DEVELOPMENT OF A PARALLEL FORTRAN PROGRAM FOR PERFORMING DENSITY DERIVED ELECTROSTATIC AND CHEMICAL ANALYSIS

BY

NIDIA GABRIELA GABALDON LIMAS

A thesis submitted to the Graduate School in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Major Subject: CHEMICAL ENGINEERING

NEW MEXICO STATE UNIVERSITY

LAS CRUCES, NEW MEXICO

MAY 2014
“Development of a Parallel Fortran Program for Performing Density Derived Electrostatic and Chemical Analysis” a thesis prepared by Nidia Gabriela Gabaldon Limas in partial fulfillment of the requirements for the degree Master of Science, has been approved and accepted by the following:

Loui Reyes
Dean of the Graduate School

Thomas A. Manz
Chair of the Examining Committee

Committee in charge:

Dr. Thomas A. Manz
Dr. Shuguang Deng
Dr. Haobin Wang
ACKNOWLEDGEMENTS

All of the calculations were performed using the computational clusters provided by Extreme Science and Engineering Discovery Environment (XSEDE project grant TG-CTS100027). XSEDE is funded by NSF grant OCI-1053575. Special thanks to Dr. Hongmei Luo, graduate program coordinator for the Chemical Engineering Department, and Carol Dyer, administrative assistant for the Chemical Engineering Department. I would also like to thank my advisor Dr. Thomas A. Manz for all the guidance and support throughout the project. I would also like to thank the other members of the examination committee, Dr. Shuguang Deng and Dr. Haobin Wang.
VITA

August 28, 1988          Born in Chihuahua, Chihuahua, Mexico

2006                      Graduated from Colegio de Bachilleres de Chihuahua (High School)
                          Chihuahua, Mexico

2010-2011                  Quality Analyst, SEGLO Logistics
                          Chihuahua, Mexico

2011-2012                  Materials Analyst, SEGLO Logistics
                          Chihuahua, Mexico

2011                      Graduated from Instituto Tecnologico de Chihuahua
                          Chihuahua, Mexico

2013-2014                  Research Assistant, Department of Chemical Engineering
                          New Mexico State University

Field of Study

Major field: Chemical Engineering
Atoms in Molecules (AIM) methods distribute the electronic density computed by quantum chemical methods among the atoms in a material to compute important atomistic descriptors. These atomistic descriptors include net atomic charges (NACs), atomic spin moments (ASMs), effective bond orders (EBOs), and atomic multipoles. NACs are important to construct force fields used in atomistic simulations such as classical molecular dynamics and Monte-Carlo simulations. In addition these atomistic descriptors are important for understanding the magnetic and electronic properties of materials.

Density Derived Electrostatic and Chemical (DDEC) analysis is an AIM method capable of computing the properties mentioned above. The DDEC method is applicable to periodic and non-periodic materials with no magnetism, collinear magnetism, and non-collinear magnetism. A new parallel Fortran program was

v
created to compute DDEC NACs, ASMs and EBOs. The Fortran program was parallelized using OpenMP. This code can be compiled and run in either parallel or serial mode. The new Fortran DDEC program in serial mode ran around 8.4 times faster than the existing Matlab serial program. The new parallel Fortran DDEC program was tested on 6 previously studied systems and the timing and parallelization results are reported. When the parallel Fortran program was run on 32 processors it showed a parallelization efficiency of around 60%. A description of the program and its main features are included, along with the parallelization strategy.

There is a need to better understand the electronic and magnetic properties of large and complex materials. As an example of a large biomolecule, a B-DNA decamer with 733 atoms was analyzed. The DDEC NACs were in good agreement with previously reported AMBER and CHARMM force field charges. As an example of a well-studied single molecule magnet, the manganese-12 acetate (Mn_{12}-ac SMM) was analyzed. The DDEC ASMs and magnetic anisotropy barrier (computed by spin orbit coupling) were in good agreement with previous experimental and computational studies. Recently, people have developed organic materials that exhibit good ferroelectric properties. As an example, the diisopropylammonium bromide (DIPAB) was studied. The ferroelectric and non-ferroelectric phases showed similar DDEC NACs. This confirms the good conformational transferability of the DDEC NACs.
# TABLE OF CONTENTS

LIST OF TABLES........................................................................................................ ix

LIST OF FIGURES ........................................................................................................ x

CHAPTER 1. INTRODUCTION.................................................................................. 1

CHAPTER 2. PROGRAM ARCHITECTURE AND PARALLELIZATION
STRATEGY .................................................................................................................. 5

2.1 Program structure........................................................................................ 5

2.2 Computational precision ........................................................................... 15

2.3 Parallelization strategy and memory management .............................. 19

CHAPTER 3. RESULTS AND DISCUSSION ......................................................... 29

3.1 Parallelization results ................................................................................ 29

3.2 B-DNA decamer (CCATTAATGG)$_2$ ....................................................... 38

3.3 Mn$_{12}$-ac single molecule magnet ........................................................... 46

3.4 Diisopropylammonium bromide ............................................................... 49

CHAPTER 4. SUMMARY AND OUTLOOK ......................................................... 53

4.1 Summary ................................................................................................... 53

4.2 Outlook ..................................................................................................... 54

APPENDIX ................................................................................................................. 55

OZONE .XYZ FILE GENERATED FOR NAC ANALYSIS ........................... 55

OZONE .XYZ FILE GENERATED FOR ASM ANALYSIS ........................... 57

OZONE .XYZ FILE GENERATED FOR EBO ANALYSIS ........................... 58
LIST OF TABLES

Table 1. DDEC features in net atomic charge analysis. ......................................... 9
Table 2. DDEC features in atomic spin moment analysis. ............................... 10
Table 3. DDEC features in effective bond order analysis. .............................. 11
Table 4. Eigenvalues predicted according to the shape of the atomic density. .... 14
Table 5. Parameters related to computational precision .................................. 16
Table 6. Parameters for computing NACs ....................................................... 17
Table 7. Parameters for computing ASMs ....................................................... 18
Table 8. Parameters for computing EBOs ....................................................... 18
Table 9. Parameters related to WFX files ....................................................... 19
Table 10. Timing results of Fortran DDEC and Matlab DDEC codes .......... 30
Table 11. Comparison of DDEC, CHARMM and AMBER NACs for the B-DNA decamer ................................................................. 42
Table 12. Accuracy of DDEC NACs for reproducing the electrostatic potential of the B-DNA decamer ................................................................. 45
Table 13. Comparison of DDEC ASMs to previously reported experimental and computational values for Mn_{12}-ac single molecule magnet. ................................. 47
Table 14. Polarization density of DIPAB ......................................................... 52
LIST OF FIGURES

Figure 1. Importance of AIM properties ........................................................................... 2
Figure 2. Types of systems previously analyzed with the DDEC methodology ................. 3
Figure 3. Flow diagram of DDEC method ........................................................................... 12
Figure 4. Instruction that is executed only if the compiler supports OpenMP, otherwise it is ignored ........................................................................................................... 20
Figure 5. Example of OpenMP directive that creates the threads and divides the work. The loop is parallelized over j .................................................................................. 21
Figure 6. Shared memory parallelization elements considered to improve DDEC Fortran ......................................................................................................................... 22
Figure 7. Example of the arrangement of loop and matrix indices to maximize cache efficiency. This structure was kept throughout the program ........................................... 24
Figure 8. Illustration of memory usage by big arrays. Green: all systems; yellow: collinear magnetism; red: non-collinear magnetism. The height of each cell is proportional to the memory usage ................................................................. 27
Figure 9. Timing and parallelization results for Ni bulk metal (1 atom/unit cell, PBE/Planewave method) ................................................................................................. 31
Figure 10. Timing and parallelization results for the non-collinear single molecule magnet Fe₄O₁₂N₄C₄₀H₅₂ (112 atoms, PW91/25/Planewave method) ........................................... 32
Figure 11. Timing and parallelization results for Zr bisperoxy complex (133 atoms, B3LYP/6-311++G** method) ................................................................................. 33
Figure 12. Time and parallelization results for Zr bisperoxy complex (133 atoms, B3LYP/LANL2DZ method) .................................................................................... 34
Figure 13. Timing and parallelization results for O₃ triplet (3 atoms, CCSD/AUG-CC-PVTZ method .wfx file input) ............................................................ 35
Figure 14. Timing and parallelization results for O₃ triplet (3 atoms, CCSD/AUG-CC-PVTZ method Gaussian cube file input) ........................................... 36
Figure 15. B-DNA decamer. The unit cell is indicated by the white parallelepiped. The atoms are colored by element (gray: C; red: O, white: H, blue: N; purple: Na). 39

