Accurate and efficient algorithm for Bader charge integration

Min Yu[1] and Dallas R. Trinkle[2,a)]
[1]*Department of Physics, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*
[2]*Department of Materials Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA*

We propose an efficient, accurate method to integrate the basins of attraction of a smooth function defined on a general discrete grid and apply it to the Bader charge partitioning for the electron charge density. Starting with the evolution of trajectories in space following the gradient of charge density, we derive an expression for the fraction of space neighboring each grid point that flows to its neighbors. This serves as the basis to compute the fraction of each grid volume that belongs to a basin (Bader volume) and as a weight for the discrete integration of functions over the Bader volume. Compared with other grid-based algorithms, our approach is robust, more computationally efficient with linear computational effort, accurate, and has quadratic convergence. Moreover, it is straightforward to extend to nonuniform grids, such as from a mesh-refinement approach, and can be used to both identify basins of attraction of fixed points and integrate functions over the basins. © *2011 American Institute of Physics*. [doi:10.1063/1.3553716]

I. INTRODUCTION

Based on density functional theory (DFT) (Ref. 1) calculations, decomposing the charge or the energy of a material into contributions from individual atoms can provide new information for material properties. Bader's "atoms in molecules" theory provides an example of a partitioning based on the charge density, and following the gradient at a particular point in space to the location of a charge density maximum centered at an atom—defining basins of attraction of fixed points of the charge density. Bader defines the atomic charges and well-defined kinetic energies as integrals over these Bader volumes,[2] Ω_ρ. Each Bader volume contains a single electron density maximum and is separated from other volumes by a zero flux surface of the gradients of the electron density, $\nabla\rho(\vec{r}) \cdot \hat{n} = 0$. Here, $\rho(\vec{r})$ is the electron density, and $\hat{n}$ is the unit vector perpendicular to the dividing surface at any surface point $\vec{r} \in \partial\Omega_\rho$. Each volume Ω_ρ is defined by a set of points where following a trajectory of maximizing ρ reaches the same unique maximum (fixed point). In practical numerical calculations, where the charge density is defined on a discrete grid of points in real space, it is very challenging to have an accurate determination of the zero flux surface.

Different approaches for condensed, periodic systems have relied on analytic expressions of the density[3,4] or discretizing the charge density trajectories.[5–10] Early algorithms were based on the electron density calculated from analytical wavefunctions of small molecules and integration along the gradient paths. Most current developments are based on a grid of electron density, which is important for DFT calculation and also applicable to analytical density function of small molecules. One octal tree algorithm[5] uses a recursive cube subdivision to find the atomic basins robustly, but practically is not applicable to complicated topologies due to huge computational cost. The "elastic sheet" method[6] defines a se-

ries of fictitious particles, which give a discrete representation of zero flux surface. Particles are relaxed according to the gradients of electron density and interparticle forces. This method will not work for complex surfaces with sharp cusps or points. Recently, Henkelman *et al.* developed an on-grid method[7] to divide an electron density grid into Bader volumes. This method can be applied to the DFT calculations of large molecules or materials. They discretize the trajectory to lie on the grid, ending at the local maximal point of the electron density. The points along each trajectory are assigned to the atom closest to the end point. Although this method is robust, and scales linearly with the grid size, it introduces a lattice bias caused by the fact that ascent trajectories are constrained to the grid points. The near-grid method[8] improves this by accumulating a correction vector—the difference between the discretized trajectory and the true trajectory—at each step. When the correction vector is sufficiently large, the discrete trajectory is corrected to a neighboring grid point. This method corrects the lattice bias and also scales linearly with respect to the size of grids. However, both grid trajectory methods require iteration to self-consistency in volume assignments. Also, the integration error scales linearly with the grid spacing, so very fine grids are required in numerical calculations to provide the correct Bader volume, reducing its applicability for accurate calculations in a large system. Lastly, a new algorithm uses a "divide and conquer" adaptive approach with tetrahedra; tetrahedra are continuously divided at the boundaries of Bader volumes, with the "weight" of each tetrahedron given by the number of vertices that belong to each volume.[9] Such an approach retains linear scaling with the grid spacing, but requires mesh refinement near boundaries to deal with the linear convergence of the error with the grid spacing. Other mesh-refinement schemes have proven very useful in the treatment of small molecules,[10] but it is less clear the applicability to condensed systems as considered here.
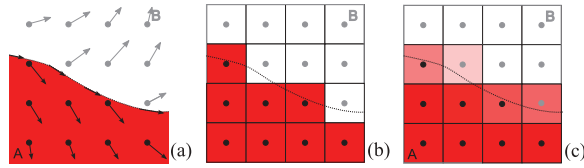
a)Electronic mail: dtrinkle@illinois.edu

FIG. 1. A schematic illustration of the zero flux surface, the near-grid algorithm, and weighted integration. The zero flux dividing surface separates volumes A and B, where arrows denote charge density gradients (a). The normal component is zero for any point on the surface $\nabla \rho \cdot \hat{n} = 0$. The near-grid method[8] gives grid-based partition (b); however, energy density integration based on this grid-based partition would cause integration error due to finite grid sizes. A weight function, (c) representing volume fractions of the cell of each grid point, is introduced to reduce the error due to a finite grid.

