Electronic Supplementary Material (ESI) for RSC Advances.
This journal is © The Royal Society of Chemistry 2016

Electronic Supplementary Information for

Introducing DDEC6 Atomic Population Analysis: Part 1. Charge Partitioning Theory and Methodology

Thomas A. Manz* and Nidia Gabaldon Limas

Department of Chemical & Materials Engineering, New Mexico State University, Las Cruces, New Mexico, 88003-8001.

*E-mail: tmanz@nmsu.edu

Contents

S1. Summary of Charge Partitioning Alternatives Evaluated

We now briefly summarize some of the alternatives investigated during the course of developing the DDEC6 method. This section's purpose is to point out strategies we tried that did not perform well or that did not result in systematic improvements. This information is important to avoid duplicative efforts that could result if investigators retried these strategies without realizing they have already been tried.

Achieving good general purpose charge assignment is a balancing act of competing demands: (a) core electrons assigned to the host atom, (b) NACs with good conformational transferability, (c) chemically meaningful NACs that describe electron transfer trends (and core electron binding energy shifts in some materials), and (d) an efficiently converging polyatomic multipole expansion that reproduces the material's electrostatic potential. To approximately correlate with spectroscopic core electron binding energy shifts in transition metal compounds, the assigned atomic charge distributions should not be too delocalized. To give NACs that approximately reproduce molecular dipole moments and the electrostatic potential surrounding a material, the assigned $\{\rho_A(\vec{r}_A)\}$ should resemble their spherical averages, $\{\rho_A^{avg}(r_A)\}$. To achieve good conformational and chemical transferability among similar materials, it is preferable for the assigned $\{\rho_A(\vec{r}_A)\}$ to resemble real atoms. This can be achieved by optimizing $\{\rho_A(\vec{r}_A)\}$ to resemble reference ion densities. Therefore, we believe the most straightforward approach to achieving an even-tempered charge assignment method involves three components: (a) integrating the electron density in the local vicinity of each atomic nucleus (where 'vicinity' refers to positions close to the volume of space dominated by that atom), (b) optimizing $\{\rho_A(\vec{r}_A)\}$ to resemble their spherical averages, and (c) optimizing $\{\rho_A(\vec{r}_A)\}$ to resemble a set of reference ions.

Following these general principles, we tested a large number of new charge assignment algorithms. For various reasons, some of these charge assignment algorithms worked much better than others. Of all the algorithms we tested, the algorithm with the best overall performance was selected to be the DDEC6 method. We now briefly describe the other algorithms we tested.

In some schemes, we defined a localized net atomic charge as

$$q_A^{loc} = z_A - \oint \frac{\left(w_A(r_A)\right)^m}{\sum_{B,L}\left(w_B(r_B)\right)^m}\rho(\vec{r})d^3\vec{r}_A \qquad \text{(S1)}$$

where $w_A(r_A)$ is a DDEC-style atomic weighting factor combining a reference density and (optionally) spherical averaging with (optionally) exponential tail constraints. Different values of m > 1 were investigated. In each charge partitioning iteration, q_A^{ref} was set equal to some linear combination of q_A^{loc}, q_A, q_A^{HD}, and (optionally) other factors. The q_A^{loc} and q_A were then updated in each iteration and iterated to convergence. This type of scheme does not work well, because in materials like boron nitride the cation is more diffuse than the anion leading to $|q_A^{loc}| > |q_A|$ resulting

in increased NAC magnitude when any q_A^{loc} is included in q_A^{ref}. (These larger NAC magnitudes degraded the quality of fitting the electrostatic potential in materials like the BN sheet and nanotube.) We tried to counterbalance this by mixing in a fraction of q_A^{HD} (which usually has $|q_A^{HD}| \le |q_A|$), but this did not produce consistently good results across a wide range of materials. We also tried variations where q_A^{ref} was set to whichever was smaller in magnitude, q_A^{loc} or q_A. We also tried variations in which q_A^{ref} was adjusted for each atom until $q_A^{loc} = q_A$. We also tried various schemes in which q_A^{ref} was adjusted for each atom until q_A^{loc} preferably lay between q_A^{HD} and q_A (or between 0 and q_A), subject to the condition that q_A^{ref} should differ from q_A by no more than a preset allowance ('trust radius'). In fact, we publically released one such scheme called DDEC4 in the CHARGEMOL 3.1 version released on ddec.sourceforge.net September 29, 2014. (This version computed the DDEC3 NACs by default, but contained the option to compute DDEC4 NACs instead.) Nevertheless, our further testing revealed more advantageous approaches, which resulted in the DDEC4 algorithm being abandoned.

After extensive trials, we finally realized that continuously updating q_A^{loc} and q_A^{ref} is inherently problematic. Specifically, some of the atoms get greedy and continuously take electrons from the other atoms. To our surprise, we found the difference $q_A^{loc} - q_A$ can remain nearly constant in some materials (e.g., TiO solid) over a large number of iterations in which q_A changes by a total of >0.5 electrons. This means that a continuously updated q_A^{loc} does not provide a reliable reference value to prevent atoms from becoming greedy. We briefly tried setting q_A^{loc} equal to the Bader charge, but abandoned this strategy due to the presence of non-nuclear attractors in some materials.

We also tested more aggressive buried tail constraints in which $P(r_A) w_A(r_A)$ was constrained to decay exponentially with increasing r_A in the atom's buried tail, where $P(r_A)$ was a polynomial of r_A. We tested a few different polynomials $P(r_A)$ chosen to reproduce the limits $P(r_A = 0) = 1$ and $P(r_A \to \infty) \propto (r_A)^p$ with p > 1. (The DDEC4 method briefly mentioned above contained this type of tail constraint.) After extensive testing, we concluded the extra complexity of $P(r_A)$ did not appreciably improve results across a wide range of materials, so we reverted to the simpler strategy of constraining just $w_A(r_A)$ to decay exponentially in the atom's buried tail. We also tried various schemes for computing the decay exponents applied to $w_A(r_A)$. Ultimately, we decided to use a strategy similar to that used in the DDEC3 method plus the addition of a constraint to prevent $w_A(r_A)$ from becoming too contracted.

We also tested strategies in which q_A^{ref} was set equal to a linear combination of q_A, $q_A^{first\_loc}$ (i.e., q_A^{loc} computed in the first charge partitioning iteration), $q_A^{second\_loc}$ (i.e., q_A^{loc} computed in the second charge partitioning iteration), q_A^{loc} (computed in the current charge cycle), q_A^{HD}, and the net atomic charge computed from the iterative-Hirshfeld like partitioning using the conditioned or unconditioned charged reference ion. After extensive testing, we could not attribute any tangible benefit to the continuous updating of q_A^{ref} in each charge cycle, so we finally replaced this kind of strategy with a fixed q_A^{ref} value.

Finally, after deciding to use a fixed q_A^{ref} value for all latter charge cycles, we tested various schemes for computing the target q_A^{ref} value. We tested a scheme similar to the DDEC5 method (see description next paragraph), except $q_A^{1,Stock}, q_A^{1,Loc}, q_A^{1,ref}, q_A^{2,Stock}, q_A^{2,Loc}, q_A^{2,ref}$ were computed based on the conditioned reference densities instead of the unconditioned reference densities. We also tried schemes that employed various combinations based on both the conditioned and unconditioned reference densities. Using conditioned reference densities to compute $q_A^{1,Stock}, q_A^{1,Loc}, q_A^{1,ref}, q_A^{2,Stock}, q_A^{2,Loc}, q_A^{2,ref}$ worsened the performance. We also tested schemes in which the ratio of $q_A^{1,Stock}$ to $q_A^{1,Loc}$ to form $q_A^{1,ref}$ was set different than the ratio of $q_A^{2,Stock}$ to $q_A^{2,Loc}$ to form $q_A^{2,ref}$. However, we did not notice any appreciable improvements and so decided on the simpler scheme of keeping these two ratios the same. We also investigated variations of this ratio before settling on the value used in the DDEC5 and DDEC6 methods.

One of the schemes we developed with fixed q_A^{ref} was called the DDEC5 method. This method used $\sigma_A(r_A) = \left(\rho_A^{cond}(r_A)\right)^{\chi} \left(\rho_A^{wavg}(r_A)\right)^{1-\chi}$ with one conditioning step (i.e., c = 1) and $\chi = 1/3$ to yield $\chi_{equiv}^{DDEC5} = 1/4$. This method used the same formula for $\rho_A^{wavg}(r_A)$ and the same reference ion charges (and $\rho_A^{cond}(r_A)$) as the DDEC6 method. DDEC5 also applied the same exponential decay constraints on $w_A(r_A)$ in the fourth and later charge cycles as the DDEC6 method. DDEC5 also applied the constraint $N_A^{val} \geq 0$. We publically released DDEC5 in the CHARGEMOL 3.2.1 version released on ddec.sourceforge.net August 12, 2015. (This version computed the DDEC3 NACs by default, but contained the option to compute DDEC5 NACs instead.) While the DDEC5 method performed well, it did not have a provably convex optimization functional or provably unique solution. We also extensively tested one algorithm with the same form as DDEC5, except using two conditioning steps (i.e., c = 2) and $\chi = 1/2$ to yield $\chi_{equiv} = 1/4$. This algorithm did not converge for the ozone+1 B3LYP system. This led us to believe the DDEC5 method might not converge for some materials, due to its optimization functional not being provably convex. Desiring a proof of unique convergence, we then developed the Convex functional and later the DDEC6 method that have proven unique solutions.

We also tested schemes similar to those described above, but differing in parameter values such as the localization exponent m, the precise formulation of the $w_A(r_A)$ tail constraints, etc. We also tested a few schemes that are quite different from those described above. A few additional schemes computed q_A^{loc} based on atom-atom overlap populations (computed via various schemes) instead of based on Eq. (S1) above. However, these were more computationally expensive than Eq. (S1), and we did not discern any performance improvements compared to Eq. (S1). We also tried schemes in which the AIM charge distribution $\rho_A(\vec{r}_A)$ was computed by using linear combinations of $\rho(\vec{r})w_A(r_A)/W(\vec{r})$ and $\rho(\vec{r})(w_A(r_A))^m / \sum_{B,L}(w_B(r_B))^m$ with m > 1 or other localization schemes instead of $(w_A(r_A))^m$.

While this list is not comprehensive of all of the charge partitioning algorithms we tested, it provides a general idea of the types of charge partitioning schemes we tested. In the end, we settled on the DDEC6 charge partitioning method, because it provided consistently good results across a wide range of material types.

In addition, we performed a large number of tests regarding optimization of the computational cost. In addition to computational tests on various materials, we developed iteration calculus with associated algebraic models (solved analytically) and finite difference numerical models (solved in spreadsheets) that accurately predicted and described the convergence performance of various computational algorithms. We used these mathematical models to derive the optimal parameters leading to fast and robust convergence. Using this iteration calculus, we designed efficient convergence accelerators (see Section S3.2.3 below) that optimize the convergence speed for self-consistent schemes. Our computational tests confirmed the theoretically derived optimal parameters and performance improvements associated with these convergence accelerators. These improvements ultimately led to the number of required charge cycles being reduced from <200 for the DDEC3 method to ~7 for the DDEC6 method. We believe that ~7 charge cycles is close to the minimum of what can be used to consistently obtain accurate results.

