

## A grid-based Bader analysis algorithm without lattice bias

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2009 J. Phys.: Condens. Matter 21 084204

(<http://iopscience.iop.org/0953-8984/21/8/084204>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 129.252.86.83

The article was downloaded on 29/05/2010 at 17:57

Please note that [terms and conditions apply](#).

# A grid-based Bader analysis algorithm without lattice bias

W Tang<sup>1</sup>, E Sanville<sup>2</sup> and G Henkelman<sup>1</sup>

<sup>1</sup> Department of Chemistry and Biochemistry, The University of Texas at Austin, Austin, TX 78712-0165, USA

<sup>2</sup> Department of Mathematical Sciences, Loughborough University, Loughborough LE11 3TU, UK

E-mail: [henkelman@mail.utexas.edu](mailto:henkelman@mail.utexas.edu)

Received 19 March 2008, in final form 6 June 2008

Published 30 January 2009

Online at [stacks.iop.org/JPhysCM/21/084204](http://stacks.iop.org/JPhysCM/21/084204)

## Abstract

A computational method for partitioning a charge density grid into Bader volumes is presented which is efficient, robust, and scales linearly with the number of grid points. The partitioning algorithm follows the steepest ascent paths along the charge density gradient from grid point to grid point until a charge density maximum is reached. In this paper, we describe how accurate off-lattice ascent paths can be represented with respect to the grid points. This improvement maintains the efficient linear scaling of an earlier version of the algorithm, and eliminates a tendency for the Bader surfaces to be aligned along the grid directions. As the algorithm assigns grid points to charge density maxima, subsequent paths are terminated when they reach previously assigned grid points. It is this grid-based approach which gives the algorithm its efficiency, and allows for the analysis of the large grids generated from plane-wave-based density functional theory calculations.

(Some figures in this article are in colour only in the electronic version)

## 1. Introduction

First principles methods, and especially density functional theory (DFT), are commonly used to calculate the many-body electronic interactions between atoms in molecules and in the solid state. Accurately describing these complex interactions is difficult, but it can also be challenging to rationalize the calculated energetics. One powerful technique for doing this is to decompose properties of the molecule or material into contributions from the individual atoms. Bader suggested an elegant way to do this partitioning [1]. His idea was to use the charge density to divide space within molecular systems into atomic (Bader) volumes. Each Bader volume contains a single charge density maximum, and is separated from other volumes by surfaces on which the charge density is a minimum normal to the surface. Typically, there is one charge density maximum at each atomic center and one Bader volume for each atom, but this is not required; there are cases in which Bader volumes do not contain a nucleus [2]. The dividing surfaces (also called zero-flux surfaces) separating these volumes lie in the bonding regions between atoms. This Bader partitioning has an advantage over other partitioning

schemes (e.g. Mulliken population analysis) in that it is based upon the charge density, which is an observable quantity that can be measured experimentally or calculated. Furthermore, in a converged electronic structure calculation, the charge density is insensitive to the basis set used. In this regard, the Bader analysis is more robust than wavefunction-based population methods [3–5].

There are several different approaches to calculating Bader volumes. Early algorithms were implemented for quantum chemistry calculations of small molecules, in which the gradient of the charge density can be calculated from derivatives of an analytic wavefunction [1, 6, 7]. These methods first find stationary points in the charge density and then follow trajectories along the density gradient from these points to map out their connectivity and the zero-flux dividing surfaces. With the dividing surfaces represented in this way, the charge in each Bader volume can be integrated radially from the charge density maximum to the surface. While this approach works for small molecules, a high density of descent trajectories is needed to accurately represent the surface away from the critical points, and the method has been criticized as being computationally expensive for large systems [8, 5].

It was also found to fail for complex bonding geometries and when the radial integration rays have multiple intersection points with the dividing surface [9–11].

Several improvements to this original approach have been proposed. Popelier developed a method to more accurately integrate the charge density within each Bader volume using a Fourier–Chebyshev fit [12] and a bisection method to analytically represent the dividing surfaces [13]. Other approaches suggested by Popelier include the use of the divergence theorem to replace the three-dimensional integration over the Bader volumes into a two-dimensional integral over the dividing surfaces [8], and the use of a tree search to treat complex bonding topologies between critical points [14, 15].

