On evaluation of electrostatic interaction energies in molecular crystals within the pseudoatom electron density formalism

by

Daniel Nguyen

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computational Science

Middle Tennessee State University

March 16, 2020

Dissertation Committee:

Dr. Anatoliy Volkov, Chair

Dr. Tibor Koritsanszky

Dr. Yuri Melnikov

Dr. Joshua L. Phillips

Dr. William Robertson

To my late father

& my wonderful family.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Anatoliy Volkov, for giving me a wonderful opportunity to work under his supervision. I appreciate his never-ending patience, time, and support that he has generously given to me for the past five years. Completing this dissertation under his guidance, I have learned so much from him, not just about doing research, but also about how to approach life. I am forever indebted to him. *Спасибо вам большое за все!*

A special thank-you also goes to Dr. Tibor Koritsanzky, my former advisor. He is the one who opened my mind to the power of science with his excellent lectures of quantum chemistry and gave me the first enthusiasm and inspiration to start and pursue my career: computational science.

I also thank the great contributions of Dr. Zbigniew Kisiel and Dr. Piero Macchi, who are co-authors in our first and third studies. Having the opportunity to collaborate with them was not only an advantage, but such a delight.

Furthermore, I am especially grateful to Professor Harris J. Silverstone for many fruitful discussions regarding the Fourier transform method, and for his help with implementation of the numerical contour integration technique in *Mathematica*, which played a vital role in our first and second studies. Big thanks are also due to Justin Marsee of the Molecular Biosciences Ph.D. Program in Middle Tennessee State University (MTSU) , who generously agreed to test the performance of our new codes in our third study on his own newly released PC (2019 AMD Ryzen 7 3700X).

I sincerely appreciate and thank my advisory committee members: Dr. Tibor Koritsanszky, Dr. Yuri Melnikov, Dr. Joshua L. Phillips, Dr. William Robertson, and Dr. John Wallin, for their insight, feedback, and advices throughout this process.

I acknowledge the Computational Science Program of MTSU for granting me the Graduate Teaching Assistantship, which helped cover the majority of my tuition and living expenses during this degree. Together with this, I would like to express my gratitude to the Chemistry Department of MTSU and various professors who gave me teaching assistant work over the past few years. Indeed, teaching has been one of the greatest learning experiences of my life.

Finally, I must thank my family for all of the endless faith, support, and love over the years. Dear Dad, Mom, Uyên (my wife), and Vũ (my brother), you have shaped me into the person I am today. You are everything to me. I love you so much.

ABSTRACT

X-ray crystallography is considered to be one of the most accurate and reliable techniques for determination of the atomic and molecular structure of crystals. Whenever high-resolution low-temperature data are available, it is also possible to model the molecular aspherical electron density distribution, $\rho(\boldsymbol{r})$. In the field of experimental X-ray charge density determination, the electron density at each point in space $\rho(\mathbf{r})$ is modelled as a superposition of atomic-like densities $\rho_a(\mathbf{r})$, called pseudoatoms [Hirshfeld, F. L. (1971). *Acta Cryst.* B**27**, 769-781; Stewart, R. F. (1976). *Acta Cryst.* A**32**, 565-574; Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921]: $\rho(\mathbf{r}) = \sum_a \rho_a(\mathbf{r} - \mathbf{R}_a)$, where $\mathbf{R}_a$ denotes the location of the nucleus of pseudoatom a.

This thesis consists of three studies that describe newly-developed techniques for fast and accurate evaluation of the electrostatic interaction energies in molecular dimers and infinite crystal structures in which the charge distributions are modelled using the Hansen-Coppens pseudoatom formalism [Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921]. For example, using a 2015 2.8 GHz Intel Xeon E3-1505M v5 computer processor, our Fortran-based implementation evaluates the electrostatic interaction energy between two monomers of a benchmark 181-atom decapeptide molecule in under 3 seconds with a precision of at least 10^{-5} kJ/mol. Using the enhanced Ewald-summation technique which includes interactions up to the hexadecapolar level, the electrostatic intermolecular interaction energy in a crystal of the same 181-atom decapeptide molecule with a total of 724 atoms in the unit cell (Z=4, space group $P2_12_12_1$) is calculated with the same precision in under 50 seconds. In addition to being fast, the described methods correctly account for the electron density penetration effects arising from overlap of the neighbouring charge distributions (a common feature of molecular crystals) which contribute 30-60% to the total electrostatic energy in the examined molecular systems, and thus cannot be neglected.

While electron densities of the benchmark compounds used in our studies were generated using the University at Buffalo theoretical databank of transferable pseudoatoms [Dominiak, P. M., Volkov, A., Li, X., Messerschmidt, M. & Coppens, P. (2007). *J. Chem. Theory Comput*. **3**, 232-247], the described implementations are directly applicable to electron densities determined from experimental X-ray diffraction experiments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

aEP	Analytical Exact Potential
aEP/MM	Analytical Exact Potential and Multipole Moment
aMM	Atomic Multiple Moment
DS	Direct Summation
DS/aMM	Atomic Multiple Moment-based Direct Summation
DS/mMM	Molecular Multiple Moment-based Direct Summation
ED	Electron Density
E_{es}	Electrostatic interaction energy
EP	Exact Potential
EP/MM	Exact Potential and Multipole Moment
ERI	Electron– electron Repulsion Integrals
ES	Ewald Summation
FT	Fourier Transform
Lα	Löwdin α-function
MM	Multipole Moment
mMM	Molecular Multipole Moment
NAI	Electron–nuclear Attraction Integrals
nEP	Numerical Exact Potential
nEP/MM	Numerical Exact Potential and Multipole Moment
STF	Slater-type function

CHAPTER I: INTRODUCTION[1]

1.1 Preface

Accurate evaluation of intermolecular interaction energies in crystals plays an important role in diverse fields of molecular modeling, crystal-structure prediction, protein folding and bending, drug design, and material science (Kitaigorodsky, 1973; Dykstra, 1993; Jeziorski *et al.*, 1994; Gatti & Macchi, 2012). In the perturbation theory approach (Stone, 2013), the total intermolecular interaction energy E_{int} can be represented as a sum of the following physically meaningful contributions:

$$E_{\mathrm{int}} = E_{\mathrm{es}} + E_{\mathrm{ind}} + E_{\mathrm{disp}} + E_{\mathrm{ex-rep}} \tag{1.1}$$

where E_{es}, E_{ind}, E_{disp}, and $E_{\mathrm{ex-rep}}$ are the electrostatic, induction, dispersion, exchange-repulsion energies, respectively. E_{es} describes the electrostatic (Coulombic) interaction between two unperturbed charge distributions (molecules), E_{ind} originates from the interaction of the unperturbed charge distribution on one molecule with the induced charge distribution on the other (and *vice versa*), E_{disp} accounts for instantaneous interactions between fluctuating charge distributions on different molecules, and $E_{\mathrm{ex-rep}}$ arises from the antisymmetrization of molecular wavefunctions, according to the Pauli exclusion principle, when two molecules are overlapping charge distribution (Bickelhaupt & Baerends, 2000, Stone, 2013).

Among the energy terms introduced above, the electrostatic interaction energy E_{es} is considered to be the most important contribution to the total intermolecular interaction

[1] Portions of this introduction also appear in *Acta Crystallographica Section A: Foundations and Advances* (Nguyen *et al.*, 2018, 2019).

energy E_{int}, especially in polar molecules (Stone, 2013). It is necessary to emphasize that most biomolecules including water, proteins, and nucleic acids are polar. Thus, an accurate description of intermolecular electrostatic interactions should be performed when dealing with the structure, function, and dynamics of biological systems. Likewise, it is well-known that intermolecular electrostatic interaction energies E_{es} along with other electrostatic properties such as electron density, electrostatic potential, electric field, and electric field gradient carry significant physicochemical information in studies of experimental X-ray charge density and molecular modeling, in particular for prediction of crystal structures (Coppens, 1997; Tsirelson & Ozerov, 1996; Gavezzotti, 2002; Flierler & Stalke, 2012; Gatti & Macchi, 2012). From a fundamental viewpoint, an accurate, computationally efficient, and numerically stable method to assess intermolecular electrostatic interaction energies E_{es} is desired.

1.2 Evaluation of electrostatic interaction energies from the pseudoatom model of electron density

The electrostatic interaction energy between two continuous molecular charge distributions (molecules) A and B, $E_{\text{es}}(AB)$, is defined as a six-dimensional (6D) integral over all space (Spackman, 1986):

$$E_{\text{es}}(AB) = \int\int \frac{\rho_A^{\text{total}}(\mathbf{R})\rho_B^{\text{total}}(\mathbf{R}')}{|\mathbf{R} - \mathbf{R}'|} d\mathbf{R}' d\mathbf{R} \tag{1.2}$$

where $\mathbf{R}$ and $\mathbf{R}'$ represent positional vectors in the three-dimensional Cartesian *global* coordinate system, and $\rho_A^{\text{total}}(\mathbf{R})$ includes both the nuclear and electronic charge distributions:

$$\rho_A^{\text{total}}(\mathbf{R}) = \rho_A^{\text{nuc}}(\mathbf{R}) - \rho(\mathbf{R}) \tag{1.3}$$

and $\rho(\mathbf{R})$ is the usual, positive, electron density (Spackman, 1986).

In the field of experimental X-ray charge density determination, the molecular electron density (ED) at each point in space $\rho(\mathbf{R})$ is represented by a superposition of atomic-like densities $\rho_a(\mathbf{r})$, called pseudoatoms (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). In the simplest case, each pseudoatom J is defined with respect a *local* coordinate system $\mathbf{r}_J$ whose axes are parallel to those of the global coordinate system, and origin moved to nucleus J located at $\mathbf{R}_J$ in the global coordinate system (*i.e.*, $\mathbf{r}_J = \mathbf{R} - \mathbf{R}_J$):

$$\rho_A^{\text{total}}(\mathbf{R}) = \sum_{J=1} \rho_J^{\text{pseudoatom}}(\mathbf{r}_J) \tag{1.4}$$

In consequence, $E_{\text{es}}(AB)$ can be expanded in terms of pairwise electrostatic interactions between pseudoatoms *a* and *b*:

$$E_{\text{es}}(AB) = \sum_{a \in A} \sum_{b \in B} E_{\text{es}}(ab) \tag{1.5}$$

where the expression for each $E_{\text{es}}(ab)$ term in the summation is similar to (1.2) with *A* → *a*, *B* → *b*, and the charge density of each pseudoatom defined in the *local* coordinate system:

$$E_{\text{es}}(ab) = \int\int \frac{\rho_a(\mathbf{r}_a)\rho_b(\mathbf{r}_b)}{|\mathbf{r}_a - \mathbf{r}_b|} d\mathbf{r}_b d\mathbf{r}_a \tag{1.6}$$

Since experimental techniques nowadays can provide quantitative and qualitative information on charge densities of molecules constituting a crystal (Koritsanszky, 2001), attaining a reliable model for pseudoatoms in molecules is possible; hence, accurate evaluation of the electrostatic interaction energies between pseudoatoms becomes achievable.

By far the most widely used aspherical pseudoatom formalism is based on the Hansen-Coppens multipolar model of ED (Hansen & Coppens, 1978; Coppens, 1997) in which each pseudoatom is modelled using the modified Laplace series:

$$\rho(\mathbf{r}) = P_c\rho_c(\mathrm{r}) + P_v\kappa^3\rho_v(\kappa\mathrm{r}) + \sum_{l=0}^{l_{max}} {\kappa'}^3 R_l(\kappa'\mathrm{r}) \sum_{m=0}^{l} P_{lm\pm} d_{lm\pm}(\theta,\phi) \qquad (1.7)$$

where $\rho_c(r)$ and $\rho_v(r)$ are the spherically averaged core and valence densities, respectively, precalculated at the Hartree-Fock (Clementi, & Roetti, 1974; Bunge *et al.*, 1992, 1993), Density Functional (Volkov & Macchi, 2005), or Dirac-Slater (Su & Coppens, 1998; Macchi & Coppens, 2001) level of theory. The population of the core P_c usually remains fixed, while the population of the spherical valence shell P_v is usually adjusted together with the expansion-contraction parameter κ (Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). The double summation in the last term describes the aspherical components of the deformation density, $\Delta\rho(\mathbf{r})$, where coefficients $P_{lm\pm}$ are the population parameters, κ' are the dimensionless adjustment coefficients of the nodeless density-normalized Slater-type (Slater, 1932) radial functions R_l (Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). In equation (1.7), r represents the radial distance from the pseudoatom center, $r = |\mathbf{r} - \mathbf{R}_a|$, while the parameters (θ, φ) are the corresponding angular coordinates, *i.e.*, $\mathbf{r} - \mathbf{R}_a \equiv (r, \theta, \varphi)$. The angular functions $d_{lm\pm}(\theta, \varphi)$ are the real density-normalized spherical harmonics (Hansen & Coppens, 1978; Paturle & Coppens, 1988; Coppens, 1997; Michael & Volkov, 2015).

1.3 The Multipole Moment (MM) method

Although (1.6) is the exact expression to compute $E_{\mathrm{es}}(ab)$, it is not readily suitable for practical applications because of the complicated 6D integration involved (Gatti & Macchi, 2012). To bypass the challenge, one of the most common practical methods is to express $E_{\mathrm{es}}(ab)$ in terms of pseudoatom atomic moments (aMM) when expanding the $|\mathbf{r}_a - \mathbf{r}_b|^{-1}$ term in Taylor series (Buckingham, 1967, 1978; Buckingham *et al.*, 1988; Stone, 1996; Coppens, 1997; Kisiel, 2001; Kisiel, 2006):

$$
\begin{aligned}
E^{\mathrm{aMM}}(ab) = {} & T_{ab}\, q_a q_b + T_{ab}^{\alpha}\left(q_a \mu_{\alpha,b} - q_b \mu_{\alpha,a}\right) \\
& + T_{ab}^{\alpha\beta}\left(\frac{1}{3} q_a \theta_{\alpha\beta,b} + \frac{1}{3} q_b \theta_{\alpha\beta,a} - \mu_{\alpha,a}\mu_{\beta,b}\right) \\
& + T_{ab}^{\alpha\beta\gamma}\left(\frac{1}{15} q_a \omega_{\alpha\beta\gamma,b} - \frac{1}{15} q_b \omega_{\alpha\beta\gamma,a} - \frac{1}{3}\mu_{\alpha,a}\theta_{\beta\gamma,b} + \frac{1}{3}\mu_{\alpha,b}\theta_{\beta\gamma,a}\right) \qquad (1.8) \\
& + T_{ab}^{\alpha\beta\gamma\delta}\left(\frac{1}{9}\theta_{\alpha\beta,a}\theta_{\gamma\delta,b} + \cdots\right) + \cdots \\
& + T_{ab}^{\alpha\beta\gamma\delta,\varepsilon\zeta\eta\lambda}\left(\frac{1}{11025}\phi_{\alpha\beta\gamma\delta,a}\phi_{\varepsilon\zeta\eta\lambda,b} + \cdots\right)
\end{aligned}
$$

where the summation, according to the Einstein notation, is carried out over the repeated indices, and $T_{ab}^{\alpha\beta\gamma\ldots\omega}$are the interaction tensors (Buckingham, 1967, 1978; Buckingham *et al.*, 1988; Stone, 1996; Coppens, 1997) for the pseudoatoms a and b,

$$T_{ab}^{\alpha\beta\gamma\ldots\omega} = \nabla_\alpha \nabla_\beta \nabla_\gamma \cdots \nabla_\omega \frac{1}{\mathbf{R}_{ab}} \qquad (1.9)$$

with $\alpha, b, \gamma \ldots \omega$ representing the Cartesian coordinates $\{x, y, z\}$, and q_i, $\mu_{\alpha,i}$, $\theta_{\alpha\beta,i}$, $\omega_{\alpha\beta\gamma,i}$ and $\phi_{\alpha\beta\gamma\delta,i}$ are the atomic multipole moments of pseudoatom i, $i \in \{a, b\}$ (Coppens, 1997). By convention, q is called either a charge or a monopole (M), μ is a dipole (D), θ – quadrupole (Q), ω – octupole (O), and ϕ – hexadecapole (H). Note a slight deviation in our notation from that in Coppens (1997) as we reserve the upper-case letters for the molecular multipole moments (mMM).