S2. Integration Routines Employed

S2.1 General Overview

In the limit of an arbitrarily fine grid spacing and sufficiently large cutoff radii, the converged DDEC6 properties should be independent of the specific choice of integration routine. The choice of integration routine primarily effects the computational efficiency and precision. The optimal integration routine depends on the type of input information available. A uniformly spaced grid is a convenient choice for quantum chemistry calculations using planewave basis sets, because this type of grid naturally lends itself to computing the electron and spin density grids via Fourier transform from the planewave coefficients. In general, using a uniformly spaced grid for charge partitioning is convenient when the quantum chemistry program (VASP, ONETEP, GPAW, etc.) used to generate the electron and spin distributions also uses this same grid type. A uniformly spaced grid is not the most computationally efficient choice for quantum chemistry calculations using

Gaussian basis sets. For quantum chemistry calculations using Gaussian basis sets, computationally efficient atom-centered overlapping[S1] and non-overlapping[S2] grid types have been extensively described in the literature. Nevertheless, we used a uniformly spaced grid for all DDEC6 calculations described in this work. This was motivated by the fact that atom-centered overlapping and non-overlapping grids have not yet been programmed into the CHARGEMOL program used to compute the DDEC6 properties.

When using uniformly spaced grids, it is sometimes best to integrate core-like and valence-like electron distributions separately. Here, the term core-like electron distribution refers to an electron distribution concentrated near atomic nuclei. Core-like electron distributions can have extremely high density values near atomic nuclei. The term valence-like electron distribution refers to an electron distribution that has a significant fraction of its electrons in the atomic valence regions without extreme density spikes near the atomic nuclei. Unless special precautions are taken, the extreme density spikes near atomic nuclei in core-like electron distributions can lead to inaccurate integration of the number of core-like electrons. Using an extremely fine uniform grid to integrate the core-like electron distribution is one possible strategy, but this strategy would be too computationally expensive and impractical. Instead, we integrate the core-like and valence-like electron distributions separately. Then we correct the core-like density grid to force it to integrate to the correct number of core-like electrons. With this correction in place, accurate integrations over the core-like density grid can be performed. The following Section S2.2 describes details of this core grid correction.

During DDEC analysis of VASP PAW quantum chemistry calculations, the precision of integrating valence-like electron distributions was improved using the valence occupancy correction and all-electron spin density approximation described in the Supporting Information Section E (pages S9–S10) of Manz and Sholl.[S3]

During DDEC analysis of GAUSSIAN 09 generated wfx files, the precision of integrating electron and spin distributions and multipole moments was improved using the valence occupancy corrections described in the Supporting Information Section F (pages S10–S11) of Manz and Sholl.[S3] In the present work, we made two additional improvements in computational efficiency. First additional improvement in computational efficiency: Each Gaussian basis set product has the general form $\left(X-X_0\right)^{\ell_1}\left(Y-Y_0\right)^{\ell_2}\left(Z-Z_0\right)^{\ell_3}\exp\left(-\alpha\left|\vec{r}-\vec{r}_0\right|^2\right)$. Here, $\vec{r}_0=\left(X_0,Y_0,Z_0\right)$ represents the center of the Gaussian basis set product. The powers $\left(\ell_1,\ell_2,\ell_3\right)$ are non-negative integers. To improve computational efficiency, we sorted all Gaussian basis set products into blocks where Gaussian basis set products in each block shared the same α and $\vec{r}_0$. The $\exp\left(-\alpha\left|\vec{r}-\vec{r}_0\right|^2\right)$, $\left(X-X_0\right),\left(Y-Y_0\right)$, and $\left(Z-Z_0\right)$ terms were computed only once for each block at each grid point. This produced computational savings by avoiding recomputing these terms for every Gaussian basis set product within each block. Second additional improvement in computational efficiency: We added a grid interpolation scheme to increase the computational efficiency of generating valence, core, and spin density grids from Gaussian basis set coefficients. This grid

interpolation decreases the computational cost by a factor of approximately five or more with negligible impact on the computational precision. Specifically, we used a set of grids explicitly including every n^{th} grid point along each lattice direction, with n = 1 (finest), 2, 3, 4, 6, 8, or 12 (coarsest). The finest grid (i.e., n = 1) had a uniform spacing of ~0.14 bohr. Each Gaussian basis set product was assigned to one of these grids according to how diffuse it was. Specifically, a Gaussian basis set product proportional to $\exp\left(-\alpha\left|\vec{r}-\vec{r}_0\right|^2\right)$ was assigned to grids according to the following scheme:

(1) If $\alpha > 5$ (atomic units), the Gaussian basis set product was assigned to the core-like density grid with analytic integration to compute the occupancy corrections. This grid had a uniform spacing of ~0.14 bohr.

(2) If $5 \geq \alpha > 0.4$ (atomic units), the Gaussian basis set product was assigned to the n=1 valence density grid. This grid had a uniform spacing of ~0.14 bohr.

(3) If $\frac{0.16}{n_i^{\,2}} \geq \alpha > \frac{0.16}{\left(n_{i+1}\right)^2}$ (atomic units), the Gaussian basis set product was assigned to the n_i valence density grid, where $n_i = 2,3,4,6,8$. The Gaussian basis set product was assigned to the n=12 (coarsest) valence density grid if $\left(0.16/144\right) \geq \alpha$ (atomic units). This scheme scales the coarseness of the grid in the exact same manner that α scales. This means the 'relative coarseness' of the grid remains approximately constant independent of the α value.

The valence and spin density contributions for each Gaussian basis set product were computed on the corresponding assigned n^{th} grid. For each Gaussian basis set product, a renormalization factor of up to ±5% was applied to ensure it integrated to the proper value over the grid.[S3] After all Gaussian basis set products were computed over the corresponding grids, the coarser grids were interpolated back onto the finer grids in the following order: (a) the n = 12 grid was interpolated back onto the n = 6 grid, (b) the n = 8 grid was interpolated back onto the n = 4 grid, (c) the n = 6 grid was interpolated back onto the n = 3 grid, (d) the n = 4 grid was interpolated back onto the n = 2 grid, (e) the n = 3 grid was interpolated back onto the n = 1 grid, and (f) the n = 2 grid was interpolated back onto the n = 1 grid. This has the effect of interpolating the n = 12 grid onto the n = 1 grid by first interpolating the n = 12 grid onto the n = 6 grid, then interpolating the n = 6 grid onto the n = 3 grid, and finally interpolating the n = 3 grid onto the n = 1 grid. By the sequence of steps (a) to (f), all of the coarser grids were finally interpolated onto the n = 1 grid. A linear interpolation was used in each of steps (a) to (f). Such a linear interpolation yields the same integral of each Gaussian basis product over the coarser and finer grids.

In this work, we used a 5 Å cutoff radius for $w_A\left(r_A\right)$, which means $\rho_A\left(\vec{r}_A\right) = 0$ for $r_A > 5$ Å. For all charge distributions depending only on r_A (i.e., spherically symmetric distributions), we used 100 radial shells evenly spaced between 0 and 5 Å.

S2.2 Core Electron Partitioning with Core Grid Correction

S2.2.1 Overview

Core electron partitioning with core grid correction assigns core-like electron distributions, $\left\{\rho_A^{core}(\vec{r}_A)\right\}$, that integrate to yield the exact analytic number of core-like electrons for each atom

$$\oint \rho_A^{core}(\vec{r}_A)\, d^3\vec{r}_A = N_A^{core}. \qquad \text{(S2)}$$

Here, N_A^{core} refers to the exact analytic number of core-like electrons that have been included in the core-like electron distribution, $\rho^{core}(\vec{r})$. In general, this depends upon the specific density grid setups not the chemical states of atoms. For example, a Mg atom could have N_A^{core} set to 0, 2, 10, or other values, depending on how many core-like electrons were written to the core density grid.

Core grid correction is not necessary to compute accurate NACs. (When comparing NACs computed with to without this core grid correction, the NACs typically change by up to ~0.002 e due to integration artifacts arising from the finite grid spacing.) For example, the paper introducing the DDEC3 method did not use core grid correction.[S3] The primary reason for including core grid correction is that it allows $\left\{\rho_A(\vec{r}_A)\right\}$ to be integrated to yield N_A without worrying about errors in the integrated number of core-like electrons. This is critical for evaluating quantities that are nonlinear functionals of $\left\{\rho_A(\vec{r}_A)\right\}$. Bond orders quantify the number of electrons exchanged between two atoms. Because the exchange interaction is a nonlinear functional of the electron density, integrals for computing bond orders must be based on $\rho_A(\vec{r}_A)$ not just the atomic valence density $\rho_A^{val}(\vec{r}_A)$. This requires that $\rho_A(\vec{r}_A)$ integrate to the correct number of electrons—hence the need for a core grid correction.

If an atomic nucleus falls directly on a grid point, the density at that grid point may be very high. During integration, the number of electrons contributed by a pixel is calculated as the electron density at that pixel times the pixel volume. For a pixel centered on an atomic nucleus, the average electron density in the volume occupied by the pixel is less than the electron density exactly at the nuclear position. Therefore, grid points centered directly at atomic nuclei will produce integration errors if the density is taken to be that at the nuclear position. This error can be removed by estimating and using the average density for each pixel volume in place of the point density at the nuclear position.

S2.2.2 Design Criteria

a) The core-like density assigned to each atom should integrate to the correct number of core-like electrons within a specified convergence tolerance (e.g., 10^{-5} e). For example, if a calculation is performed with 2 core electrons in Mg, the assigned core density for this atom should integrate to between 1.99999 and 2.00001 e. This is done by correcting the core density for pixels with the highest core density (i.e., the nuclear cusps).

b) The core grid correction should never produce a negative core-like electron density for any grid point.

c) For a particular atom, the core grid correction should not change the relative ordering of grid point core-like densities. Specifically, if grid point 1 contains a higher core-like density assigned to atom A than grid point 2, then after the correction is applied this should still be the case.

d) Because nuclear cusps contribute most of the integration error, the core grid correction should be localized to those grid points with the highest core-like densities (i.e., those closest to atomic nuclei).

e) The core grid correction should not interfere with the exponential decay constraint applied to the $\left\{w_A^{core}\left(r_A\right)\right\}$. Recall that atomic core densities decay at least as fast as exp(-2r_A) where r_A is in bohr. This constraint is applied during the core partitioning. Consider two grid points near nucleus A such that grid point 1 is closer to nucleus A than grid point 2. Then,

$$w_A^{core}\left(r_2\right) \le w_A^{core}\left(r_1\right)\exp\left(-2\left(r_2 - r_1\right)\right) \qquad \text{(S3)}$$

where r_1 and r_2 are the distances from grid points 1 and 2, respectively, to nucleus A.

S2.2.3 Iterative Algorithm

Two separate sets of iterations are performed: (i) a first set of iterations to determine a pre-corrected $w_A^{core}\left(r_A\right)$ and (ii) a second set of iterations to correct the core-like electron density grid to yield the correct number of core-like electrons for each atom.

S2.2.3.1 Iterations to determine a pre-corrected $w_A^{core}\left(r_A\right)$

For atoms having no core-like electrons assigned to the core grid (e.g., if $N_A^{core} < 10^{-10}$), the assigned $w_A^{core}\left(r_A\right)$ and $\rho_A^{core,avg}\left(r_A\right)$ are set to zero. For all remaining atoms, the following sequence of steps is performed starting with the initial estimate $w_A^{core}\left(r_A\right) = \rho_A^{ref}\left(r_A, q_A^{ref} = 0\right)$.

a) For each grid point:

$$W^{core}\left(\vec{r}\right) = \sum_{A,L} w_A^{core}\left(r_A\right) \qquad \text{(S4)}$$

b) The spherical average core density is computed for each atom,

$$\rho_A^{core,avg}\left(r_A\right) = \left\langle \frac{w_A^{core}\left(r_A\right)}{W^{core}\left(\vec{r}\right)} \rho^{core}\left(\vec{r}\right) \right\rangle_{r_A} \qquad \text{(S5)}$$

c) If $\oint \rho_A^{core,avg}\left(r_A\right) d^3\vec{r}_A$ changes between successive core iterations are $< 10^{-5}$ e for every atom and at least five prior core iterations have been performed, the calculation is considered converged and exits. Otherwise, the calculation continues.

d) Starting with $w_A^{core}\left(r_A\right) = \rho_A^{core,avg}\left(r_A\right)$ as the initial guess, $w_A^{core}\left(r_A\right)$ is updated to satisfy constraint (S3) by recursively setting

$$w_A^{core}\left(r_A\right) = \min\left(w_A^{core}\left(r_A\right), w_A^{core}\left(r_A - \Delta r_A\right)\exp\left(-2\Delta r_A\right)\right) \qquad \text{(S6)}$$

beginning with the second radial shell and continuing outward to the last radial shell. The calculation then repeats the sequence of steps b) to d) until it converges and exits in step c).