Most current implementations of Bader’s analysis are based upon a grid of charge density values [6, 16, 17, 11, 18, 19]. This is particularly important for plane-wave-based DFT calculations, because it allows for the analysis of condensed phase systems with many atoms. The algorithm described in this work is an improvement upon such a grid-based method [18]. In that original work, an algorithm was introduced in which ascent trajectories along the charge density were followed between grid points to determine the Bader volumes. The constraint of ascent trajectories to the grid means that each point need be considered only once. The method is highly efficient, scales linearly with system size, and is robust to complex bonding topology found in condensed systems. Recently, however, Sanville *et al* showed that this algorithm introduces a bias, causing the Bader surfaces to artificially follow the orientation of the lattice [19]. A modified algorithm was implemented by these authors in which ascent trajectories along interpolated gradients of the charge density were not constrained to the grid. This algorithm removes the bias and retains the linear scaling of the original method. Here, we present a modified version of our original algorithm which also removes the lattice bias in a somewhat different way. Trajectories are constrained to grid points at which charge density gradients are evaluated, and a correction vector is calculated which points from the nearest grid point to the unbiased (off-lattice) trajectory. In this way, lattice bias is removed and we have an algorithm which retains the efficiency, linear scaling, and robustness of the original grid-based method.

## 2. Methodology

This method for finding Bader volumes without lattice bias is an improvement to an earlier grid-based method [18]. We first describe the original ‘on-grid’ method, the bias that it has, and then this new ‘near-grid’ modification which removes the bias.

The input to these methods is a grid of charge density values defined on an orthogonal lattice. The generalization to non-orthogonal lattices and specific boundary conditions follows without significant complication, and has been done in the software available from [20]. The charge density grid can be obtained from *ab initio* calculations or from experiment. The method described here is particularly well suited to DFT calculations of large molecules or materials, with many atoms and complex bonding geometries.

### 2.1. On-grid method

The following steps summarize the on-grid Bader analysis method. Additional details are provided in [18].

- (i) First, an initial grid point is chosen. To associate this point with a Bader volume, a path of steepest ascent is followed between neighboring grid points along the charge density gradient.
- (ii) From each grid point along the path,  $(i, j, k)$ , the projection of the charge density gradient is calculated along the direction to each of the 26 neighboring grid points,

$$\nabla\rho(i, j, k) \cdot \hat{r}(di, dj, dk) = \frac{\Delta\rho}{|\Delta\hat{r}|}. \quad (1)$$

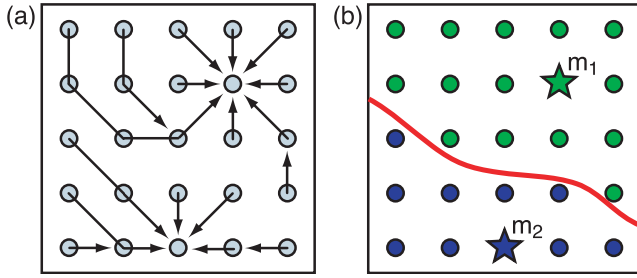
Here,  $(di, dj, dk)$  is a vector of integers describing the step along the grid to the neighbor. The integers  $di$ ,  $dj$ , and  $dk$  can each take the values  $\{-1, 0, 1\}$ , excluding  $di = dj = dk = 0$ . There are 26 neighbors to each grid point. The vector  $\hat{r}(di, dj, dk)$  is the normalized direction to the neighbor reached by the grid step  $(di, dj, dk)$ . The gradient projections in equation (1) are calculated using finite difference, where the change in charge density to the neighbor is

$$\Delta\rho = \rho(i + di, j + dj, k + dk) - \rho(i, j, k), \quad (2)$$

and the distance to the neighbor is

$$|\Delta\vec{r}| = |\vec{r}(i + di, j + dj, k + dk) - \vec{r}(i, j, k)|. \quad (3)$$

- (iii) One of the 26 neighbors is then determined as the next point along the ascent path. This neighbor,  $(i + di, j + dj, k + dk)$ , is the one which maximizes the gradient projection from equation (1). This gradient must be a positive value in order to make the step. If there are no positive values, the point  $(i, j, k)$  is a charge density maximum.
- (iv) The steepest ascent path is followed until a charge density maximum is found. Each new maximum found is assigned an integer value corresponding to the order in which it was found. In an array with the same dimensions as the charge density grid (initialized to zero), we take the number of the maximum to which the trajectory terminated, and assign that integer to all the grid points along the trajectory. We can do this because an ascent trajectory from each point along the path terminates at that same maximum. In this way, all points along the path are associated with a Bader volume around the charge density maximum.
- (v) Each grid point is analyzed in this way. The order in which grid points are analyzed is not particularly important, and it is easiest to cycle through them in a loop over all values of  $i$ ,  $j$ , and  $k$ . Grid points that are already assigned to a Bader volume are skipped, so that ascent trajectories are only followed from unassigned grid points. Trajectories are terminated if they either reach a grid point which has already been assigned, or when a new charge density maximum is found. Figure 1 illustrates both kinds of trajectories. If the trajectory terminates at an assigned grid