Since the evaluation of atomic moments from Hansen-Coppens pseudoatom parameters (Coppens, 1997) and the implementation of interaction tensors $T_{ab}^{\alpha\beta\gamma\ldots\omega}$ at vector **r** (Abramov *et al.*, 2000; Kisiel, 2001, 2004) are straightforward, the resulting $E_{es}^{MM}(ab)$ can then be evaluated quite easily. Unfortunately, the MM method is valid only when applied to long-range interactions. When atoms (or molecules) are close to each other, *i.e.* their electron densities are overlapped, the multipole expansion of the electrostatic interactions becomes divergent (Gatti & Macchi, 2012). In consequence, the evaluated electrostatic energies systematically underestimate those derived by the exact integrations in (1.6). The difference, $E_{es} - E_{es}^{MM}$, is called penetration energy (Spackman, 2007; Gatti & Macchi, 2012). Several strategies have been proposed to account for penetration energy in electrostatic interaction energy evaluations. Some approaches use off-atom centered and damping functions (Stone, 1982, 2013; Freitag *et al.*, 2000); Piquemal et al., 2003), whereas others employ simplified charge density treatments (Kairys & Jensen, 1999; Volkov *et al.*, 2004; Qian & Krim, 2006; Spackman, 2006; Gatti & Macchi, 2012).

1.4 The Exact Potential and Multipole Moment (EP/MM) method

In their earlier study, Volkov, Koritsanszky, and Coppens (2004) proposed a method, called exact potential and multipole moment (EP/MM), for an accurate calculation of $E_{\mathrm{es}}(ab)$, and consequently $E_{\mathrm{es}}(AB)$. At high internuclear separations R_{ab} of pseudoatoms a and b, the integral in expression (1.6) can be evaluated essentially exactly via a MM expansion. When R_{ab} is below a user-defined criterion R_{cutoff} (for the second-row atoms, the optimal R_{cutoff} value was found to be 4 Å), the exact potential (EP) integral in (1.6) is first conveniently expanded into the nuclear-nuclear (E_{nn}), electron-nuclear (E_{en}), and electron-electron (E_{ee}) interaction terms:

$$\begin{aligned} E_{\mathrm{es}}(ab) &= E_{\mathrm{nn}} + E_{\mathrm{en}}(ab) + E_{\mathrm{en}}(ba) + E_{\mathrm{ee}} \\ &= \frac{Z_a Z_b}{R_{ab}} - \int \rho_a(\mathbf{r}_a) V_b^{\mathrm{nuc}}(\mathbf{r}_a) d\mathbf{r}_a - \int \rho_b(\mathbf{r}_b) V_a^{\mathrm{nuc}}(\mathbf{r}_b) d\mathbf{r}_b \\ &\quad + \int\int \frac{\rho_a(\mathbf{r}_a)\rho_b(\mathbf{r}_b)}{|\mathbf{r}_a - \mathbf{r}_b|} d\mathbf{r}_b d\mathbf{r}_a \end{aligned} \tag{1.10}$$

where Z_J and V_J^{nuc} are the nuclear charge and nuclear potential of pseudoatom *J*, and each $E_{\mathrm{en}}(JK)$ term represents the interaction of the electron charge of pseudoatom *J* with the nuclear charge of pseudoatom *K*.

The EP/MM method has been incorporated into the *XDPROP* program of the *XD* package (Volkov *et al.*, 2016). In the original EP/MM implementation, the $E_{\mathrm{en}}(ab)$, $E_{\mathrm{en}}(ba)$, and E_{ee} terms were evaluated numerically (this approach is called the *numerical* exact potential, nEP) on a nuclei-centered grid of points via a three-dimensional (3D) quadrature integration (Becke, 1988). If an adequate R_{cutoff} is provided, the MM component of the EP/MM method is relatively fast and accurate. However, the nEP method is severely limited in terms of speed and accuracy by the numerical 3D integration since those terms are directly related to the number of radial and angular grid points (Becke, 1988), something that was not fully investigated by Volkov, Koritsanszky, and Coppens (2004). Indeed, decreasing the number of grid points speeds up the calculation but may lead to a significant loss of accuracy, and *vice versa*.

1.5 Motivation and context of the thesis

One way to increase the speed of a numerical integration without loss of accuracy is to employ the so-called “grid pruning” technique (Murray *et al*., 1993; Gill *et al*., 1993) that has been extensively and successfully used in quantum-mechanical calculations. Alternatively, Spackman (2006) suggested an elegant way to correct for the penetration energy term, missing in the MM expansion (Buckingham, 1967, 1978; Buckingham *et al*., 1988), with the help of promolecular (Hirshfeld & Rzotkiewicz, 1974; Spackman & Maslen, 1986) charge densities. Although the implementations of these two methods are relatively straightforward, we instead seek an analytical solution.

We note that the electron density of each pseudoatom *J*, $\rho_J(\mathbf{r}'_J)$, can be written as a linear combination of the nucleus-centered density-normalized Slater-type functions (Slater, 1932) $\chi_J(\mathbf{r}'_J)$, of the general form (Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996):

$$\chi_J(\mathbf{r}'_J) = \chi_J(r'_J, \theta'_J, \phi'_J) = N(n_J, \zeta_J) r_J'^{n_J} \exp(-\zeta_J r'_J) d_{l_J m_J}(\theta'_J, \phi'_J) \qquad (1.11)$$

where vector $\mathbf{r}'_J \equiv (r'_J, \theta'_J, \phi'_J)$ is defined in the *local* coordinate system centered on nucleus *J* with coordinate axes parallel to those of the global coordinate system; n_J, l_J, and m_J are the integers similar to quantum numbers, ζ_J is the effective exponent, $N(n_J, \zeta_J)$ is the normalization coefficient, and $d_{l_J m_J}(\theta'_J, \phi'_J)$ is the real density-normalized spherical harmonic function (Stewart, 1976; Hansen & Coppens, 1978; Paturle & Coppens, 1988; Coppens, 1992; Michael & Volkov, 2015). With that consideration, the evaluation of all the electron-nuclear interaction integrals E_{en} and electron-electron repulsion integrals E_{ee} in equation (1.10) immediately reduce to the evaluation of two types of integrals well-known in quantum mechanics: two-center, one-electron nuclear attraction integrals (NAI) and two-center, two-electron Coulomb electron repulsion integrals (ERI) (Barnett & Coulson, 1951; Harris & Michels, 1967; McLean & Yoshimine, 1968; Weatherford & Jones, 1982):

$$\mathrm{NAI} = Z_b \int \frac{\chi_a(\mathbf{r}'_a)}{|\mathbf{r}'_a - \mathbf{R}'_b|} d\mathbf{r}'_a = Z_b V_{\chi_a}^{\mathrm{elec}}(\mathbf{R}'_b) \tag{1.12}$$

$$\mathrm{ERI} = \int \int \frac{\chi_a(\mathbf{r}'_a)\chi_b(\mathbf{r}'_b)}{|\mathbf{r}'_a - \mathbf{r}'_b|} d\mathbf{r}'_b d\mathbf{r}'_a \tag{1.13}$$

where vector $\mathbf{R}'_b$ is the location of nucleus *b* that carries positive charge Z_b and $V_{\chi_a}^{\mathrm{elec}}(\mathbf{R}'_b)$ is the electronic potential at $\mathbf{R}'_b$ due to the density function χ_a of pseudoatom *a*. Each density function is defined in its *local* coordinate system with axes parallel to those of the global system:

$$\chi_a(\mathbf{r}'_a) = r_a'^{n_a} \exp\left(-\zeta_a r'_a\right) d_{l_a m_a}(\theta'_a, \phi'_a) \tag{1.14}$$

$$\chi_b(\mathbf{r}'_b) = r_b'^{n_b} \exp\left(-\zeta_b r'_b\right) d_{l_b m_b}(\theta'_b, \phi'_b) \tag{1.15}$$

We also note that a more-or-less comprehensive summary of various techniques for evaluation of $V_{\chi_a}^{\mathrm{elec}}(\mathbf{R}'_b)$ within the pseudoatom formalism is given by Volkov *et al*. (2006) and Spackman (2007). In this thesis, we have elected to use the electronic potential equations expanded in terms of incomplete gamma function as given in Volkov *et al*. (2006). Since the electronic potential integrals have already been programmed in *XDPROP* (Volkov *et al*., 2016), evaluation of all NAI becomes trivial.

Evaluation of two-electron Coulomb electron repulsion integrals ERI in equation (1.13) is more complicated. And there is a significant body of literature devoted to this topic. For example, Barnett & Coulson (1951), Ruedenberg *et al.* (1956), Wahl *et al.* (1964), Geller (1962, 1963*a,b*, 1964*a,b*), O-Ohata & Ruedenberg (1966), Silverstone

(1966, 1967*a,b,c*, 2014), Huzinaga (1967), Harris & Michels (1967), McLean & Yoshimine (1968), Todd *et al.* (1970), Guseinov (1970), Zimont & Mar'yaskin (1972), Filter & Steinborn (1978), Weatherford & Jones (1982), Weniger & Steinborn (1983), Trivedi & Steinborn (1983), Grotendorst & Steinborn (1985), Weniger *et al*. (1986), Shestakov (1992), Fernandez Rico *et al.* (2000), Öztekin *et al*. (2001), Guseinov & Mamedov (2002), Berlu (2004), Öztekin (2004), Öztekin & Özcan (2007), and Lesiuk & Moszynski (2014). The list is not exhaustive – it highlights the most relevant (from our point of view) to our case studies. From this plethora of methods, we have chosen to use the so-called Löwdin α-function (Lα) (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993) and the Fourier transform (FT) technique (Silverstone, 1966, 1967*a,b,c*, 2014; Geller, 1962, 1963*a,b*, 1964*a,b*; Geller & Griffith, 1964; Harris & Michels, 1967). These two methods are (*i*) ideally suited for evaluation of integrals (1.13), and (*ii*) relatively simple to program in Fortran so that they can be later incorporated into the *XDPROP* program of the *XD* package (Volkov *et al*., 2016).

1.6 Objectives of the thesis

The main objectives of the thesis are:

- to develop fully analytical, fast and precise methods (aEP) that can replace the numerical (quadrature) evaluation of the electron–nuclear attraction integrals (NAI) and electron– electron repulsion integrals (ERI) in the exact potential (EP) part in the original EP/MM implementation (Volkov, Koritsanszky & Coppens, 2004) for the calculation of intermolecular electrostatic interaction energies.

- to implement the aEP and their hybrid counterpart (aEP/MM) methods using Fortran90 and incorporate them into the *XDPROP* module of the *XD* software package (Volkov *et al.*, 2016) in order to test them on necessary benchmark systems, compare with the reference data, and present exhaustive analysis on their performance.
- to extend the usage of the new analytical EP/MM methods to the evaluation of the intermolecular electrostatic energies in molecular crystals.

1.7 Three-Article Dissertation Structure

This thesis is presented using an article-style format, consisting of our three articles included in chapters II – IV. The first two articles are original research papers published in *Acta Crystallographica Section A: Foundations and Advances* (Nguyen, Kisiel & Volkov, 2018; Nguyen & Volkov, 2019), and the last one is currently in the publication preparation process. The three studies together demonstrate how we achieve our goals outlined in Section 1.5.

The first article (Chapter II) introduces an accurate, stable, and fast analytical technique that improves the existing EP/MM method (Volkov, Koritsanszky *et al.*, 2004) in terms of speed and accuracy by replacing the numerical (quadrature) evaluation of the electron-nuclear attraction integrals (NAI) and electron-electron repulsion integrals (ERI) of the exact potential (EP) for the calculation of electrostatic interaction energies from pseudoatom-based molecular electron densities. The NAI are simply evaluated using the electronic potential, $V^{\text{elec}}(\mathbf{r})$, formulas derived in (Volkov *et al.*, 2006) while the ERI are evaluated with the help of the Löwdin α-function (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978) and C-matrix of Jones and Weatherford (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993).

The second article (Chapter III) consists of three parts. The first part describes a simple implementation of the Fourier transform method (Silverstone, 1966-2014; Geller, 1962-1964; Geller & Griffith, 1964; Harris & Michels, 1967) as another accurate and reliable analytical approach for evaluation of the electron-electron repulsion integrals (ERI). In the second part, we explore the computational aspects of both the Löwdin α-function and Fourier transform techniques with a goal to increase the resulting precision without a major loss of speed. Finally, we compare the Löwdin α-function and Fourier transform implementations and report our decision as to which method is better suited for evaluation of the ERI within the pseudoatom electron-density formalism.

The third article (Chapter IV) presents an extension of the analytical exact potential and multipole moment (EP/MM) method to the evaluation of the intermolecular electrostatic energies in an infinite crystal. In this study, we develop and perform a comprehensive numerical analysis of the Ewald (ES) and direct (DS) summation methods which use the exact potential method to correct for the short-range penetration effects originating from the overlap of neighbouring molecular charge densities.

The final chapter (Chapter V) summaries the results of our studies, includes general concluding remarks and an outlook for our future studies.

CHAPTER II: ARTICLE 1 - FAST ANALYTICAL EVALUATION OF INTERMOLECULAR ELECTROSTATIC INTERACTION ENERGIES USING THE PSEUDOATOM REPRESENTATION OF THE ELECTRON DENSITY. I. THE LÖWDIN *α*-FUNCTION METHOD[2]

Abstract

Previously reported (Volkov *et al.* (2004). *Chem. Phys. Lett.* **391**, 170–175) exact potential and multipole moment (EP/MM) method for evaluation of intermolecular electrostatic interaction energies using the nuclei-centered pseudoatom representation of electron densities is significantly improved in terms of both speed and accuracy by replacing the numerical quadrature integration of the exact potential with a fully analytical technique. The resulting approach, incorporated in the *XDPROP* module of the software package *XD*, has been tested on several molecular systems ranging in size from water–water to dodecapeptide–dodecapeptide dimers using electron densities constructed *via* the University at Buffalo Aspherical Atom Databank. The improved hybrid method provides electrostatic interaction energies within the uncertainty of ≤ 0.2 kJ/mol for all benchmark systems. The running time for a dimer of a sizable, 225-atom dodecapeptide is under 4 seconds on a 2012 central processing unit (2.8 GHz AMD Opteron 6348), and under 3 seconds on a relatively modern processor (2.8 GHz Intel Xeon E3-1505M v5).

[2] This chapter also appears in *Acta Crystallographica Section A: Foundations and Advances* [Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536].

2.1 Introduction

Essentially all experimental X-ray charge density studies rely on the accurate evaluation of electrostatic properties that carry some physicochemical significance (Coppens, 1997; Tsirelson & Ozerov, 1996; Flierler & Stalke, 2012; Gatti & Macchi, 2012). For example, the topological features of the electron density and the analysis of the electrostatic potential have been found to reveal the nature and strength of interatomic and intermolecular interactions (Bader, 1990; Coppens, 1997; Tsirelson & Ozerov, 1996; Flierler & Stalke, 2012; Gatti & Macchi, 2012). Another important property in molecular crystals that is intimately related to those mentioned above is the electrostatic interaction energy, E_{es}, which is often considered to be the most important contribution to the total interaction energy (Stone, 2013).

The electrostatic interaction energy between two continuous molecular charge distributions (molecules) A and B, $E_{\text{es}}(AB)$, is defined as a six-dimensional integral over all space:

$$E_{\text{es}}(AB) = \int\int \frac{\rho_A^{\text{total}}(\mathbf{R})\rho_B^{\text{total}}(\mathbf{R}')}{|\mathbf{R}-\mathbf{R}'|} d\mathbf{R}' d\mathbf{R} \tag{2.1}$$

where $\mathbf{R}$ and $\mathbf{R}'$ represent positional vectors in the three-dimensional Cartesian *global* coordinate system, and $\rho_A^{\text{total}}(\mathbf{R})$ includes both the nuclear and electronic charge distributions:

$$\rho_A^{\text{total}}(\mathbf{R}) = \rho_A^{\text{nuc}}(\mathbf{R}) - \rho(\mathbf{R}) \tag{2.2}$$

and $\rho(\mathbf{R})$ is the usual, positive, electron density.