S2.2.3.2 Iterations to correct the core density grid

For atoms having no core-like electrons assigned to the core grid (e.g., if $N_A^{core} < 10^{-10}$), the assigned $w_A^{core}(r_A)$ and $\rho_A^{core,avg}(r_A)$ are set to zero. For all remaining atoms, the following sequence of steps is performed to correct the core grid.

a) In each correction iteration i, a real variable K_A is computed for each atom using the following equations

$$\rho_A^{core}(\vec{r}_A)\big|_i = \frac{w_A^{core}(r_A)\big|_i}{W^{core}(\vec{r})\big|_i}\rho^{core}(\vec{r})\big|_i \qquad \text{(S7)}$$

$$K_A\big|_i = \min\left(\left(\frac{N_A^{core} - \oint \rho_A^{core}(\vec{r}_A)\big|_i \, d^3\vec{r}_A}{\oint\left(\rho_A^{core}(\vec{r}_A)\big|_i\right)^3 d^3\vec{r}_A}\right), \frac{0.25}{\left(\rho_A^{core}\big|_i(\max)\right)^2}\right) \qquad \text{(S8)}$$

where $\rho_A^{core}\big|_i(\max)$ is the largest value of $\rho_A^{core}(\vec{r}_A)\big|_i$ over the set of all grid points.

b) $\rho^{core}(\vec{r})$ is updated by

$$\tilde{\rho}_A^{core}(\vec{r}_A)\big|_{i+1} = \frac{\rho_A^{core}(\vec{r}_A)\big|_i}{\sqrt{1 - 2K_A\big|_i\left(\rho_A^{core}(\vec{r}_A)\big|_i\right)^2}} \qquad \text{(S9)}$$

$$\rho^{core}(\vec{r})\big|_{i+1} = \sum_{A,L}\tilde{\rho}_A^{core}(\vec{r}_A)\big|_{i+1}. \qquad \text{(S10)}$$

c) $\rho_A^{core,avg}(r_A)$ is updated by

$$\rho_A^{core,avg}(r_A)\big|_{i+1} = \left\langle \frac{w_A^{core}(r_A)\big|_i}{W^{core}(\vec{r})\big|_i}\rho^{core}(\vec{r})\big|_{i+1} \right\rangle_{r_A} \qquad \text{(S11)}$$

d) If $\left|N_A^{core} - \oint \rho_A^{core,avg}(r_A)\big|_{i+1} d^3\vec{r}_A\right| < 10^{-5}$ for every atom, the calculation is considered converged and exits. Otherwise, the calculation continues.

e) Starting with $w_A^{core}(r_A)\big|_{i+1} = \rho_A^{core,avg}(r_A)\big|_{i+1}$ as the initial guess, $w_A^{core}(r_A)\big|_{i+1}$ is updated to satisfy constraint (S3) by recursively setting

$$w_A^{core}(r_A)\big|_{i+1} = \min\left(w_A^{core}(r_A)\big|_{i+1}, w_A^{core}(r_A)\big|_{i+1}\exp(-2\Delta r_A)\right) \qquad \text{(S12)}$$

beginning with the second radial shell and continuing outward to the last radial shell.

f) $W^{core}(\vec{r})$ is updated:

$$W^{core}(\vec{r})\big|_{i+1} = \sum_{A,L} w_A^{core}(r_A)\big|_{i+1} \qquad \text{(S13)}$$

The calculation then repeats the sequence of steps a) to f) until it converges and exits in step d).

S2.2.4 Proof this Iterative Algorithm Satisfies the Design Criteria

a) The iterative scheme converges to the desired solution. *Proof*: Near the solution, we have

$$K_A\big|_i = \frac{N_A^{core} - \oint \rho_A^{core}\left(\vec{r}_A\right)\big|_i \, d^3\vec{r}_A}{\oint \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^3 d^3\vec{r}_A} \text{ and } \left|K_A\big|_i \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^2\right| \ll 1 \qquad \text{(S14)}$$

Therefore, we can expand Eq. (S9) as a Taylor series to give

$$\tilde{\rho}_A^{core}\left(\vec{r}_A\right)\big|_{i+1} = \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i + \left(\frac{N_A^{core} - \oint \rho_A^{core}\left(\vec{r}_A\right)\big|_i \, d^3\vec{r}_A}{\oint \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^3 d^3\vec{r}_A} \right) \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^3 \right) + \text{residual} \qquad \text{(S15)}$$

Integrating Eq. (S15) yields

$$\oint \tilde{\rho}_A^{core}\left(\vec{r}_A\right)\big|_{i+1} d^3\vec{r}_A = N_A^{core} + \text{residual} \qquad \text{(S16)}$$

which shows $\oint \tilde{\rho}_A^{core}\left(\vec{r}_A\right)\big|_{i+1} d^3\vec{r}_A$ converges to N_A^{core}. Combining Eqs. (S9) and (S16) gives

$$\oint \left(\frac{\rho_A^{core}\left(\vec{r}_A\right)\big|_i}{\sqrt{1 - 2K_A\big|_i \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^2}} \right) d^3\vec{r}_A = N_A^{core} + \text{residual} \qquad \text{(S17)}$$

which shows the left side of Eq. (S17) converges to N_A^{core}. This can only be true if $\lim_{i\to\infty} K_A = 0$, because otherwise $\rho_A^{core}\left(\vec{r}_A\right)\big|_i$ would increase ($\lim_{i\to\infty} K_A > 0$) or decrease ($\lim_{i\to\infty} K_A < 0$) without bound. With Eq. (S8) this means

$$\lim_{i\to\infty} \oint \rho_A^{core}\left(\vec{r}_A\right)\big|_i \, d^3\vec{r}_A = N_A^{core} \qquad \text{(S18)}$$

$$\lim_{K_A\to 0} \tilde{\rho}_A^{core}\left(\vec{r}_A\right)\Big|_i = \rho_A^{core}\left(\vec{r}_A\right)\big|_i \qquad \text{(S19)}$$

so the iterative process converges to the desired solution.

b) The corrected core density is nonnegative at every grid point. *Proof*: The minimum of the factor $\sqrt{1 - 2K_A\big|_i \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^2}$ occurs when $K_A\big|_i > 0$ and for the grid point $\rho_A^{core}\left(\vec{r}_A\right)\big|_i(\max)$. From Eq. (S8), it follows $K_A\big|_i \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^2 \le 0.25$. Therefore, $\sqrt{1 - 2K_A\big|_i \left(\rho_A^{core}\left(\vec{r}_A\right)\big|_i\right)^2} \ge \sqrt{1/2}$. Examining Eq. (S9), this means $\tilde{\rho}_A^{core}\left(\vec{r}_A\right)\big|_{i+1} \ge 0$.

c) For a particular atom, the correction does not change the relative ordering of grid point core densities. *Proof*: Consider the function

$$\Xi(s) = \frac{s}{\sqrt{1-2Ks^2}} \qquad \text{(S20)}$$

which has the derivative

$$\frac{d\Xi}{ds} = \frac{1}{\left(1-2Ks^2\right)^{3/2}} \qquad \text{(S21)}$$

$\Xi(s)$ is a monotonically increasing function of s over the range $Ks^2 < 0.5$. Because Eq. (S9) has the functional form $\Xi(s)$, the relative ordering of grid point core densities is preserved for each atom.

d) <u>The correction is localized to those grid points with the highest core densities (i.e., those closest to atomic nuclei)</u>. *Proof*: Combining

$$\rho^{core}(\vec{r})\big|_i = \sum_{A,L} \rho_A^{core}(\vec{r}_A)\big|_i \qquad \text{(S22)}$$

with Eq. (S10) gives

$$\rho^{core}(\vec{r})\big|_{i+1} - \rho^{core}(\vec{r})\big|_i = \sum_{A,L} \rho_A^{core}(\vec{r}_A)\big|_i \left(\frac{1}{\sqrt{1-2K_A\big|_i \left(\rho_A^{core}(\vec{r}_A)\big|_i\right)^2}} - 1 \right) \qquad \text{(S23)}$$

Consider the function

$$g(s) = s\left(\frac{1}{\sqrt{1-2Ks^2}} - 1 \right) = \Xi(s) - s \qquad \text{(S24)}$$

which has the derivative

$$\frac{dg}{ds} = \frac{1}{\left(1-2Ks^2\right)^{3/2}} - 1. \qquad \text{(S25)}$$

$\frac{dg}{ds} = 0$ if and only if K = 0 or s=0. Moreover, $g(s)$ is a monotonically increasing (when K > 0) or decreasing (when K < 0) function of s over the range $Ks^2 < 0.5$. For small s, $g(s)$ expands a Taylor series to

$$g(s) = Ks^3 + \text{residual} \le \frac{0.25}{\left(s_{max}\right)^2} s^3 + \text{residual} \qquad \text{(S26)}$$

The inequality on the right-most side of Eq. (S26) arises from Eq. (S8). Due to the cubic dependence of $g(s)$ on s for small s, the points with largest $\rho_A^{core}(\vec{r}_A)\big|_i$ dominate the core correction for atom A. These points are typically located close to nucleus A.

e) <u>The correction does not interfere with the exponential decay constraint applied to $\left\{w_A^{core}(r_A)\right\}$</u>. Consider two grid points near nucleus A such that grid point 1 is closer to nucleus A than grid point 2. From Eq. (S9),

$$\ln\left(\frac{\tilde{\rho}_A^{core}(\vec{r}_1)\big|_{i+1}}{\tilde{\rho}_A^{core}(\vec{r}_2)\big|_{i+1}}\right)=\ln\left(\frac{\rho_A^{core}(\vec{r}_1)\big|_i}{\rho_A^{core}(\vec{r}_2)\big|_i}\right)+\frac{1}{2}\ln\left(\frac{1-2K_A\big|_i\left(\rho_A^{core}(\vec{r}_2)\big|_i\right)^2}{1-2K_A\big|_i\left(\rho_A^{core}(\vec{r}_1)\big|_i\right)^2}\right) \qquad \text{(S27)}$$

***case 1*:** $K_A>0$. In this case, the last term in Eq. (S27) increases $\ln\left(\frac{\tilde{\rho}_A^{core}(\vec{r}_1)\big|_{i+1}}{\tilde{\rho}_A^{core}(\vec{r}_2)\big|_{i+1}}\right)$ so the core density increase performed during the core grid correction does not cause $\ln\left(\frac{\tilde{\rho}_A^{core}(\vec{r}_1)\big|_{i+1}}{\tilde{\rho}_A^{core}(\vec{r}_2)\big|_{i+1}}\right)$ to be $\le\exp\left(2\left(r_2-r_1\right)\right)$.

***case 2*:** $K_A<0$. This occurs when the core density grid assigns too much core density to atom A. This can be due to a nucleus falling directly on a grid point, in which case the core density near this nucleus is too high and Eq. (S27) appropriately decreases it. Alternatively, this case can arise if the assigned core density is too diffuse for any reason. The lowering of core density near this nucleus together with constraint (S3) corrects the problem of too much core density being assigned to this atom.