**Figure 1.** An illustration of the steepest ascent paths (a) on a charge density grid to find the Bader volumes using the on-grid analysis method. These ascent trajectories are constrained to the grid points, moving at each step to the neighboring grid point towards which the charge density gradient is maximized. Each trajectory either terminates at a new charge density maximum,  $m_i$ , or at a grid point which has already been assigned. After all grid points are assigned (b), the set of points which terminate at each maximum (green to  $m_1$  and blue to  $m_2$ ) constitute that Bader volume. The Bader surfaces (red curved line) separate the volumes.

point, the initial point and all points along the trajectory are assigned to that same Bader volume. If the trajectory terminates at a new charge density maximum, a new entry is made in the list of known charge density maxima, and all points along the trajectory are assigned to that new Bader volume number.

- (vi) After analyzing all grid points in this way, each is assigned to a Bader volume. The total charge in each Bader region is found by integrating the charge density over the grid points assigned to that region. The surface around a Bader volume can be visualized by plotting the charge density of that individual Bader volume.

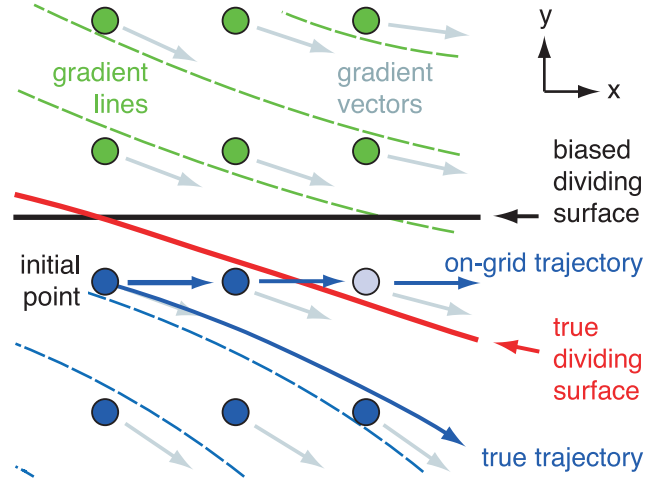
### 2.2. Lattice bias in the on-grid method

The on-grid method is simple to implement, efficient, and robust. Recently, however, Sanville *et al* showed that the method results in some systematic error of the Bader surfaces and population analysis [19]. This error is caused by the fact that the ascent trajectories in the on-grid method are constrained to the grid points. With trajectories following grid directions, the Bader surfaces become artificially angular with facets parallel to the grid. This bias remains in the limit of a fine charge density grid.

The lattice bias in the on-grid method is illustrated in figure 2. Ascent trajectories step between grid points in the direction that is most aligned with the charge density gradient (see equation (1)). Since there are only 26 such directions (8 in two-dimensions), the trajectories accumulate error when the gradients lines do not run parallel to the lattice. In figure 2, an on-grid trajectory is shown crossing the true dividing surface, resulting in points (e.g. the light-blue point) that are assigned to the incorrect Bader volume, and a surface which is artificially aligned along the lattice.

### 2.3. Near-grid method without lattice bias

Here, we describe a method which fixes the lattice bias error in the on-grid method. This new method is described as the



**Figure 2.** Illustration of lattice bias in the on-grid method. The true dividing surface (red) runs parallel to the gradient lines, but the on-grid ascent trajectories follow the lattice direction along which the projection of the charge density gradient is maximized. Starting from the initial point, this direction is along  $+x$  and the trajectory moves from grid point to grid point in this direction. The error can be seen as the on-grid trajectory (straight blue arrows through the light-blue point) deviates from the true trajectory (solid blue curved arrow). The resulting dividing surface follows the  $x$  lattice direction instead of the true dividing surface.

near-grid method; trajectories still follow from grid point to grid point, but a correction vector is remembered, which points to the location of the actual trajectory as it passes near each grid point. The near-grid method eliminates the lattice bias because trajectories are not constrained to grid points. It also retains the simplicity and linear scaling with system size of the on-grid method.

The following steps describe the near-grid algorithm.