The pseudoatom model (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996) represents molecular charge distributions as a superposition of nuclei-centered pseudoatoms. In the simplest case, each pseudoatom *J* is defined with respect to a *local* coordinate system $\mathbf{r'}_J$ (the need for the 'prime' notation will be evident shortly) whose axes are parallel to those of the global coordinate system, and origin moved to nucleus *J* located at $\mathbf{R}_J$ in the global coordinate system. As such, $E_{\mathrm{es}}(AB)$ can be expanded in terms of pairwise electrostatic interactions between pseudoatoms *a* and *b*:

$$E_{\mathrm{es}}(AB) = \sum_{a \in A} \sum_{b \in B} E_{\mathrm{es}}(ab) \tag{2.3}$$

where the expression for each $E_{\mathrm{es}}(ab)$ term in the summation is similar to (2.1) with $A \rightarrow a$, $B \rightarrow b$, and the charge density of each pseudoatom defined in the *local* coordinate system:

$$E_{\mathrm{es}}(ab) = \int \int \frac{\rho_a^{\mathrm{total}}(\mathbf{r}'_a)\rho_b^{\mathrm{total}}(\mathbf{r}'_b)}{|\mathbf{r}'_a - \mathbf{r}'_b|} d\mathbf{r}'_b d\mathbf{r}'_a \tag{2.4}$$

In our earlier study (Volkov, Koritsanszky *et al.*, 2004), we reported a simple method, called EP/MM, for an accurate calculation of $E_{\mathrm{es}}(ab)$, and consequently $E_{\mathrm{es}}(AB)$. At high internuclear separations of pseudoatoms *a* and *b*, *i.e.* when $R_{ab} = |\mathbf{R}_a - \mathbf{R}_b|$ is large, integral (2.4) can be evaluated essentially exactly *via* a multipole moment (MM) expansion (Buckingham, 1967, 1978; Buckingham *et al.*, 1988). When R_{ab} is below a user-defined criterion R_{cutoff} (for the second-row atoms, the optimal R_{cutoff} value was found to be 4 Å), the exact potential (EP) integral (2.4) is first conveniently expanded into the nuclear-nuclear (E_{nn}), electron-nuclear (E_{en}), and electron-electron (E_{ee}) interaction terms:

$$\begin{aligned} E_{\mathrm{es}}(ab) &= E_{\mathrm{nn}} + E_{\mathrm{en}}(ab) + E_{\mathrm{en}}(ba) + E_{\mathrm{ee}} \\ &= \frac{Z_a Z_b}{R_{ab}} - \int \rho_a(\mathbf{r}_a')V_b^{\mathrm{nuc}}(\mathbf{r}_a')d\mathbf{r}_a' - \int \rho_b(\mathbf{r}_b')V_a^{\mathrm{nuc}}(\mathbf{r}_b')d\mathbf{r}_b' \\ &+ \int\int \frac{\rho_a(\mathbf{r}_a')\rho_b(\mathbf{r}_b')}{|\mathbf{r}_a' - \mathbf{r}_b'|} d\mathbf{r}_b' d\mathbf{r}_a' \end{aligned} \tag{2.5}$$

where Z_J and V_J^{nuc} are the nuclear charge and nuclear potential of pseudoatom *J*, respectively, and each $E_{\mathrm{en}}(JK)$ term represents the interaction of the electron charge of pseudoatom *J* with the nuclear charge of pseudoatom *K*. Then, the $E_{\mathrm{en}}(ab)$, $E_{\mathrm{en}}(ba)$, and E_{ee} terms are evaluated numerically on a nuclei-centered grid of points *via* a three-dimensional quadrature integration (Becke, 1988). For the E_{ee} term, this is achieved by rewriting it as a function of the electronic potential, V^{elec}:

$$E_{\mathrm{ee}} = \int\int \frac{\rho_a(\mathbf{r}_a')\rho_b(\mathbf{r}_b')}{|\mathbf{r}_a' - \mathbf{r}_b'|} d\mathbf{r}_b' d\mathbf{r}_a' = \int \rho_a(\mathbf{r}_a')V_b^{\mathrm{elec}}(\mathbf{r}_a')d\mathbf{r}_a' = \int \rho_b(\mathbf{r}_b')V_a^{\mathrm{elec}}(\mathbf{r}_b')d\mathbf{r}_b' \tag{2.6}$$

where

$$V_b^{\mathrm{elec}}(\mathbf{r}_a') = \int \frac{\rho_b(\mathbf{r}_b')}{|\mathbf{r}_a' - \mathbf{r}_b'|} d\mathbf{r}_b' \quad \text{and} \quad V_a^{\mathrm{elec}}(\mathbf{r}_b') = \int \frac{\rho_a(\mathbf{r}_a')}{|\mathbf{r}_a' - \mathbf{r}_b'|} d\mathbf{r}_a' \tag{2.7}$$

In general, our technique is similar to the Fast Multipole Method (Greengard & Rokhlin, 1987) and the QM/MM approach (Warshel & Levitt, 1976) in the sense that an accurate approximation is introduced at higher internuclear separations leading to a significant computational speedup without loss of accuracy. The EP/MM method, as outlined above, has been incorporated in the *XDPROP* program of the *XD* package (Volkov *et al.*, 2016).

The original implementation of the multipole moment (MM) component of the EP/MM method was based on generic yet relatively slow evaluation of interaction tensors *T* at vector **R** (Buckingham, 1967, 1978; Buckingham *et al.*, 1988), $T_{\alpha\beta\dots\nu} =$

$(4\pi\epsilon_0)^{-1}\nabla_\alpha\nabla_\beta \dots \nabla_\nu\mathbf{R}^{-1}$, as implemented in the *XDINTER* program (Abramov *et al.*, 2000). To speed up the MM calculations, we replaced the *XDINTER*-based code for evaluation of the *T* tensors with highly optimized subroutines from the program *MIN16* (Kisiel, 2001, 2004); *MIN16* was originally developed for evaluation of intermolecular electrostatic interaction energies within the framework of the distributed multipole analysis (DMA) model (Buckingham & Fowler, 1985; Buckingham *et al.*, 1987) and was used, for example, to rationalize geometries of clusters of ethylene derivatives with HCl (Kisiel, Fowler & Legon, 1990; Kisiel, Fowler, Legon, Devanne *et al.*, 1990).

Provided an adequate R_{cutoff} is used, the MM component of the EP/MM method is now extremely fast and accurate. However, the original implementation (Volkov, Koritsanszky *et al.*, 2004) of the EP integrals (2.5) is limited in terms of speed and accuracy by the numerical three-dimensional integration.

The three-dimensional quadrature integration is performed on a set of radial and angular grid points (Becke, 1988). The orientation of the angular grid is not expected to be an issue as the Lebedev quadrature (Lebedev & Laikov, 1999) employed for this purpose possesses octahedral rotation and inversion symmetry. However, the speed and accuracy of the numerical integration are directly related to the number of radial and angular grid points, something that was not fully investigated by Volkov, Koritsanszky *et al.* (2004). Indeed, decreasing the number of grid points speeds up the calculation but may lead to a significant loss of accuracy, and *vice versa*. We also note that in order to obtain a single $E_{\text{es}}(ab)$ value, numerical integrations over both pseudoatoms a and b are required, which increases the numerical EP (nEP) calculation time by a factor of two:

(i) integration over pseudoatom *a*:

$$E_{\mathrm{en}}(ab) = \int \rho_a(\mathbf{r}'_a) V_b^{\mathrm{nuc}}(\mathbf{r}'_a) d\mathbf{r}'_a \quad \text{and} \quad E_{\mathrm{ee}}(ab) = \int \rho_a(\mathbf{r}'_a) V_b^{\mathrm{elec}}(\mathbf{r}'_a) d\mathbf{r}'_a \tag{2.8}$$

(ii) integration over pseudoatom *b*:

$$E_{\mathrm{en}}(ba) = \int \rho_b(\mathbf{r}'_b) V_a^{\mathrm{nuc}}(\mathbf{r}'_b) d\mathbf{r}'_b \quad \text{and} \quad E_{\mathrm{ee}}(ba) = \int \rho_b(\mathbf{r}'_b) V_a^{\mathrm{elec}}(\mathbf{r}'_b) d\mathbf{r}'_b \tag{2.9}$$

The integration over pseudoatom b cannot be eliminated because the required $E_{\mathrm{en}}(ba)$ term is difficult to describe via numerical integration over a quadrature grid centered on nucleus a (and if one could devise such a grid, the computational cost would likely be prohibitive). The integrals $E_{\mathrm{ee}}(ab)$ and $E_{\mathrm{ee}}(ba)$ represent the electron-electron repulsion energy and should agree within a desired limit. In the original implementation (Volkov, Koritsanszky *et al.*, 2004), the comparison of $E_{\mathrm{ee}}(ab)$ and $E_{\mathrm{ee}}(ba)$ integrals served as an accuracy check for the numerical integration, as were the integrals

$$\int \rho_a(\mathbf{r}'_a) d\mathbf{r}'_a \quad \text{and} \quad \int \rho_b(\mathbf{r}'_b) d\mathbf{r}'_{\boldsymbol{b}}, \tag{2.10}$$

that are expected to return the exact electron count for each pseudoatom, known from the sum of core and valence density population parameters (Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996).

One way to increase the speed of a numerical integration without the loss of accuracy is to employ the so-called “grid pruning” technique (Murray *et al.*, 1993; Gill *et al.*, 1993) that has been extensively and successfully used in quantum-mechanical calculations. Though the implementation of “grid pruning” is relatively straightforward, this path was rejected in a search for an analytical solution.

Alternatively, Spackman (2006) suggested an elegant way to correct for the penetration energy term, missing in the MM expansion (Buckingham, 1967, 1978;

Buckingham *et al.*, 1988), with the help of promolecular (Hirshfeld & Rzotkiewicz, 1974; Spackman & Maslen, 1986) charge densities. The resulting approximate technique was found to be fast and produced E_{es} values within 0.2-8.4 kJ/mol from those obtained *via* the nEP method for 11 dimers of α-glycine, L-(+)-lactic acid and *N*-acetylglycine (Spackman, 2006). While it would have been relatively easy to incorporate Spackman's approach in *XDPROP*, it was not used for the same reason as grid pruning.

In the following, we present a fully analytical approach for evaluation of all integrals in the EP part [equation (2.5)]. This technique, dubbed analytical EP (aEP), when used in combination with the MM (aEP/MM) method instead of the nEP integration has been found to be very accurate and numerically stable, while at the same time being fast enough for calculation of E_{es} for large (tested up to the dodecapeptide size) molecules.

2.2 Analytical Exact Potential (aEP) method

We start by noting that the electron density of each pseudoatom *J*, $\rho_J(\mathbf{r}'_J)$, can be written as a linear combination of the nucleus-centered density-normalized Slater-type functions (Slater, 1932) $\chi_J(\mathbf{r}'_J)$, of the general form (Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996):

$$\chi_J(\mathbf{r}'_J) = \chi_J(r'_J, \theta'_J, \phi'_J) = N(n_J, \zeta_J) r'^{n_J}_J \exp\left(-\zeta_J r'_J\right) d_{l_J m_J}(\theta'_J, \phi'_J) \qquad (2.11)$$

where vector $\mathbf{r}'_J \equiv (r'_J, \theta'_J, \phi'_J)$ is defined in the *local* coordinate system centered on nucleus *J* with coordinate axes parallel to those of the global coordinate system; n_J, l_J, and m_J are the integers similar to quantum numbers, ζ_J is the effective exponent, $N(n_J, \zeta_J)$ is the normalization coefficient, and $d_{l_J m_J}(\theta'_J, \phi'_J)$ is the real density-normalized spherical harmonic function (Stewart, 1976; Hansen & Coppens, 1978; Paturle & Coppens, 1988; Coppens, 1992; Michael & Volkov, 2015). Because $N(n_J, \zeta_J)$ is a constant, we shall exclude it from all the derivations below since it is trivial to account for it in any expression, as is the case for the linear expansion coefficients (multipole populations $P_{l_J m_J}$).

2.2.1 Evaluation of the electron-nuclear interaction energy, $\boldsymbol{E}_{\mathbf{en}}$

Considering the electron density as a linear combination of functions (2.11), all electron-nuclear interaction integrals E_{en} in equations (2.5), (2.8), and (2.9) immediately reduce to a well-known in quantum mechanics two-center, one-electron nuclear attraction integral (NAI) (Barnett & Coulson, 1951; Harris & Michels, 1967; McLean & Yoshimine, 1968; Weatherford & Jones, 1982):

$$\text{NAI} = Z_b \int \frac{\chi_a(\mathbf{r}'_a)}{|\mathbf{r}'_a - \mathbf{R}'_b|} d\mathbf{r}'_a = Z_b V^{\text{elec}}_{\chi_a}(\mathbf{R}'_b) \qquad (2.12)$$

where vector $\mathbf{R}'_b$ is the location of nucleus *b* that carries positive charge Z_b and $V^{\text{elec}}_{\chi_a}(\mathbf{R}'_b)$ is the electronic potential at $\mathbf{R}'_b$ due to the density function χ_a of pseudoatom *a*. A more-or-less comprehensive summary of various techniques for evaluation of $V^{\text{elec}}_{\chi_a}(\mathbf{R}'_b)$ within the pseudoatom formalism is given by Volkov *et al.* (2006) and Spackman (2007). In this study, we have elected to use the electronic potential equations expanded in terms of incomplete gamma function as given in Volkov *et al.* (2006). Since the electronic potential integrals have already been programmed in *XDPROP* (Volkov *et al.*, 2016), evaluation of all NAI integrals becomes trivial.

2.2.2 Evaluation of the electron-electron interaction energy, $E_{\mathbf{ee}}$

Evaluation of the electron-electron repulsion integrals E_{ee} in equation (2.6) is more complicated. When each of the electron densities of pseudoatoms a and b is represented *via* a linear combination of density functions (2.11), the problem is reduced to evaluation of a number of well-known in quantum mechanics two-center, two-electron Coulomb electron repulsion integrals (ERI) (Barnett & Coulson, 1951; Harris & Michels, 1967; McLean & Yoshimine, 1968; Weatherford & Jones, 1982) defined in terms of Slater-type functions (Slater, 1932):

$$\text{ERI} = \int\int \frac{\chi_a(\mathbf{r}_a')\chi_b(\mathbf{r}_b')}{|\mathbf{r}_a' - \mathbf{r}_b'|} d\mathbf{r}_b' d\mathbf{r}_a' \tag{2.13}$$

where, as before, each function is defined in its *local* coordinate system with axes parallel to those of the global system:

$$\chi_a(\mathbf{r}_a') = r_a'^{n_a} \exp\left(-\zeta_a r_a'\right) d_{l_a m_a}(\theta_a', \phi_a') \tag{2.14}$$

$$\chi_b(\mathbf{r}_b') = r_b'^{n_b} \exp\left(-\zeta_b r_b'\right) d_{l_b m_b}(\theta_b', \phi_b') \tag{2.15}$$

There is a significant body of literature devoted to this topic. For example, Barnett & Coulson (1951), Ruedenberg *et al.* (1956), Wahl *et al.* (1964), Geller (1964), O-Ohata & Ruedenberg (1966), Silverstone (1966), Harris & Michels (1967), McLean & Yoshimine (1968), Todd *et al.* (1970), Guseinov (1970), Zimont & Mar'yaskin (1972), Filter & Steinborn (1978), Weatherford & Jones (1982), Shestakov (1992), Fernandez Rico *et al.* (2000) Lesiuk & Moszynski (2014). The list is not exhaustive – it highlights the most relevant to our case (from our point of view) studies. From this plethora of methods, we have chosen to use the so-called Löwdin α-function expansion technique (Löwdin, 1956;

Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978). This method is (*i*) ideally suited for evaluation of integrals (2.13), and (*ii*) relatively simple to code.