S2.2.5 What features cause this scheme to converge rapidly and robustly?

a) The correction is localized to regions with highest $\rho_A^{core}(\vec{r}_A)$ where typically $w_A(r_A)/W(\vec{r})\approx 1$. This makes corrections for different atoms almost independent of each other.

b) The relative ordering of core density values for an atom is preserved. This ensures a smooth behavior.

c) Convergence is rapid near the solution, as evidenced by the Taylor series expansion in Eqs. (S15) and (S16).

d) Examining Eqs. (S8) and (S9), the density changes are bounded by

$$\frac{1}{\sqrt{3}}\le\frac{\tilde{\rho}_A^{core}(\vec{r}_A)\big|_{i+1}}{\rho_A^{core}(\vec{r}_A)\big|_i}\le\sqrt{2} \qquad \text{(S28)}$$

The extreme values occur for the grid point corresponding to $\rho_A^{core}\big|_i(\max)$ when $\oint\left(\rho_A^{core}(\vec{r}_A)\big|_i\right)^3 d^3\vec{r}_A$ is dominated by $\rho_A^{core}\big|_i(\max)$ such that $\oint\left(\rho_A^{core}(\vec{r}_A)\big|_i\right)^3 d^3\vec{r}_A\approx\left(\rho_A^{core}\big|_i(\max)\right)^3\times V_{pixel}$ (where V_{pixel} is the pixel volume) and under the condition that $\left|N_A^{core}-\oint\rho_A^{core}(\vec{r}_A)\big|_i\,d^3\vec{r}_A\right|$ is large. Under these conditions, $K_A>0$ gives the limiting behavior

$$\tilde{\rho}_A^{core}\left(\vec{r}_A\right)\Big|_{i+1}\left(max\right)=\frac{\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)}{\sqrt{1-2\dfrac{0.25}{\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)^2}\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)^2}}. \qquad (S29)$$

$$=\sqrt{2}\,\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)$$

Under these conditions, $K_A<0$ gives the limiting behavior

$$\tilde{\rho}_A^{core}\left(\vec{r}_A\right)\Big|_{i+1}\left(max\right)=\frac{\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)}{\sqrt{1-2\dfrac{\left(N_A^{core}-\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)V_{pixel}\right)}{\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)^3 V_{pixel}}\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)^2}} \qquad (S30)$$

$$\approx\frac{\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)}{\sqrt{3}}$$

when $\left(\rho_A^{core}\left(\vec{r}_A\right)\Big|_i\left(max\right)\right)V_{pixel}\gg N_A^{core}$.

e) Noting that $\left(\sqrt{2}\right)^{20}=1024$, this means about 20 iterations are required to increase a grid point density by a factor of 10^3. Noting that $\left(\sqrt{3}\right)^{13}=1262.665$, this means about 13 iterations are required to decrease a grid point density by a factor of 10^3. Because the approach to convergence is smooth and the core density assigned to each grid point is never off by more than a factor of 10^6, this means convergence is always achieved in fewer than 40 iterations. In practice, convergence is nearly always achieved in fewer than 20 iterations.

S3. Computational Algorithm for Convex Functional

S3.1 Iterative Algorithm

The complex functional was optimized using the following procedure. First, q_A^{ref} was computed in the first two charge cycles as described in Section 3.2 of the main text. Second, the conditioned reference ion density, $\rho_A^{cond}\left(r_A\right)$, was computed in the third charge cycle as described in Section 3.3 of the main text.

The fourth charge cycle used the following procedure to compute the fixed reference density

$$\rho_A^{fixed\_ref}\left(r_A\right)=H_A\left(r_A\right) \qquad (S31)$$

in the Convex functional. First, we computed

$$\sigma_A\left(r_A\right)=\rho_A^{cond}\left(r_A\right)\left\langle\rho\left(\vec{r}\right)/\rho^{cond}\left(\vec{r}\right)\right\rangle_{r_A}. \qquad (S32)$$

Then, $\sigma_A\left(r_A\right)$ is reshaped to form $G_A\left(r_A\right)$ exactly as described in item i) in Section 3.4 of the main text. Then, $G_A\left(r_A\right)$ is reshaped to form $H_A\left(r_A\right)$ exactly as described in item ii) of Section 3.4 of the main text.

The fifth (i.e., $i=5$) and latter charge cycles form an iterative process to achieve a self-consistent solution. The fifth charge cycle starts with the following initial estimates: $w_A^5(r_A)=\rho_A^{fixed\_ref}(r_A)$, $\kappa_A^5=0$, and $C_A^5=1$. These are refined to self-consistency using the following sequence of steps in the fifth and latter charge cycles:

1. In the first loop over grid points and atoms, the following sum is computed at each grid point:
$$W(\vec{r})=\sum_{B,L} w_B(r_B) \qquad \text{(S33)}$$

2. In the second loop over grid points and atoms, the following quantities are computed:
$$\rho_A(\vec{r}_A)=w_A(r_A)\rho(\vec{r})/W(\vec{r}) \qquad \text{(S34)}$$
$$\rho_A^{avg}(r_A)=\left\langle \rho_A(\vec{r}_A)\right\rangle_{r_A} \qquad \text{(S35)}$$
$$N_A=\oint \rho_A(\vec{r}_A)d^3\vec{r}_A \qquad \text{(S36)}$$
$$u_A=2\oint\left(1-\frac{w_A(r_A)}{W(\vec{r})}\right)\left(1+\frac{w_A(r_A)}{W(\vec{r})}\right)\rho_A(\vec{r}_A)d^3\vec{r}_A \qquad \text{(S37)}$$
All of these quantities except $\{\rho_A(\vec{r}_A)\}$ are stored.

3. If the N_A and $\{\rho_A^{avg}(r_A)\}$ changes between successive charge cycles were less than 10^{-5}e and 10^{-5}e/bohr^3, respectively, for each atom two consecutive times in a row then the calculation is considered converged. Starting with the 10th charge cycle, the calculation breaks at this point if it is considered converged. If it is not converged or the charge cycle is less than 10, the calculation proceeds to # 4 below.
4. At the end of the ith charge cycle $(i\geq 5)$, the updated atomic weighting factors are given by
$$w_A^{i+1}(r_A)=C_A^i\, e^{\kappa_A^{i+1}}\sqrt{\rho_A^{fixed\_ref}(r_A)\rho_A^{avg}(r_A)} \qquad \text{(S38)}$$
where
$$\kappa_A^{i+1}=\left(\max\left(0,\kappa_A^i-const\cdot\left(N_A^{val}/u_A\right)\right)\right) \qquad \text{(S39)}$$
if $u_A\geq 10^{-7}$, and $\kappa_A^{i+1}=0$ if $u_A<10^{-7}$. The constant appearing in Eq. (S39) affects only the convergence speed and robustness without affecting the converged solution. The approximately optimal value of $2-\sqrt[3]{4}$ is derived in Section S3.2.2 below. The

convergence accelerator, C_A^i, minimizes the number of required charge cycles without changing the converged solution. For $i \geq 6$,

$$C^i = \frac{e^{\kappa_A^{i+1}} \sqrt{\rho_A^{fixed\_ref}(r_A)\rho_A^{avg}(r_A)}}{\left(2-\sqrt{2}\right)e^{\kappa_A^{i+1}} \sqrt{\rho_A^{fixed\_ref}(r_A)\rho_A^{avg}(r_A)} + \left(\sqrt{2}-1\right)w_A^i(r_A)} . \qquad \text{(S40)}$$

As explained in Section S3.2 below, this form of the convergence accelerator maximizes the convergence speed. The convergence accelerator equals one when the calculation converges.

After generating $w_A^{i+1}(r_A)$ for each atom, the calculation returns to step #1 above and starts the next (i.e., $(i+1)$) charge cycle using $w_A^{i+1}(r_A)$ as the new estimate for $w_A(r_A)$ in Eq. (S33). This iterative process is continued until the calculation satisfies the convergence criteria and breaks in step #3 above.

We now show that convergence of $\{N_A\}$ and $\{\rho_A^{avg}(r_A)\}$ can occur only if $\{\kappa_A\}$ are also converged. Following Manz and Sholl,[S3] we define

$$J_{AB} = \frac{\partial N_A}{\partial \kappa_B} = \oint \left(\frac{w_A(r_A)}{W(\vec{r})}\delta_{AB} - \sum_L \frac{w_A(r_A)w_B(r_B)}{\left(W(\vec{r})\right)^2} \right) \rho(\vec{r})\, d^3\vec{r} . \qquad \text{(S41)}$$

The $\sum_L$ in Eq. (S41) accounts for periodic images (if any) of atom B. For a process in which $\{\rho_A^{avg}(r_A)\}$ are converged, changes in $\{w_A(r_A)\}$ can arise only from changes in $\{\kappa_A\}$. For such a process,

$$\sum_A d\kappa_A dN_A = \sum_A \sum_B d\kappa_A J_{AB} d\kappa_B \qquad \text{(S42)}$$

Inserting Eq. (S41) into (S42) and rearranging gives

$$\sum_A d\kappa_A dN_A = \frac{1}{2}\sum_A \sum_{B,L} \left[\left(d\kappa_A - d\kappa_B\right)^2 \oint \left(\frac{w_A(r_A)w_B(r_B)}{\left(W(\vec{r})\right)^2} \right) \rho(\vec{r})\, d^3\vec{r} \right] \geq 0 . \qquad \text{(S43)}$$

If $\{N_A\}$ are converged, then both sides of Eq. (S43) are identically zero. For every pair of atoms A and B, this requires either $\left(d\kappa_A - d\kappa_B\right)^2 = 0$ or else the AB overlap integral in Eq. (S43) is zero. Thus, for any sets of atoms with non-zero overlaps, $\{d\kappa_A\} = 0$ when $\{dN_A\} = 0$. For an atom without any overlaps (i.e., isolated atomic ion limit), $u_A = 0$ and the converged $\rho_A(\vec{r}_A)$ is independent of κ_A. Therefore, when u_A is negligible (e.g., $u_A < 10^{-7}$) we set $\kappa_A = 0$. For all other atoms, $\left(d\kappa_A - d\kappa_B\right)^2 = 0$ when $\{dN_A\} = 0$. Therefore, $\{N_A\}$ cannot be converged between

successive charge cycles unless $\{\kappa_A\}$ are converged between successive charge cycles. Therefore, $\{w_A(r_A)\}$ are converged in the Convex functional method if and only if $\{N_A\}$ and $\{\rho_A^{avg}(r_A)\}$ are converged.