Figure 16. Timing and parallelization results for the B-DNA decamer (733 atoms/unit cell, PBE/Planewave method). 40

Figure 17. Comparison of CHARMM and DDEC NACs for B-DNA. 43

Figure 18. Comparison of AMBER and DDEC NACs for B-DNA. 44

Figure 19. Mn_{12}-ac single molecule magnet. Blue: Mn type 1; red: Mn type 2; orange: Mn type 3. 47

Figure 20. Spin orbit coupling potential energy surface of the Mn_{12}-ac molecule. Shown here are 62 points on the unit sphere, which represent 62 possible orientations of the global spin quantization axis (red: lowest energy, blue: highest energy). 48

Figure 21. DIPAB molecular crystals. Top: colored by element (gray: C; white: H; red: Br; blue: N). Bottom: colored by NAC (green: -0.21 to -0.23; yellow: -0.40 to -0.42; red: -0.59 to -0.65; light blue: 0.04 to 0.05; medium blue: 0.11 to 0.13; dark blue: 0.22 to 0.23). 50
CHAPTER 1. INTRODUCTION

Computational Chemistry is based on computations performed in two areas: *ab initio* Quantum Chemistry (QC) and classical methods (such as Molecular Dynamics (MD) and Monte-Carlo (MC)). As illustrated in Figure 1, AIM methods use the information obtained by QC calculations to compute properties like the net atomic charges (NACs), atomic spin moments (ASMs), effective bond orders (EBOs), atomic multipoles and electron cloud parameters of a system among others properties. These descriptors are fundamental in understanding why materials behave the way they do. They can also be used to construct force fields used in atomistic simulations. For example, NACs can be used in force fields to reproduce the electrostatic potential surrounding a material.

One of the AIM methods that has been developed is the Density Derived Electrostatic and Chemical (DDEC) method. It can compute the NACs\(^1,2\), ASMs\(^3\) and EBOs\(^4\) of a system from the electronic density and spin distributions and works with a wide variety of systems, such as periodic and non-periodic materials, porous or dense materials. It can be applied to non-magnetic systems or those with collinear or non-collinear magnetism. This method computes conformationally transferable NACs that can be used to constructs flexible force fields.\(^2,5,6\) The computed ASMs are similar to polarized neutron diffraction experiments.\(^3\)
Figure 2 illustrates the different kind of systems DDEC has been used for: (a) organometallic complexes, such as a Zr bisperoxy complex\(^2\), (b) highly correlated materials, such as the Fe\(_3\)O\(_4\) (001) surface antiphase domain boundary\(^7\), (c) several kinds of biomolecules, such as the ubiquitin molecule (when DDEC was implemented in ONETEP program)\(^5\), (d) materials with non-collinear magnetism such as the Fe\(_4\)O\(_{12}\)N\(_4\)C\(_{40}\)H\(_{52}\) single molecule magnet\(^3\), (e) small molecules such as the O\(_3\) triplet\(^3\).
(f) catalytic processes such as C-H bond cleavage of propane σ-complexes on PdO (101)\(^8\), (g) gas adsorption in porous materials such as CO\(_2\) adsorption in ZIF-8 metal organic frameworks (MOFs)\(^9\). DDEC has also been compared to spectroscopy results such as those of Mo-containing compounds\(^10\).

**Figure 2.** Types of systems previously analyzed with the DDEC methodology.

The overall objective of this project was to write a parallel Fortran code to perform DDEC analysis. The previous version of the code was written in Matlab, an interpreted computer language. We hypothesized that translating the code into a
compiled language would result in a large speedup. The design criteria for the parallel Fortran program were:

1) The code must be written to run either as a parallel or serial program.
2) The one-processor parallelized version, should run at least 85% as quickly as the serial version.
3) When running on 16 processors, the parallelization efficiency should be 75% on average. The program will be tested up to 32 processors, but the required parallelization efficiency will be based on 16 processors.
4) The code should compile correctly on a variety of compilers that support Fortran 2008 (e.g., GNU, Intel, and PGI).
5) Parallelized version must be tested on several different computational clusters.
6) Some of the systems to be tested should contain at least 2000 atoms.
7) The Fortran code should be able to read all of the input file types that could be read by the Matlab code.
8) The program should be easy to install on Linux (as a makefile) and Windows systems. A possibility for Windows would be to compile as binary the serial code.
9) The code must compute NACs, ASMs, EBOs and atomic dipoles and quadrupoles.
CHAPTER 2. PROGRAM ARCHITECTURE AND PARALLELIZATION STRATEGY

2.1 Program structure

The program reads the system’s information from an input file with the electron density and spin distributions. The program uses a uniform grid. For a periodic system the grid points fill a parallelepiped corresponding to a unit cell. For a non-periodic system the grid points are distributed over a parallelepiped enclosing the entire molecule. DDEC follows the stockholder partitioning:

\[ \rho_A(\vec{r}_A) = w_A(\vec{r}_A) \frac{\rho(\vec{r})}{W(\vec{r})} \]  

(1)

where \( w_A \) represents the relative amount of the density that will be assigned to a particular atom.

\[ W(\vec{r}) = \sum_{A,k} w_A(\vec{r}_A) \]  

(2)

where summation over A is all atoms in the unit cell and summation over k is all of the periodic images.

In 1977 Hirshfeld proposed to calculate \( w_A \) from a set of reference densities based on the density of isolated neutral atoms,\(^{11}\) so

\[ w_A = \rho_A^{ref}(r_A, z_A) \]  

(3)
this scheme provided a good approximation when computing the NACs but it also proved to be challenging to represent real ionic states and had difficulties reproducing the electrostatic potential outside the electron cloud, as it often underestimated the NACs.\textsuperscript{12}

Bultinck et al. proposed in 2007 the Iterative Hirshfeld method (IH). This method calculates $w_A$ from a set of reference densities based on isolated ions with the same net charge as in the molecule.\textsuperscript{13}

$$w_A = \rho_A^{ref}(\mathbf{r}_A, n_A) \hspace{1cm} (4)$$

An iterative process is used to optimize the weighting assigned to each atom. While this scheme can represent the electrostatic potential $V(\mathbf{r})$ outside the electron cloud, it has difficulties with obtaining accurate NACs in condensed materials.\textsuperscript{1,14-16}

In 2008 Lillestolen and Wheatley introduced the Iterative Stockholder Approach (ISA). It uses spherical averages of the electron density assigned to an atom to approximate $w_A$ by an iterative process:\textsuperscript{17}

$$w_A = \rho_A^{avg}(\mathbf{r}_A) \hspace{1cm} (5)$$

This scheme minimizes the atomic multipoles, so it can reproduce $V(\mathbf{r})$. However, as the electron density is not constrained to decay as a real atom’s, ISA has problems to assign the proper weight to buried atoms.\textsuperscript{1}
Several methods compute the ASMs, among them Mulliken, Bader and Natural Population Analysis (NPA). Mulliken is too basis set sensitive. Bader is capable of computing chemically meaningful NACs and reliable ASMs for periodic and non-periodic materials that may have collinear or non-collinear magnetism, but it cannot accurately reproduce the electrostatic potential outside the electron distribution. NPA can also calculate reliable ASMs and is not overly basis set sensitive, but it cannot be applied on periodic materials with non-collinear magnetism.

Manz and Sholl proposed in 2010 the Density Derived Electrostatic and Chemical method (DDEC)/c2. This scheme combines IH and ISA methodologies and achieves simultaneous optimization. It minimizes atomic multipoles thanks to the spherical averaging and represents real atomic states due to the reference densities:

$$w^\text{DDEC/c2}_A = \left( \rho^\text{ref}_A \right)^x \left( \rho^\text{avg}_A \right)^{1-x}$$

DDEC reference densities are computed with charge compensated reference ions, to account for the dielectric screening. However, DDEC/c2 could not obtain the proper charges for dense materials, such as Mo$_2$C and Pd$_3$V.