Figure 1 illustrates the partition of real space into two Bader volumes A and B by a zero flux dividing surface. The component of $\nabla \rho$ along the surface normal $\hat{n}$ is zero for any point on the surface ∂A or ∂B. A grid-based partition algorithm, such as the near-grid method, divides space into volume surrounding around each grid point and assigns each grid volume to a particular Bader volume. Even though grid points may be assigned to Bader volumes correctly, the density integration based on the grid-based partition would bring in numerical integration error that scales linearly with the grid spacing. Introducing a weight integrand representing the fraction of grid volume that belongs to a particular Bader volume smooths out the grid-based partition and improves the integration accuracy, and scales quadratically with the grid spacing. The atomic contribution is neither 1 nor 0 at the dividing surfaces, but fractional. In Fig. 1, red represents a weight of 1 to atom A for grid points closer to atom A, and transitions to white for a weight of 0 for grid points away from atom A.

The grid-based weight representing volume fractions of each grid volume assigned to different atoms and gives a more accurate integrand for the integration of either charge density or of kinetic energy density over the Bader volumes. The weight is computed from the total integrated flux of trajectories in a grid volume to neighboring grid volumes. The algorithm is robust, efficient with linear computing time in the number of grid points, and more accurate than the near-grid algorithm. Surprisingly, it combines both better error scaling—quadratic in the grid spacing—and improved computational efficiency. We have implemented and tested the algorithm for regular real-space grids, as result from plane-wave DFT calculations. However, it is straightforward to apply to nonuniform grids, such as would result from an adaptive mesh-refinement approach. In Sec. II, we derive the algorithm to construct a grid-based weight to perform numerical integrals over Bader volumes. Section III presents examples including three-dimensional charge density from three Gaussian functions in an FCC cell, TiO$_2$ bulk, NaCl crystal, and for a Si self-interstitial. Finally, we show the improved computational efficiency in Sec. IV. The end result is a simple, extendable, computationally efficient algorithm with quadratic integration error.

II. THE WEIGHT METHOD

The Bader partitioning of space defines volumes by the endpoint of a trajectory following the gradient flow of the
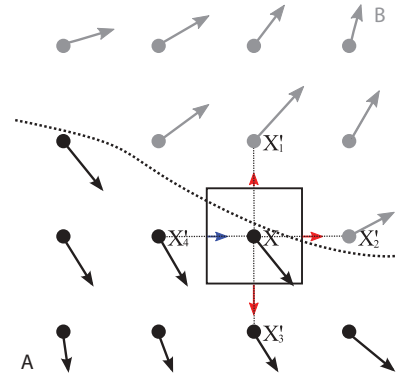

FIG. 2. Schematic illustration of the weight method. The volume of the cell of a grid point flows to its neighbors with larger charge density magnitude. Flowing flux is shown as directional map, either flowing from X to X' as red arrows or flowing from X' to X as blue arrows.

charge density, $\nabla \rho$. We assume that ρ has continuous first and second derivatives throughout all space of interest, and has a set of discrete local maxima (fixed points) $\vec{x}_1$, $\vec{x}_2$, etc., where $\nabla \rho = 0$ and the matrix $\nabla \nabla \rho$ is negative-definite. The basin of attraction A_n, of a fixed point $\vec{x}_n$, is the set of points which flow to the fixed point $\vec{x}_n$ along the charge density gradient. That is, for any point $\vec{r}$, we can integrate the trajectory given by $\dot{\vec{x}}(t) = \nabla \rho(\vec{x})$, with the initial condition $\vec{x}(0) = \vec{r}$, to find $\lim_{t \to \infty} \vec{x}(t)$. Each trajectory will end at a fixed point $\vec{x}_n$, and except for a set of points with zero volume in space, the extremum is a local maximum; the basin of attraction A_n are all points $\vec{r}$ whose trajectory $\lim_{t \to \infty} \vec{x}(t)$ ends at $\vec{x}_n$. Note also that if point $\vec{r}_0 \in A$, and the trajectory starting from $\vec{r}_1$ reaches $\vec{r}_0$ in a finite time t, then $\vec{r}_1 \in A$. This set defines a partitioning of space, where $A_n \cap A_m = \emptyset$ when $n \neq m$ and $\cup_n A_n = \Omega$. Finally, each basin A_n is such that wherever the normal $\hat{n}$ to the bounding surface ∂A_n is well-defined, $\hat{n} \cdot \nabla \rho = 0$. If ρ is the charge density, then A_n are the Bader volumes; but this definition is applicable to any sufficiently smooth function with a discrete set of local maxima. As the definition of the basins A_n derives from trajectories, it is not possible in general to determine if two neighboring points $\vec{r}$ and $\vec{r}'$ belong to the same or different basins based only on local information.

Figure 2 shows the reformulation for an approximate fractional partitioning of a real-valued function evaluated at a set of discrete points, X. The grid points X partition space into Voronoi polyhedra[11] V_X covering each grid point X, where a point in space $\vec{r}$ belongs to the volume V_X if X is the closest point in Cartesian space to $\vec{r}$. Each polyhedra is defined by the nearest neighboring points X' that are a distance $\ell_{X \to X'}$ away; the Voronoi polyhedron at X has facets $\partial V_{X \to X'}$ with normal $\hat{n}_{X \to X'}$ pointing from X to X' and area $a_{X \to X'}$. Moreover, the facet is at the midpoint between X and X'. Our goal is to define for each grid point X, a weight $w^A(X)$ between 0 and 1 such that $\sum_A w^A(X) = 1$ for all X, and the discrete approximation to the integral over the basin A

$$\int_A d^3 r f(\vec{r}) \approx \sum_X V_X w^A(X) f(X) \tag{1}$$

converges quadratically in the grid spacing for smooth functions $f(\vec{r})$. The weight, in this case, is the fraction of points

in V_X whose trajectory ends in the basin A. Note that if the points X form a regular periodic grid, the Voronoi volumes, facet areas, and neighbor distances need only be computed for the Wigner–Seitz cell around a single grid point.