S3.2 Derivation of Optimal Convergence Parameters

S3.2.1 Derivation of the form of u_A in Eq. (S37)

The careful reader will observe the quantity u_A appearing in Eq. (S37) for the Convex functional has a different form than the quantity u_A appearing in Eq. (77) of the main text for the DDEC6 method. During each DDEC6 charge partitioning step, $H_A(r_A)$ is computed first and then κ_A is adjusted if necessary; that is, κ_A is optimized while keeping $H_A(r_A)$ constant. In contrast, when optimizing the Convex functional, κ_A affects $\rho_A^{avg}(r_A)$ which in turn affects $w_A(r_A)$; that is, κ_A is updated while varying $\rho_A^{avg}(r_A)$. For generality, we consider an atomic weighting factor of the form

$$w_A^{convex}(r_A) = e^{\kappa_A}\left(\rho_A^{fixed\_ref}(r_A)\right)^{\chi}\left(\rho_A^{avg}(r_A)\right)^{1-\chi} \qquad \text{(S44)}$$

where $0<\chi\le 1$. Note that $\chi = 1/2$ for the Convex functional described in Section S3.1 above. Inserting Eq. (S44) into Eq. (46) of the main text and rearranging yields

$$\rho_A^{avg}(r_A) = e^{\kappa_A/\chi}\rho_A^{fixed\_ref}(r_A)\left(\left\langle \rho(\vec{r})/W(\vec{r})\right\rangle_{r_A}\right)^{1/\chi}. \qquad \text{(S45)}$$

Case 1: When $W(\vec{r})\approx\rho(\vec{r})$, changes in $w_A^{convex}(r_A)$ tend to be absorbed by the other atoms. In this case, taking the partial derivative of Eq. (S45) yields

$$\left(\frac{\partial\rho_A^{avg}(r_A)}{\partial\kappa_A}\right)_{\kappa_{B\neq A}} \approx \frac{\rho_A^{avg}(r_A)}{\chi} \qquad \text{(S46)}$$

which combined with Eq. (S44) yields

$$\left(\frac{\partial w_A^{convex}(r_A)}{\partial\kappa_A}\right)_{\kappa_{B\neq A}} \approx \frac{w_A^{convex}(r_A)}{\chi}. \qquad \text{(S47)}$$

Case 2: At the other extreme, where changes in $w_A^{convex}(r_A)$ are not absorbed by the other atoms, then

$$\begin{aligned}\left(\frac{\partial N_A}{\partial\kappa_A}\right)_{\kappa_{B\neq A}} &= \frac{\partial}{\partial\kappa_A}\oint w_A(r_A)\rho(\vec{r})/W(\vec{r})\,d^3\vec{r}_A \\ &= \oint\frac{\partial w_A(r_A)}{\partial\kappa_A}\frac{\rho(\vec{r})}{W(\vec{r})}\left(1-\frac{w_A(r_A)}{W(\vec{r})}\right)d^3\vec{r}_A\end{aligned}. \qquad \text{(S48)}$$

Taking the partial derivative of Eq. (S44) with the help of Eq. (S45) yields

$$\left(\frac{\partial w_A^{convex}}{\partial\kappa_A}\right)_{\kappa_{B\neq A}} \lesssim \frac{w_A^{convex}(r_A)}{\chi}. \qquad \text{(S49)}$$

Substituting Eq. (S49) into (S48) gives

$$\left(\frac{\partial N_A}{\partial \kappa_A}\right)_{\kappa_{B\neq A}} \lesssim \frac{1}{\chi}\oint w_A^{convex}(r_A)\frac{\rho(\vec{r})}{W(\vec{r})}\left(1-\frac{w_A^{convex}(r_A)}{W(\vec{r})}\right)d^3\vec{r}_A \,. \quad \text{(S50)}$$

Cases 1 and 2 can be combined by noting that case 1 dominates when $w_A^{convex}(r_A) \ll W(\vec{r})$, and case 2 dominates when $w_A^{convex}(r_A) \approx W(\vec{r})$. Assuming a linear interpolation where $\left(1-\frac{w_A^{convex}(r_A)}{W(\vec{r})}\right)$ is the fraction assigned to case 1 and $\frac{w_A^{convex}(r_A)}{W(\vec{r})}$ is the fraction assigned to case 2 yields:

$$u_A = \left(\frac{\partial N_A}{\partial \kappa_A}\right)_{\kappa_{B\neq A}} \lesssim \frac{1}{\chi}\oint w_A^{convex}(r_A)\frac{\rho(\vec{r})}{W(\vec{r})}\left(1-\frac{w_A^{convex}(r_A)}{W(\vec{r})}\right)\left(1+\frac{w_A^{convex}(r_A)}{W(\vec{r})}\right)d^3\vec{r}_A \,. \quad \text{(S51)}$$

The safest approach corresponds to setting u_A to its approximate upper bound. (This corresponds to estimating κ_A changes conservatively to minimize overshoot.) This explains the basis for the form of u_A appearing in Eq. (S37).

S3.2.2 Derivation of the constant value in Eq. (S39)

To derive this constant, we construct a convergence model in which the error at charge cycle i is characterized by

$$\varepsilon_i = \min\left(-N_A^{val}\big|_i, 0\right) \quad \text{(S52)}$$

where ε_i is some positive number. (Note that ε_i has a different value for each atom.)

Substituting Eq. (S52) into (S39) yields

$$\kappa_A^{i+1} - \kappa_A^i = const\cdot\varepsilon_i / u_A \,. \quad \text{(S53)}$$

During charge cycle i+1, this change in κ_A impacts $w_A(r_A)$ only through the e^{κ_A} prefactor and not through the $\rho_A^{avg}(r_A)$ term, because the $\rho_A^{avg}(r_A)$ term will be impacted only during the i+2 and subsequent charge cycles. From Eq. (S44), the change in $w_A^{convex}(r_A)$ at constant $\rho_A^{avg}(r_A)$ is

$$\left(\frac{\partial w_A^{convex}(r_A)}{\partial \kappa_A}\right)_{\rho_A^{avg}(r_A),\kappa_{B\neq A}} = w_A^{convex}(r_A) \,. \quad \text{(S54)}$$

Comparing Eqs. (S54) and (S47) gives

$$\left(\frac{\partial w_A^{convex}(r_A)}{\partial \kappa_A}\right)_{\rho_A^{avg}(r_A),\kappa_{B\neq A}} \approx \chi\left(\frac{\partial w_A^{convex}(r_A)}{\partial \kappa_A}\right)_{\kappa_{B\neq A}} \,. \quad \text{(S55)}$$

Eq. (S55) therefore implies that within overlapping atomic regions

$$\left(\frac{\partial N_A^{val}}{\partial \kappa_A}\right)_{\rho_A^{avg}(r_A),\kappa_{B\neq A}} \approx \chi\left(\frac{\partial N_A^{val}}{\partial \kappa_A}\right)_{\kappa_{B\neq A}} = \chi u_A \quad \text{(S56)}$$

where we have used the definition

$$u_A = \left(\frac{\partial N_A^{val}}{\partial \kappa_A} \right)_{\kappa_{B \neq A}} . \qquad \text{(S57)}$$

Rearranging Eq. (S56) gives the change in N_A^{val} between successive iterations due to changes in κ_A made during the present iteration

$$N_A^{val,i+1} - N_A^{val,i} \approx \left(\frac{\partial N_A^{val}}{\partial \kappa_A} \right)_{\rho_A^{avg}(r_A), \kappa_{B \neq A}} \left(\kappa_A^{i+1} - \kappa_A^{i} \right) \approx \chi u_A \left(\kappa_A^{i+1} - \kappa_A^{i} \right) \qquad \text{(S58)}$$

where $\left\{ N_A^{val,i} \right\}$ is the valence population obtained by using $\left\{ w_A^i(r_A) \right\}$ as the atomic weighting factors.

Combining Eqs. (S53) and (S58), the impact of the κ_A change felt during the i+1 charge cycle is

$$\left(\frac{\partial N_A}{\partial \kappa_A} \right)_{\rho_A^{avg}(r_A), \kappa_{B \neq A}} \left(\kappa_A^{i+1} - \kappa_A^{i} \right) = const \cdot \varepsilon_i \cdot \chi . \qquad \text{(S59)}$$

Thus, neglecting the $\rho_A^{avg}(r_A)$ changes, the error during charge cycle i+1 will be

$$\varepsilon_{i+1} = \varepsilon_i (1 - const \cdot \chi) . \qquad \text{(S60)}$$

Applying once more,

$$\varepsilon_{i+2} = \varepsilon_{i+1} (1 - const \cdot \chi) = \varepsilon_i (1 - const \cdot \chi)^2 . \qquad \text{(S61)}$$

Combining Eqs. (S53) with (S60) and (S61) yields

$$\kappa_A^{i+2} - \kappa_A^{i+1} = const \cdot \varepsilon_i (1 - const \cdot \chi) / u_A \qquad \text{(S62)}$$

$$\kappa_A^{i+3} - \kappa_A^{i+2} = const \cdot \varepsilon_i (1 - const \cdot \chi)^2 / u_A . \qquad \text{(S63)}$$

Therefore, the sum of κ_A changes over three successive charge cycles is

$$\kappa_A^{i+3} - \kappa_A^{i} = const \cdot \varepsilon_i \left[(1 - const \cdot \chi)^2 + (1 - const \cdot \chi) + 1 \right] / u_A . \qquad \text{(S64)}$$

The key to deriving an appropriate value for the constant is to note that eventually the $\rho_A^{avg}(r_A)$ changes will kick in and affect $w_A(r_A)$. A reasonable value of the constant will correspond to the sum of κ_A changes over three successive charge cycles not overshooting the eventual N_A^{val} changes even when including the eventual effects of changing $\rho_A^{avg}(r_A)$ on $w_A(r_A)$. Since the eventual N_A^{val} change is u_A times the κ_A change, the non-overshoot condition derived from Eq. (S64) is

$$const \left[(1 - const \cdot \chi)^2 + (1 - const \cdot \chi) + 1 \right] \leq 1 . \qquad \text{(S65)}$$

The approximately optimal constant corresponds to the equality condition. Solving Eq. (S65) yields

$$\text{const} = \frac{1}{\chi} - \sqrt[3]{\frac{1}{\chi^3} - \frac{1}{\chi^2}} \,. \qquad \text{(S66)}$$

For $\chi = 1/2$, we have $\text{const} = 2 - \sqrt[3]{4} = 0.41259...$

We performed computational tests (using model systems in spreadsheet) with different χ values and different constant values, which verified Eq. (S66) yields nearly optimal convergence performance. With the constant value from Eq. (S66), the system converges rapidly with highly damped overshoot if the $N_A^{val} \geq 0$ constraint is binding. Using $\chi = 1/2$, $\text{const} = 2 - \sqrt[3]{4} = 0.41259...$, and the convergence accelerator described in Section S3.2.3, convergence to 10^{-5} electrons is achieved in approximately 20 total charge cycles for the Convex functional when the $N_A^{val} \geq 0$ constraint is binding and in approximately 14 charge cycles when the $N_A^{val} \geq 0$ constraint is not binding.

S3.2.3 Derivation of the optimal convergence accelerator

When using an atomic weighting factor of the form shown in Eq. (S44), the spherical average atomic density, $\rho_A^{avg}(r_A)$, is computed based on the $w_A(r_A)$ that used the prior $\rho_A^{avg}(r_A)$. This causes the situation that if the estimate for $\rho_A^{avg}(r_A)$ in charge cycle i is too large (small), the new estimate for $\rho_A^{avg}(r_A)$ in charge cycle i+1 will also be too large (small). In general, therefore, it will take many charge cycles to work off errors in the estimated $\left\{\rho_A^{avg}(r_A)\right\}$. The higher the proportion of spherical averaging (i.e., the smaller the χ value) in $w_A(r_A)$, the more pronounced this problem will be.