- (i) Starting from an initial grid point,  $(i, j, k)$ , the trajectory of steepest ascent is followed up the charge density. Each step is made to one of the 26 neighboring grid points.
- (ii) In order to make a step from point  $(i, j, k)$ , the charge density gradient,  $\nabla\rho$ , is calculated from the charge density at the six closest neighbors using a central finite difference scheme. The components of the charge density in the  $x$ ,  $y$ , and  $z$  directions are:

$$\nabla\rho_x = \frac{\rho(i+1, j, k) - \rho(i-1, j, k)}{|\vec{r}(i+1, j, k) - \vec{r}(i-1, j, k)|}, \quad (4a)$$

$$\nabla\rho_y = \frac{\rho(i, j+1, k) - \rho(i, j-1, k)}{|\vec{r}(i, j+1, k) - \vec{r}(i, j-1, k)|}, \quad (4b)$$

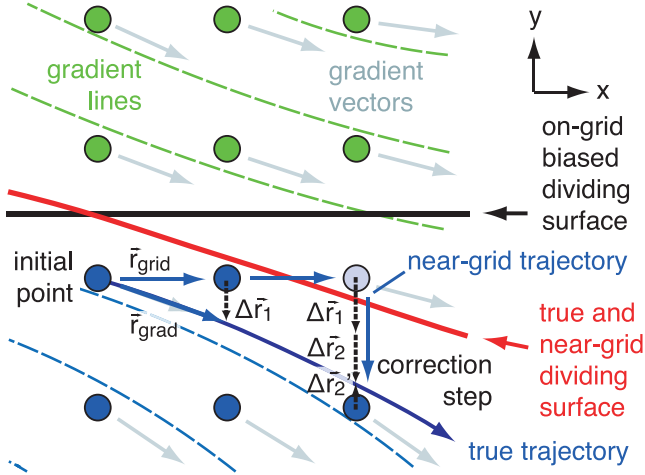
$$\nabla\rho_z = \frac{\rho(i, j, k+1) - \rho(i, j, k-1)}{|\vec{r}(i, j, k+1) - \vec{r}(i, j, k-1)|}. \quad (4c)$$

- (iii) To follow the gradient, a step along the gradient vector,

$$\vec{r}_{\text{grad}} = c(\nabla\rho_x, \nabla\rho_y, \nabla\rho_z), \quad (5)$$

should be taken. For this step to advance the trajectory by one grid point, and no more in any direction, the constant is chosen as

$$c = \min(dx/|\nabla\rho_x|, dy/|\nabla\rho_y|, dz/|\nabla\rho_z|), \quad (6)$$



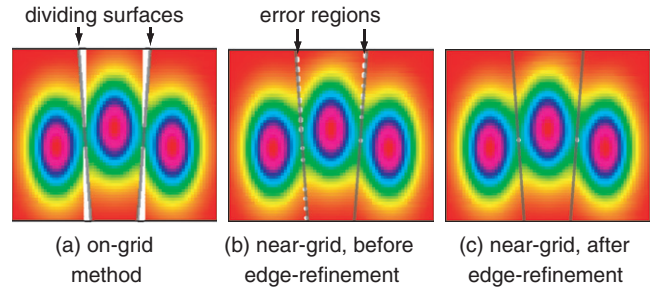
**Figure 3.** An illustration of the near-grid step and how a correction vector points to the true trajectory. Steps are taken between grid points. The first step between grid points,  $\vec{r}_{\text{grid}}$  and along the gradient,  $\vec{r}_{\text{grad}}$ , differ by the correction vector  $\Delta\vec{r}_1$ . After the second step, the difference is  $\Delta\vec{r}_2$ , and the total correction vector is  $\Delta\vec{r} = \Delta\vec{r}_1 + \Delta\vec{r}_2$ . Now, the  $y$  component of the correction vector is larger than half of the grid spacing so that a correction step is taken in the  $-y$  direction, and the correction vector is recalculated as  $\Delta\vec{r}'_2$ .

where  $dx$ ,  $dy$ , and  $dz$  are the grid spacings along the  $x$ ,  $y$ , and  $z$  Cartesian directions, respectively. The step  $\vec{r}_{\text{grad}}$  takes the trajectory (in general) off the grid. To retain the efficiency of the on-grid method, we then jump to the nearest grid point, so that the trajectory can be described as a hopping process between grid points. If we label this step between grid points as  $\vec{r}_{\text{grid}}$ , we can then describe a correction vector (initially zero) from the final grid point to the location of the true trajectory,

$$\Delta\vec{r} = \Delta\vec{r} + (\vec{r}_{\text{grad}} - \vec{r}_{\text{grid}}). \quad (7)$$

In figure 3 the first  $\vec{r}_{\text{grid}}$  step is along  $+x$ , and the correction vector,  $\Delta\vec{r}_1$ , points along  $-y$  to the true trajectory.