To transition from the continuum definition of spatial partitioning to our Voronoi partitioned definition, we introduce the continuum probability density for our trajectories, $P(\vec{r}, t)$. From the trajectory equation, the probability flux at any point and time is $\vec{j}(\vec{r}, t) = P(\vec{r}, t)\nabla\rho(\vec{r})$. Then, the probability distribution evolves in time according to a continuity equation

$$\frac{\partial P(\vec{r}, t)}{\partial t} + \nabla \cdot (P(\vec{r}, t)\nabla\rho(\vec{r})) = 0. \tag{2}$$

This equation represents the combined evolution of a distribution of points in space; we use it to determine how the points in V_X distribute to neighboring volumes $V_{X'}$. Define the volume probability

$$P_X(t) = \int_{V_X} d^3 r\, P(\vec{r}, t), \tag{3}$$

then the evolution from the initial condition

$$P(\vec{r}, 0) = \begin{cases} V_X^{-1} & : \vec{r} \in V_X \\ 0 & : \vec{r} \notin V_X \end{cases} \tag{4}$$

is given by

$$\frac{dP_X(t)}{dt} = -\int_{V_X} d^3 r\, \nabla \cdot (P(\vec{r}, t)\nabla\rho(\vec{r}))$$

$$= -\sum_{X'} \int_{\partial V_{X \to X'}} P(\vec{r}, t)\nabla\rho \cdot \hat{n}_{X \to X'} d^2 r$$

$$\approx -P_X(t) V_X^{-1} \sum_{X'} \int_{\partial V_{X \to X'}} \nabla\rho \cdot \hat{n}_{X \to X'} d^2 r$$

$$\approx -P_X(t) \sum_{X'} \frac{a_{X \to X'}}{V_X} \cdot \frac{R(\rho_{X'} - \rho_X)}{\ell_{X \to X'}}$$

$$\equiv -P_X(t) \sum_{X'} \tau_{X \to X'}, \tag{5}$$

where $R(u) = u\theta(u)$ is the ramp function, so that $\tau_{X \to X'} \geq 0$ and is zero when $\rho_{X'} < \rho_X$; this is a consequence of our initial conditions where $P(r, 0)$ is only nonzero in the interior of V_X. The first approximation ignores spatial variation of $P(r, t)$ through the volume V_X (an error linear in the grid spacing), and the second approximation ignores spatial variation of $\nabla\rho$ along a facet $\partial V_{X \to X'}$, and approximates the gradient at the midpoint between X and X' with the finite difference value (also with an error that is linear in the grid spacing). The solution to Eq. (5) is $P_X(t) = \exp(-t \sum_{X'} \tau_{X \to X'})$. For that solution, the time-integrated flux of probability from V_X to $V_{X'}$ through the facet $\partial V_{X \to X'}$ is

$$J_{X \to X'} = \int_0^\infty dt \int_{\partial V_{X \to X'}} P(\vec{r}, t)\nabla\rho \cdot \hat{n}_{X \to X'} d^2 r$$

$$\approx \int_0^\infty dt\, P_X(t) V_X^{-1} \int_{\partial V_{X \to X'}} \nabla\rho \cdot \hat{n}_{X \to X'} d^2 r$$

$$\approx \int_0^\infty dt\, P_X(t)\tau_{X \to X'}$$

$$= \frac{a_{X \to X'}\ell_{X \to X'}^{-1} R(\rho_{X'} - \rho_X)}{\sum_{X'} a_{X \to X'}\ell_{X \to X'}^{-1} R(\rho_{X'} - \rho_X)}, \tag{6}$$

where we have used the same approximations as above. This flux defines the total fraction of points inside V_X that transition to volume $V_{X'}$ through $\partial V_{X \to X'}$. Note that $\sum_{X'} J_{X \to X'} = 1$, unless X is a local (discrete) maxima, where $\rho_X > \rho_{X'}$ for all neighbors X'. Finally, as the weight $w^A(X)$ represents the volume fraction of points in volume V_X whose trajectory ends inside basin A, then

$$w^A(X) = \sum_{X'} J_{X \to X'} w^A(X'). \tag{7}$$

Note that if for all X' where $\rho(X') > \rho(X)$, $\sum_A w^A(X') = 1$, then as $\sum_{X'} J_{X \to X'} = 1$, Eq. (7) guarantees that $\sum_A w^A(X) = 1$. Appendix A shows that the error in the weight of linear order in the grid spacing produces a quadratic order error in the integration.

Forward substitution solves Eq. (7) after the grid points are sorted from highest to lowest density $\rho(X)$. Sequentially, each point X is either

(1) A local maxima: $\rho(X) > \rho(X')$ for all neighbors X'. This grid point corresponds to a new basin A, and we assign $w^A(X) = 1$.
(2) An interior point: for all X' where $\rho(X') > \rho(X)$, the weights have been assigned and $w^A(X') = 1$ for the *same* basin A. Then Eq. (7) assigns X to basin A as well: $w^A(X) = 1$.
(3) A boundary point, with weights between 0 and 1 for multiple basins assigned by Eq. (7).