This problem can be solved using a convergence accelerator. A convergence accelerator causes changes in the estimated $\left\{\rho_A^{avg}(r_A)\right\}$ to take effect in fewer charge cycles, thereby allowing the calculation to be converged in fewer charge cycles. All feasible functional forms of a convergence accelerator become linear as the change to $\left\{\rho_A^{avg}(r_A)\right\}$ becomes relatively small. Therefore, we choose the linear form

$$w_A^{i+1}(r_A) = \psi_A^i(r_A) + m\left(\psi_A^i(r_A) - \psi_A^{i-1}(r_A)\right) \quad \text{(S67)}$$

where m is a constant and

$$\psi_A^i(r_A) = e^{\kappa_A^{i+1}} \left(\rho_A^{fixed\_ref}(r_A)\right)^{\chi} \left(\rho_A^{avg}(r_A)\Big|_i\right)^{(1-\chi)} . \qquad \text{(S68)}$$

We begin by defining a set of variables that quantify the approach to convergence:

$$y1_A^i(r_A) = \ln\left(\rho_A^{avg}(r_A)\Big|_i \Big/ \rho_A^{avg}(r_A)\Big|_{converged}\right) \qquad \text{(S69)}$$

$$y2_A^i(r_A) = \ln\left(w_A^i(r_A) \Big/ w_A^{converged}(r_A)\right) \qquad \text{(S70)}$$

where $\rho_A^{avg}(r_A)\big|_i$ is the value of $\rho_A^{avg}(r_A)$ computed during charge cycle i, and $\rho_A^{avg}(r_A)\big|_{converged}$ and $w_A^{converged}(r_A)$ are the final converged values of $\rho_A^{avg}(r_A)$ and $w_A(r_A)$, respectively. Combining Eqs. (S68) and (S69) yields

$$\ln\left(\psi_A^i(r_A)/\psi_A^{converged}(r_A)\right) = (1-\chi)\,y1_A^i(r_A) + \left(\kappa_A^{i+1} - \kappa_A^{converged}\right). \quad \text{(S71)}$$

According to Eq. (S67),

$$\psi_A^{converged}(r_A) = w_A^{converged}(r_A). \quad \text{(S72)}$$

In regions where atom-atom overlaps are significant (i.e., $w_A(r_A) \ll W(\vec{r})$), the spherical average computed during charge cycle i (i.e., $\rho_A^{avg}(r_A)\big|_i$) is proportional to $w_A^i(r_A)$:

$$\frac{\partial \ln\left(\rho_A^{avg}(r_A)\big|_i\right)}{\partial \ln\left(w_A^i(r_A)\right)} \approx 1 \ . \quad \text{(S73)}$$

From Eq. (S73), it directly follows that

$$y1_A^i(r_A) \approx y2_A^i(r_A) \approx \varepsilon_A^i(r_A) \quad \text{(S74)}$$

where $\varepsilon_A^i(r_A)$ quantifies the approach to convergence.

Computational tests on real systems studied with the CHARGEMOL program, as well as numerical model systems studied in spreadsheet, showed that convergence in the fewest number of charge cycles is achieved when the constant m is set to the largest value giving non-oscillatory convergence. For steady non-oscillatory convergence, the errors are reduced to a nearly constant fraction between successive charge cycles:

$$f \approx \frac{\varepsilon_A^i(r_A)}{\varepsilon_A^{i-1}(r_A)}. \quad \text{(S75)}$$

Dividing Eq. (S67) by $\psi_A^{converged}(r_A)$ and simplifying in the limit of small $\varepsilon(r_A)$ and small κ_A yields:

$$\varepsilon_A^{i+1}(r_A) = (1-\chi)\varepsilon_A^i(r_A) + m(1-\chi)\left(\varepsilon_A^i(r_A) - \varepsilon_A^{i-1}(r_A)\right) + \left(\kappa_A^{i+1} - \kappa_A^{converged}\right) + m\left(\kappa_A^{i+1} - \kappa_A^i\right). \quad \text{(S76)}$$

We optimize the convergence parameter m for the case where the $N_A^{val} \geq 0$ applies but is non-binding; that is, for the case where $\kappa_A^{i+1} = \kappa_A^i = \kappa_A^{converged} = 0$. In this case, substituting Eq. (S75) into (S76) gives

$$f^2\varepsilon_A^{i-1}(r_A) = (1-\chi)f\varepsilon_A^{i-1}(r_A) + m(1-\chi)\left(f\varepsilon_A^{i-1}(r_A) - \varepsilon_A^{i-1}(r_A)\right) \quad \text{(S77)}$$

which simplifies to the characteristic equation

$$f^2 - (1+m)(1-\chi)f + m(1-\chi) = 0. \quad \text{(S78)}$$

Solving Eq. (S78) gives

$$f = \frac{(1+m)(1-\chi) \pm \sqrt{(1+m)^2(1-\chi)^2 - 4m(1-\chi)}}{2}. \quad \text{(S79)}$$

The limiting value of f occurs when the discriminant (i.e., quantity under square root) is zero:

$$(1+m)^2(1-\chi)^2 - 4m(1-\chi) = 0 \qquad \text{(S80)}$$

which yields

$$m = \frac{\left(1-\sqrt{\chi}\right)^2}{1-\chi}. \qquad \text{(S81)}$$

Substituting Eq. (S81) into (S79) yields the limiting value of f:

$$f = 1-\sqrt{\chi}. \qquad \text{(S82)}$$

We chose the following form for our convergence accelerator:

$$w_A^{i+1}(r_A) = \frac{e^{2\kappa_A^{i+1}}\left(\rho_A^{\text{fixed\_ref}}(r_A)\right)^{2\chi}\left(\rho_A^{\text{avg}}(r_A)\right)^{2(1-\chi)}}{(1-a)e^{\kappa_A^{i+1}}\left(\rho_A^{\text{fixed\_ref}}(r_A)\right)^{\chi}\left(\rho_A^{\text{avg}}(r_A)\right)^{(1-\chi)} + (a)w_A^i(r_A)} = \frac{\left(\psi_A^i(r_A)\right)^2}{(1-a)\psi_A^i(r_A) + (a)w_A^i(r_A)} \qquad \text{(S83)}$$

where $0 \le a < 1$ is a constant. We chose this form, because it has the key advantage of guaranteeing $w_A^{i+1}(r_A) \ge 0$. Dividing both sides of Eq. (S83) by $w_A^{\text{converged}}(r_A) = \psi_A^{\text{converged}}(r_A)$ (Eq. (S72)) and taking the logarithm yields

$$\ln\left(\frac{w_A^{i+1}(r_A)}{w_A^{\text{converged}}(r_A)}\right) = \ln\left(\frac{\psi_A^i(r_A)}{\psi_A^{\text{converged}}(r_A)}\right) + \ln\left(\frac{\psi_A^i(r_A)}{(1-a)\psi_A^i(r_A) + (a)w_A^i(r_A)}\right). \qquad \text{(S84)}$$

Inserting Eqs. (S70) and (S71) into (S84) and simplifying for $\kappa_A = 0$ yields

$$y2_A^{i+1}(r_A) = (1-\chi)\,y1_A^i(r_A) + \ln\left(\frac{\psi_A^i(r_A)}{(1-a)\psi_A^i(r_A) + (a)w_A^i(r_A)}\right). \qquad \text{(S85)}$$

Inserting Eqs. (S74) and (S75) into (S85) yields

$$f\varepsilon_A^i(r_A) \approx (1-\chi)\varepsilon_A^i(r_A) + \ln\left(\frac{\psi_A^i(r_A)}{(1-a)\psi_A^i(r_A) + (a)w_A^i(r_A)}\right). \qquad \text{(S86)}$$

For small $\left|\varepsilon_A^i(r_A)\right|$, expanding Eq. (S71) for $\kappa_A = 0$ gives

$$\psi_A^i(r_A)/\psi_A^{\text{converged}}(r_A) \approx 1 + (1-\chi)\,y1_A^i(r_A) \approx 1 + (1-\chi)\varepsilon_A^i(r_A) \qquad \text{(S87)}$$

where the rightmost side follows from Eq. (S74). For small $\left|\varepsilon_A^i(r_A)\right|$, expanding Eq. (S70) gives

$$w_A^i(r_A)/w_A^{\text{converged}}(r_A) = w_A^i(r_A)/\psi_A^{\text{converged}}(r_A) \approx 1 + y2_A^i(r_A) \approx 1 + \varepsilon_A^i(r_A) \qquad \text{(S88)}$$

where the rightmost side follows from Eqs. (S72) and (S74). Using Eqs. (S87) and (S88) gives

$$\ln\left(\frac{\psi_A^i(r_A)}{(1-a)\psi_A^i(r_A) + (a)w_A^i(r_A)}\right) \approx \ln\left(\frac{1 + (1-\chi)\varepsilon_A^i(r_A)}{(1-a)\left(1 + (1-\chi)\varepsilon_A^i(r_A)\right) + (a)\left(1 + \varepsilon_A^i(r_A)\right)}\right) \qquad \text{(S89)}$$

which when expanded at first-order for $\left|\varepsilon_A^i(r_A)\right| \ll 1$ yields

$$\ln\left(\frac{1+(1-\chi)\varepsilon_A^i(r_A)}{(1-a)\left(1+(1-\chi)\varepsilon_A^i(r_A)\right)+(a)\left(1+\varepsilon_A^i(r_A)\right)}\right)\approx -a\chi\varepsilon_A^i(r_A). \qquad \text{(S90)}$$

Substituting Eq. (S90) into (S86) yields

$$f\varepsilon_A^i(r_A)\approx\varepsilon_A^i(r_A)\left((1-\chi)-a\chi\right). \qquad \text{(S91)}$$

Substituting Eq. (S82) into (S91) and solving gives the optimized convergence accelerator parameter

$$a=\sqrt{\frac{1}{\chi}}-1\,. \qquad \text{(S92)}$$

Notably, the rate of convergence is practically independent of the material. The largest number of charge cycles required to converge the NACs and $\left\{\rho_A^{avg}(r_A)\right\}$ within convergence_threshold e and e/bohr^3, respectively, is

$$charge\_cycles\le\frac{\ln(convergence\_threshold)-\ln(\Delta q_A)}{\ln(f)}+4+2 \qquad \text{(S93)}$$

where Δq_A is the maximum NAC magnitude error on the fourth charge cycle. (Recall that the first three charge cycles are used to compute the conditioned reference density; therefore, the self-consistent cycles begin with the fourth charge cycle.) convergence_threshold is the error on the (last -2)th charge cycle. Two final charge cycles are required to demonstrate the NAC and $\left\{\rho_A^{avg}(r_A)\right\}$ changes between successive charge cycles are below the convergence_threshold two times in a row. The first four and last two charge cycles are thus added in Eq. (S93). We used convergence_threshold = 10^{-5} e and e/bohr^3 on the NACs and $\left\{\rho_A^{avg}(r_A)\right\}$, respectively. A reasonable approximation is that the NACs on the fourth charge cycle are within ~±0.2 e of the final NACs. Substituting into Eq. (S93) with $f=1-\sqrt{1/2}$ yields $charge\_cycles\le 14$. Indeed, more than 99% of the materials studied in this paper converged within 14 charge cycles when using the Convex functional with the convergence accelerator.

We performed an extensive set of computational tests confirming all aspects of the theory described above. These computational tests included both tests on real materials using the CHARGEMOL code as well as numerical finite difference models in spreadsheet. All aspects of the above theory were doubly confirmed (i.e., both for the real materials and for the finite difference models), including:

1. The errors between successive charge cycles follows a nearly constant ratio f.
2. We compared f values for m = 0 and the optimal m value (i.e., also a = 0 and the optimal a value) for both χ = 1/2 and 1/3. All of the computational results were in precise agreement with Eq. (S79). In these cases, the number of charge cycles required for convergence closely followed Eq. (S93).
3. As m and a are decreased below their optimal values, the calculation takes more charge cycles to converge. As m and a are increased above their optimal values, the calculations

do not converge in fewer charge cycles. As m and a are increased to unreasonably large values (i.e., many times larger than their optimal values) the calculations begin to oscillate notably.