The integration in (2.13) is not straightforward because $\chi_a(\mathbf{r}'_a)$ and $\chi_b(\mathbf{r}'_b)$ are defined with respect to different centers. As such, before carrying out the integration in (2.13), the functions are manipulated as follows:

(i) The functions $\chi_a(\mathbf{r}'_a)$ and $\chi_b(\mathbf{r}'_b)$ are rotated (Cromer *et al.*, 1976; Su & Coppens, 1994) to a new coordinate system with the origin defined at nucleus a [thus assigning coordinates (0, 0, 0) to nucleus a], and the new *Z*-axis directed along the $a \to b$ vector so that nucleus b is located at (0, 0, *z*). The resulting functions are $\chi_a(\mathbf{r}_a)$ and $\chi_b(\mathbf{r}_b)$. However, because nucleus a is now located at the origin, $\mathbf{r}_a \to \mathbf{r}$, where $\mathbf{r} \equiv (r, \theta, \phi)$ is defined in this new coordinate system. Thus, $\chi_a(\mathbf{r}_a) \to \chi_a(\mathbf{r})$.

(ii) The function $\chi_b(\mathbf{r}_b)$ centered at (0, 0, *z*), is expanded about the new origin (located at nucleus a) in an infinite series of spherical harmonics with functional coefficients, Löwdin α-functions (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978), determined *via* the Jones and Weatherford *C*-matrix method (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993). The resulting function $\chi_b(\mathbf{r})$ is now defined in *the same* coordinate system as function $\chi_a(\mathbf{r})$.

As a result of these transformations, we have for $\chi_b(\mathbf{r})$ (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993):

$$\chi_b(\mathbf{r}) \;=\; \chi_b(r,\theta,\phi) = \frac{1}{{\zeta_b}^{n_b}}\left[\frac{(2l_b+1)(l_b+m_b)!}{4\pi(l_b-m_b)!}\right]^{1/2}$$

$$\times \sum_{l'=m_b}^{\infty}\left(\frac{4\pi(l'+m_b)!}{(2l'+1)(l'-m_b)!}\right)^{1/2} \times \alpha_{l'}^{n'_b l_b m_b}(\zeta_b z, \zeta_b r) \times d_{l' m_b}(\theta,\phi) \tag{2.16}$$

where $n'_b = n_b + 1$. The general form of the Löwdin α-function is given by (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993)

$$\alpha_{l'}^{nlm}(\zeta z, \zeta r) = \frac{(2l'+1)(l'-m)!}{2(l'+m)!}$$

$$\times \sum_{i=0}^{n+l+l'}\sum_{j=0}^{n+l'} C_{l'}^{nlm}(i,j) \times H_{ij}(\zeta z, \zeta r) \times (\zeta z)^{i-l-l'-1} \times (\zeta r)^{j-l'-1} \tag{2.17}$$

where (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993)

$$H_{ij}(\zeta z, \zeta r) = \begin{cases} \exp(-\zeta z)\left[(-1)^j \exp(\zeta r) - \exp(-\zeta r)\right], & r < z \\ \exp(-\zeta r)\left[(-1)^i \exp(\zeta z) - \exp(-\zeta z)\right], & r > z \end{cases} \tag{2.18}$$

and $C_{l'}^{nlm}(i,j)$ is the (i, j)-th element of the C-matrix generated by the following expression (Jones, 1984, 1992, 1993):

$$\sum_{i=0}\sum_{j=0} C_{l'}^{nlm}(i,j)\, z^i r^j = \sum_{p=0}^{\frac{l+m}{2}}\; \sum_{q=0}^{l+m-2p}\; \sum_{v=0}^{l+m-2p-q}\; \sum_{p'=0}^{\frac{l'-m}{2}}\; \sum_{q'=0}^{l'-m-2p'}\; \sum_{v'=0}^{l'-m-2p'-q'}\; \sum_{k=0}^{t}\sum_{k'=0}^{t-k}$$

$$\times \frac{z^x r^y (-1)^{v+q'+p+p'+l}}{4^{l+l'+p-p'}(l-p)!\,p!\,p'!\,q!\,q'!\,v!\,v'!} \tag{2.19}$$

$$\times \frac{(2l-2p)!\;(2l'-2p')!\;t!}{(l+m-2p-q-v)!\,(l'-p')!\,(l'-m-2p'-q'-v')!\;k'!\,(t-k-k')!}$$

where (Jones, 1984, 1992, 1993)

$$x = n + l + 2l' - 2p' - 2v' - 2v - k - k'$$

$$y = 2p' + 2v + 2v' + k'$$

$$t = n - l + 2p + 2q + 2q'$$

In a simplified form, $\chi_b(\mathbf{r}_b)$ can now be written as:

$$\chi_b(r,\theta,\phi) = k1_b \sum_{l'=m_b}^{\infty} k2_b(l') \times\ k3_b(l') \times\ k4_b(l') \times d_{l'm_b}(\theta,\phi)$$
$$\times \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} C_{l'}^{n'_b l_b m_b}(i,j) \times H_{ij}(\zeta_b z, \zeta_b r) \times\ \zeta_b^{i+j} z^i\, r^{j-l'-1} \tag{2.20}$$

where

$$k1_b\ = \frac{1}{{\zeta_b}^{n_b}}\left[\frac{(2l_b+1)(l_b+m_b)!}{(l_b-m_b)!}\right]^{1/2} \tag{2.21}$$

$$k2_b(l') = \left[\frac{(l'+m_b)!}{(2l'+1)(l'-m_b)!}\right]^{1/2} \tag{2.22}$$

$$k3_b(l') = \frac{(2l'+1)(l'-m_b)!}{2(l'+m_b)!} \tag{2.23}$$

$$k4_b(l') = z^{-l_b-l'-1}\zeta_b^{-l_b-2l'-2} \tag{2.24}$$

and

$$i_{max} = n'_b + l_b - |m_b| + l'$$

$$j_{max} = {n'}_b + l'$$

With the two functions $\chi_a(\mathbf{r})$ and $\chi_b(\mathbf{r})$ defined in the same coordinate system, it is obviously easier to obtain the electronic potential of $\chi_a(\mathbf{r})$,

$$V_a^{\text{elec}}(\mathbf{r}) = \int \frac{\chi_a(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' \tag{2.25}$$

so, the expression for ERI becomes:

$$\text{ERI} = \int \chi_b(\mathbf{r}) V_a^{\text{elec}}(\mathbf{r}) d\mathbf{r} \tag{2.26}$$

While for $V_a^{\text{elec}}(\mathbf{r})$ we could have used equations from Volkov *et al.* (2006), we have decided to keep using Jones' approach (Jones, 1981, 1991, 1993):

$$V_a^{\text{elec}}(\mathbf{r}) = V_a(\mathbf{r}) = k_a d_{l_a m_a}(\theta, \phi)[V1_a(r) + V2_a(r) + V3_a(r)] \tag{2.27}$$

where

$$k_a = \frac{4\pi}{(2l_a + 1)} \tag{2.28}$$

$$V1_a(r) = \frac{n!}{{\zeta_a}^{n+1}}\, r^{-l_a - 1} \tag{2.29}$$

$$V2_a(r) = \ -\exp\left(-\zeta_a r\right) \sum_{k=0}^{n} \frac{n!}{(n-k)!\ {\zeta_a}^{k+1}}\, r^{n-k-l_a-1} \tag{2.30}$$

$$V3_a(r) = \ \exp\left(-\zeta_a\, r\right) \sum_{k=0}^{m} \frac{m!}{(m-k)!\ {\zeta_a}^{k+1}}\, r^{m-k+l_a} \tag{2.31}$$

where $n = n_a + l_a + 2$ and $m = n_a - l_a + 1$. Combining (2.20) and (2.27), ERI now becomes:

$$\begin{aligned} \text{ERI} &= \int \chi_b(\mathbf{r}) V_a^{\text{elec}}(\mathbf{r}) d\mathbf{r} \\ &= k_a\, k1_b \sum_{l'=m_b}^{\infty} k2_b(l') \times\, k3_b(l') \times\, k4_b(l') \\ &\times \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} \Big\{ C_{l'}^{n'_b l_b m_b}(i,j) \times\, \zeta_b^{i+j} z^i \\ &\qquad \times \int r^{j-l'-1} \times H_{ij}(\zeta_b z, \zeta_b r) \times [V1_a(r) + V2_a(r) + V3_a(r)] \\ &\qquad \times\, d_{l' m_b}(\theta, \phi)\, d_{l_a m_a}(\theta, \phi) d\mathbf{r} \Big\} \end{aligned} \tag{2.32}$$

The orthonormality of spherical harmonics immediately eliminates all terms in the outer sum in which $l' \neq l_a$ or $m_b \neq m_a$. Hence, ERI can be reduced to a summation of finite *non-zero* terms, where $l' = l_a$ and $m_b = m_a$, as:

$$\begin{aligned}
\mathrm{ERI} &= k_a\, k1_b \times k2_b(l_a) \times\ k3_b(l_a) \times\ k4_b(l_a) \\
&\quad \times \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} \left\{ C_{l_a}^{n'_b l'_b m'_b}(i,j)\ \times\ \zeta_b^{i+j} z^i \right. \\
&\qquad\qquad \times \int_0^{\infty} r^{j-l_a-1} \times H_{ij}(\zeta_b z, \zeta_b r) \times [V1_a(r) + V2_a(r) + V3_a(r)] \\
&\qquad\qquad \left. \times\ r^2 dr \right\} \\
&=\ k_0\ (I_1 + I_2 + I_3)
\end{aligned} \tag{2.33}$$

where $i_{max} = n'_b + l_b - |m_b| + l_a$, $j_{max} = n'{}_b + l_a$, and

$$\begin{aligned}
k_0\ &= k_a\, k1_b\, k2_b(l_a) \times\ k3_b(l_a) \times\ k4_b(l_a) \\
&= \frac{2\pi}{z^{l_b+l_a+1}\, {\zeta_b}^{n'_b+l_b+2l_a+1}} \left(\frac{(2l_b+1)(l_b+m_b)!\,(l_a-m_b)!}{(2l_a+1)(l_b-m_b)!\,(l_a+m_b)!} \right)^{1/2}
\end{aligned} \tag{2.34}$$

$$\begin{aligned}
I_1 &= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} C_{l_a}^{n'_b l_b m_b}(i,j)\ \times\ \zeta_b^{i+j} z^i \\
&\qquad\qquad \times \int_0^{\infty} r^{j-l_a-1} \times H_{ij}(\zeta_b z, \zeta_b r) \times V1_a(r) \times r^2 dr \\
&= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}}\ C_{l_a}^{n'_b l_b m_b}(i,j)\ \times\ \zeta_b^{i+j} z^i\ \times\ \frac{n!}{\zeta_a^{n+1}} \\
&\qquad \times \left\{ (-1)^j \exp\,(-\zeta_b z) \int_0^{z} r^{j-2l_a} \exp\,(\zeta_b r)\ dr \right. \\
&\qquad\qquad +\ (-1)^i \exp\,(\zeta_b z) \int_z^{\infty} r^{j-2l_a} \exp\,(-\zeta_b r)\ dr \\
&\qquad\qquad \left. -\ \exp\,(-\zeta_b z) \int_0^{\infty} r^{j-2l_a} \exp\,(-\zeta_b r)\ dr \right\}
\end{aligned} \tag{2.35}$$

$$
\begin{aligned}
I_2 &= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} C_{l_a}^{n_b' l_b m_b}(i,j) \times \zeta_b^{i+j} z^i \\
&\qquad \times \int_0^\infty r^{j-l_a-1} \times H_{ij}(\zeta_b z, \zeta_b r) \times V2_a(r) \times r^2 dr \\
&= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} \sum_{k=0}^{n} C_{l_a}^{n_b' l_b m_b}(i,j) \times \zeta_b^{i+j} z^i \times \frac{n!}{(n-k)!\ \zeta_a^{k+1}} \qquad (2.36) \\
&\quad \times \left\{ -(-1)^j \exp(-\zeta_b z) \int_0^z r^{j-2l_a+n-k} \exp[-(\zeta_a - \zeta_b) r] dr \right. \\
&\qquad - (-1)^i \exp(\zeta_b r) \int_z^\infty r^{j-2l_a+n-k} \exp[-(\zeta_a + \zeta_b) r] dr \\
&\qquad \left. + \exp(-\zeta_b z) \int_0^\infty r^{j-2l_a+n-k} \exp[-(\zeta_a + \zeta_b) r] dr \right\}
\end{aligned}
$$

$$
\begin{aligned}
I_3 &= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} C_{l_a}^{n_b' l_b m_b}(i,j) \times \zeta_b^{i+j} z^i \\
&\qquad \times \int_0^\infty r^{j-l_a-1} \times H_{ij}(\zeta_b z, \zeta_b r) \times V3_a(r) \times r^2 dr \\
&= \sum_{i=0}^{i_{max}} \sum_{j=0}^{j_{max}} \sum_{k=0}^{m} C_{l_a}^{n_b' l_b m_b}(i,j) \times \zeta_b^{i+j} z^i \times \frac{m!}{(m-k)!\ \zeta_a^{k+1}} \qquad (2.37) \\
&\qquad \times \left\{ (-1)^j \exp(-\zeta_b z) \int_0^z r^{j+1+m-k} \exp[-(\zeta_a - \zeta_b) r] dr \right. \\
&\qquad + (-1)^i \exp(\zeta_b r) \int_z^\infty r^{j+1+m-k} \exp[-(\zeta_a + \zeta_b) r] dr \\
&\qquad \left. + \exp(-\zeta_b z) \int_0^\infty r^{j+1+m-k} \exp[-(\zeta_a + \zeta_b) r] dr \right\}
\end{aligned}
$$

To carry out the integration in equations (2.35)-(2.37), the following basic formulas (Jones, 1981, 1993) are used:

$$\int_0^a r^n dr = \begin{cases} \dfrac{a^{n+1}}{n+1}, & n > -1 \text{ or } n < -1 \\ 0, & n = -1 \end{cases} \tag{2.38}$$

$$\int_0^a r^n \exp(-br)\, dr$$
$$= \begin{cases} \dfrac{n!}{b^{n+1}} - \dfrac{n!\, exp(-ba)}{b^{n+1}} \displaystyle\sum_{t=0}^{n} \left(\dfrac{(ba)^{n-t}}{(n-t)!} \right), & n > -1 \\ (-b)^{-n-1} \displaystyle\sum_{t=1}^{-n-1} \left(\dfrac{-exp(-ba)\,(-n-t-1)!}{(-ba)^{-n-t}\,(-n-1)!} + \dfrac{(-n-t-1)!}{(-n-1)!\,(-n-t)!} \right), & n < -1 \\ 0, & n = -1 \end{cases} \tag{2.39}$$

$$\int_a^\infty r^n \exp(-br)\, dr$$
$$= \begin{cases} \dfrac{exp(-ba)}{b^{n+1}} \displaystyle\sum_{t=0}^{n} \dfrac{n!}{(n-t)!} (ba)^{n-t}, & n > -1 \\ -(-b)^{-n-1} \displaystyle\sum_{t=1}^{-n-1} \left(\dfrac{-exp(-ba)\,(-n-t-1)!}{(-ba)^{-n-t}\,(-n-1)!} \right), & n < -1 \\ 0, & n = -1 \end{cases} \tag{2.40}$$

$$\int_0^\infty r^n \exp(-br)\, dr = \begin{cases} \dfrac{n!}{b^{n+1}}, & n > -1 \\ \displaystyle\sum_{t=1}^{-n-1} \left(\dfrac{(-b)^{-n-1}\,(-n-t-1)!}{(-n-1)!\,(-n-t)!} \right), & n < -1 \\ 0\,, & n = -1 \end{cases} \tag{2.41}$$

It has been empirically found that when $n > -1$ and the absolute value of ba is very small compared to n, the evaluation of the integral $\int_0^a r^n \exp(-br) dr$ using formula (2.39) leads to a significant loss of precision. In such cases, the integral is obtained via numerical evaluation of the generalized lower incomplete gamma function, $\gamma_\mu(p, x) = \int_0^x s^{p-1} \exp(-\mu s) ds$, using the continued fraction technique recently proposed by Abergel & Moisan (2016):

$$\int_0^a r^n \exp(-br)\,dr = \gamma_b(n+1,a) \qquad n > -1 \tag{2.42}$$

As discussed by the authors, the generalized lower incomplete gamma function can be evaluated with a mantissa-exponent representation: $\gamma_\mu(p,x) = m(\mu x, p)\exp(-\mu x + p\log x)$, and the mantissa $m(\mu x, p)$ is expressed as a continued fraction (cfrac) and computed by using the modified Lentz's method (Lentz,1976):

$$m^{\mathrm{cfrac}}(\mu x, p) = \frac{a_1}{b_1 +}\,\frac{a_2}{b_2 +}\,\frac{a_3}{b_3 +}\,\dots = \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cdots}}} \tag{2.43}$$

where $a_1 = 1$, and $\forall n \geq 1, a_{2n} = -(p-1+n)\mu x,\ a_{2n+1} = n\mu x$, and $b_n = p-1+n$.