In 2012 Manz and Sholl improved their method with DDEC/c3. This improved methodology introduced conditioned reference densities, which allows to use the same optimal ratio of reference density to spherical average density in the
wA’s for all systems. It also introduced an exponential tail constraint that ensures the electronic density decays exponentially even in the buried tails regions².

Besides computing NACs, DDEC can compute ASMs³ and EBOs⁴. DDEC uses theoretical concepts to achieve higher accuracy and rapid convergence. Table 1-Table 3 show the main features of DDEC NACs, ASMs and EBOs analysis, respectively. Next to each feature is listed the feature’s purpose.

The program is divided into modules that are arranged to compute the NACs, ASMs and EBOs in that order as shown in Figure 3. The way each block works is described below.

**Read input files:** Currently, the program can read several formats containing the electronic density: VASP4, VASP5,¹⁹-²² .wfx, .xsf and .cube files. DDEC identifies the type of input file and begins to read with the according module. At this point the program might read the electron density and spin distribution in terms of grid points or in terms of basis set coefficients.

**Generate density grids:** In cases where the basis set coefficients are read from the input file, the program will choose the size of the grid with a preferred grid spacing. The program will compute the valence and spin densities. The density is represented as the linear combination of basis set products. If the basis function product is diffuse, the contribution to the valence and spin density will be calculated with a coarser grid and then interpolated back onto the regularly spaced grid.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stockholder type partitioning</td>
<td>Atomic densities sum to $\rho(\vec{r})$ at every point</td>
</tr>
<tr>
<td>Reference densities included in atomic weights</td>
<td>Assigned ${\rho_A(\vec{r}_A)}$ resemble real atoms/ions</td>
</tr>
<tr>
<td>Reference densities computed with charge</td>
<td>Accounts for dielectric screening in extended materials</td>
</tr>
<tr>
<td>compensation</td>
<td></td>
</tr>
<tr>
<td>Cutoff radius</td>
<td>Linear scaling</td>
</tr>
<tr>
<td>Exponential decay constraint</td>
<td>Capability to analyze nonporous dense solids with short bond lengths</td>
</tr>
<tr>
<td>Core-valence partitioning</td>
<td>Capability to use uniform grids</td>
</tr>
<tr>
<td>Spherical averaging included in atomic weights</td>
<td>Minimizes atomic multipoles to more accurately reproduce $V(\vec{r})$ with NACs</td>
</tr>
<tr>
<td>Reference density conditioning</td>
<td>Ideal reference density fraction is 3/14, independent of material</td>
</tr>
<tr>
<td>Occupancy corrections</td>
<td>Improves integration precision</td>
</tr>
<tr>
<td>Constrains $n_A \geq 0$</td>
<td>Resemble real atomic state</td>
</tr>
</tbody>
</table>
Table 2. DDEC features in atomic spin moment analysis.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stockholder type partitioning</td>
<td>( \bar{m}(\vec{r}) = \sum_{A,k} \bar{m}_A(\vec{r}_A) )</td>
</tr>
<tr>
<td>Spherical averaging included in optimization</td>
<td>ASMs more accurately reproduce ( B^{spin}(\vec{r}) ) around material</td>
</tr>
<tr>
<td>functional</td>
<td></td>
</tr>
<tr>
<td>Constraint ensures ( m_A(\vec{r}_A) \leq \rho_A(\vec{r}_A) )</td>
<td>Ensures chemical feasibility</td>
</tr>
<tr>
<td>Spin magnetization density represented as a vector</td>
<td>Works for collinear and non-collinear magnetism</td>
</tr>
<tr>
<td>Cutoff radius</td>
<td>Linear scaling computational cost</td>
</tr>
<tr>
<td>Proportional spin partitioning included in</td>
<td>Helps to ensure chemically reasonable results</td>
</tr>
<tr>
<td>optimization functional</td>
<td></td>
</tr>
<tr>
<td>Optimization functional based on</td>
<td>Unique and rapid convergence</td>
</tr>
<tr>
<td>information distance</td>
<td></td>
</tr>
</tbody>
</table>

**Add missing core**: If the input file lacks the information about the core electron density (e.g., if the electron density was obtained with a pseudopotential) the program will add back in the missing core density in order to perform an all-electron calculation.
**Table 3.** DDEC features in effective bond order analysis.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute atom-atom overlaps to determine which EBOs are negligible and uses cutoff radii</td>
<td>Linearly scaling computational cost</td>
</tr>
<tr>
<td>Uses spherical compartments to model the exchange hole</td>
<td>Reduces integration cost</td>
</tr>
<tr>
<td>Exchange hole model is normalized to obey Pauli exclusion principle</td>
<td>Improves accuracy</td>
</tr>
<tr>
<td>Exchange interactions formulated in vector form</td>
<td>Applies to collinear and non-collinear magnetism</td>
</tr>
<tr>
<td>Localized exchange hole (delta function) used to construct EBO upper and lower bounds</td>
<td>Improves accuracy</td>
</tr>
<tr>
<td>Coefficients for exchange hole delocalization computed with a self-consistent statistical model</td>
<td>Improves accuracy</td>
</tr>
<tr>
<td>Computes EBOs with different ratios of pair to total density (bond rescaling scheme)</td>
<td>Allows both EBO ratios and SEBOs to be accurately computed</td>
</tr>
<tr>
<td>EBOs and SEBOs computed as functionals of $(\rho(\vec{r}), m(\vec{r}))$</td>
<td>Ensures consistent results across various exchange-correlation theories</td>
</tr>
<tr>
<td>Based on DDEC/c3 $\rho_A(\vec{r}_A)$ and $\bar{m}_A(\vec{r}_A)$</td>
<td>Chemically meaningful results</td>
</tr>
</tbody>
</table>
Figure 3. Flow diagram of DDEC method.
Core partitioning: The program uses an iterative scheme to assign a core electron density to each atom and make sure it integrates to the correct number of core electrons for each atom.

Charge partitioning: This module performs an iterative process to obtain \( w_A \) and the density assigned to each atom. It makes sure the valence electron density integrates to the correct number of valence electrons.

Multipole moment analysis: The program computes the atomic dipole and quadrupole moments. If the material is non-periodic, then the total dipole and quadrupole moments are calculated about the system’s center-of-mass. This module also computes the eigenvalues of the atomic quadrupole traceless matrix. Table 4 shows the different cases of the eigenvalues. For completely spherical electronic density distributions, the eigenvalues will be zero. If it has an oblate spheroidal (stretched sphere) shape, it will have a negative eigenvalue and two positive eigenvalues with half the magnitude of the first one. If it has a prolate spheroidal (deflated sphere) shape, one eigenvalue will be positive and the other two will be negative with half the magnitude of the first one. If it has an ellipsoidal shape, all of the eigenvalues will have a different magnitude.

Generate net atomic charge file: An .xyz file is generated and is structured so it can be read by the Jmol program. It contains information about the atomic coordinates, NACs and atomic multipoles. An example of an .xyz file with NACs is included in the Appendix.
**Table 4.** Eigenvalues predicted according to the shape of the atomic density.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Eigenvalues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spherical</td>
<td>(0, 0, 0)</td>
</tr>
<tr>
<td>Oblate spheroidal</td>
<td>(-2(\lambda), +(\lambda), +(\lambda))</td>
</tr>
<tr>
<td>Prolate spheroidal</td>
<td>(+2(\lambda), -(\lambda), -(\lambda))</td>
</tr>
<tr>
<td>Ellipsoidal (three unequal eigenvalues)</td>
<td>((\lambda_1), (\lambda_2), (\lambda_3))</td>
</tr>
</tbody>
</table>
**Spin partitioning:** The ASMs will be computed if the input files have electron spin density information (collinear or non-collinear magnetism). The resulting ASMs are capable of reproducing the system’s magnetic field to good accuracy. When convergence is achieved, the program will write an .xyz file with the atomic coordinates and ASMs. The Appendix contains an example of an .xyz file with ASMs.

**Prepare EBO density grids:** The DDEC program will correct the density grids to include the occupancy correction.

**EBO calculation and generate EBO file:** The EBO modules calculates: (a) the amount of density that overlaps two atoms, (b) the EBOs, and (c) the sum of EBOs (SEBOs) for each atom. The EBOs and SEBOs are printed in an .xyz file. The Appendix contains an example of an .xyz file with the SEBO and EBO information.