S3.2.4 Convergence speed of spin partitioning

Finally, we note that the DDEC spin partitioning method[S4] follows a similar convergence law as that noted above for the Convex functional. Specifically, the spin partitioning method uses $\chi_{spin}=1/2$ which utilizes a geometric average between $\vec{m}_A^{avg}(r_A)$ and $\vec{m}_A^{proportional}(\vec{r}_A)$.[S4] The DDEC spin partitioning method uses an optimized convergence algorithm that achieves convergence as rapidly as feasible.[S4] For the same reasons as described above, the DDEC spin partitioning method converges at the same rate for all materials with a constant error fraction between successive spin cycles. Theoretical analysis shows the optimal f value depends only on χ independent of the particulars of the optimization scheme. Specifically, the analog of Eq. (S82) is

$$f_{spin}=1-\sqrt{\chi_{spin}}\ . \qquad \text{(S94)}$$

Using $\chi_{spin}=1/2$, this means the ASM errors on spin cycle i+1 are only about 29% as large as the errors on spin cycle i. We confirmed this prediction using numerous computational tests on real collinear and non-collinear magnetic materials. For both collinear and non-collinear magnetism, the required number of spin cycles follows this analog of Eq. (S93):

$$spin\_cycles \le \frac{\ln(spin\_threshold)-\ln(\Delta M_A)}{\ln(f_{spin})}+1+1 \qquad \text{(S95)}$$

where ΔM_A is the maximum ASM magnitude error on the first spin cycle. In Eq. (S95), the first +1 accounts for the first spin cycle. spin_threshold is the error on the (last -1)th spin cycle. A final spin cycle is required to demonstrate the ASM changes are below the spin_threshold; this accounts for the second +1 appearing in Eq. (S95). For spin partitioning, we used a spin_threshold of 5×10^{-5} electrons with $\chi_{spin}=1/2$. Substituting these values into Eq. (S95) yields

$$spin\_cycles \le 9\,. \qquad \text{(S96)}$$

Indeed, all of the collinear and non-collinear magnetic materials we have examined to date followed Eq. (S96). All of the DDEC methods use the same spin partitioning algorithm.[S4] Thus, the spin partitioning convergence properties are equivalent whether using the DDEC3, DDEC6, or Convex functional for charge partitioning. *In summary, a key advantage of our methodology is that both the charge and spin partitioning converge within a small number of cycles for all materials.*

S4. Flow Diagrams for the DDEC6 Method

S4.1 Algorithm for computing $\{\rho_A^{cond}(r_A)\}$

As shown in Figure S1, a robust and rapidly converging iterative algorithm was used to compute $\{\rho_A^{cond}(r_A)\}$. In each iteration, an estimate of Φ_A^I is used to compute the estimate

$$\rho_A^{cond}(r_A) = Y_A^{avg}(r_A) + \Phi_A^I \sqrt{Y_A^{avg}(r_A)}. \qquad \text{(S97)}$$

Then, constraint

$$\phi^I(r_A) = \frac{d\rho_A^{cond}(r_A)}{dr_A} \le 0 \qquad \text{(S98)}$$

is enforced by recursively setting

$$\rho_A^{cond}(r_A) = \min\left(\rho_A^{cond}(r_A), \rho_A^{cond}(r_A - \Delta r_A)\right) \qquad \text{(S99)}$$

beginning with the second radial shell and continuing outward until the last radial shell. To complete one iteration, φ_A^I is computed via

$$\varphi_A^I = \int_0^{r_{cutoff}} \rho_A^{cond}(r_A) 4\pi (r_A)^2 dr_A - z_A + q_A^{ref} \to 0. \qquad \text{(S100)}$$

A scheme is required to generate a new Φ_A^I estimate for the next iteration. Since φ_A^I is a monotonically increasing function of Φ_A^I, Φ_A^I should be increased (decreased) when $\varphi_A^I < 0$ ($\varphi_A^I > 0$). Convergence is achieved when $\left|\varphi_A^I\right|$ is less than some zero tolerance (e.g., 10^{-10} electrons).

The update scheme we used contained the following sequence of steps. In the first iteration, we set $\Phi_A^{I\,(1)} = 0$. Then we set

$$\Phi_A^{I\,(i+1)} = 2\Phi_A^{I\,(i)} - \frac{\varphi_A^{I\,(i)}}{\int_0^{r_{cutoff}} \sqrt{Y_A^{avg}(r_A)} 4\pi (r_A)^2 dr_A} \qquad \text{(S101)}$$

where $\varphi_A^{I\,(i)}$ is the result of using $\Phi_A^{I\,(i)}$. Because $\varphi_A^{I\,(1)} < 0$ if constraint (S98) is binding, repetitively applying Eq. (S101) increases $\Phi_A^{I\,(i+1)}$ until $\varphi_A^{I\,(i)} > 0$. Since $\Phi_A^{I\,(i+1)}$ more than doubles between successive iterations, the number of iterations required to reach this upper bound is small. At this point, we stop using Eq. (S101) and set $\Phi_A^{I\,(upper)}$ as the smallest known value yielding $\varphi_A^{I\,(upper)} > 0$. We also set $\Phi_A^{I\,(lower)}$ as the largest known value yielding $\varphi_A^{I\,(lower)} < 0$. The remainder of the steps are simply aimed at squeezing the upper and lower bounds as quickly as possible. First, we try the midpoint $\Phi_A^{I\,(mid)} = \left(\Phi_A^{I\,(upper)} + \Phi_A^{I\,(lower)}\right)/2$ to get $\varphi_A^{I\,(mid)}$. Then, we fit the triple of points $\left(\Phi_A^{I\,(lower)}, \varphi_A^{I\,(lower)}\right)$, $\left(\Phi_A^{I\,(mid)}, \varphi_A^{I\,(mid)}\right)$, $\left(\Phi_A^{I\,(upper)}, \varphi_A^{I\,(upper)}\right)$ to a parabola. Then, we set $\Phi_A^{I\,(parabolic)}$ equal to the root of the parabola where $\varphi_A^{I\,(parabolic)}$ is predicted to be zero. After computing the actual value of $\varphi_A^{I\,(parabolic)}$ via Eq. (S100), we identify which points among $\varphi_A^{I\,(lower)}$, $\varphi_A^{I\,(mid)}$, $\varphi_A^{I\,(upper)}$, and $\varphi_A^{I\,(parabolic)}$ are the closest to zero from above (i.e., $\varphi_A^{I\,(above)}$) and below (i.e., $\varphi_A^{I\,(below)}$). A linear interpolation between these two closest-to-zero points is performed to identify

$\Phi_A^{I\,(\text{corrector})}$ as the point where $\varphi_A^{I\,(\text{corrector})}$ is predicted to be: (a) $-\varphi_A^{I\,(\text{above})}$ if $3\left|\varphi_A^{I\,(\text{above})}\right| < \left|\varphi_A^{I\,(\text{below})}\right|$, (b) $-\varphi_A^{I\,(\text{below})}$ if $3\left|\varphi_A^{I\,(\text{below})}\right| < \left|\varphi_A^{I\,(\text{above})}\right|$, and (c) $\left(\varphi_A^{I\,(\text{above})} + \varphi_A^{I\,(\text{below})}\right)/2$ otherwise. This procedure places $\varphi_A^I = 0$ approximately half-way between $\varphi_A^{I\,(\text{corrector})}$ and $\varphi_A^{I\,(\text{above})}$ or $\varphi_A^{I\,(\text{below})}$ (whichever is smaller in magnitude), subject to the constraint that the resulting interval size is cut in half or better. At the close of this iteration, we identify the new lower (upper) bound as the largest (smallest) Φ_A^I among $\Phi_A^{I\,(\text{lower})}$, $\Phi_A^{I\,(\text{mid})}$, $\Phi_A^{I\,(\text{upper})}$, $\Phi_A^{I\,(\text{parabolic})}$, and $\Phi_A^{I\,(\text{corrector})}$ yielding a corresponding $\varphi_A^I < 0$ ($\varphi_A^I > 0$). Having refined the lower and upper bounds, we repeat the bisection, parabolic fitting, and linear interpolation steps in the next iteration to reach a tighter yet lower and upper bound. This process is repeated until convergence. Because this algorithm cuts the size of the search domain by better than half in each reshaping cycle, it is mathematically guaranteed to always converge to the correct solution in a few reshaping cycles. In practice, we found this process converges magnificently, with one or two cycles of parabolic fitting and linear interpolation usually sufficient to achieve a precision of 10^{-10} electrons.

Set $\Phi_A^{I\,(1)} = 0$

Find $\Phi_A^{I\,(i+1)} = 2\Phi_A^{I\,(i)} - \dfrac{\varphi_A^{I\,(i)}}{\int_0^{r_{cutoff}} \sqrt{Y_A^{avg}(r_A)}\,4\pi(r_A)^2\,dr_A}$ until $\varphi_A^{I\,(i)} > 0$

Set upper bound as $\Phi_A^{I\,(upper)}$ =smallest value yielding $\varphi_A^{I\,(upper)} > 0$

Set lower bound as $\Phi_A^{I\,(lower)}$ =largest value yielding $\varphi_A^{I\,(lower)} < 0$

Is $\left|\varphi_A^{I\,(upper)}\right| < 10^{-10}$ or $\left|\varphi_A^{I\,(lower)}\right| < 10^{-10}$?

Yes → End

No

Get midpoint $\varphi_A^{I\,(mid)}$ with $\Phi_A^{I\,(mid)} = \left(\Phi_A^{I\,(upper)} + \Phi_A^{I\,(lower)}\right)/2$

Is $\left|\varphi_A^{I\,(mid)}\right| < 10^{-10}$?

Yes → End

No

Parabolic fitting of $\left(\Phi_A^{I\,(lower)}, \varphi_A^{I\,(lower)}\right)$, $\left(\Phi_A^{I\,(mid)}, \varphi_A^{I\,(mid)}\right)$, $\left(\Phi_A^{I\,(upper)}, \varphi_A^{I\,(upper)}\right)$

$\Phi_A^{I\,(parabolic)}$ = root of the fitted parabola. Use this value to compute $\varphi_A^{I\,(parabolic)}$

Is $\left|\varphi_A^{I\,(parabolic)}\right| < 10^{-10}$?

Yes → End

No

Identify which $\varphi_A^{I\,(lower)}$, $\varphi_A^{I\,(mid)}$, $\varphi_A^{I\,(upper)}$, and $\varphi_A^{I\,(parabolic)}$ are the closest to zero from above and below

Is $3\left|\varphi_A^{I\,(above)}\right| < \left|\varphi_A^{I\,(below)}\right|$?

Yes → Choose $\Phi_A^{I\,(corrector)} = \Phi_A^{I\,(above)} - 2\varphi_A^{I\,(above)} \dfrac{\left(\Phi_A^{I\,(above)} - \Phi_A^{I\,(below)}\right)}{\left(\varphi_A^{I\,(above)} - \varphi_A^{I\,(below)}\right)}$

No

Is $3\left|\varphi_A^{I\,(below)}\right| < \left|\varphi_A^{I\,(above)}\right|$?

Yes → Choose $\Phi_A^{I\,(corrector)} = \Phi_A^{I\,(below)} - 2\varphi_A^{I\,(below)} \dfrac{\left(\Phi_A^{I\,(above)} - \Phi_A^{I\,(below)}\right)}{\left(\varphi_A^{I\,(above)} - \varphi_A^{I\,(below)}\right)}$

No

$\Phi_A^{I\,(corrector)} = \left(\Phi_A^{I\,(above)} + \Phi_A^{I\,(below)}\right)/2$

Identify new lower and upper bounds as the largest and smallest among $\Phi_A^{I\,(lower)}$, $\Phi_A^{I\,(mid)}$, $\Phi_A^{I\,(upper)}$, $\Phi_A^{I\,(parabolic)}$, and $\Phi_A^{I\,(corrector)}$ yielding $\varphi_A^I < 0$ and $\varphi_A^I > 0$, respectively.