Then $w^A(X)$ is known from $w^A(X')$ where $\rho(X') > \rho(X)$ for each basin A (as $J_{X \to X'} \neq 0$ only if $\rho(X') > \rho(X)$). Note also that the weight for a particular basin A_n is assigned *without* reference to any other basin A_m; once the set of time-integrated fluxes $J_{X \to X'}$ are known and the densities sorted in descending order, the solution for each basin is straightforward, and Eq. (7) is only needed on the boundary points.

This algorithm solves several issues with the near-grid method. It requires no self-consistency, which improves the computational scaling. Moreover, the introduction of smooth functions that define the volume fraction of points in each basin produces less error and faster convergence with additional grid points. The algorithm is also readily applicable to nonuniform grids, such as an adaptive meshing scheme—it only requires computation of the Voronoi volumes and facets for the grid points. In one dimension, Eq. (1) has quadratic convergence in the grid spacing (cf. Appendix A); we now demonstrate the quadratic convergence and improved integration accuracy for three-dimensional problems.

III. EVALUATION OF NUMERICAL CONVERGENCE

One determination of the accuracy of Bader volume integration is the vanishing of the volume integration of the Laplacian of charge density $\nabla^2\rho(\mathbf{r})$. The nonzero value of

the Laplacian of charge density integration within each Bader volume is our atomic integration error and can be used as an estimate of the error of the integration of the kinetic energy density. We construct the zero flux surface of the gradients of charge density and evaluate the integration error with both the weight- and near-grid methods for several cases. First, we consider an analytic charge density with known boundaries in an orthogonal and a nonorthogonal cell. Next, we calculate real systems: an ionic compound and a semiconductor. We also evaluate the Bader charge of Na atom in a NaCl crystal by integrating the charge density within Bader volume and compare the convergence with the near-grid method. Finally, we demonstrate the improved error and computational efficiency on a large calculation: a Si self-interstitial, with a 217-atom supercell and $\sim 10^7$ grid points.

A. Gaussian densities

Figure 3 shows an example of misassignment of the grid points to basins from the near-grid method. Misassignment occurs for the grid points close to the dividing surfaces with the gradients of charge density almost parallel to the surfaces. In this example, a three-dimensional model charge density is constructed from three Gaussian functions in simple cubic unit cell, $\rho(\vec{r}) = \sum_{i=1,3} e^{(-\vec{r}-\vec{r}_i)^2/W^2}$. The $\vec{r}_i$ are $(0.25N, 0.25N, 0.4N)$, $(0.5N, 0.5N, 0.5N)$, and $(0.75N, 0.75N, 0.4N)$, with width $W = N/10$. Figure 3 shows the charge density distribution on $(1\bar{1}0)$ plane. Due to the symmetry of charge density distribution, the true dividing surfaces along charge density saddle points are known analytically and shown as two black lines on $(1\bar{1}0)$ plane. The grid points marked by orange circles are assigned to the wrong basins by the near-grid method, different from the partition of spatial points by true dividing surfaces. The gradients of charge density shown in arrows for these misassigned surface points have small normal components, and we believe this is the cause of the misassignment.

To test integration accuracy beyond simple cubic lattices, we map this model charge density onto an FCC unit cell shown in Fig. 4. The three-dimensional model charge density is constructed from three Gaussian functions, $\rho(\mathbf{r}) = \sum_{i=1,3} e^{(-\mathbf{r}-\mathbf{r}_i)^2/W^2}$. The $\mathbf{r}_i$ are located at $(0.25N, 0.25N, 0.4N)$; $(0.5N, 0.5N, 0.5N)$; $(0.75N, 0.75N, 0.4N)$ where N^3 is the number of grid points in the FCC unit cell and $W = N/10$. We vary N from 20 to 100. The Voronoi cell of the FCC lattice has 12 neighbors, where all facets have the same area. The atomic weights on every grid represent the fraction of Voronoi volume of that grid point flowing to specific atom through its neighbors. By calculating on a set of grid sizes, one obtains the maximal atomic integration errors from the near-grid method and the weight method.

Figure 4 shows a reduction in error of 3 orders of magnitude from the near-grid method. Fitting data to a nonlinear function $y = a N^{-r}$ gives a convergence rate of 0.71 for the weight method, and 0.45 for the near-grid method. The exponent of 0.71 is close to the 2/3 expected for quadratic convergence, and 0.45 is close to the 1/3 expected for linear convergence. The weight method has both better absolute
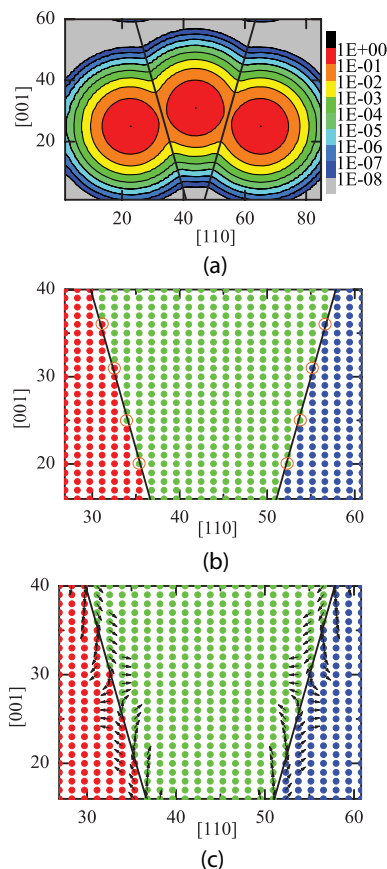