- (iv) From each new point along the ascent trajectory, the vectors  $\vec{r}_{\text{grad}}$  and  $\vec{r}_{\text{grid}}$  are calculated, and the correction vector is accumulated so that it always points to the true trajectory. When the length of any component of  $\Delta\vec{r}$  is larger than half of the grid spacing, a correction step is taken in that direction. The correction vector is then recalculated by subtracting the correction step. In this way, the true trajectory is never more than half a grid point from the current grid point in any direction.
- (v) The ascent trajectory is terminated when one of two criteria is met: (i) when it reaches a charge density maximum, or (ii) when it reaches a point for which the point itself and all of its neighbors are assigned to the same Bader region. In either case, all the grid points along the ascent trajectory are assigned to the same Bader region as the end points of the trajectory.
- (vi) An ascent trajectory is calculated from each grid point until all are assigned to a Bader volume. From each new trajectory, the correction vector  $\Delta\vec{r}$  is reset to  $(0, 0, 0)$ .



**Figure 4.** Comparison of the on-grid and near-grid methods for a two-dimensional charge density surface constructed from three Gaussian functions. By construction, the true dividing surface (gray lines in (a)) are at a small angle to the lattice. The on-grid method results in biased dividing surfaces that follow the grid; error regions, in which the Bader volume is incorrectly assigned, are colored white. In a single iteration of the near-grid method (b), the error regions are confined to the Bader surfaces. A subsequent refinement iteration (c) corrects all but two grid points, which are mis-assigned because of the low resolution of the grid. As the grid density is increased, the near-grid method finds the exact Bader volumes.

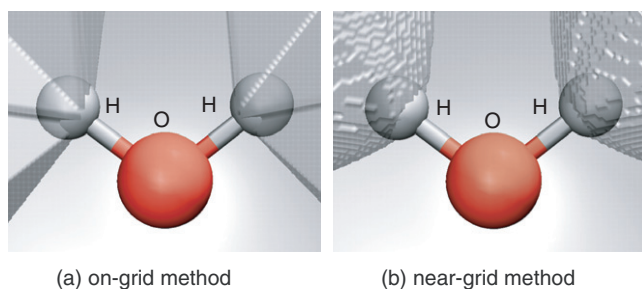
- (vii) When all the points are assigned, a final refinement of the grid points adjacent to the Bader surfaces is required. Depending upon the order in which grid points are analyzed, the grid point adjacent to the Bader surface can be assigned to one of the two volumes on either side of the dividing surface. This ambiguity is due to the fact that the trajectory between grid points deviates from the true trajectory by up to half a grid step. The refinement step corrects this by first identifying all grid points on the boundary of a Bader volume. An ascent trajectory is followed from each of these boundary grid points to determine in which volume it belongs. In this case, only the initial point is assigned, so that (in the case of smooth charge density grids) the refinement process need not be repeated.

### 3. Results

#### 3.1. Two-dimensional model

The lattice bias in the on-grid method is particularly severe when the true dividing surface runs at a small angle to the grid over long distances. In figure 4 we have constructed such a two-dimensional charge density grid from the sum of three Gaussian functions. The on-grid method finds vertical dividing surfaces that deviate from the true (gray) dividing surface, resulting in the (white) error regions. In these regions the method has assigned charge to the wrong Bader region.

The near-grid method corrects this lattice bias. After ascent trajectories are followed from each grid point, error regions are localized to grid points adjacent to the true dividing surface. A subsequent refinement step eliminates these errors, and gives the correct Bader volumes in the limit of a sufficiently fine grid.



**Figure 5.** Comparison of Bader dividing surfaces found with the on-grid method (left) and the near-grid method (right). The on-grid surfaces are angular with facets oriented along the grid directions. This bias is removed in the near-grid method.