To be more consistent and precise in determining whether formula (2.39) or (2.42) should be used to evaluate the integer $\int_0^a r^n \exp(-br)dr$ for $n > -1$, a parametric number p_{lim}, following a suggestion by Abergel & Moisan (2016), is introduced:

$$p_{\mathrm{lim}}(\mathrm{ba}) = \begin{cases} 5\sqrt{|ba|} - 5, & ba < -9 \\ 0, & -9 \leq ba \leq 0 \\ ba, & ba > 0 \end{cases}$$

Therefore,

$$\int_0^a r^n \exp(-br)dr$$

$$= \begin{cases} \dfrac{n!}{b^{n+1}} - \dfrac{n!\exp(-ba)}{b^{n+1}} \displaystyle\sum_{t=0}^{n} \left[\dfrac{(ba)^{n-t}}{(n-t)!}\right], & n > -1 \ and\ n < p_{\mathrm{lim}} \\ \gamma_b^{\mathrm{cfrac}}(n+1,a), & n > -1\ \ and\ n \geq\ p_{\mathrm{lim}} \end{cases} \tag{2.44}$$

2.3 Benchmark systems

The aEP and its hybrid counterpart (aEP/MM) have been implemented in the in-house version of program *XDPROP*, a part of the *XD* package (Volkov *et al.*, 2016). The speed and accuracy of the methods have been tested on a total of 28 dimers of five molecular systems (Table 2.1). The size of each monomer ranged from three atoms (water) to 225 atoms (dodecapeptide). Among the benchmark systems were six dimers of α-glycine, included for consistency with the two previous studies (Volkov, Koritsanszky *et al.*, 2004; Spackman, 2006). As before, crystal structures of the benchmark systems were extracted from the Cambridge Structural Database (Groom *et al.*, 2016). The program *LSDB* (Volkov, Li *et al.*, 2004; Dominiak *et al.*, 2007; Volkov *et al.*, 2007) was used to extend hydrogen atoms to the standard neutron distances (Allen, 1986), and assign pseudoatom parameters (at the hexadecapole level, $l_{\max} = 4$, for the non-hydrogen atoms and at the quadrupole level, $l_{\max} = 2$, for the hydrogen atoms) from the University at Buffalo Aspherical Atom Databank (Volkov, Li *et al.*, 2004; Dominiak *et al.*, 2007). All atom types present in the benchmark systems were found in the databank. The total electron count for each system predicted by the databank was always within 1% from the “true” electron count (0.28% for leu-enkephalin, 0.52% for nanopeptide, 0.44% for decapeptide, and 0.60% for dodecapeptide). Nevertheless, to satisfy the total electroneutrality, valence population parameters for all atoms were scaled in *LSDB* using the Faerman and Price method (Faerman & Price, 1990). Molecular dimers used for benchmark calculations were identified using the program *PLATON* (Spek, 2009).

Table 2.1 Molecular systems used for benchmark calculations.

The molecular geometries of generated dimers are available in the form of XYZ files in the Supporting information.

Crystal Structure	Dimer Label	Dimer Description	Number of atoms	
			monomer 1	monomer 2
α-Glycine †	Gly1…6	glycine dimers	10	10
Leu-enkephalin ‡	Lenk1…3	leu-enkephalin dimers	77	77
	Lenk4…10	leu-enkephalin-water dimers	77	3
	Lenk11	water-water dimer	3	3
Nonapetide §	Nonapep1…2	nonapeptide dimers	174	174
	Nonapep3	nonapeptide-dimethylacetamide dimer	174	15
Decapeptide ¶	Decapep1	decapeptide dimer	181	181
	Decapep2…3	decapeptide-methanol dimers	181	6
	Decapep4…5	decapeptide-water dimers	181	3
Dodecapeptide ††	Dodecapep1	dodecapeptide dimer	225	225
	Dodecapep2…3	dodecapeptide-2-butanone dimers	225	13

† CSD code: GLYCIN85 (Destro *et al.*, 2000). ‡ Tyrosyl-glycyl-glycyl-phenylalanyl-leucine trihydrate (Tyr–Gly–Gly–Phe–Leu).3H_2O; CSD code: GEWWAG01 (Pichon-Pesme *et al.*, 1992; Wiest *et al.*, 1994). § Methyl *N*-(t-butoxycarbonyl)leucylleucyl-2-methylalanylleucylleucyl-2-methylalanylleucylleucyl-2-methylalaninate *N,N*-dimethylacetamide solvate; Boc- (L-Leu-L-Leu-Aib)2-L-Leu-D-Leu-Aib-OMe, where Aib is α-aminoisobutyric acid, Leu leucine, Boc *tert*-butyloxycarbonyl; CSD code: DOWCEZ (Demizu, Yamashita, Misawa *et al.*, 2015). ¶ Methyl *N*-(*t*-butoxycarbonyl)leucyl-2-methylalanylleucyl-2-methylalanylleucyl-2-methyl-alanylleucyl-2-methylalanylleucyl-2-methylalaninate methanol solvate hydrate; Boc-(L-Leu-Aib)$_5$-OMe; CSD code: IPUNAK (Demizu et al., 2016). †† Methyl *N*-(*t*-butoxycarbonyl)leucyl-leucyl-2-methylalanyl-leucyl-leucyl-2-methylalanyl-leucyl-leucyl-2-methyl-alanyl-leucyl-leucyl-2-methylalaninate butan-2-one solvate; Boc-L-Leu-L-Leu-Aib-(D-Leu-D-Leu-Aib)$_2$-L-Leu-L-Leu-Aib-OMe; CSD code: VUQZUE (Demizu, Yamashita, Doi *et al.*, 2015).

2.4 Performance of the analytical Exact Potential (aEP) method

2.4.1 Intermolecular electrostatic interaction energies

To assess the numerical accuracy of the aEP method, the electrostatic interaction energies (E_{es}) for all dimers calculated with aEP were compared with those evaluated using the numerical nEP method with four different sets of (radial × angular) points: 50×194, 70×302, 200×1202, and 400×5810 (Table 2.2). As in the previous study (Volkov, Koritsanszky *et al.*, 2004), Gauss-Chebyshev (Becke, 1988) and Lebedev (Lebedev & Laikov, 1999) quadratures were used for the radial and angular parts, respectively. Radial coordinates and weights were remapped using the formula (M4) of Treutler and Ahlrichs (1995).

As expected, the aEP results show the best agreement with those from nEP integrations with the largest (400×5810) grid (columns 2 and 3 of Table 2.2). The root mean square (RMS) difference between these methods over all benchmark systems is only 0.4 kJ/mol, with the largest discrepancies of 1.2 kJ/mol for the first dodecapeptide dimer and 0.9-1.0 kJ/mol for three other (Nonapep1, Nonapep2, and Decapep1) dimers. These discrepancies are likely due to deficiency of numerical integrations over pseudoatoms at high separations since all these discrepancies are essentially removed when nEP is replaced with nEP/MM (see column 4 of Table 2.3).

Table 2.2 Deviations for E_{es} (in kJ/mol) evaluated using the nEP method with the different quadrature grids (number of radial points × number of angular points) relative to those obtained with the newly developed aEP approach.
Deviations larger than 1 kJ/mol are shown in *italic underlined* font; those above 2 kJ/mol are also **bold underlined** font.

Dimers	Analytical EP (aEP) ‡	Numerical EP (nEP) †			
		400×5810	200×1202	70×302	50×194
Gly1	-115.8	0.0	0.4	*1.5*	0.5
Gly2	-29.1	0.1	0.4	-0.3	**2.2**
Gly3	-87.4	0.0	0.1	0.6	-0.3
Gly4	-163.7	0.1	-1.0	*1.4*	*1.9*
Gly5	46.5	0.0	0.1	-0.1	0.3
Gly6	-23.3	0.0	0.0	0.1	0.1
Lenk1	-313.0	0.3	-0.5	0.3	0.2
Lenk2	-156.8	0.1	0.5	*-1.2*	1.0
Lenk3	-15.7	0.3	0.4	0.7	0.3
Lenk4	-64.2	0.0	0.6	0.0	**2.1**
Lenk5	-62.3	0.0	0.3	*-1.2*	0.8
Lenk6	-44.6	0.1	0.0	0.7	*1.5*
Lenk7	-53.8	0.0	-0.6	**-4.5**	*1.3*
Lenk8	-35.4	0.0	-0.1	0.6	0.2
Lenk9	-23.8	0.1	0.2	-0.1	*1.4*
Lenk10	-29.9	0.0	0.0	-0.1	0.3
Lenk11	-42.9	0.0	0.3	-0.2	0.0
Nonapep1	-135.1	1.0	*1.1*	0.1	*1.7*
Nonapep2	-11.5	1.0	*1.1*	*1.2*	0.8
Nonapep3	-40.2	0.1	0.2	0.3	*1.4*
Decapep1	-198.2	0.9	1.0	*1.3*	0.8
Decapep2	-58.7	0.1	0.3	0.8	*1.8*
Decapep3	-29.3	0.1	-0.3	0.9	*1.4*
Decapep4	-44.7	0.0	*-1.1*	*-1.8*	*1.7*
Decapep5	-12.4	0.0	0.0	0.1	0.1
Dodecapep1	-202.7	*1.2*	*1.5*	*1.5*	**2.8**
Dodecapep2	-14.6	0.1	0.2	0.1	*-1.9*
Dodecapep3	-13.0	0.1	0.2	0.2	0.7
RMS §	—	0.4	0.6	1.2	1.3

‡ E_{es}(aEP) † ΔE_{es} relative to E_{es}(aEP) § Root mean square ΔE_{es} relative to E_{es}(aEP)

Table 2.3 Deviations for E_{es} (in kJ/mol) evaluated using the analytical and numerical (with different radial × angular quadrature grids) EP/MM methods relative to those obtained with the aEP approach.

All EP/MM calculations were performed with R_{cutoff} = 5 Å. Deviations larger than 1 kJ/mol are shown in *italic underlined* font; those above 2 kJ/mol are in **bold underlined** font.

Dimers	Analytical EP (aEP) ‡	Analytical EP/MM (aEP/MM) †	Numerical EP/MM (nEP/MM) †			
			400×5810	200×1202	70×302	50×194
Gly1	-115.8	0.0	0.0	0.4	*1.5*	0.5
Gly2	-29.1	0.0	0.1	0.4	-0.2	**2.1**
Gly3	-87.4	0.0	0.0	0.1	0.6	-0.3
Gly4	-163.7	0.0	0.1	-1.0	*1.5*	*1.9*
Gly5	46.5	0.0	0.0	0.1	-0.1	0.3
Gly6	-23.3	0.0	0.0	0.0	0.0	0.1
Lenk1	-313.0	0.1	0.1	-0.7	0.0	0.0
Lenk2	-156.8	0.1	-0.2	0.3	*-1.5*	0.8
Lenk3	-15.7	0.1	0.1	0.2	0.5	0.1
Lenk4	-64.2	0.0	0.0	0.6	0.0	*2.0*
Lenk5	-62.3	0.0	0.0	0.3	*-1.2*	0.8
Lenk6	-44.6	0.0	0.1	0.0	0.7	*1.5*
Lenk7	-53.8	0.0	0.0	-0.6	**-4.5**	*1.3*
Lenk8	-35.4	0.0	0.0	-0.1	0.6	0.2
Lenk9	-23.8	0.0	0.1	0.2	-0.1	*1.4*
Lenk10	-29.9	0.0	0.0	0.0	-0.1	0.3
Lenk11	-42.9	0.0	0.0	0.3	-0.2	0.0
Nonapep1	-135.1	0.2	0.2	0.3	-0.7	0.9
Nonapep2	-11.5	0.0	0.0	0.1	0.2	-0.2
Nonapep3	-40.2	0.0	0.0	0.1	0.2	*1.3*
Decapep1	-198.2	0.1	0.0	0.2	0.5	0.0
Decapep2	-58.7	0.0	0.1	0.3	0.8	*1.8*
Decapep3	-29.3	0.0	0.1	-0.3	0.9	*1.4*
Decapep4	-44.7	0.0	0.0	*-1.2*	*-1.8*	*1.7*
Decapep5	-12.4	0.0	0.0	0.0	0.1	0.0
Dodecapep1	-202.7	0.1	0.1	0.4	0.4	*1.7*
Dodecapep2	-14.6	0.0	0.1	0.2	0.1	*-2.0*
Dodecapep3	-13.0	0.1	0.1	0.2	0.2	0.7
RMS §	—	0.1	0.1	0.4	1.1	1.2

‡ E_{es}(aEP) † ΔE_{es} relative to E_{es}(aEP) § Root mean square ΔE_{es} relative to E_{es}(aEP)

Reducing the number of quadrature points in nEP, in general, increases the discrepancy between the aEP and nEP values (columns 4-6 of Table 2.2). The RMS differences for nEP with 200×1202, 70×302 and 50×194 points are 0.6, 1.2 and 1.3 kJ/mol, respectively. Discrepancies of 1-2 kJ/mol are common with only few extending beyond 2 kJ/mol. The differences of 2.1 kJ/mol (Gly2) and 2.8 kJ/mol (Dodecapep1) for the 50×194 nEP integrations are not surprising, which is not the case for the largest difference of 4.5 kJ/mol obtained with the 70×302 grid for the Lenk7 dimer. It shows that increasing the number of the quadrature grids points does not always lead to better energies, a definite weakness of the numerical integration technique.

The results presented in Table 2.2 clearly demonstrate the superiority of the analytical EP method over the numerical EP method, as expected.

2.4.2 Individual Coulomb integrals

In the process of the manuscript review, one of the referees mentioned that the accuracy of the analytical EP (aEP) method cannot be tested against the numerical EP (nEP) approach, as both may provide biased results, and suggested that external reference is needed. To address this very much justified concern, we have performed an additional analysis of the accuracy *and precision* of the aEP energies.

When considering the expansion of E_{es} [equation (2.5)] we note that the E_{nn} term in both methods (nEP and aEP) is evaluated exactly. The two electron-nuclear interaction terms, $E_{\mathrm{en}}(ab)$ and $E_{\mathrm{en}}(ba)$, are also evaluated in aEP essentially exactly within the double-precision (64-bit) floating point arithmetic and thus should be more accurate than *any* numerical integration in nEP. As such, the only term that is evaluated with some degree of uncertainty is E_{ee}, which is a sum of two-center Coulomb integrals over *all* core, valence, and deformation Slater-type functions (Coppens, 1997) for each pair of atoms in a dimer.