FIG. 3. Charge density distribution constructed from three Gaussian functions, and basin identification with the near-grid method and the weight method. The grid size is $N = 60$, and the charge density is shown on $(1\bar{1}0)$ plane; this is coplanar with the centers of the three Gaussian functions. The true dividing surfaces (known by symmetry) are indicated by two black lines. The basin assignment of grid points is given: red dots for ion I, green dots for ion II, and blue dots for ion III; for the weight method, a single color is assigned to the maximum weight at each point. Basin assignment from the near-grid method is given in the middle panel; basins with maximal weight on every grid points from the weight method are indicated in the bottom panel. Orange circles in the middle panel indicate the grid points misassigned by the near-grid method and corrected by the weight method. Arrows in the bottom panel denote the directions of the gradients of charge density, which can be used to verify the correctness of basin assignment.
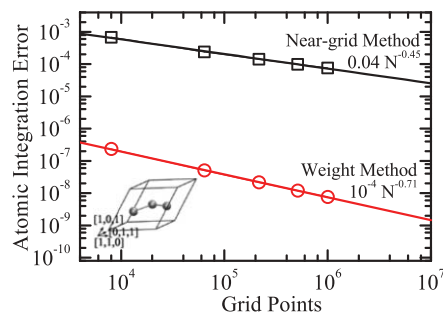

FIG. 4. Comparison of the near-grid method and the weight method on atomic integration errors in an FCC cell. The maximal atomic volume integrations of the Laplacian of charge density within Bader volumes using the near-grid method and the weight method are denoted by squares and circles, respectively. We calculate charge density grids ranging from 20^3 points to 100^3. Our algorithm gives atomic integration errors 3 orders of magnitude lower than the near-grid method and converges faster than the near-grid method.

error and converges faster than the near-grid method; in addition, there is no crossover point at large grid spacing where near-grid has smaller errors. The several orders of magnitude improvement in integration error is, in part, a consequence of the very smooth Gaussian functions. For the following cases of charge densities computed from Kohn–Sham orbitals, we will continue to see the same scaling of error with grid density, and a reduction of 1 to 2 order of magnitude in integration error. However, the difference in error reduction for charge densities computed from Kohn-Sham orbitals is less than for a smooth model system because of the larger magnitude higher-order spatial derivatives (cf. Appendix A).

B. TiO₂ bulk

For a real charge density, we perform density-functional theory (DFT) calculations on TiO_2 bulk with the projector augmented wave (PAW) (Ref. 12) method and with the generalized gradient approximation (GGA) of Perdew, Burke, and Ernzerhof (PBE)[13] for the exchange-correlation energy. DFT calculations are performed with the Vienna ab initio simulation package (VASP) (Refs. 14 and 15) using a plane-wave basis with the PAW method, with potentials generated by Kresse and Joubert.[16] Atomic configurations for Ti and O are $[Ne]3s^2 3p^6 4s^2 3d^2$ with cutoff radius 1.22 Å, and $[He]2s^2 2p^4$ with cutoff radius 0.58 Å, respectively. We use a plane-wave basis set with cutoff energy of 900 eV. The tetragonal unit cell of rutile TiO_2 (see Fig. 5) contains two Ti atoms and four O atoms. Monkhorst–Pack k-point method with $4 \times 4 \times 6$ k-points for six-atom cell is used for Brillouin-zone integration with a Gaussian smearing of 0.1 eV for electronic occupancies. Theoretically optimized lattice constant are $a = 4.649$ Å, $c = 2.970$ Å, $u = 0.305$ agreeing with experimental lattice constants of $a = 4.584$ Å, $c = 2.953$ Å, $u = 0.305$.[17] A set of charge density grids ranging from $45 \times 45 \times 30$, $60 \times 60 \times 40$, $75 \times 75 \times 50$, $90 \times 90 \times 60$, $120 \times 120 \times 80$ to $150 \times 150 \times 100$ points are calculated. For the energy cutoff of 900 eV, a grid of $45 \times 45 \times 30$ is required to eliminate wrap-around errors and is the minimum size used by an accurate VASP calculation.

Figure 5 shows maximal atomic integration errors as a function of grid sizes. The weight method gives maximal
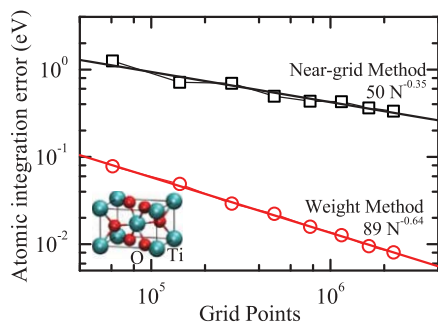

FIG. 5. Maximal atomic integration error on rutile TiO_2 with respect to the charge density grids. A set of charge density grids ranging from $45 \times 45 \times 30$ points to $150 \times 150 \times 100$ points are calculated. The weight method reports maximal atomic integration error at least 1 order of magnitude smaller than the near-grid method. The weight method is practically useful for calculations with a small grid size.
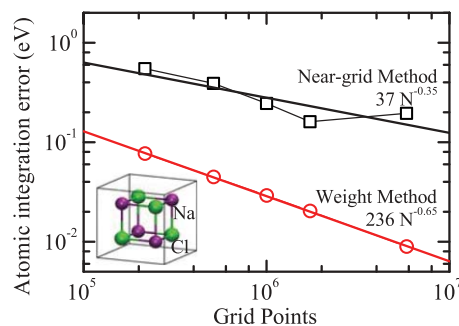

FIG. 6. Comparison of the near-grid method and the weight method for maximal atomic integration error of NaCl crystal. A set of charge density grids ranging from 60^3 points to 180^3 points are calculated. Comparing to the near-grid method, the weight method reduces the integration error remarkably.

atomic integration error 1 order of magnitude lower than the near-grid method systematically. The atomic integration error larger than 1.0 eV on the minimal grid size $45 \times 45 \times 30$ from the near-grid method is unacceptably large. Again, the convergence rate of the error goes as $\sim 2/3$ for the weight method—corresponding to quadratic convergence—and $\sim 1/3$ for the near-grid method—corresponding to linear convergence. Both the improved error and faster convergence allows for more accurate density integration with fewer grid points than near grid.