### 2.2 Computational precision

To keep computational accuracy DDEC predefines some values. Table 5- Table 9 list the most important parameters used throughout the program for general purposes, NACs, ASMs, EBOs, and .wfx files, respectively. These parameters help to ensure linear scaling and accuracy of results.
Table 5. Parameters related to computational precision.

<table>
<thead>
<tr>
<th>General parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>dp</td>
</tr>
<tr>
<td>nshells</td>
</tr>
<tr>
<td>zero_tolerance</td>
</tr>
<tr>
<td>integration_tolerance</td>
</tr>
<tr>
<td>integration_tolerance_percent</td>
</tr>
<tr>
<td>pixel_integration_tolerance</td>
</tr>
<tr>
<td>cutoff_radius</td>
</tr>
</tbody>
</table>
### Table 6. Parameters for computing NACs.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Function</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference_weighting</td>
<td>3/14</td>
<td>Proportion of reference ion density to include in atomic weights</td>
<td>N/A</td>
</tr>
<tr>
<td>charge_convergence_tolerance</td>
<td>10⁻⁵</td>
<td>All NACs must vary less than this to converge</td>
<td>Electrons</td>
</tr>
<tr>
<td>density_decaying_exponent</td>
<td>1.75</td>
<td>Decay exponent for electron density of buried atom tails</td>
<td>bohr⁻¹</td>
</tr>
<tr>
<td>wA_renormalization_max</td>
<td>10</td>
<td>Defines the non-overlapping atom tolerance to ensure a non-zero number of electrons are assigned to an atom</td>
<td>N/A</td>
</tr>
<tr>
<td>rmin_cloud_penetration (does not affect assigned NACs or atomic electron densities)</td>
<td>200</td>
<td>The spherical average density used to calculate exponential tail decay exponent between r_{min cloud penetration} and r_{cutoff}</td>
<td>picometers</td>
</tr>
</tbody>
</table>
Table 7. Parameters for computing ASMs.

<table>
<thead>
<tr>
<th>Spin related parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>spin_convergence_tolerance</td>
</tr>
<tr>
<td>spin_ref_fraction</td>
</tr>
<tr>
<td>num_lookup_points</td>
</tr>
<tr>
<td>Xi_threshold</td>
</tr>
<tr>
<td>Xi_zero_tolerance</td>
</tr>
</tbody>
</table>

Table 8. Parameters for computing EBOs.

<table>
<thead>
<tr>
<th>EBO related parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable</strong></td>
</tr>
<tr>
<td>EBO_print_cutoff</td>
</tr>
<tr>
<td>C</td>
</tr>
</tbody>
</table>
Table 9. Parameters related to WFX files.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Function</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>preferred_grid_spacing</td>
<td>0.14</td>
<td>Preferred spacing between grid points</td>
<td>bohr</td>
</tr>
<tr>
<td>periodic_cutoff_length</td>
<td>28</td>
<td>Basis set functions that are further apart than this are neglected when computing basis set products</td>
<td>bohr</td>
</tr>
<tr>
<td>coarser_grid_alpha_cutoff</td>
<td>0.4</td>
<td>Smaller alpha values will integrate primitive pairs over coarser grids</td>
<td>bohr(^2)</td>
</tr>
<tr>
<td>analytic_alpha_cutoff</td>
<td>5</td>
<td>Primitive pairs with alpha sum greater than this are treated analytically</td>
<td>bohr(^2)</td>
</tr>
<tr>
<td>gaussian_overlap_tolerance</td>
<td>(10^{-12})</td>
<td>Gaussian basis function overlaps smaller than this are neglected</td>
<td>N/A</td>
</tr>
</tbody>
</table>

2.3 Parallelization strategy and memory management

To achieve faster results compared to Matlab, the existing Matlab DDEC code was translated into Fortran 2008 code. The advantage of compiled languages (such as
Fortran) is that they are usually faster than interpreted languages (such as Matlab). Fortran has the advantage that it can be easily parallelized.

Two main schemes are available to parallelize a program: (a) using distributed memory (for example, message passing interface, or MPI), and (b) using shared memory (for example, open multi-processing, or OpenMP).

The parallelization scheme chosen was to use shared memory. It is easy to create a shared memory program if the serial program already exists. The only thing needed is to add OpenMP directives. Another advantage is that the user does not have to compile extra libraries to get the OpenMP parallelization. If the compiler does not support OpenMP, the program will be compiled in the serial mode and the parallelization directives will be ignored. Examples of OpenMP directives are shown in Figure 4 and Figure 5.

![chunk_size = ceiling((2*delta_nc+1)/real(number_of_threads*6))](image)

**Figure 4.** Instruction that is executed only if the compiler supports OpenMP, otherwise it is ignored.
Figure 5. Example of OpenMP directive that creates the threads and divides the work. The loop is parallelized over j.

The limitation of using just OpenMP as a parallelization scheme is that the systems analyzed are restricted to a single node. The limiting factor is the number of processors it can be parallelized over. A node with two processors can run efficiently with up to two threads. Also, the memory available on a single node is a limiting factor. The nodes will not communicate with other nodes to get the extra memory the program needs if it is dealing with a system that requires more memory than the one the node has. The distributed memory protocol MPI can solve this problem. However, we used OpenMP, because it is easier to program than MPI.

In order to create an efficient shared memory parallel program the objectives listed in Figure 6 had to be achieved: (a) set big matrices as shared variables, (b) parallelize over grid points and bond pairs, (c) order the loop indices, (d) minimize
thread creation, (e) minimize number of critical directives, and (f) construct modules with global variables.

**Figure 6.** Shared memory parallelization elements considered to improve DDEC Fortran.
Tests were performed to analyze the effect of having a list of arguments passed in and out of the subroutines. The best time performance was shown when no arguments were used. Thus, the program was designed to work with global variables.

Cache is an intermediary between the processors and main memory. It stores temporary information so the processor does not have to travel back and forth to the main memory. However, cache is small and sometimes it cannot store all the information the program needs. In order to maximize cache efficiency, the number of fetches to main memory the processor makes should be kept to a minimum. Ordinarily, a fetch from main memory will retrieve 8 sequential double precision numbers. To maximize performance, the program should use all of these numbers before making the next fetch. This can be done by setting the innermost loops to change the fastest index of the matrices. In Fortran, the fastest index of a matrix is the first index. The organization of the loops in the program is represented in Figure 7.

Each variable in a parallel region of a code must be declared as shared or private. Shared variables are accessed by all the threads, while private variables have a different value for each thread. Private variables will not retain their value when the threads are destroyed. Temporary variables, index numbers, loop iteration indices, and other local variables were set as private. Big arrays (such as valence and spin density matrices) are treated as shared variables. Whenever a big matrix was no longer needed, it was deallocated to free up some memory space.
DO kc = 1,totnumC  
  DO kb = 1,totnumB  
    DO ka = 1,totnumA  \textbf{\scriptsize \textit{ensures \textcolor{green}{“ka” changes fastest}}}
      \textbf{\scriptsize \textit{multiple \textcolor{green}{“ka” values are fetched by cache}}}
        \textbf{\scriptsize \textit{ensures “ka” changes fastest}}
      \textbf{\scriptsize \textit{multiple “ka” values are fetched by cache}}
      \textbf{\scriptsize \textit{ensures “ka” changes fastest}}
      \textbf{\scriptsize \textit{multiple “ka” values are fetched by cache}}
    END DO  
  END DO  
END DO

\textbf{Figure 7.} Example of the arrangement of loop and matrix indices to maximize cache efficiency. This structure was kept throughout the program.

The largest matrices in the program (i.e., the ones that run over grid points) are shown in Figure 8. Each row stands for an array and each column stands for a module in the program (not all of the modules are represented, only the ones when the variables are allocated or deallocated). The continuous colored line represents when the variable exists. Green blocks are used by all of the systems; yellow blocks are used only by those systems with collinear magnetism; red blocks are used only by those systems with non-collinear magnetism. A tall cell equals three times the size of a small cell, because the information contained by that matrix is represented for each of the three Cartesian axes. The computation of the dominant atom volume is only calculated if the density was generated using the PAW method. As the figure suggests, the computation of the ASMs requires the largest amount of memory. The
largest memory requirement will be the valence iterator if the system is non-magnetic.