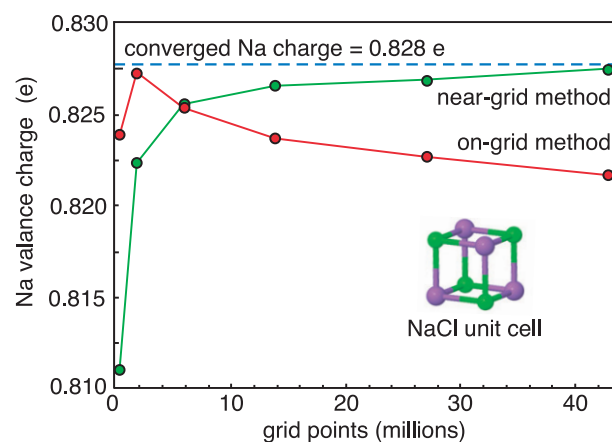
### 3.2. A water molecule

Finding the Bader volumes in a water molecule is a more realistic test. The geometry and charge density of the molecule was calculated with Gaussian 98 [21] using the aug-cc-pVDZ basis set at the MP2 level of perturbation theory. The charge density was written on a  $257^3$  orthogonal grid. The dividing surfaces found with the on-grid and near-grid methods are shown in figure 5. The lattice bias is apparent in the on-grid method from the angular shape of the surfaces (see also [18]). This bias is removed with the near-grid method and the surfaces become smooth. The ripples in the surface shown in figure 5 are due to the finite resolution of the grid.

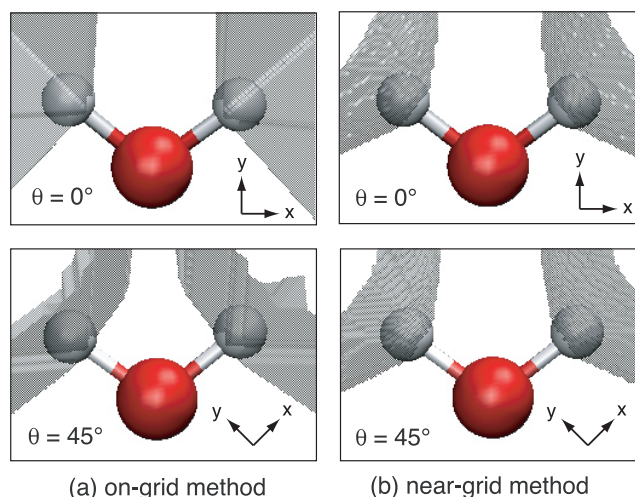
### 3.3. Ionic charge in a NaCl crystal

These grid-based analysis methods can be used for solid-state systems as well as molecular ones. In order to show convergence of the method with respect to grid density, we have calculated the Bader charges in a NaCl salt crystal, using the eight-atom unit cell illustrated in figure 6 (inset). The charge density was generated using the VASP plane-wave-based DFT code [22], using the PW91 generalized gradient functional [23]. Frozen core charges were represented with pseudopotentials of the Vanderbilt form [24] using the projector augmented wave (PAW) framework [25]. Including this frozen core charge in the charge density grid is important for calculating accurate Bader charges. A plane wave energy cutoff of 262.5 eV was used, and the Brillouin zone was sampled with a  $3 \times 3 \times 3$  Monkhorst–Pack [26]  $k$ -point mesh. An optimal lattice constant of 5.86 Å was determined for the NaCl crystal.

A set of charge density grids ranging from  $60^3$  points to  $350^3$  points were calculated. The grid spacing is 0.095 Å in the former case and 0.016 Å in the latter. The calculated valance charge on the Na ions is shown in figure 6. The near-grid method shows monotonic and smooth convergence to a valance charge of 0.828  $e$ , whereas the on-grid method has systematic errors in the limit of a fine grid. Although the systematic error of less than 0.01  $e$  in the on-grid method is acceptable for most calculations, the near-grid method should be used for its improved accuracy and systematic convergence.



**Figure 6.** Convergence with respect to grid density of the Bader charges for ions in a NaCl crystal. A fully converged charge on each Na ion is 0.828  $e$  (blue dashed line). The (old) on-grid method deviates from this value by ca 0.01  $e$  for a fine grid with 40 million points. The (new) near-grid method rapidly converges to the correct value.

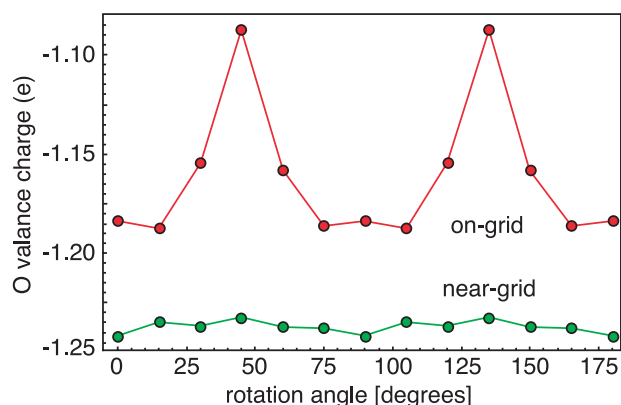


**Figure 7.** The Bader surface in  $\text{H}_2\text{O}$  is strongly dependent upon the orientation of the molecule with the on-grid method. This orientation dependence is due to the bias in the method which tends to orient the Bader surfaces along the grid directions. In the near-grid method, this bias is removed and the Bader surfaces are insensitive to the orientation of the molecule.