Is $\left|\varphi_A^{I\,(upper)}\right| < 10^{-10}$ or $\left|\varphi_A^{I\,(lower)}\right| < 10^{-10}$?

No → Get midpoint …

Yes → End

NOTE: Every time Φ_A^I is updated, it is used to compute

$$\rho_A^{cond}(r_A) = Y_A^{avg}(r_A) + \Phi_A^I \sqrt{Y_A^{avg}(r_A)}$$

and we enforce the constraint

$$\phi^I(r_A) = \frac{d\rho_A^{cond}(r_A)}{dr_A} \le 0$$

by

$$\rho_A^{cond}(r_A) = \min\left(\rho_A^{cond}(r_A), \rho_A^{cond}(r_A - \Delta r_A)\right)$$

to finally find

$$\varphi_A^I = \int_0^{r_{cutoff}} \rho_A^{cond}(r_A)\,4\pi(r_A)^2\,dr_A - z_A + q_A^{ref}$$

Figure S1: Flow diagram for computing the conditioned reference densities subject to the applied constraints.

S4.2 Summary of the seven charge partitioning steps comprising the DDEC6 method

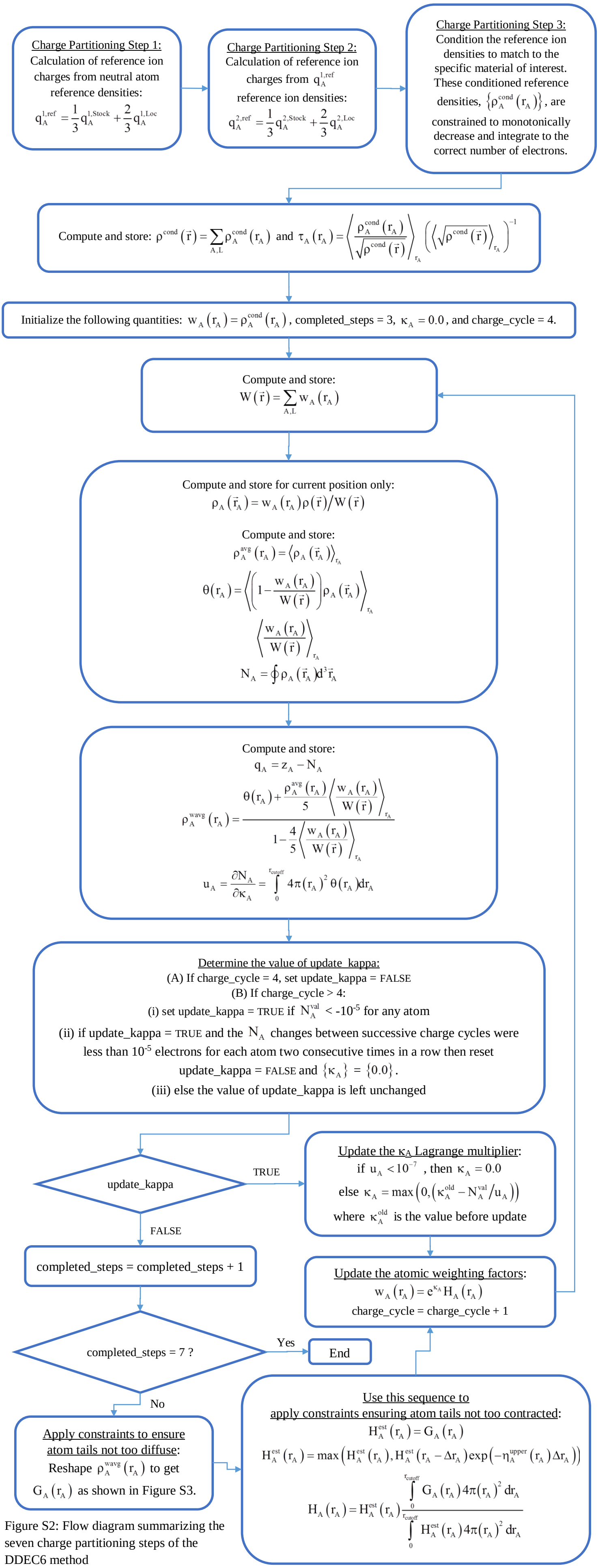

Figure S2: Flow diagram summarizing the seven charge partitioning steps of the DDEC6 method

S4.3 Algorithm for computing $\{G_A(r_A)\}$

As shown in Figure S3, a robust and rapidly converging iterative algorithm was used to compute $\{G_A(r_A)\}$. In each iteration, an estimate of Φ_A^{II} is used to compute the estimate

$$G_A(r_A) = \sigma_A(r_A) + \Phi_A^{II}\sqrt{\sigma_A(r_A)}. \qquad \text{(S102)}$$

Then, the constraint

$$\phi^{II}(r_A) = \frac{dG_A(r_A)}{dr_A} + \eta_A^{lower}(r_A)G_A(r_A) \le 0 \qquad \text{(S103)}$$

$$\eta_A^{lower}(r_A) = \left(1.75\,\text{bohr}^{-1}\right)\left(1-\left(\tau_A(r_A)\right)^2\right) \qquad \text{(S104)}$$

is enforced by recursively setting

$$G_A(r_A) = \min\left(G_A(r_A), G_A(r_A - \Delta r_A)\exp\left(-\eta_A^{lower}(r_A)\Delta r_A\right)\right) \qquad \text{(S105)}$$

beginning with the second radial shell and continuing outward until the last radial shell. To complete one iteration, φ_A^{II} is computed via

$$\varphi_A^{II} = \int_0^{r_{cutoff}} \left(G_A(r_A) - \sigma_A(r_A)\right) 4\pi (r_A)^2 \, dr_A . \qquad \text{(S106)}$$

A scheme is required to generate a new Φ_A^{II} estimate for the next iteration. Since φ_A^{II} is a monotonically increasing function of Φ_A^{II}, Φ_A^{II} should be increased (decreased) when $\varphi_A^{II} < 0$ ($\varphi_A^{II} > 0$). Convergence is achieved when $\left|\varphi_A^{II}\right|$ is less than some zero tolerance (e.g., 10^{-10} electrons). The update scheme we used contained the sequence of steps shown in Figure S3. These steps are analogous to those described in Section S4.1 above, except that Φ_A^{II}, φ_A^{II}, $G_A(r_A)$, and $\sigma_A(r_A)$ replace Φ_A^{I}, φ_A^{I}, $\rho_A^{cond}(r_A)$, and $Y_A^{avg}(r_A)$. Because our algorithm cuts the size of the search domain by better than half in each reshaping cycle, it is mathematically guaranteed to always converge to the correct solution in a few reshaping cycles. In practice, we found this process converges magnificently, with one to three cycles of parabolic fitting and linear interpolation usually sufficient to achieve a precision of 10^{-10} electrons.

Set $\Phi_A^{II(1)} = 0$

Find $\Phi_A^{II(i+1)} = 2\Phi_A^{II(i)} - \dfrac{\varphi_A^{II(i)}}{\int_0^{r_{cutoff}} \sqrt{\sigma_A(r_A)}\,4\pi(r_A)^2\,dr_A}$ until $\varphi_A^{II(i)} > 0$

Set upper bound as $\Phi_A^{II(upper)}$ =smallest value yielding $\varphi_A^{II(upper)} > 0$

Set lower bound as $\Phi_A^{II(lower)}$ =largest value yielding $\varphi_A^{II(lower)} < 0$

Is $\left|\varphi_A^{II(upper)}\right| < 10^{-10}$ or $\left|\varphi_A^{II(lower)}\right| < 10^{-10}$?

Yes → End

No

Get midpoint $\varphi_A^{II(mid)}$ with $\Phi_A^{II(mid)} = \left(\Phi_A^{II(upper)} + \Phi_A^{II(lower)}\right)/2$

Is $\left|\varphi_A^{II(mid)}\right| < 10^{-10}$?

Yes → End

No

Parabolic fitting of $\left(\Phi_A^{II(lower)}, \varphi_A^{II(lower)}\right)$, $\left(\Phi_A^{II(mid)}, \varphi_A^{II(mid)}\right)$, $\left(\Phi_A^{II(upper)}, \varphi_A^{II(upper)}\right)$

$\Phi_A^{II(parabolic)}$ = root of the fitted parabola. Use this value to compute $\varphi_A^{II(parabolic)}$

Is $\left|\varphi_A^{II(parabolic)}\right| < 10^{-10}$?

Yes → End

No

Identify which $\varphi_A^{II(lower)}$, $\varphi_A^{II(mid)}$, $\varphi_A^{II(upper)}$, and $\varphi_A^{II(parabolic)}$ are the closest to zero from above and below

Is $3\left|\varphi_A^{II(above)}\right| < \left|\varphi_A^{II(below)}\right|$?

Yes → Choose $\Phi_A^{II(corrector)} = \Phi_A^{II(above)} - 2\varphi_A^{II(above)} \dfrac{\left(\Phi_A^{II(above)} - \Phi_A^{II(below)}\right)}{\left(\varphi_A^{II(above)} - \varphi_A^{II(below)}\right)}$

No

Is $3\left|\varphi_A^{II(below)}\right| < \left|\varphi_A^{II(above)}\right|$?

Yes → Choose $\Phi_A^{II(corrector)} = \Phi_A^{II(below)} - 2\varphi_A^{II(below)} \dfrac{\left(\Phi_A^{II(above)} - \Phi_A^{II(below)}\right)}{\left(\varphi_A^{II(above)} - \varphi_A^{II(below)}\right)}$

No

$\Phi_A^{II(corrector)} = \left(\Phi_A^{II(above)} + \Phi_A^{II(below)}\right)/2$

Identify new lower and upper bounds as the largest and smallest among $\Phi_A^{II(lower)}$, $\Phi_A^{II(mid)}$, $\Phi_A^{II(upper)}$, $\Phi_A^{II(parabolic)}$, and $\Phi_A^{II(corrector)}$ yielding $\varphi_A^{II} < 0$ and $\varphi_A^{II} > 0$, respectively.

Is $\left|\varphi_A^{II(upper)}\right| < 10^{-10}$ or $\left|\varphi_A^{II(lower)}\right| < 10^{-10}$?

No → Get midpoint

Yes → End

NOTE: Every time Φ_A^{II} is updated, it is used to compute

$$G_A(r_A) = \sigma_A(r_A) + \Phi_A^{II}\sqrt{\sigma_A(r_A)}$$

and we enforce the constraint

$$\phi^{II}(r_A) = \frac{dG_A(r_A)}{dr_A} + \eta_A^{lower}(r_A)\,G_A(r_A) \le 0$$

$$\eta_A^{lower}(r_A) = \left(1.75\ \text{bohr}^{-1}\right)\left(1 - \left(\tau_A(r_A)\right)^2\right)$$

by

$$G_A(r_A) = \min\left(G_A(r_A), G_A(r_A - \Delta r_A)\exp\left(-\eta_A^{lower}(r_A)\Delta r_A\right)\right)$$

to finally find

$$\varphi_A^{II} = \int_0^{r_{cutoff}} \left(G_A(r_A) - \sigma_A(r_A)\right)4\pi(r_A)^2\,dr_A$$

Figure S3: Flow diagram for computing $G_A(r_A)$

References:

S1. A. D. Becke, *J. Chem. Phys.* 1988, **88**, 2547-2553.
S2. G. T. Velde and E. J. Baerends, *J. Comput. Phys.* 1992, **99**, 84-98.
S3. T. A. Manz and D. S. Sholl, *J. Chem. Theory Comput.* 2012, **8**, 2844-2867.
S4. T. A. Manz and D. S. Sholl, *J. Chem. Theory Comput.* 2011, **7**, 4146-4164.