The approximated memory needed in order to run the program is derived from the system size and Figure 8. It is calculated from the number of big matrices that exist at a given time plus two to ensure the smaller matrices will be taken into account in the memory assignment:

- Non-magnetic materials will need the arrays in the modules in green in Figure 8:

\[(6 + 2) \times npoints \times 8 \text{ bytes} \quad (7)\]

- Collinear magnetism materials will need the arrays in the modules in green and yellow in Figure 8:

\[(9 + 2) \times npoints \times 8 \text{ bytes} \quad (8)\]

- Non-collinear magnetism materials will need the arrays in the modules in green and red in Figure 8:

\[(23 + 2) \times npoints \times 8 \text{ bytes} \quad (9)\]

Where \( npoints \) stands for the total number of points in the density grid and can be obtained from the input files.
The new Fortran DDEC program also has new modules compared to the Matlab code: (a) matrix operations because Fortran does not contain built-in functions for some matrix operations, (b) a module specifying double precision arithmetic, and (c) a string utilities module to modify character strings.

The OpenMP clause SCHEDULE was specified to control the way the work was distributed among the threads. The keyword STATIC was used when the processors would perform the same number of operations in each loop operation, ensuring balance of work among the processors. DYNAMIC was set when the number of operations per loop iteration could vary. In DYNAMIC scheduling, the next loop iteration is assigned to the first free processor. DYNAMIC scheduling achieves better load balancing but has higher overhead than STATIC scheduling.

The threads had to be created the least amount of times as possible. Overhead arises when time is spent in the creation and coordination of the threads. To achieve this, when feasible, thread creation was placed around the outermost loop index.

To ensure the use of all of the processors in the parallel regions, the loops were parallelized over grid points and bond pairs. This ensured that no processors would stay idle even when performing calculations over a single atom.

Several OpenMP directives are designed to prevent racing conditions. A racing condition occurs if two processors try to write to the same memory location at the same time. The ATOMIC directive was set when the value of a variable had to be updated in main memory by different processors. The REDUCTION directive was
**Figure 8.** Illustration of memory usage by big arrays. Green: all systems; yellow: collinear magnetism; red: non-collinear magnetism. The height of each cell is proportional to the memory usage.
used to parallelize sums, subtractions, finding a maximum or a minimum value, etc. The CRITICAL directive allows only one thread at a time to perform a set of code operations. ATOMIC, CRITICAL and REDUCTION were tested to see which one was the least computationally expensive. REDUCTION was the least time consuming. These directives were kept to a minimum to avoid overhead cost.
CHAPTER 3. RESULTS AND DISCUSSION

3.1 Parallelization results

The Fortran DDEC program was created based on the Matlab DDEC code (T. A. Manz, Chargemol program Matlab versions, ddec.sourceforge.net). The parallel Fortran DDEC code was tested on the Trestles cluster\textsuperscript{23} at the San Diego Supercomputing Center (SDSC) and the Stampede cluster at the Texas Advanced Computing Center through an XSEDE allocation. The final program was compiled with the module GNU/4.8.2. The parallel Fortran code was benchmarked against the Matlab code to make sure the same numerical results were obtained.

The parallel code was created after the Fortran DDEC code was working in serial version. Several tests were performed on Trestles\textsuperscript{23} on systems previously reported in order to make sure the program was working correctly. The tests were run on serial, 1, 2, 4, 8, 16 and 32 processors.

The program performance on previously studied methods is shown in Figures 9-14. These figures plot the results of running the program in serial, and 1, 2, 4, 8, 16, and 32 processors on Ni bulk metal, the Fe$_4$O$_{12}$N$_4$C$_{40}$H$_{52}$ single molecule magnet,\textsuperscript{3} Zr bisperoxy complex with two levels of theory: 6-311++G** (a large basis set)\textsuperscript{2} and LANL2DZ (a smaller basis set)\textsuperscript{2} and O$_3$ triplet.\textsuperscript{2} The ozone triplet at CCSC/aug-cc-pvtz level of theory was analyzed using two different types of input: (a) .wfx file (Figure 13) and (b) Gaussian .cube files (total, valence, and spin densities on a uniform grid with 0.2 bohr between adjacent grid points, Figure 14). The time on the
left y-axis represents the number of seconds it took to complete the different modules per atom (red: reading input files and generating the density grids; green: charge partitioning; blue: spin partitioning; yellow: EBO analysis). The title shows the number of atoms in the unit cell of each system and the level of theory used to compute the electron density files. The right y-axis shows the parallelization efficiency for each number of processors.

Table 10 compares the timing results of the Matlab and Fortran DDEC codes. In all of the systems tested, the Fortran program had the same precision as the Matlab program completed the NACs, ASMs and EBOs. The new Fortran DDEC program in serial mode ran around 8.4 times faster than the existing Matlab serial program.

**Table 10.** Timing results of serial Fortran DDEC and serial Matlab DDEC codes.

<table>
<thead>
<tr>
<th>System</th>
<th>Fortran time (min)</th>
<th>Matlab time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ni bulk metal</td>
<td>1.55</td>
<td>16.17</td>
</tr>
<tr>
<td>$\text{Fe}<em>4\text{O}</em>{12}\text{N}_4\text{C}<em>4\text{H}</em>{52}$ single molecule magnet</td>
<td>182.62</td>
<td>2684.83</td>
</tr>
<tr>
<td>Zr bisperoxy complex (B3LYP/6-311++G**)</td>
<td>1109.59</td>
<td>6585.73</td>
</tr>
<tr>
<td>Zr bisperoxy complex (B3LYP/LANL2DZ)</td>
<td>414.58</td>
<td>2743.78</td>
</tr>
<tr>
<td>$\text{O}_3$ triplet (wfx CCSD/AUG-CC-PVTZ)</td>
<td>7.50</td>
<td>49.52</td>
</tr>
<tr>
<td>$\text{O}_3$ triplet (cube CCSD/AUG-CC-PVTZ)</td>
<td>0.70</td>
<td>6.27</td>
</tr>
</tbody>
</table>
Figure 9. Timing and parallelization results for Ni bulk metal (1 atom/unit cell, PBE/Planewave method).
Figure 10. Timing and parallelization results for the non-collinear single molecule magnet Fe$_4$O$_{12}$N$_4$C$_{40}$H$_{52}$ (112 atoms/unit cell, PW91$^{24}$/Planewave method).
**Figure 11.** Timing and parallelization results for Zr bisperoxy complex (133 atoms, B3LYP/6-311++G** method).
Figure 12. Time and parallelization results for Zr bisperoxy complex (133 atoms, B3LYP/LANL2DZ method).
Figure 13. Timing and parallelization results for O$_3$ triplet (3 atoms, CCSD/AUG-CC-PVTZ method, wfx file input).
Figure 14. Timing and parallelization results for $O_3$ triplet (3 atoms, CCSD/AUG-CC-PVTZ method Gaussian cube file input).
According to the results, the program took about 100 seconds or less to compute the NACs, ASMs and EBOs. However, those calculations which read the Gaussian09\textsuperscript{25} generated .wfx files took significantly longer, because the density grids had to be computed from the basis set coefficients. When reading .wfx files, the time the parallel DDEC program spent per atom varied with the basis set used. A larger basis set resulted in a longer time compared to the smaller basis set: the parallel DDEC program took around 500 seconds per atom for Zr bisperoxy 6-311++G** on serial and 25 seconds per atom on 32 processors. The parallel DDEC program took 190 seconds per atom for Zr bisperoxy LANL2DZ on serial, and 10 seconds per atom on 32 processors.

Systems with one to three atoms (in the unit cell) showed good parallelization efficiency on 2–8 processors, but poor parallelization efficiency on 32 processors. The systems with one or three atoms in them took less than a minute to complete the DDEC analysis on 32 processors. Ni metal (1 atom) went from 93 seconds on serial to 9.4 seconds on 32 processors, while the ozone triplet CCSD file (.wfx input) completed the calculation in 30.7 seconds on 32 processors. The ozone triplet cube file finished the calculation in 5.3 seconds on 32 processors. The drop of efficiency in Ni is explained with the parallelization scheme: it switched from parallelizing bond pairs to parallelizing grid points when the number of bond pairs was smaller than the number of processors. Parallelizing over grid points represents a greater overhead compared to parallelizing over bond pairs, as the threads have to be created a greater
number of times. This approach was taken so no processors stayed idle in the calculation when computing the EBOs.