### 3.4. Variation of charge with molecular orientation

The lattice bias in the on-grid method can be seen by comparing the Bader volumes and charges for a molecule aligned at different orientations with respect to the charge density lattice. A converged calculation should not depend upon this orientation. For this test, we have calculated the Bader volumes and charges around an  $\text{H}_2\text{O}$  molecule using the VASP code. The details of these calculations are the same as for the NaCl calculations with the exception of a 250 eV energy cutoff and a single  $\Gamma$ -point for the isolated molecule.

Figure 7 shows a change in the shape of the Bader volumes for the on-grid method as the  $\text{H}_2\text{O}$  molecule is rotated by  $45^\circ$  in the plane of the figure. With the



**Figure 8.** The calculated Bader charge on the O atom in a  $\text{H}_2\text{O}$  molecule as it is rotated with respect to the charge density grid. The biased Bader surfaces in the on-grid method give rise to both systematic and orientation dependent errors as compared to the near-grid method, for which the Bader surfaces and O valance charge remain constant with orientation.

near-grid method, the shape of the volumes remains constant—as it should.

The sensitivity of the Bader charges to molecular orientation is compared quantitatively in figure 8. Not only does the on-grid method systematically underestimate the charge transfer from hydrogen to oxygen, it also varies by ca  $0.1 e$  as the molecule is rotated by  $45^\circ$ . With the near-grid method, this variation is largely removed; the shape and integrated charge of the Bader volumes is insensitive to the orientation of the molecule with respect to the charge density grid.

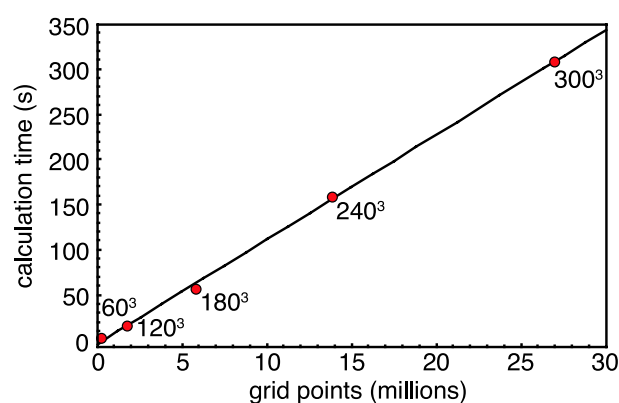
### 3.5. Scaling of computational effort

A strength of the on-grid method is that there is a fixed computational effort per charge density grid point, and therefore the total computational time scales linearly with the number of grid points [18]. This property of the method, as well as the robustness, makes the algorithm applicable to large systems which can have complex bonding geometries. These are features of the method that we want to retain in this improved near-grid method.

To test this, the computer time required to analyze charge density files was compared for various grid sizes for the NaCl system described in section 3.3. Cubic grids ranging from  $60^3$  to  $300^3$  points were analyzed. Each analysis required a single refinement iteration. Figure 9 shows that the computer time required to complete the analysis scales linearly with the number of charge density grid points. The slope of the line corresponds to analyzing a million grid points in 11.5 s on a 2.5 GHz G5 PowerPC.

## 4. Concluding remarks

The Bader analysis algorithm presented here removes the lattice bias of a constrained grid-based algorithm [18] allowing convergence in the limit of a fine charge density grid. The algorithm is suitable for large DFT calculations, and can be



**Figure 9.** Computer time required to analyze the charge density grid for the eight-atom NaCl cell with the near-grid algorithm. The computational cost scales linearly with the number of grid points in the charge density file, as with the on-grid method.

used for plane-wave-based calculations of condensed phase systems.

## Acknowledgments

This research was supported by the National Science Foundation from the NSF-CAREER award CHE-0645497 and the Robert A Welch Foundation. We also gratefully acknowledge the computational resources provided by the Texas Advanced Computing Center at the University of Texas at Austin.