C. NaCl crystal

In this example, we evaluate the Bader charge (valence electron density integration within Bader volume) of Na atom in a NaCl crystal by integrating the charge density within Bader volume and compare the value with the near-grid method. We perform DFT calculations by use of the PAW method, the GGA of Perdew–Wang (PW91)[18] for the exchange-correlation energy. Atomic configurations for Na and Cl are $[He]2s^2 2p^6 3s^1$ with cutoff radius 0.77 Å, and $[Ne]3s^2 3p^5$ with cutoff radius 1.00 Å, respectively. A plane-wave basis set with cutoff energy of 500 eV is applied. The NaCl unit cell contains four Na atoms and four Cl atoms. Monkhorst–Pack k-point method with $3 \times 3 \times 3$ k-points for eight-atom cell is used for Brillouin-zone integration with a Gaussian smearing of 0.2 eV for electronic occupancies. The optimized lattice constant of 5.67 Å agrees with the experimental lattice constant of 5.64 Å. A set of charge density grids of 60^3, 80^3, 100^3, 120^3, to 180^3 points are calculated.

Figure 6 shows the maximal atomic integration error as a function of various grid sizes. The weight method again shows maximal atomic integration error at least 1 order of magnitude lower than the near-grid method systematically. The scaling of the error goes as the $\sim 2/3$ power for the weight method, showing continued quadratic convergence, while the near-grid method error scales as the $\sim 1/3$ power, which is linear convergence.

Figure 7 shows that Bader charge of Na atom evaluated on various charge grids. The weight method computes a Bader charge of Na atom slightly larger than the near-grid method. Fitting the data to $\rho = \rho_0 + (C/N_{grid}^\alpha)$, we find converged Bader charge values of 0.878, 0.881 e, for the
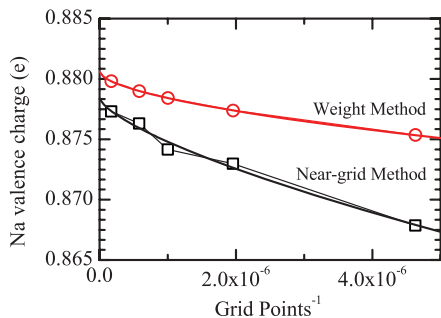
FIG. 7. Comparison of the near-grid method and the weight method on convergence of Bader charges of Na in a NaCl crystal. The Bader charge of Na is calculated for a set of density grids ranging from 60^3 points to 180^3 points. Both methods give monotonic and smooth convergence.

near-grid method and the weight method, respectively. We believe this is due to a systematic misassignment for the near-grid method, as shown for the Gaussian charge density case. This suggests that the misassignment may not be improved by increasing the density of grid points in the near-grid method. This suggests that the divide and conquer approach using continually refined grids can face potential difficulty. For 60^3 grid points, the near-grid method underestimates the Bader charge by 0.01 e, while the weight method underestimates it by 0.005 e, again showing faster convergence.

D. Si self-interstitial

For a real charge density of a large-scale (> 100) atom calculation, we study a Si tetrahedral self-interstitial with the PAW[12] method, the GGA of PW91 (Ref. [18]) for the exchange-correlation energy. Atomic configurations for Si is [Ne]$3s^2 3p^2$ with cutoff radius 1.01 Å. We use a plane-wave basis set with cutoff energy of 417 eV, giving a lattice constant of 5.4674 Å. For our $3 \times 3 \times 3$ simple cubic supercell of 216+1 atoms, we use a Monkhorst–Pack k-point method with $4 \times 4 \times 4$ with a Gaussian smearing of 0.15 eV for electronic occupancies. A set of charge density grids ranging from 128^3, 150^3, 180^3, to 210^3 points are calculated. This gives an interstitial formation energy of 4.02 eV.

Figure [8] shows maximal atomic integration errors as a function of grid sizes. The weight method gives maximal
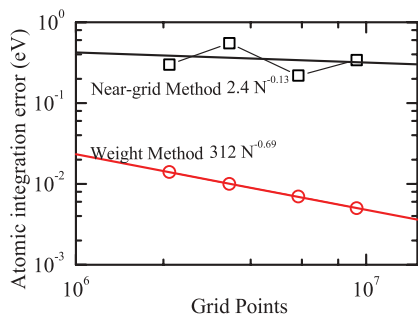

FIG. 8. Maximal atomic integration error on a 216+1 atom supercell calculation of a Si self-interstitial with respect to the charge density grids. A set of charge density grids ranging from 128^3 points to 210^3 points are calculated. The weight method reports maximal atomic integration error at least 1 order of magnitude smaller than the near-grid method.

atomic integration error 1 order of magnitude lower than the near-grid method systematically and shows the same scaling of error with number of grid points corresponding to quadratic convergence. The near-grid method has more than 1 order of magnitude larger error, but also shows unusual nonmonotonic behavior of the error for the same charge density. Such behavior cannot be due to normal integration error, but must be coming from misassignment of grid points to the wrong basin. Both the improved error, faster convergence allows for more accurate density integration with far fewer grid points than near grid.