### 3.2 B-DNA decamer (CCATTAATGG)$_2$

The study of biological molecules is important to learn about the prevention and treatment of diseases. Computational chemistry can be a valuable tool to simulate how the biomolecules will behave under different conditions. Classical molecular dynamics and Monte-Carlo simulations require force-fields to describe the interactions between atoms in these systems. These force-fields typically contain NACs to describe the electrostatic interactions and additional parameters to describe other kinds of interactions.

We studied the B-DNA decamer (CCATTAATGG)$_2$ as an example of a biological molecule. The structure was obtained from a neutron diffraction experiment performed by Arai et al. (PDB ID: 1WQZ). Similar to previous studies, we placed a Na$^+$ ion near each phosphate group to simulate the B-DNA being in a real solution. The positions of the Na$^+$ ions were optimized with VASP, and all other atoms were fixed at the neutron diffraction positions. Figure 15 shows the B-DNA decamer structure with the unit cell. The electron density distribution was calculated with the PBE functional.
Figure 15. B-DNA decamer. The unit cell is indicated by the white parallelepiped. The atoms are colored by element (gray: C; red: O, white: H, blue: N; purple: Na).

The resulting files were used as input for the parallel Fortran DDEC code. The complete charge analysis in serial (32 processors) was finished in 20.69 hr (52 minutes). Figure 16 shows the parallelization profile of the B-DNA decamer. The parallelization efficiency on 32 processors was >70%. This shows that the code parallelizes very efficiently for large systems.
Figure 16. Timing and parallelization results for the B-DNA decamer (733 atoms/unit cell, PBE/Planewave method).
Several studies have previously computed NACs for B-DNA decamers.\textsuperscript{29-32} To validate our results, the DDEC NACs were compared to the NACs obtained with AMBER and CHARMM methods. The first column in Table 11 lists the type of atom analyzed. Each atom type was defined according to which atoms were directly bonded to it. For example, C-C2HO refers to a carbon atom that is bound to two other carbon atoms, one hydrogen atom, and one oxygen atom. Column 2 lists the average of DDEC NACs for each atom type. Columns 3, 4, and 5 list the standard deviation, maximum, and minimum of the DDEC NACs for each atom type. The CHARMM and AMBER NACs are listed in columns 6 to 8. The dash in the CHARMM column indicates that the article had no O-CH group, because the structures reported did not include a terminating chain.\textsuperscript{31} The dash in the first Amber column indicates that there was no carbon in the deoxyribose group connected to a base pair nitrogen, as the structures reported substituted each with a hydrogen.\textsuperscript{30}

Figure 17 and Figure 18 compare the DDEC NACs to the CHARMM and AMBER NACs, respectively. For comparison, the $y=x$ line is illustrated. The charges showed good agreement, except for the P NAC. For P, the CHARMM NAC was significantly higher than the DDEC NAC. For P, the AMBER NAC was slightly lower than the DDEC NAC. This shows that the DDEC NACs computed for B-DNA are reasonable.
Table 11. Comparison of DDEC, CHARMM and AMBER NACs for the B-DNA decamer.

<table>
<thead>
<tr>
<th>Group</th>
<th>DDEC PBE/planewave</th>
<th>CHARMM CM2 AM1&lt;sup&gt;a&lt;/sup&gt;</th>
<th>AMBER RESP HF/6-31G*&lt;sup&gt;b&lt;/sup&gt;</th>
<th>AMBER RESP HF/6-31G*&lt;sup&gt;c&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-C2H</td>
<td>-0.447</td>
<td>-0.332</td>
<td>-0.48</td>
<td>-0.522</td>
</tr>
<tr>
<td>C-C2H2</td>
<td>-0.368</td>
<td>-0.202</td>
<td>-0.055</td>
<td>-0.085</td>
</tr>
<tr>
<td>C-C2HO</td>
<td>0.173</td>
<td>0.103</td>
<td>0.281</td>
<td>0.117</td>
</tr>
<tr>
<td>C-C2N</td>
<td>-0.087</td>
<td>-0.057</td>
<td>0.243</td>
<td>0.136</td>
</tr>
<tr>
<td>C-C3</td>
<td>-0.129</td>
<td>-0.216</td>
<td>0.021</td>
<td>0.003</td>
</tr>
<tr>
<td>C-CH2O</td>
<td>0.069</td>
<td>0.074</td>
<td>0.107</td>
<td>-0.007</td>
</tr>
<tr>
<td>C-CH3</td>
<td>-0.301</td>
<td>-0.104</td>
<td>-0.28</td>
<td>-0.227</td>
</tr>
<tr>
<td>C-CHN</td>
<td>0.077</td>
<td>0.208</td>
<td>-0.07</td>
<td>-0.12</td>
</tr>
<tr>
<td>C-CHON</td>
<td>0.319</td>
<td>0.289</td>
<td>-</td>
<td>0.034</td>
</tr>
<tr>
<td>C-CN2</td>
<td>0.429</td>
<td>0.486</td>
<td>0.66</td>
<td>0.524</td>
</tr>
<tr>
<td>C-CON</td>
<td>0.552</td>
<td>0.681</td>
<td>0.459</td>
<td>0.506</td>
</tr>
<tr>
<td>C-HN2</td>
<td>0.225</td>
<td>0.407</td>
<td>0.226</td>
<td>0.269</td>
</tr>
<tr>
<td>C-N3</td>
<td>0.703</td>
<td>0.736</td>
<td>0.571</td>
<td>0.743</td>
</tr>
<tr>
<td>C-ON2</td>
<td>0.645</td>
<td>0.778</td>
<td>0.685</td>
<td>0.682</td>
</tr>
<tr>
<td>H-C</td>
<td>0.09</td>
<td>0.136</td>
<td>0.123</td>
<td>0.127</td>
</tr>
<tr>
<td>H-N</td>
<td>0.338</td>
<td>0.419</td>
<td>0.397</td>
<td>0.405</td>
</tr>
<tr>
<td>N-C2</td>
<td>-0.502</td>
<td>-0.603</td>
<td>-0.605</td>
<td>-0.693</td>
</tr>
<tr>
<td>N-C2H</td>
<td>-0.488</td>
<td>-0.692</td>
<td>-0.389</td>
<td>-0.679</td>
</tr>
<tr>
<td>N-C3</td>
<td>-0.153</td>
<td>-0.507</td>
<td>-0.384</td>
<td>-0.007</td>
</tr>
<tr>
<td>N-CH2</td>
<td>-0.673</td>
<td>-0.739</td>
<td>-0.912</td>
<td>-0.938</td>
</tr>
<tr>
<td>O-C</td>
<td>-0.533</td>
<td>-0.563</td>
<td>-0.58</td>
<td>-0.592</td>
</tr>
<tr>
<td>O-C2</td>
<td>-0.342</td>
<td>-0.373</td>
<td>-0.489</td>
<td>-0.369</td>
</tr>
<tr>
<td>O-CH</td>
<td>-0.538</td>
<td>-</td>
<td>-0.664</td>
<td>-0.593</td>
</tr>
<tr>
<td>O-CP</td>
<td>-0.506</td>
<td>-0.792</td>
<td>-0.517</td>
<td>-0.509</td>
</tr>
<tr>
<td>O-P</td>
<td>-0.942</td>
<td>-1.23</td>
<td>-0.797</td>
<td>-0.776</td>
</tr>
<tr>
<td>P-O4</td>
<td>1.53</td>
<td>2.54</td>
<td>1.237</td>
<td>1.166</td>
</tr>
</tbody>
</table>

<sup>a</sup> CM2 AM1 Hamiltonian is the method used by Khandogin before using CHARMM.<sup>31</sup>  
<sup>b</sup> RESP HF/631G* used by Jayaram to compute charges for AMBER.<sup>30</sup>  
<sup>c</sup> RESP 6-31G* used by Cornell to calculate partial charges for AMBER.<sup>29</sup>
Figure 17. Comparison of CHARMM and DDEC NACs for B-DNA.
Figure 18. Comparison of AMBER and DDEC NACs for B-DNA.
The root mean squared error (RMSE) in the electrostatic potential quantifies how accurately a force-field model can reproduce the electrostatic potential surrounding a material. The relative root mean squared error (RRMSE) is the ratio of the RMSE using NACs to the RMSE using no charges. A RMSE analysis was performed to determine how accurately our NACs would construct a force field. The results are displayed in Table 12 and show a RRMSE of ~0.58. This result shows that using NACs and atomic dipoles improved the description of the electrostatic potential compared to no charges. However, the improvement was not as much as we desired.