## References

- [1] Bader R F W 1990 *Atoms in Molecules: a Quantum Theory* (New York: Oxford University Press)
- [2] Madsen G K H, Gatti C, Iversen B B, Damjovic Lj, Stucky G D and Srdanov V I 1999 F center in sodium electrosodalite as a physical manifestation of a non-nuclear attractor in the electron density *Phys. Rev. B* **59** 12359
- [3] Wiberg K B and Rablen P R 1993 Comparison of atomic charges derived via different procedures *J. Comput. Chem.* **14** 1504–18
- [4] Angyan J G, Jansen G, Loos M, Hattig C and Hess B A 1994 Distributed polarizabilities obtained using a constrained density-fitting algorithm *Chem. Phys. Lett.* **219** 267
- [5] De Proft F, Van Alsenoy C, Peeters A, Langenaeker W and Geerlings P 2002 Atomic charges, dipole moments, and Fukui functions using the Hirshfeld partitioning of the electron density *J. Comput. Chem.* **23** 1198
- [6] Popelier P L A 1998 *Morphy98* A program written by P L A Popelier with a contribution from R G A Bone, UMIST, Manchester, England
- [7] Stefanov B B and Cioslowski J 1995 An efficient approach to calculation of zero-flux atomic surfaces and generation of atomic integration data *J. Comput. Chem.* **16** 1394–404
- [8] Popelier P L A 2001 A fast algorithm to compute atomic charges based on the topology of the electron density *Theor. Chem. Acc.* **105** 393–9
- [9] Biegler König F W, Bader R F W and Tang T 1982 Calculation of the average properties of atoms in molecules. II *J. Comput. Chem.* **3** 317–28

- [10] Uberuaga B P, Batista E R and Jónsson H 1999 Elastic sheet method for identifying atoms in molecules *J. Chem. Phys.* **111** 10664–9
- [11] Katan C, Rabiller P, Lecomte C, Guezo M, Oison V and Souhassou M 2003 Numerical computation of critical properties and atomic basins from three-dimensional grid electron densities *J. Appl. Crystallogr.* **36** 65
- [12] Popelier P L A 1994 An analytical expression for interatomic surfaces in the theory of atoms in molecules *Theor. Chim. Acta* **87** 465–76
- [13] Popelier P L A 1998 A method to integrate an atom in a molecule without explicit representation of the interatomic surface *Comput. Phys. Commun.* **108** 180
- [14] Malcolm N O J and Popelier P L A 2003 An improved algorithm to locate critical points in a 3d scalar field as implemented in the program morphy *J. Comput. Chem.* **24** 437
- [15] Malcolm N O J and Popelier P L A 2003 An algorithm to delineate and integrate topological basins in a three-dimensional quantum mechanical density function *J. Comput. Chem.* **24** 1276
- [16] Noury S, Krokidis X, Fuster F and Silvi B 1999 Computational tools for the electron localization function topological analysis *Comput. Chem.* **23** 597–604
- [17] Biegler König F W, Schönbohm J and Bayles D 2001 Aim2000—a program to analyze and visualize atoms in molecules *J. Comput. Chem.* **36** 65
- [18] Henkelman G, Arnaldsson A and Jónsson H 2006 A fast and robust algorithm for bader decomposition of charge density *Comput. Mater. Sci.* **36** 354–60
- [19] Sanville E, Kenny S D, Smith R and Henkelman G 2007 Improved grid-based algorithm for bader charge allocation *J. Comput. Chem.* **28** 899–908
- [20] Arnaldsen A, Tang W and Henkelman G *Bader Charge Analysis* <http://theory.cm.utexas.edu/bader/>
- [21] Frisch M J *et al* 1998 *Gaussian 98, Revision A.7* (Pittsburgh, PA: Gaussian)
- [22] Kresse G and Hafner J 1993 *Ab initio* molecular dynamics for liquid metals *Phys. Rev. B* **47** R558–61
- [23] Perdew J P 1991 Unified theory of exchange and correlation beyond the local density approximation *Electronic Structure of Solids* ed P Ziesche and H Eschrig (Berlin: Akademie Verlag) p 11
- [24] Vanderbilt D 1990 Soft self-consistent pseudopotentials in a generalized eigenvalue formalism *Phys. Rev. B* **41** 7892–5
- [25] Kresse G and Joubert J 1999 From ultrasoft pseudopotentials to the projector augmented wave method *Phys. Rev. B* **59** 1758
- [26] Monkhorst H J and Pack J D 1976 Special points for Brillouin-zone integrations *Phys. Rev. B* **13** 5188–92