009**, 472(1), 149.
9. M. Yu, D. R. Trinkle, *J. Chem. Phys.* **2011**, 134(6), 064111.
10. A. Otero-de-la-Roza, V. Luaña, *J. Comput. Chem.* **2011**, 32(2), 291.
11. J. I. Rodríguez, A. M. Köster, P. W. Ayers, A. Santos-Valle, A. Vela, G. Merino, *J. Comput. Chem.* **2009**, 30(7), 1082.
12. P. M. Polestshuk, *J. Comput. Chem.* **2013**, 34(3), 206.
13. G. Henkelman, A. Arnaldsson, H. Jónsson, *Comput. Mater. Sci.* **2006**, 36(3), 354.
14. P. L. A. Popelier, *Mol. Phys.* **1996**, 87(5), 1169.
15. D. Kozlowski, J. Pilmé, *J. Comput. Chem.* **2011**, 32(15), 3207.
16. P. L. A. Popelier, *Chem. Phys. Lett.* **1994**, 228(1), 160.
17. H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, P.-T. Bremer, *J. Comput. Chem.* **2018**, 39(16), 936.
18. F. W. Biegler-Konig, T. T. Nguyen-Dang, Y. Tal, R. F. W. Bader, A. J. Duke, *J. Phys. B: At. Mol.* **1981**, 14(16), 2739.
19. P. L. A. Popelier, *Coordination Chemistry Reviews* **2000**, 197(1), 169.
20. N. O. J. Malcolm, P. L. A. Popelier, *J. Comput. Chem.* **2003**, 24(10), 1276.
21. J. I. Rodríguez, *J. Comput. Chem.* **2013**, 34(8), 681.
22. R. Hernández-Esparza, S.-M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, J.-M. Hernández-Pérez, R. Vargas, J. Garza, *J. Comput. Chem.* **2014**, 35(31), 2272.
23. A. I. Stash, V. G. Tsirelson, *Journal of Applied Crystallography* **2022**, 55(2), 420.
24. L. Palatinus, S. J. Prathapa, S. v. Smaalen, *Journal of Applied Crystallography* **2012**, 45, 575.
25. E. Sanville, S. D. Kenny, R. Smith, G. Henkelman, *J. Comput. Chem.* **2007**, 28(5), 899.
26. W. Tang, E. Sanville, G. Henkelman, *Journal of Physics: Condensed Matter* **2009**, 21(8), 084204.
27. S. Noury, X. Krokidis, F. Fuster, B. Silvi, *Computers & Chemistry* **1999**, 23(6), 597.

28. M. J. Hutcheon, A. M. Teale, *J. Chem. Theory Comput.* **2022**.
29. P. Soler, F. Fuster, H. Chevreau, *J. Comput. Chem.* **2004**, 25(15), 1920.
30. T. A. Keith. In Department of Chemistry; McMaster University, Canada., **1993**.
31. T. Lu, F. Chen, *J. Comput. Chem.* **2012**, 33(5), 580.
32. M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, J. E. P. Jr., F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox. Gaussian, Inc., Wallingford CT, **2009**.
33. H. B. Schlegel, M. J. Frisch, *Int. J. Quantum Chem.* **1995**, 54(2), 83.
34. P. L. A. Popelier, *J. Comput. Chem.* **2018**, 39(10), 604.
35. J. Duchi, E. Hazan, Y. Singer, *Journal of Machine Learning Research* **2011**, 12(61), 2121.
36. Y. Yong, Z. Xiaodong, M. Qian, *IEEE Transactions on Software Engineering* **1997**, 23(1), 4.
37. X. Fradera, M. A. Austen, R. F. W. Bader, *J. Phys. Chem. A* **1998**, 103(2), 304.
38. A. Savin, B. Silvi, F. Colonna, *Canadian Journal of Chemistry* **1996**, 74(6), 1088.