IV. COMPUTATIONAL EFFORT

The weight method is computationally efficient, requiring overall effort that scales linearly with the number of grid points. The total computer time is comprised of two primary tasks: the sorting of charge density costs $O(N \log N)$ with N grid points and the atomic weight evaluation on the sorted grid points beginning from grid point with maximum density requires at most $N \times N_{atom}$ computer time. The computational effort is smaller than that, as only the surface grid points which have fractional atomic weights require N_{atom} calculations, while each interior grid point requires only one calculation. Generally, the number of surface grid points is a small fraction of the number of total grid points and scales as $N^{2/3}$. For example, the ratio of the number of surface grid points to the number of total grid points is 14% in NaCl with total grid sizes 60^3.

In a calculation with charge density grid sizes approaching $\sim 10^7$ grid points and hundreds of atoms in large supercells, we find our algorithm is not only more accurate, but more efficient than the near-grid method. Both methods scale linearly with the number of grid points. Figure [9] shows the linear scaling of computer time required to analyze the charge density grid for an eight-atom NaCl and a 216+1 atom Si supercell with the number of grid points. The improved efficiency of our algorithm appears to originate from the lack of a self-consistent refinement of basin assignment. Comparing to the near-grid method, which needs refinement of assignments, our weight method has small prefactor, although both are linearly scaled. The primary computational effort in both cases is the computation of weights at each grid point; hence, if only a few specific basins are needed, the computational effort can be further reduced. It is also worth noting that the algorithm is quite simple compared with trajectory following algorithms.[10]

V. CONCLUSIONS

We develop a weight method to integrate functions defined on a discrete grid over basins of attraction (such as Bader volumes) in an efficient and accurate manner. The weight method works with the density on a discrete grid and assigns volume fractions of the Voronoi cell of each grid point to surrounding basins. Starting from the local density maxima, all the grid points are sorted in density descending order. Grid points can then be fractionally weighted from the weights of its neighbors with larger density. This method depends upon
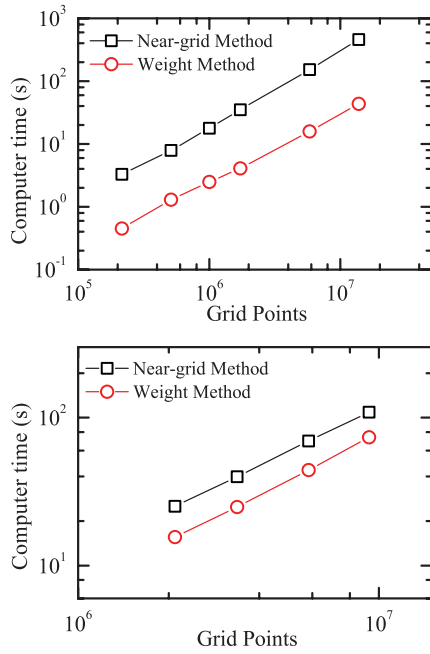
FIG. 9. Computer time required to analyze the charge density grid for an eight-atom NaCl cell (top) and a 216+1 atom Si supercell with the self-interstitial (bottom). The calculations were performed using an Intel(R)-Xeon(R) L5320, with a clockspeed of 1.86 GHz. The computer time scales linearly with respect to the number of charge density grid sizes with the weight method, as with the near-grid method. The weight method has a smaller prefactor than the near-grid method.

the formulation of flow across that dividing surfaces between the cells of two neighboring grid points and can be applied to uniform or nonuniform grids.

We perform tests on model three-dimensional charge density constructed from Gaussian functions in an FCC cell. The weight method shows that the atomic integration error is inversely proportional to the 2/3 power of grid points, while the integration error is inversely proportional to the 1/3 power of grid points using the near-grid method. We also perform tests on more realistic systems, such as TiO_2 bulk, NaCl crystal, and a Si self-interstitial. In both cases, the weight method reports maximal atomic integration error at least 1 order of magnitude smaller than the near-grid method systematically. Furthermore, we calculate the Bader charge of NaCl crystal using these two methods, both give monotonic and smooth convergence with respect to the increasing grid sizes, while they converge to slight different values, by 0.003 e. The weight method is more accurate than the near-grid method that requires very fine grids.

ACKNOWLEDGMENTS

APPENDIX: QUADRATIC ERROR IN ONE DIMENSION

The weight method for integration of the Bader charge volume has error that is quadratic in the grid spacing in one dimension. Consider the charge density $\rho(x)$ evaluated on a regular grid with spacing h. In Eq. (1), there are only two grid points where $w^A(X)$ is not exactly 0 or 1; these are the boundary points, and each is adjacent to a local minimum. In one dimension, the contributions to Eq. (1) that could produce errors linear in h only come from those points; the integration of the interior produces a total error that is quadratic in h. Hence, without loss of generality, we consider a single boundary point and show that its contribution to Eq. (1) produces an error that is of the order h^2, rather than h.