Table 12. Accuracy of DDEC NACs for reproducing the electrostatic potential of the B-DNA decamer.

<table>
<thead>
<tr>
<th></th>
<th>RMSE (kcal/mol)</th>
<th>RRMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No charges</td>
<td>23.66</td>
<td>1.00</td>
</tr>
<tr>
<td>DDEC NACs</td>
<td>13.89</td>
<td>0.59</td>
</tr>
<tr>
<td>DDEC NACs and atomic dipoles</td>
<td>13.76</td>
<td>0.58</td>
</tr>
</tbody>
</table>
3.3 Mn$_{12}$-ac single molecule magnet

The Mn$_{12}$-ac single molecule magnet (Mn$_{12}$-ac SMM) has been studied extensively since its synthesis and discovery by Lis.$^{33}$ This structure was obtained from the crystal databank (CDB ID: BESXAA by Farrell et al.$^{34}$) and the electronic density was obtained with Gaussian09$^{25}$ (PBE/LANL2DZ) and VASP5 (PBE/Planewave). The files were analyzed with the parallel DDEC Fortran code. The program took around 36 minutes to calculate the NACs, ASMs and EBOs for the Gaussian files on 32 processors, while it took 13 minutes to calculate the same properties for the VASP5 files on 32 processors.

Figure 19 shows the Mn$_{12}$-ac SMM representing the Mn atoms as balls. The ASMs of the central Mn atoms point towards the reader while the ASMs of the outer Mn atoms point away from the reader. As shown in Table 13, the DDEC ASMs are in good agreement with the ASMs of Robinson et al.’s neutron diffraction experiment$^{35}$ and Pederson and Khanna’s DFT calculation$^{36}$.

The magnetic anisotropy barrier is the energy required to flip the magnetic orientation of a molecular magnet in the absence of a magnetic field. VASP5 was used to perform a spin orbit coupling calculation to obtain the Mn$_{12}$-ac SMM anisotropy barrier. The energy profile is represented in Figure 20. The red points lay along the easy magnetization axis (the axis along which the molecule points its ASM spontaneously). To rotate the magnetic moment requires a maximum energy of 5.13 meV (represented by the dark blue band) which is when the magnetic moment would
Figure 19. Mn$_{12}$-ac single molecule magnet. Blue: Mn type 1; red: Mn type 2; yellow: Mn type 3.

Table 13. Comparison of DDEC ASMs to previously reported experimental and computational values for Mn$_{12}$-ac single molecule magnet.

<table>
<thead>
<tr>
<th>Type of atom</th>
<th>DDEC PBE Planewave</th>
<th>DDEC PBE LANL2DZ</th>
<th>Polarized neutron diffraction$^a$</th>
<th>Pederson PBE$^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mn type 1</td>
<td>-2.79</td>
<td>-2.55</td>
<td>-2.34±0.13</td>
<td>-2.6</td>
</tr>
<tr>
<td>Mn type 2</td>
<td>3.81</td>
<td>3.61</td>
<td>3.79±0.12</td>
<td>3.6</td>
</tr>
<tr>
<td>Mn type 3</td>
<td>3.81</td>
<td>3.55</td>
<td>3.69±0.14</td>
<td>3.6</td>
</tr>
</tbody>
</table>

$^a$ Robinson et al.$^{35}$. $^b$ Pederson and Khanna using integration of the spin density over spheres of 2.5 bohr radius to compute the ASMs.$^{36}$
Figure 20. Spin orbit coupling potential energy surface of the Mn12-ac molecule. Shown here are 62 points on the unit sphere, which represent 62 possible orientations of the global spin quantization axis (red: lowest energy, blue: highest energy).

be pointing 90° with respect to the easy axis. This magnetic anisotropy barrier corresponds to a temperature of 59.5 K, and is in good agreement with the experimental value of 60-62 K by Fort et al.\textsuperscript{37} and the computed value of 55.6-55.8 K by Pederson and Khanna\textsuperscript{36}. 

48
3.4 Diisopropylammonium bromide

The last system studied was diisopropylammonium bromide (DIPAB). The structures of the following conformations were obtained from the crystal data base: $P2_1$ (293 K), $P2_12_12_1$ (293 K), and $P2_1/m$ (438 K). Phase $P2_1$ is ferroelectric. $P2_12_12_1$ becomes $P2_1$ when heated and $P2_1$ becomes $P2_1/m$ when heated more. $P2_1/m$ becomes $P2_1$ when cooled but not $P2_12_12_1$ if cooled further.

We performed DFT calculations on these three phases using VASP5. For $P2_1$ and $P2_12_12_1$, the lattice constants and angles where held at the experimental values and the positions of all atoms were fully relaxed. For $P2_1/m$, the experimental X-ray structure shows disordered atoms, so we created a super-cell including both components of the disordered structure and relaxed the atomic positions. The three structures were then analyzed with the parallel Fortran DDEC code to get their NACs and EBOs. Figure 21 shows the structure of the three systems and how the NACs stayed stable when the phase changed.

A ferroelectric phase has a net dipole moment per unit cell. The polarization density $P$ is the average dipole moment per unit volume. For an ionic material such as DIPAB, the polarization density depends upon the surface termination. Different surface terminations will yield different polarization densities. Fu et al. performed measurements and calculations of the polarization density of DIPAB phase $P2_1$. Using a Sawyer-Tower circuit, they measured the polarization of approximately 0.11 C/m$^2$. Using a pyroelectric technique, they measured the polarization of
approximately 0.2 C/m². Using DFT calculations, they computed the polarization of 0.23 C/m², however, the methodology was not explained in the paper. Fu et al. did not specify the surface terminations associated with their polarization density measurements and calculations.

Figure 21. DIPAB molecular crystals. Top: colored by element (gray: C; white: H; red: Br; blue: N). Bottom: colored by NAC (red: -0.59 to -0.65; yellow: -0.38 to -0.42; green: -0.21 to -0.23; light blue: 0.04 to 0.06; medium blue: 0.10 to 0.14; dark blue: 0.21 to 0.26).

To better understand these materials, we computed the polarization density of the three DIPAB phases using the DDEC NACs and (optionally) atomic dipoles.
When atomic dipoles are included, the total dipole moment exactly reproduces the system’s dipole moment up to a small integration error.\textsuperscript{2} Table 14 summarizes the results. Structures $P2_1\bar{2}12_1$ and $P2_1/m$ resulted in a total dipole moment close to zero, a value expected as these structures do not present any ferroelectric properties. The total dipole moment of structure $P2_1$ was calculated using both lower and higher polarity surface terminations. The lower polarity structure has a cation and an anion in the same surface layer. The higher polarity structure has anions on one surface of the slab and cations on the other surface of the slab. For the lower polarity termination, the value obtained using just the DDEC NACs was 0.0348 C/m\textsuperscript{2}. For the higher polarity termination, the value obtained using just the DDEC NACs was 0.1495 C/m\textsuperscript{2}. Including atomic dipoles had negligible effect on the polarization densities. The value obtained for the lower polarity surface termination was smaller than the values reported by Fu et al. Our value obtained for the higher polarity surface termination was similar to the results of Fu et al. using the Sawyer-Tower circuit and substantially lower than their pyroelectric technique and DFT results.
Table 14. Polarization density of DIPAB.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Termination</th>
<th>Polarization density (C/m²) computed using</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NACs</td>
</tr>
<tr>
<td>$P2_1$</td>
<td>Lower polarity</td>
<td>0.0348</td>
</tr>
<tr>
<td>$P2_1$</td>
<td>Higher polarity</td>
<td>0.1495</td>
</tr>
<tr>
<td>$P2_1212_1$</td>
<td></td>
<td>0.000</td>
</tr>
<tr>
<td>$P2_1/m$</td>
<td></td>
<td>0.000</td>
</tr>
</tbody>
</table>
CHAPTER 4. SUMMARY AND OUTLOOK

4.1 Summary

The approach to parallelize the DDEC code using Fortran 2008 was presented in order to obtain NACs, ASMs, EBOs, and atomic dipoles and quadrupoles from the same input file types that the Matlab DDEC code could read. The parallel Fortran code showed parallelization efficiencies of around 60% for 32 processors and around 65% for 16 processors. The OpenMP parallelized code can also be compiled in serial mode. Therefore, the code does not require anything beyond the standard Fortran 2008 compiler.