To check the accuracy and numerical precision of the E_{ee} part of our aEP code, it was initially decided to evaluate a number of published two-center Coulomb integrals and compare our results with the literature values (Jones, 1993; Magnasco *et al.*, 1998; Magnasco & Rapallo, 2000; Guseinov & Mamedov, 2000; Özmen *et al.*, 2003; Gümüş, 2005; Yakar *et al.*, 2006). However, it turns out that (*i*) many of those integrals include a very short internuclear distance and (*ii*) for many integrals the published values are not very consistent. To circumvent the problem, we followed the recently published approach by Silverstone (2014) and created a *Mathematica* (Wolfram Research, Inc., 2016) code that is capable of evaluating two-center Coulomb integrals to the desired precision (Volkov *et al.*, 2018). In brief, the code uses a combination of the Fourier-transform technique (Prosser & Blanchard, 1962; Geller, 1964; Silverstone, 1966; Harris & Michels, 1967) and numerical contour integration available in *Mathematica*. The number of significant digits in the calculated integrals are controlled using *Mathematica*'s `WorkingPrecision` and `AccuracyGoal` options (Wolfram Research, Inc., 2016). The original method (Silverstone, 2014) was shown to easily give 36-digit accuracy and beyond for overlap integrals. Using our own contour integration implemented in *Mathematica*, we were able to reproduce all digits in the 12 overlap integrals with the integer-*n* Slater-type functions listed by Silverstone (2014), which in turn are in excellent agreement with the values published earlier by Bağci & Hoggan (2014).

The *Mathematica* contour integration code (with the `WorkingPrecision` option set at 90, which makes *Mathematica* maintain 90 digits in all internal computations, and the AccuracyGoal option set at 70, which requests 70 digits in the final numerically evaluated integral) was used to evaluate all 89,875 two-center Coulomb integrals between Slater-type functions in the first glycine dimer (Gly1) with at least 36-digit accuracy. To simplify the analysis, the Coulomb integrals were separated into three distinct groups:

(i) 70,225 integrals between the spherical core/valence functions of the two atoms (*a* and *b*). For this type of integrals, the aEP method in *XDPROP* was able to reproduce on average 13±1 digits. The minimum number of reproducible digits was found to be 9 in only 19 out of all 70,225 integrals in this group.

(ii) 15,900 integrals between the spherical core/valence functions of atom *a* (*b*), and the deformation density functions of atom *b* (*a*). The average number of digits reproduced by our aEP method for this type of integrals was 10±2. Out of 15,900 integrals, only 2 integrals were found to be determined to 3 digits, 72 integrals to 4 digits, and 408 integrals to 5 digits. All the remaining integrals in this group were determined to 6 or more digits.

(iii)3,750 integrals between the deformation density functions of atoms *a* and *b*. The aEP method was able to reproduce on the average 9±2 digits for this type of integrals. Only two integrals ended up having 2 reproducible digits, 12 integrals – 3 digits, and 51 integrals – 4 digits. The remaining integrals were all determined to ≥ 5 digits.

The results of this comparison are conveniently summarized in Figure S2.1 (in Appendix A), in which the number of digits in individual Coulomb integrals over pairs of Slater functions reproduced by the aEP method in *XDPROP* is plotted a function of the integral magnitude (represented using the log-10 scale). It shows that the most important

Coulomb integrals (*i.e.* those with the largest magnitude) belong to the first group (core/valence - core/valence group) and are determined to 10 or more digits. The number of digits lost in the very small number of integrals belonging to the other two groups is not of much concern, as these integrals have markedly smaller values (note the log-10 scale), and therefore are not significant.

The performed analysis confirms (independently from the comparison with the nEP values) conclusions regarding the accuracy of the aEP electrostatic interaction energies (E_{es}) drawn in the previous section.

We note that when calculating values for some of the published two-center Coulomb integrals (Jones, 1993; Magnasco *et al.*,1998; Magnasco & Rapallo, 2000; Guseinov & Mamedov, 2000; Özmen *et al.*, 2003; Gümüş, 2005; Yakar *et al.*, 2006) we discovered that our aEP code does not work well for integrals with small internuclear separations (below ≈0.3 Å) and very high values of *n*, *l,* and *m* [equation (2.11)]. However, since the proposed analytical EP method is designed for calculation of *intermolecular* E_{es}, the internuclear separation is never lower than 0.3 Å. Because the level of the multipolar expansion in *XD* (and thus, in *XDPROP*) is currently limited to $l_{\mathrm{max}} = 4$ (hexadecapoles), the issues with high values of *l,* and *m* can also be disregarded. Finally, the pseudoatom density functions with large values of *n* are primarily confined to the first group of integrals (spherical core/valence – spherical core/valence) for which l=0 and m=0, which are evaluated well by our code. That said, we are in the process of testing the analytical contour integration technique for evaluation of Coulomb integrals (Todd *et al.*, 1970) which should remove limitations of the current implementation.

2.5 The choice of R_{cutoff} in the analytical EP/MM (aEP/MM) method

In the previous study (Volkov, Koritsanszky *et al.*, 2004) the optimal internuclear cutoff distance R_{cutoff} that is used in the EP/MM method to switch from the EP integration to the MM approximation was found to be 4 Å. To verify this numerical value, we compared the aEP results for the six glycine dimers with those obtained *via* the aEP/MM approach at different R_{cutoff} values: 4.0, 4.2, …, 5.2 Å (Figure 2.1).

As expected, the absolute differences $\Delta E_{\text{es}} = |E_{\text{es}}(\text{aEP}) - E_{\text{es}}(\text{aEP/MM})|$, averaged over all glycine dimers, decrease on increasing the R_{cutoff} value. The drop is steep between 4.0 and 4.4 Å, but levels off beyond 4.6 Å. That said, even at 4.0 Å, the mean deviation introduced in aEP/MM is under 0.2 kJ/mol.

Figure 2.1 also illustrates the dependence of the computational cost of the aEP/MM on the R_{cutoff} parameter. As expected, the aEP/MM computation time increases with R_{cutoff}. For the glycine dimers, when R_{cutoff} = 4.0 - 4.2 Å the aEP/MM calculation is essentially two times faster than the full aEP calculation but slows down as R_{cutoff} is increased. Depending on the desired accuracy of E_{es}, the appropriate value of R_{cutoff} may be chosen anywhere between 4 and 5 Å, as illustrated in Figure 2.1. We note that once the heavier elements are included in the charge density model, the R_{cutoff} parameter will likely need to be increased.

In all subsequent aEP/MM and nEP/MM calculations presented in this study, the R_{cutoff} value was set at 5 Å.

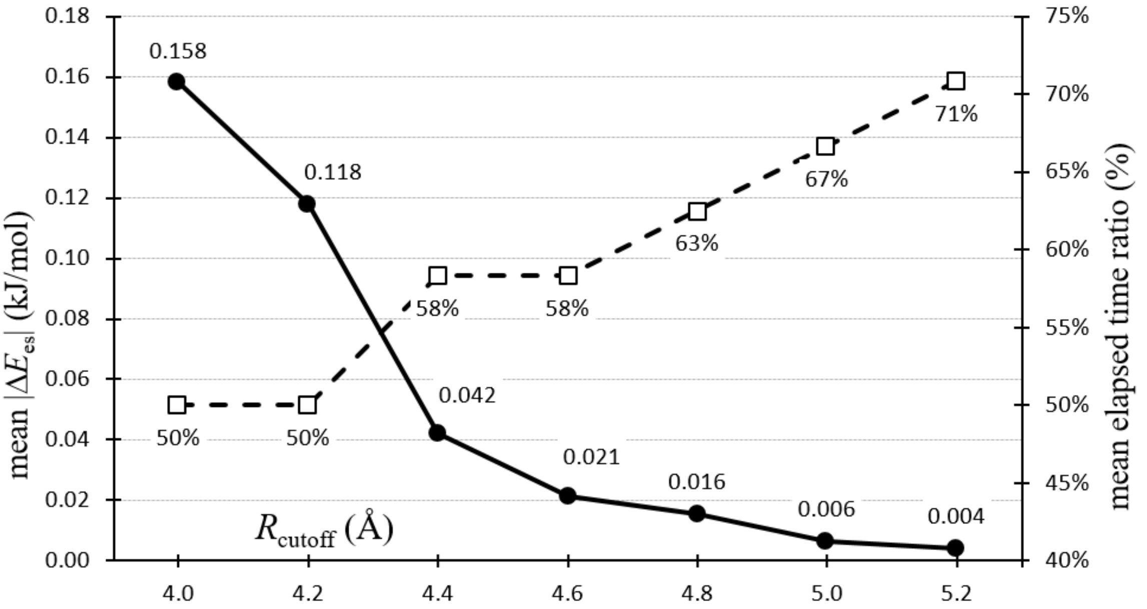


Figure 2.1 The mean absolute difference between E_{es} (kJ/mol) (full circles / solid line) from the aEP and aEP/MM methods for six dimers of α-glycine plotted as a function of R_{cutoff} (Å) at which the aEP/MM method switches from the aEP to the MM approximation. Also, the mean ratio of the elapsed times of aEP/MM and aEP (in %) is shown as a function of R_{cutoff} (empty squares / dashed line).

2.6 Performance of the analytical EP/MM (aEP/MM) method

2.6.1 Accuracy

The accuracy of the hybrid aEP/MM method has been evaluated *via* comparison with the "new" aEP and "old" nEP/MM techniques (Table 2.3). The agreement between aEP, aEP/MM and nEP/MM/400×5810 (with R_{cutoff} = 5 Å for the two hybrids) is excellent – the RMS deviations are on the order of 0.1 kJ/mol. The agreement between aEP and aEP/MM is not unexpected – it is what the hybrid method is designed for. Perhaps the most surprising result is a significant improvement in the numerical EP/MM (nEP/MM) energies compared to nEP (Table 2.2), especially evident for the Decapep1, Dodecapep1, Nonanep1 and Nonapep2 dimers. For these systems, the error drops from about 1 kJ/mol in nEP/400×5810 down to 0.2 kJ/mol and below in nEP/MM/400×5810. As mentioned above, this can be attributed to some difficulties for the numerical EP procedure (even with an extended grid) to evaluate Coulomb integrals over pseudoatoms at high separations. The MM approach (despite being an approximation) helps to correct for that. In fact, one can possibly argue that for large separations the MM approximation may even be more accurate than the analytical EP procedure since the evaluation of the exponential terms $\exp(-\zeta r)$ and $\exp(-\zeta r)$ in the analytical integrals may lead to numerical instabilities (Kay & Silverstone, 1970). The analysis of the total intermolecular E_{es} energies and Coulomb integrals over the individual Slater functions clearly shows that the analytical EP (aEP) method is reliable even at large separations as long as *n*, *l*, and *m* [equation (2.11)] are not too high (see discussion at the end of section 2.4.2). Note that for the two largest dimers in our benchmark study (decapeptide-decapeptide dimer 1, Decapep1, in which the interatomic separations, R_{ab}, are between 2.0 and 36 Å, and dodecapeptide-dodecapeptide dimer 1, Dodecapep1, in which 1.9 Å ≤ R_{ab} ≤ 40 Å) the aEP and aEP/MM results are in excellent (≈0.1 kJ/mol) agreement!

The hybrid scheme also improves several E_{es} values for the numerical EP (nEP) method with smaller grids (*i.e.* 200×1202, 70×302, and 50×194) – the results for the Decapep1, Dodecapep1, Nonanep1 and Nonapep2 dimers are improved by about 1 kJ/mol. However, there is no improvement for some other dimers (such as Gly2, Lenk4, Lenk7 *etc.*) which means that there is a problem with the numerical integration over pseudoatoms at separations less than 5 Å. At this time, there is really no need for further investigation as the implemented aEP method is clearly superior in terms of accuracy to the “old” nEP technique.

2.6.2 Speed

While accuracy is undoubtedly essential, it is not of much use if the accurate method is too computationally demanding or too time-consuming. For example, the numerical EP and EP/MM methods with the 400×5810 grid are rather accurate, but slow, especially nEP.

While we have collected timings for all nEP runs, we shall omit those from our discussion below as this technique is clearly cost prohibitive for large systems. For example, it took 2.6 days (d) to perform a nEP/MM/400×5810 calculation for *each* enkephalin-enkephalin dimer, and whopping 19 d to complete the dodecapeptide-dodecapeptide calculation.

Table 2.4 lists elapsed times (in seconds) for all aEP, aEP/MM, and nEP/MM calculations of the E_{es} values listed in Table 2.3. All runs were performed using a 2.8 GHz AMD Opteron 6348 processor (Piledriver-based “Abu Dhabi” model released in 2012). No special optimization options beyond –O2 were used in GFortran when compiling *XDPROP*. Note that the single-threaded Passmark score (PassMark Software, 2018) for this 6-year old processor is not very high (about 1200), while that for an entry-level AMD Ryzen 3 1300X processor released in 2017 is 50% higher (1879). This is to show that we did not artificially inflate performance of our aEP and aEP/MM methods by using one of the latest processors, such as, for example, Intel Core i7-7700K whose single-threaded Passmark score is 2582 (PassMark Software, 2018). That said, in order to estimate the speedup, one would achieve when using a modern processor, we have included timings for the largest calculations obtained with a 2.8 GHz Intel Xeon E3-1505M v5 “Skylake” processor with a Passmark score of 1905 (PassMark Software, 2018).

The timings presented in Table 2.4 clearly show the superiority of the newly developed aEP/MM method: it is hands down the fastest of all tested techniques. For larger molecules, where the computational time is crucial, the aEP/MM method more than ten times faster than the closest competitor, nEP/MM/50×194, all the while providing more accurate electrostatic interaction energies. At no times, the aEP/MM computation exceeds 4 s on the AMD Opteron 6348 CPU, even when processing interactions in the decapeptide-decapeptide and dodecapeptide dimers. Using a faster Xeon E3-1505M v5 processor (whose PassMark score is about 1.6 times higher than that of Opteron 6348) speeds up the aEP/MM calculations by factor of ≈1.3, which is not unusual.

It is expected that for even larger systems, such as proteins and enzymes, the aEP/MM calculation timings will not drastically increase as in most of the cases there is a limited number of atoms in the two monomers with separations below 5 Å.

We note that the timings of all EP/MM (both analytical and numerical) computations can be further reduced by lowering the R_{cutoff} criterion from 5 Å down to 4 - 4.2 Å at the expense of a slight reduction in accuracy of the calculated E_{es} values (Figure 2.1).

Table 2.4 Elapsed time (seconds) for calculation of E_{es} using the aEP method, aEP/MM method, and nEP/MM method with grids of various sizes.

All calculations were performed using a 2.8 GHz AMD Opteron 6348 processor (Piledriver-based “Abu Dhabi” model released in 2012). No special optimization options beyond –O2 were used in GFortran when compiling *XDPROP*.

Dimers	Analytical EP (aEP)	Analytical EP/MM (aEP/MM)	Numerical EP/MM (nEP/MM)			
			400×5810	200×1202	70×302	50×194
Gly1	0.9	0.3	1188	133	10.6	4.9
Gly2	0.9	0.4	1336	152	12.6	5.7
Gly3	0.9	0.6	2204	238	21	9.2
Gly4	0.9	0.8	3067	330	31	12.9
Gly5	0.9	0.7	2281	241	21	9.3
Gly6	0.9	0.4	1315	150	12.3	5.6
Lenk1	50	3.3 (2.7†)	15045	1815	145	72
Lenk2	50	1.6	5305	555	52	25
Lenk3	50	1.7	6646	706	66	31
Lenk4	1.5	0.3	1065	101	9.7	4.8
Lenk5	1.5	0.2	739	77	7.4	3.6
Lenk6	1.5	0.3	1288	134	13.0	6.2
Lenk7	1.5	0.3	928	101	9.6	4.6
Lenk8	1.5	0.3	758	86	8.1	3.2
Lenk9	1.5	0.6	2623	253	26	11.2
Lenk10	1.5	0.4	1456	134	14.2	6.0
Lenk11	0.1	0.1	241	27	2.9	1.2
Nonapep1	184	3.3 (2.4†)	14249	1741	141	74
Nonapep2	176	2.1	7766	868	76	35
Nonapep3	14.8	1.2	5901	615	51	24
Decapep1	198	3.4 (2.7†)	13325	1621	144	69
Decapep2	5.5	0.5	2397	250	22	11.0
Decapep3	5.5	0.8	4692	470	42	19.8
Decapep4	2.9	0.7	3848	404	37	16.8
Decapep5	2.8	0.3	1583	163	15	6.9
Dodecapep1	307	3.6 (2.8†)	12238	1153	134	64
Dodecapep2	13.9	1.0	5798	500	57	30
Dodecapep3	13.8	1.2	5378	563	50	30

† elapsed time (seconds) when using Intel Xeon E3-1505M v5 2.8 GHz processor

Surprisingly, even the aEP method holds its ground against the hybrid nEP/MM techniques, in most cases being faster than nEP/MM/200×1202 and even nEP/MM/50×194. For smaller molecules, *i.e.* glycine and water-water dimers, the aEP method is significantly faster than the nEP/MM/50×194 method, and sometimes (for example, glycine dimers) is almost as fast as aEP/MM. That said, there is no reason to use the aEP method for production runs as the aEP/MM approach (when combined with the appropriate R_{cutoff} criterion) provides just as accurate E_{es} energies at a significantly lower computational cost.