Let X be an boundary point, where the basin A lies to its left. This requires that $\rho(X - h) > \rho(X)$, and $\rho(X + 2h) > \rho(X + h)$. Finally, in order for $w^A(X)$ to not be identically 1, $\rho(X) < \rho(X + h)$. This means that there is a point $X + \delta$ for $\delta \in [0, h]$, such that $\rho'(X + \delta) = 0$. Then, the flux from Eq. (6) is

$$J_{X \to X-h} = \frac{\rho(X - h) - \rho(X)}{\rho(X - h) + \rho(X + h) - 2\rho(X)} \quad \text{(A1)}$$

and $J_{X \to X+h} = 1 - J_{X \to X-h}$; finally, as $\rho(X) < \rho(X + h)$, $J_{X+h \to X} = 0$. Then, $w^A(X - h) = 1, w^A(X + h) = 0$, and so $w^A(X) = J_{X \to X-h}$. Finally, the contribution to Eq. (1) from X is

$$\int_{X-h/2}^{X+\delta} f(x)dx \approx h J_{X \to X-h} f(X) \quad \text{(A2)}$$

To evaluate the integration error, we use a Taylor expansion for ρ and f around the grid point X. We write $\rho^{(n)} = d^n \rho / dx^n(X)$ and $f^{(n)} = d^n f / dx^n(X)$. Note that $\rho^{(1)}$ has to scale as h in order for the dividing point $X + \delta$ to lie between X and $X + h$. The Taylor expansion of $w^A(X) = J_{X \to X-h}$ from Eq. (A1) to linear order in h is

$$w^A(X) \approx \left[\frac{1}{2} - \frac{\rho^{(1)}h^{-1}}{\rho^{(2)}}\right] - h\frac{\rho^{(3)}}{6\rho^{(2)}} + O(h^2), \quad \text{(A3)}$$

so our integration contribution is

$$h\left[\frac{1}{2} - \frac{\rho^{(1)}h^{-1}}{\rho^{(2)}}\right]f(X) - h^2\frac{\rho^{(3)}}{6\rho^{(2)}}f(X) + O(h^3). \quad \text{(A4)}$$

To find the true value of the expression, we need to determine δ to at least quadratic order in h; write $\delta = \delta^{(1)}h + \delta^{(2)}h^2$, and we have

$$\rho'(X + \delta) = \rho^{(1)} + \delta \cdot \rho^{(2)} + \frac{1}{2}\delta^2 \cdot \rho^{(3)} + O(h^3)$$

$$0 = h(\rho^{(1)}h^{-1}) + h\delta^{(1)}\rho^{(2)}$$

$$+ h^2\delta^{(2)}\rho^{(2)} + \frac{1}{2}h^2(\delta^{(1)})^2\rho^{(3)} + O(h^3), \quad \text{(A5)}$$

which is solved by

$$\delta^{(1)} = -\frac{\rho^{(1)}h^{-1}}{\rho^{(2)}}$$

$$\delta^{(2)} = -\frac{(\rho^{(1)}h^{-1})^2\rho^{(3)}}{2(\rho^{(2)})^3}. \quad \text{(A6)}$$

With our quadratic approximation for δ, we can integrate $f(x)$ as

$$
\int_{X-h/2}^{X+\delta} f(x)dx = (x-X)f(X) + \frac{1}{2}(x-X)^2 f^{(1)}
$$

$$
+ O((x-X)^3)\Big|_{X-h/2}^{X+\delta}
$$

$$
= h\left[\frac{1}{2} - \frac{\rho^{(1)}h^{-1}}{\rho^{(2)}}\right]f(X) + \frac{h^2}{2}\left(\frac{\rho^{(1)}h^{-1}}{\rho^{(2)}}\right)^2
$$

$$
\times \left[f^{(1)} - f(X)\frac{\rho^{(3)}}{\rho^{(2)}}\right] + O(h^3), \qquad (A7)
$$

which agrees with the contribution from our weight integration in Eq. (A4) up to an error of order h^2. As a special case, consider $f(x) = 1/h$; then the integral

$$
\frac{1}{h}\int_{X-h/2}^{X+\delta} dx = w^A(X) + O(h) \qquad (A8)
$$

which shows that the weight is the volume fraction of the Voronoi volume belonging to basin A to first order in h.

[1] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[2] R. F. Bader, *Atoms in Molecules: A Quantum Theory* (Oxford University Press, Oxford, 1990).
[3] F. W. Biegler-König, T. T. Nguyen-Dang, Y. Tal, R. F. W. Bader, and A. J. Duke, J. Phys. B: At. Mol. Phys. **14**, 2739 (1981).
[4] P. L. A. Popelier, Comput. Phys. Commun. **93**, 212 (1996).
[5] N. O. J. Malcolm and P. L.A. Popelier, J. Comput. Chem. **24**, 1276 (2003).
[6] B. P. Uberuaga, E. R. Batista, and H. Jónsson, J. Chem. Phys. **111**, 10664 (1999).
[7] G. Henkelman, A. Arnaldsson, and H. Jónsson, Comput. Mater. Sci. **36**, 354 (2006).
[8] W. Tang, E. Sanville, and G. Henkelman, J. Phys.: Condens. Matter **21**, 084204 (2009).
[9] A. Otero-de-la-Roza and V. Luaña, J. Comput. Chem. **32**, 291 (2010).
[10] J. I. Rodríguez, A. M. Köster, P. W. Ayers, A. Santos-Valle, A. Vela, and G. Merino, J. Comput Chem. **30**, 1082 (2009).
[11] G. Voronoi, J. Reine Angew. Math. **133**, 97 (1908).
[12] P. E. Blöchl, Phys. Rev. B **50**, 17953 (1994).
[13] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[14] G. Kresse and J. Hafner, Phys. Rev. B **47**, 558 (1993).
[15] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[16] G. Kresse and D. Joubert, Phys. Rev. B **59**, 1758 (1999).
[17] U. Diebold, Surf. Sci. Rep. **48**, 53 (2003).
[18] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1992).