This Parallel Fortran DDEC code was tested on the Trestles and Stampede clusters. Six systems were studied previously and were used to test Fortran DDEC parallelization efficiency: (a) the non-collinear single molecule magnet Fe₄O₁₂N₄C₄₀H₅₂, (b) Ni metal, (c) Zr bisperoxy complex (6-311++G**), (d) Zr bisperoxy complex (LANL2DZ), and (e) ozone triplet (.wfx and .cube). The tests were performed on the Trestles cluster using serial and one, two, four, eight, sixteen and thirty-two processors. For non-periodic systems, preparing the density grids from the .wfx files proved to be very time consuming and represents an opportunity to improve the time spent preparing those grids. The DDEC NACs, ASMs and EBOs were computed for the first time for three systems: (a) B-DNA decamer, (b) Mn12-ac single molecule magnet, and (c) diisopropylammonium bromide. The results are compared to previous experimental and computational studies.
4.2 Outlook

For molecular systems, the time the program spent generating the density grids from .wfx files was too high. This problem could be solved using variable space atom-centered grids instead of uniform grids. This would allow computing the density near the center of the atoms with fewer points and higher accuracy.

In order to broaden DDEC use, the program could be tested on compilers besides GNU, such as Intel or PGI. Windows users might be interested in running an .exe version of the code, so an executable file could be distributed. A makefile can also be created to make compilation easier to the user.

More than 2000 atoms could not be tested because using Gaussian09\textsuperscript{25} and VASP\textsuperscript{19-22} on large systems greatly increases the time required to obtain the electronic density. One solution could be using a linearly scaling DFT code to generate the electron density. Another solution would be to use a less accurate but more computationally efficient method than DFT to generate the electron density.
APPENDIX

OZONE .XYZ FILE GENERATED FOR NAC ANALYSIS

3
Nonperiodic system
O  0.000000  -0.000000  0.579101  0.229616
O  0.000000   0.990774 -0.289550 -0.114808
O  0.000000  -0.990774 -0.289550 -0.114808

See ddec.sourceforge.net for latest version.

Computational parameters:
Reference_weighting = 0.214286
Number of radial integration shells = 100
Cutoff radius (pm) = 500
Error in the integrated total number of electrons before renormalization (e) = 2.8468E-09
Charge convergence tolerance = 1.0000E-05
Minimum radius for electron cloud penetration fitting (pm) = 200
Decay exponent for electron density of buried atom tails = 1.7500
Number of iterations to convergence = 22

The following XYZ coordinates are in angstroms. The atomic dipoles and quadrupoles are in atomic units.

atom number, atomic symbol, x, y, z, net_charge, dipole_x, dipole_y, dipole_z, dipole_mag, Qxy, Qxz, Qyz, Q(x^2-y^2), Q(3z^2-R^2), three eigenvalues of traceless quadrupole moment tensor

1 O  0.000000  -0.000000  0.579101  0.229616  -0.000000  -0.000000  -0.161890  0.161890  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
-0.060077  -0.059434   0.119511

2 O  0.000000   0.990774 -0.289550 -0.114808  0.000000  -0.126438 -0.333035  0.073788
0.053938  0.055161  -0.183830  -0.179203
-0.176332  -0.051897   0.228229

3 O  0.000000  -0.990774 -0.289550 -0.114808  0.000000  -0.126438 -0.333035  0.073788
0.053938  0.055161  -0.183830  -0.179203
-0.176332  -0.051897   0.228229

The sperically averaged electron density of each atom fit to a function of the form exp(a - br) for r >=min_cloud_penetration
atom number, atomic symbol, x, y, z, a, b, Rsquared where a and b are in atomic units and Rsquared is the squared correlation coefficient
Since the system is non-periodic, the total dipole and quadrupole moments were computed relative to the system center of mass.

Multipole moments (in atomic units) using the net atomic charges:

Partial_charge_dipole_x = 0.0000  Partial_charge_dipole_y = 0.0000
Partial_charge_dipole_z = 0.3769  Partial_charge_dipole_magnitude = 0.3769
Partial_charge_quadrupole_xy = 0.0000  Partial_charge_quadrupole_xz = 0.0000  Partial_charge_quadrupole_yz = -0.0000
Partial_charge_quadrupole_x2minusy2 = 0.8049
Partial_charge_quadrupole_3z2minusr2 = 1.2174

Eigenvalues of traceless quadrupole moment tensor = -0.6054  0.1996  0.4058

Multipole moments (in atomic units) using the net atomic charges and atomic dipoles:

Dipole_x = -0.0000  Dipole_y = 0.0000  Dipole_z = 0.3229
Dipole_magnitude = 0.3229
Quadrupole_xy = 0.0000  Quadrupole_xz = -0.0000  Quadrupole_yz = -0.0000
Quadrupole_x2minusy2 = 0.8915  Quadrupole_3z2minusr2 = 0.3592

Eigenvalues of traceless quadrupole moment tensor = -0.5056  0.1197  0.3859

Multipole moments (in atomic units) using the net atomic charges, atomic dipoles, and atomic quadrupoles:

Dipole_x = -0.0000  Dipole_y = 0.0000  Dipole_z = 0.3229
Dipole_magnitude = 0.3229
Full_quadrupole_xy = 0.0000  Full_quadrupole_xz = 0.0000
Full_quadrupole_yz = -0.0000  Full_quadrupole_x2minusy2 = 0.0416
Full_quadrupole_3z2minusr2 = 0.3275

Eigenvalues of traceless quadrupole moment tensor = -0.0754  -0.0338  0.1092
OZONE .XYZ FILE GENERATED FOR ASM ANALYSIS

3
Nonperiodic system
O  0.000000  -0.000000   0.579101   0.425102
O  0.000000   0.990774  -0.289550   0.787449
O  0.000000  -0.990774  -0.289550   0.787449

Collinear spin population analysis was performed
The total spin magnetic moment of the unit cell is  2.000000

See ddec.sourceforge.net for latest version.

Computational parameters:
Spin_ref_fraction = 0.50
Number of radial integration shells = 100
Cutoff radius (pm) = 500
Spin convergence tolerance = 0.000050
Number of iterations to convergence =  7

1 O  0.000000  -0.000000   0.579101   0.425102
2 O  0.000000   0.990774  -0.289550   0.787449
3 O  0.000000  -0.990774  -0.289550   0.787449
2014/03/11  19:19:46
3 nonperiodic system
O  0.000000  -0.000000  0.579101  2.990753
O  0.000000   0.990774  -0.289550  1.753922
O  0.000000  -0.990774  -0.289550  1.753922

See ddec.sourceforge.net for latest version.

The sum of effective bond orders (SEBO) for each atom in the unit cell are listed above.
All effective bond orders (EBOs) greater than 0.001000 are printed below.

=============================================================
Printing EBOs for ATOM # 1 (O) in the reference unit cell.
Bonded to the (0, 0, 0) translated image of atom number 2 (O) with EBO = 1.4954
Bonded to the (0, 0, 0) translated image of atom number 3 (O) with EBO = 1.4954
The sum of effective bond orders for this atom is SEBO = 2.990753

=============================================================
Printing EBOs for ATOM # 2 (O) in the reference unit cell.
Bonded to the (0, 0, 0) translated image of atom number 1 (O) with EBO = 1.4954
Bonded to the (0, 0, 0) translated image of atom number 3 (O) with EBO = 0.2585
The sum of effective bond orders for this atom is SEBO = 1.753922

=============================================================
Printing EBOs for ATOM # 3 (O) in the reference unit cell.
Bonded to the (0, 0, 0) translated image of atom number 1 (O) with EBO = 1.4954
Bonded to the (0, 0, 0) translated image of atom number 2 (O) with EBO = 0.2585
The sum of effective bond orders for this atom is SEBO = 1.753922

2014/03/11 19:19:49
REFERENCES