2.7 Concluding remarks

The existing EP/MM for the calculation of electrostatic interaction energies from pseudoatom-based molecular electron densities is further improved in terms of speed and accuracy by replacing the numerical (quadrature) evaluation of the electron-nuclear attraction (NAI) and electron-electron repulsion integrals (ERI) of the exact potential with accurate, stable, and fast analytical technique (aEP).

The electron-nuclear attraction integrals are evaluated using the electronic potential, $V^{\text{elec}}(\mathbf{r})$, formulas derived in our previous study (Volkov *et al.*, 2006), though any other fast and accurate analytical method is applicable; see, for example, any of the Jones' formulas (Jones, 1981, 1991, 1993) or methods listed by Volkov *et al.* (2006), Spackman (2007).

The electron-electron repulsion integrals are evaluated with the help of the Löwdin α function (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978) and *C*-matrix of Jones and Weatherford (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993). For a two-atom system, this technique shifts the origin of the coordinate system to one of the atoms and expands functions of the other about this common origin in an infinite series of spherical harmonics. Following the Jones approach (Jones & Weatherford, 1978; Jones, 1980, 1981, 1984, 1991, 1992, 1993), we were able to obtain analytical, numerically stable (at the desired internuclear distances and not too high values of *n*, *l*, and *m*) implementation for evaluation of the electron-electron repulsion integrals.

The resulting standalone version of the analytical Exact Potential method (aEP) and its combination with the Multipole Moment (MM) approximation, named aEP/MM, have been tested on a number of molecular systems composed of the H, C, N and O atoms, ranging from water-water to dodecapeptide-dodecapeptide dimers, whose aspherical molecular electron densities were constructed using University at Buffalo Aspherical Atom Databank (Volkov, Koritsanszky *et al.*, 2004; Dominiak *et al.*, 2007).

A series of hybrid aEP/MM calculations performed at different values of R_{cutoff} show that with R_{cutoff} = 5 Å (which was found to give slightly more accurate energies for this type of molecular systems than the previously recommended value of 4 Å; the value of R_{cutoff} will likely need to be increased if heavier elements are present), the aEP/MM method produces results that are within 0.2 kJ/mol from those obtained *via* a fully-analytical integration (aEP) and the nEP/MM method with the 400×5810 grid (the latter has no practical applications due to extremely high computational cost). On the contrary, the aEP/MM method is numerically reliable and fast, even for large systems. The benchmark aEP/MM calculations never exceeded 4 s when running on a 6-year old 2.8 GHz AMD Opteron 6348 processor, even for nonapetide, decapetide, and dodecapeptide dimers. On a more modern processor (2.8 GHz Intel Xeon E3-1505M v5), no benchmark aEP/MM calculation exceeded 3 s.

As such, the newly developed hybrid aEP/MM scheme is highly recommended for calculation of electrostatic interaction energies in molecular systems of various sizes as long as their electron densities are defined within the pseudoatom electron density formalism. The latest version of the *XDPROP* code incorporating the described technique is available upon request from one of the authors (AV).

References

Abergel, R. & Moisan, L. (2016). *Fast and accurate evaluation of a generalized incomplete gamma function*. MAP5 2016-14. 2016. 〈hal-01329669〉 *http://hal.archives-ouvertes.fr/hal-01329669*

Abramov, Y. A., Volkov, A., Wu, G. & Coppens, P. (2000). *Acta Cryst. A* **56**, 585-591.

Allen, F. H. (1986). *Acta Cryst.* B**42**, 515–522.

Bader, R. F. (1990). *Atoms in Molecules: A Quantum Theory*. Oxford: Clarendon Press.

Bağci, A. & Hoggan, P. E. (2014). *Phys. Rev. E* **89**, 053307.

Barnett, M. P. & Coulson C. A. (1951). *Phil. Trans. Royal Soc. London. Series A, Math. Phys. Sci.* **243**, 221-249.

Becke, A. D. (1988). *J. Chem. Phys.* **88**, 2547-2553.

Buckingham, A. D. (1967). *Adv. Chem. Phys.* **12**, 107-142.

Buckingham, A. D. (1978). In *Intermolecular Forces: From Diatomics to Biopolymers*; edited by B. Pullman, New York: Wiley.

Buckingham, A.D. & Fowler, P.W. (1985). *Can. J. Chem.* **63**, 2018-2025.

Buckingham, A. D., Fowler, P. W. & Galwas, P. A. (1987). *Chem. Phys.* **112**, 1–14.

Buckingham, A. D., Fowler, P. W. & Hutson, J. M. (1988). *Chem. Rev.* **88**, 963–988.

Coppens, P. (1992) *The structure factor. International Tables for Crystallography, Vol. B: Reciprocal space*, 1st ed., edited by U. Shmueli, ch. 1.2. Dordrecht: Kluwer Academic Publishers.

Coppens, P. (1997). *X-ray Charge Densities and Chemical Bonding.* New York: Oxford University Press.

Cromer, D. T., Larson, A. C. & Stewart, R. F. (1976). *J. Phys. Chem.* **65**, 336-349.

Demizu, Y., Okitsu, K., Yamashita, K., Doi, M., Misawa, T., Oba, M., Tanaka, M. & Kurihara, M. (2016). *Eur. J. Org. Chem.* **2016**, 2815-2820.

Demizu, Y, Yamashita, H., Doi, M., Misawa, T., Oba, M., Tanaka, M. & Kurihara, M. (2015). *J. Org. Chem.* **80**, 8597-8603.

Demizu, Y., Yamashita, H., Misawa, T., Doi, M., Tanaka, M. & Kurihara, M. (2015). *Chem. Pharm. Bull.* **63**, 218-224.

Destro, R., Roversi, P., Barzaghi, M. & Marsh, R. E. (2000). *J. Phys. Chem. A* **104**, 1047-1054.

Dominiak, P. M., Volkov, A., Li, X., Messerschmidt, M. & Coppens, P. (2007). *J. Chem. Theory Comput.* **3**, 232-247.

Faerman, C. H. & Price, S. L. (1990). *J. Am. Chem. Soc.* **112**, 4915-4926.

Fernandez Rico, J., Fernandez, J. J., Lopez, R. & Ramirez, G. (2000). *Int. J. Quantum. Chem.* **78**,137-145.

Filter, E. & Steinborn, E. O. (1978). *Phys. Rev. A.* **18**, 1-11.

Flierler, U. & Stalke, D. (Eds.) (2012). *Electron Density and Chemical Bonding I: Experimental Charge Density Studies.* Berlin, Heidelberg: Springer-Verlag.

Gatti, C. & Macchi, P. (2012). Editors. *Modern Charge-Density Analysis*. Netherlands: Springer.

Geller, M. (1964). *J. Chem. Phys.* **41**, 4006-4007.

Gill, P. M. W., Johnson, B. G. & Pole, J. A. (1993). *Chem. Phys. Lett.* **209**, 506-512.

Greengard, L. & Rokhlin, V. (1987). *J. Comp. Phys.* **73**, 325-348.

Groom, C. R., Bruno, I. J., Lightfoot, M. P. & Ward, S. C. (2016). *Acta Cryst.* B**72**, 171-179.

Gümüş, S. (2005). *Z. Naturforsch. Teil A.* **60**, 477-483.

Guseinov, I. I. (1970). *J. Phys. B. At. Mol. Phys.* **3**, 1399-1412.

Guseinov, I. I. & Mamedov, B. A. (2000). *Int. J. Quantum Chem*. **78**, 146-152.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Harris, F. E. & Michels, H. H. (1967). *Adv. Chem. Phys.* **13**, 205-266.

Hirshfeld, F. L. (1971) *Acta Cryst.* B**27**, 769-781.

Hirshfeld, F. L. & Rzotkiewicz, S. (1974). *Mol. Phys.* **27**, 1319-1343.

Jones, H. W. (1980). *Int. J. Quantum Chem.* **18**, 709-713.

Jones, H. W. (1981). *Int. J. Quantum Chem.* **20**, 1217-1224.

Jones, H. W. (1984). *Phys. Rev. A* **30**, 1-4.

Jones, H. W. (1991). *J. Comp. Chem.* **12**, 1217-1222.

Jones, H. W. (1992). *Int. J. Quantum Chem.* **41**, 749-754.

Jones, H. W. (1993). *Int. J. Quantum Chem.* **45**, 21-30.

Jones, H. W. & Weatherford, C. A. (1978). *Int. J. Quantum Chem. Symp.* **12**, 483-488.

Jones, H. W. & Weatherford, C. A. (1989). *J. Mol. Struct. (Theochem)* **199**, 233-243.

Kay, K. G. & Silverstone, H. J. (1970). *J. Chem. Phys.* **53**, 4269-4285.

Kisiel, Z. (2001). In *Spectroscopy from Space*, edited by J. Demaison, K. Sarka, E. A. Cohen. Dordrecht: Kluwer Academic Publishers.

Kisiel, Z. (2004*). MIN16. Geometries of hydrogen-bonded dimers from the electrostatic model of Buckingham and Fowler*. http://info.ifpan.edu.pl/~kisiel/model/model.htm#min16.

Kisiel, Z., Fowler, P.W. & Legon, A.C. (1990). *J. Chem. Phys.* **93**, 3054-3062.

Kisiel, Z., Fowler, P.W., Legon, A.C., Devanne, D. & Dixneuf, P. (1990). *J. Chem. Phys.* **93**, 6249-6255.

Lebedev, V. I. & Laikov, D. N. (1999). *Dokl. Math*. **59**, 477-481.

Lentz, W. J. (1976). *Appl. Opt*. **15**, 668–671.

Lesiuk, M. & Moszynski, R. (2014). *Phys. Rev. E.* **90**, 063318.

Löwdin, P. O. (1956). *Adv. Phys.* **5**, 1-171.

Magnasco, V., Casanova, M. & Rapallo, A. (1998). *Chem. Phys. Lett*. **289**, 81-89.

Magnasco, V. & Rapallo, A. (200). *Int. J. Quant. Chem.* **79**, 91-100.

McLean, A. D. & Yoshimine, M. (1968). *IBM J. Res. Dev.* **12**, 206-233.

Michael, J. R. & Volkov, A. (2015). *Acta Cryst.* A**71**, 245–249.

Murray, C. W., Handy, N. C. & Laming, G. J. (1993). *Mol. Phys.* **78**, 997-1014.

O-Ohata, K. & Ruedenberg, K. (1966). *J. Math. Phys.* **7**, 547-559.

Özmen, A., Karakaş, A, Atav, Ü. & Yakar, Y. (2003). *Int. J. Quantum Chem.* **91**, 13-19.

PassMark Software (2018). PassMark Software Pty Ltd, Redwood City, California, USA. http://www.cpubenchmark.net Accessed 21 February, 2018.

Paturle, A. & Coppens, P. (1988). *Acta Cryst.* A**44**, 6-7.

Pichon-Pesme, V., Lecomte, C., Wiest, R. & Benard, M. (1992). *J. Am. Chem. Soc.* **114**, 2713-2715.

Prosser, F. P. & Blanchard, C. H. (1962). *J. Phys. Chem*. **36**, 1112.

Ruedenberg, K., Roothaan, C. C. J. & Jaunzemis, W. (1956). *J. Chem. Phys.* **24**, 201-220.

Sharma, R. R. (1976). *Phys. Rev. A.* **13**, 517-527.

Shestakov, A. F. (1992). *J. Struct. Chem.* **33**, 131-133.

Silverstone, H. J. (1966). *J. Phys. Chem.* **45**, 4337-4341.

Silverstone, H. J. (2014). *J. Phys. Chem. A* **118**, 11971-11974.

Silverstone, H. J. & Moats, R. K. (1977). *Phys. Rev. A.* **16**, 1731-1732.

Slater, J. C. (1932). *Phys. Rev.* **42**, 33-43.

Spackman, M. A. (2006). *Chem. Phys. Lett.* **418**, 158-162.

Spackman, M. A. (2007). *Acta Cryst.* A**63**, 198-200.

Spackman, M. A. & Maslen, E. N. (1986). *J. Phys. Chem.* **90**, 2020-2027.

Spek, A. L. (2009). *Acta Cryst.* D**65**, 148-155.

Stewart, R. F. (1976). *Acta Cryst.* A**32**, 565-574.

Stone, A. (2013). *The Theory of Intermolecular Forces*, 2nd ed. Oxford University Press.

Su, Z. & Coppens, P. (1994). *Acta Cryst.* A**50**, 636-643.

Todd, H. D., Kay, K. G. & Silverstone, H. J. (1970). *J. Chem. Phys.* **53**, 3951-3956.

Treutler, O. & Ahlrichs, R. (1995). *J. Chem. Phys.* **102**, 346-354.

Tsirelson, V. G. & Ozerov, R. P. (1996). *Electron Density and Bonding in Crystals*. Bristol, England / Philadelphia, USA: Institute of Physics Publishing.

Volkov, A., King, H. F., Coppens, P. & Farrugia, L. J. (2006). *Acta Cryst.* A**62**, 400–408.

Volkov, A., Koritsanszky T. S. & Coppens, P. (2004). *Chem. Phys. Lett.* **391**, 170–175.

Volkov, A., Li, X., Koritsanszky, T. S. & Coppens, P. (2004). *J. Phys. Chem. A,* **108**, 4283-4300.

Volkov, A., Macchi, P., Farrugia, L. J., Gatti, C., Mallinson, P., Richter, T. & Koritsanszky, T. (2016). *XD2016 - A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors*.

Volkov, A., Messerschmidt, M. & Coppens, P. (2007). *Acta Cryst.* D**63**, 160-170.

Volkov A., Nguyen, D. & Silverstone H. J. (2018). Manuscript in preparation.

Wahl, A. C., Cade, P. E. & Roothaan, C. C. J. (1964). *J. Chem. Phys.* **41**, 2578-2599.

Warshel, A & Levitt, M. (1976). *J. Mol. Biol.* **103**, 227–249.

Weatherford, C. A. & Jones, H. W. (1982). Editors. *ETO Multicenter Molecular Integrals Proceedings of the First International Conference* held at Florida A&M University, Tallahassee, Florida, U.S.A., August 3–6, 1981. Netherlands: Springer.

Wiest, R., Pichon-Pesme, V., Benard, M. & Lecomte, C. (1994). *J. Phys. Chem*. **98**, 1351–1362.

Wolfram Research, Inc. (2016). *Mathematica, Version 10.4*. Champaign, Illinois, USA.

Yakar, Y., Özmen, A. & Atav, Ü. (2006). *Chin. J. Chem.* **24**, 603-608.

Zimont, S. L. & Mar'yaskin, N. Ya. (1972). *Dokl. Akad. Nauk SSSR,* **205**, 1059-1062.

APPENDIX A: Supporting Information

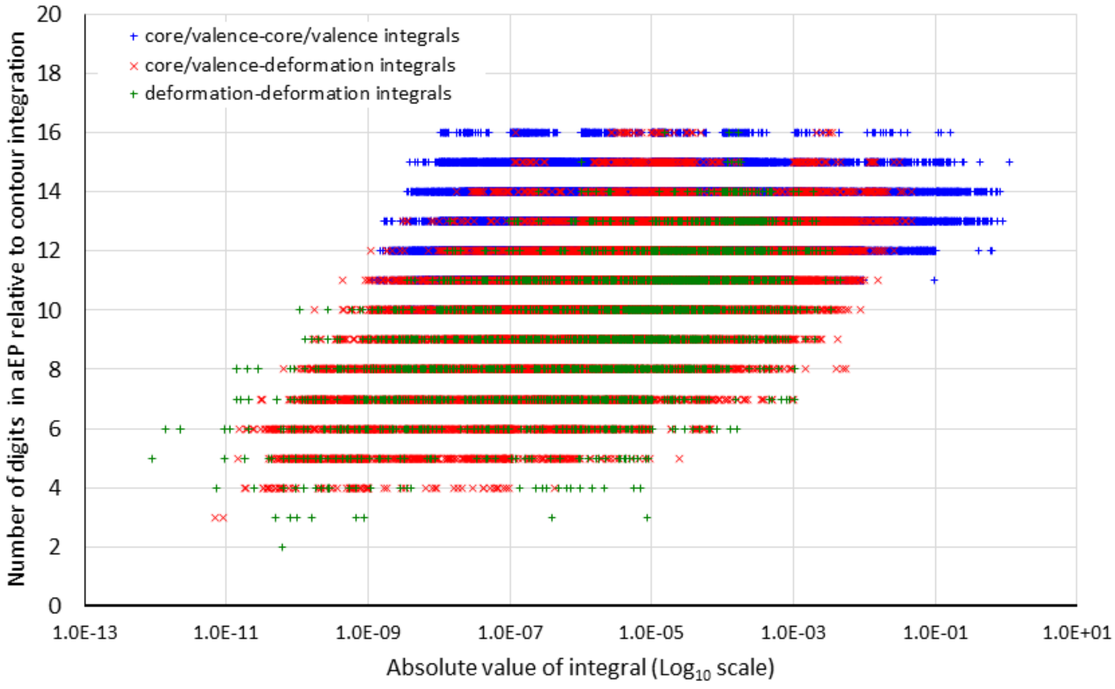


Figure S2.1 Number of digits in the two-center Coulomb integrals in glycine dimer 1 (Gly1) reproduced using the analytical Exact Potential (aEP) method implemented in *XDPROP* relative to the numerical contour integration in *Mathematica* (`WorkingPrecision` → 90, `AccuracyGoal` → 70) plotted as a function of the absolute integral value (Log-10 scale).

CHAPTER III: *ARTICLE 2* - FAST ANALYTICAL EVALUATION OF INTERMOLECULAR ELECTROSTATIC INTERACTION ENERGIES USING THE PSEUDOATOM REPRESENTATION OF THE ELECTRON DENSITY. II. THE FOURIER TRANSFORM METHOD[3]

Abstract

The Fourier transform method for analytical determination of the two-center Coulomb integrals needed for evaluation of the electrostatic interaction energies between pseudoatom-based charge distributions is presented, and its Fortran-based implementation using the 128-bit floating-point arithmetic in the XDPROP module of the XD software is described. In combination with mathematical libraries included in the Lahey/Fujitsu LF64 Linux compiler, the new implementation outperforms the previously reported Löwdin α-function technique [Nguyen *et al.* (2018). *Acta Cryst.* A**74**, 524–536] in terms of precision of the determined individual Coulomb integrals regardless of whether the latter uses the 64-, 80- or 128-bit precision floating-point format, all the while being only marginally slower. When the Löwdin α-function or Fourier transform method is combined with a multipole moment approximation for large interatomic separations (such a hybrid scheme is called the analytical exact potential and multipole moment method, aEP/MM) the resulting electrostatic interaction energies are evaluated with a precision of $\leq 5\times10^{-5}$ kJ/mol for the current set of benchmark systems composed of H, C, N and O atoms and ranging in size from water–water to dodecapeptide–dodecapeptide dimers. Using a 2012 4.0 GHz

[3] This chapter also appears in *Acta Crystallographica Section A: Foundations and Advances* [Nguyen, D., & Volkov, A. (2019) *Acta Cryst.* A**75**, 448-464].

AMD FX-8350 computer processor, the two recom- mended aEP/MM implementations, the 80-bit precision Löwdin α-function and 128-bit precision Fourier transform methods, evaluate the total electrostatic interaction energy between two 225-atom monomers of the benchmark dodecapeptide molecule in 6.0 and 7.9 s, respectively, versus 3.1 s for the previously reported 64-bit Löwdin α-function approach.

3.1 Introduction

In the previous paper (Nguyen *et al.*, 2018; called Paper I in the following), we presented the implementation of the Löwdin α-function technique (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993; Suzuki, 1984, 1985, 1987, 1992; Antone, 1985; Bouferguene, 2005; Bouferguene & Safouhi, 2006; Mamedov & Çopuroğlu, 2011) for analytical evaluation of the two-center Coulomb integrals that appear in calculations of the intermolecular electrostatic interaction energy (E_{es}) between the pseudoatom-based molecular charge distributions (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). The basic form of the Coulomb integral, C, evaluated is

$$C(\mathbf{R}_a, \mathbf{R}_b) = \int\int \frac{\chi_a(\mathbf{r}_a - \mathbf{R}_a)\chi_b(\mathbf{r}_b - \mathbf{R}_b)}{|\mathbf{r}_b - \mathbf{r}_a|} d\mathbf{r}_b d\mathbf{r}_a \qquad (3.1)$$

where χ_a is a Slater-type function (STF; Slater, 1932) centered on nucleus a located at $\mathbf{R}_a$, χ_b is a STF centered on nucleus b located at $\mathbf{R}_b$, and vectors $\mathbf{R}_a$ and $\mathbf{R}_b$ are defined in the global coordinate system. The general form of a three-dimensional wavefunctionnormalized STF in which the angular part is represented by the real spherical harmonic function $y_{l,m}(\theta,\phi)$ is

$$\chi(\mathbf{r}) = \chi(r,\theta,\phi) = \mathcal{N}_{n,\zeta} r^{n-1} \exp\,(-\zeta r) y_l^m(\theta,\phi) \tag{3.2}$$

$$y_l^m(\theta,\phi) = (-1)^{|m|} N_{l,m} P_l^{|m|}(\cos\theta) \begin{Bmatrix} \cos(|m|\phi), m > 0 \\ \sin(|m|\phi), m < 0 \\ 1, m = 0 \end{Bmatrix} \tag{3.3}$$

$$N_{l,m} = \left[\frac{(2l+1)}{2(1+\delta_{m,0})\pi} \frac{(l-|m|)!}{(l+|m|)!} \right]^{1/2} \tag{3.4}$$

where $\delta_{i,j}$ is the Kronecker delta function, and functions $P_l^m(x)$ are the associated Legendre polynomials (Press *et al.*, 1992; Tam, 2008; Wolfram Research, 2018). We follow the notation used in *Mathematica* (Wolfram Research, 2018; Tam, 2008) according to which the Condon-Shortley phase $(-1)^m = (-1)^{|m|}$ (Condon & Shortley, 1959) is included in the definition of the associated Legendre polynomials (Press *et al.*, 1992):

$$P_l^{|m|}(x) = (-1)^{|m|}(1-x^2)^{|m|/2} \frac{d^{|m|} P_l(x)}{dx^{|m|}} \tag{3.5}$$

$$P_l^{-|m|}(x) = (-1)^{|m|} \frac{(l-|m|)!}{(l+|m|)!} P_l^{|m|}(x) \tag{3.6}$$

Note that Coppens (1997) follows the Arfken (1985) convention and excludes the Condon-Shortley phase $(-1)^m$ from the definition of the associated Legendre polynomials. The function $y_l^m(\theta,\phi)$ in equation (3.3) agrees with the definition of real spherical harmonics in Coppens (1997), Ivanic & Ruedenberg (1996), and Blanco *et al.* (1997), but is different from that in Homeier & Steinborn (1996) by a factor of $(-1)^m$. As mentioned by Blanco *et al.* (1997) with a reference to Chisholm (1976), such definition is advantageous as it produces "*signless expressions for the real spherical harmonics*".

The general form of the density-normalized STF is (Hansen & Coppens, 1978; Coppens, 1997):

$$\chi'(\mathbf{r}) = \chi'(r,\theta,\phi) = \mathcal{N}'_{n,\zeta} r^{n'} e^{-\zeta' r} d_{l,m}(\theta,\phi) \tag{3.7}$$

where $d_{l,m}(\theta,\phi)$ is the density-normalized spherical harmonic function (Hansen & Coppens, 1978; Coppens, 1997; Michael & Volkov, 2015):

$$d_{l,m}(\theta,\phi) = (-1)^{|m|} N'_{l,m} P_l^{|m|}(\cos\theta) \begin{Bmatrix} \cos(|m|\phi), m > 0 \\ \sin(|m|\phi), m < 0 \\ 1, m = 0 \end{Bmatrix} \tag{3.8}$$

and the normalization coefficients $N'_{l,m}$ are defined as described in the work of Hansen & Coppens (1978), Paturle & Coppens (1988), Coppens (1997) and Michael & Volkov (2015).

The Löwdin α-function technique (referred to in Paper I as the analytical exact potential method, aEP) was programmed using a standard Fortran90 in the in-house version of the *XDPROP* program of the *XD* software (Volkov *et al.*, 2016), and tested against the numerical three-dimensional quadrature integration (numerical exact potential, nEP; Volkov, Koritsanszky *et al.*, 2004) for 28 molecular systems ranging in size from water–water to dodecapeptide–dodecapeptide dimers with electron densities constructed *via* the University at Buffalo Aspherical Atom Databank (Volkov, Li *et al.*, 2004; Dominiak *et al.*, 2007). The Löwdin α-function method showed an excellent (within 0.2 kJ/mol) agreement with results from a high-level numerical (quadrature) integration, all the while being significantly faster. For example, when combined with the multipole moment approximation for interatomic separations beyond 5 Å (such a hybrid method has been abbreviated as aEP/MM), it took only 3.6 s on a 2012 2.8 GHz AMD Opteron computer processor (CPU) to calculate E_{es} between two monomers of a 225-atom dodecapeptide molecule (Demizu *et al.*, 2015), while the numerical integration method needed ~3.4 h using the same CPU to achieve the same level of accuracy.

While the total intermolecular electrostatic interaction energies (E_{es}) for all benchmark systems calculated in Paper I with the analytical Löwdin α-function integration were undoubtedly correct, one of the referees pointed out (justifiably so) that we did not verify the precision of the determined E_{es} against another (reference) technique. To remedy the issue, we compared (Nguyen *et al.*, 2018) all 89 875 two-center Coulomb integrals in one of the glycine dimers (Gly1) evaluated using the Löwdin α-function technique in *XDPROP* with results from the numerical contour integration approach in *Mathematica* recently proposed by Silverstone (2014). In the process, it was discovered that the number of reproducible digits recovered by the *XDPROP*-based Löwdin α-function technique shows a significant variation. While the average number of reproducible digits was found to be acceptable (13±1, 10±2 and 9±2 digits for core/valence ↔ core/valence, core/valence ↔ deformation and deformation ↔ deformation density integrals, respectively), a number of integrals were determined to as few as three significant digits in the core/valence ↔ deformation density integrals, and two significant digits in the deformation ↔ deformation density set of the integrals. Luckily, those imprecise integrals all had relatively small absolute values, and thus did not significantly affect the resulting electrostatic energies.

The issues with the numerical stability and precision when evaluating Coulomb integrals have been traced to evaluation of (*a*) the *C*-matrix elements for values of *n*, *l*, and *m* greater than approximately eight, and, especially, (*b*) the integrals related to the generalized lower incomplete gamma function (Abergel & Moisan, 2016; Olver *et al.*, 2018), *i.e.*

$$\gamma_{\mu}(p,x) = \int_{0}^{x} s^{p-1} e^{-\mu s} ds \tag{3.9}$$

The issues we encountered are certainly not new. Harris (2003) reiterated that "*analytic expressions for Coulomb integrals* [do] *not need to involve functions of exponential-integral (Ei) or logarithmic type*". However, later studies (for example, Yakar *et al*., 2006) showed that auxiliary functions "*closely related to the* […] *incomplete gamma functions*" may still be used for evaluation of various two-center integrals (overlap, Coulomb, and hybrid). Indeed, our own results presented in Paper I clearly demonstrate the usefulness of such an approach as long as the values of n and l in equations (3.2) and (3.7) are not too high. We also note that the Löwdin α-function method when programmed in Fortran is extremely fast.

The current study consists of three parts. In the first part, we describe a simple implementation of the Fourier transform method for evaluation of the two-center Coulomb (and overlap) integrals. Then, we explore the computational aspects of both the Löwdin α-function and Fourier transform techniques with a goal to increase the resulting precision without a major loss of speed. Finally, we compare the Löwdin α-function and Fourier transform implementations and report our decision as to which method is better suited for evaluation of the two-center Coulomb integrals within the pseudoatom electron-density formalism.

To conclude this introductory section, we note yet another popular technique for evaluation of two-center integrals over STFs that is based on integration in the ellipsoidal coordinates and the Neumann expansion of $\mathbf{r}^{-1}$ [see, for example, Huzinaga (1967), Harris & Michels (1967)]. The most complete and thorough analysis of this technique was given by Harris (2002) who also made his *Maple V* code available for download (Harris, 2002). However, because Harris' *Maple* code (just like our *Mathematica*-based numerical contour integration program) relies heavily on the arbitrary-precision arithmetic built into the computer algebra software, it is not readily translatable to Fortran/C/C++ unless an arbitrary-precision library is available (see the discussion in Section 3.5.2).

For a more complete literature list on the existing methods for evaluation of two-center integrals over STFs we refer to a recent paper by Lesiuk & Moszynski (2014).

3.2 The Fourier transform method for evaluation of the Coulomb integral

3.2.1 General

The history of the Fourier transform method for evaluation of two-center integrals over STFs starts in 1960s with publications by Prosser and Blanchard (1962), Geller (1962, 1963*a,b*, 1964*a,b*), Geller & Griffith (1964), Silverstone (1966, 1967*a,b,c*), O-Ohata & Ruedenberg (1966), Harris & Michels (1967), Todd *et al.* (1970), and it is still going strong (Niehaus & Rico, 2008; Özcan & Öztekin, 2009; Ozdogan & Nalcaci, 2012; Özay & Öztekin, 2013; Silverstone, 2014). While it is nearly impossible to list every single study that employs the Fourier transform method, we note (in addition to the publications listed above) the following papers related to the subject of our study: Weniger & Steinborn (1983), Trivedi & Steinborn (1983), Grotendorst & Steinborn (1985), Weniger *et al.* (1986), Shestakov (1992), Öztekin *et al.* (2001), Guseinov & Mamedov (2002), Berlu (2004), Öztekin (2004), and Öztekin & Özcan (2007).

The Fourier transform implementation that we use in this work is essentially a combination (hybrid) of the approaches proposed by Silverstone (1966, 1967*a,b,c*, 2014), Geller (1962, 1963*a,b*, 1964*a,b*), and Harris & Michels (1967). We start with a simplification of the Coulomb integral formula (3.1) by shifting the origin of the Cartesian coordinate system to the nucleus/center a and denoting the location of the nucleus b as $\mathbf{r}$ (this greatly simplifies the notation used throughout the paper):

$$C(\mathbf{r}) = \int\int \frac{\chi_a(\mathbf{r}_a)\chi_b(\mathbf{r}_b - \mathbf{r})}{|\mathbf{r}_b - \mathbf{r}_a|} d\mathbf{r}_b d\mathbf{r}_a \tag{3.10}$$

where $\mathbf{r} \equiv (r, \theta, \phi)$. Following Geller (1962, 1963*a,b*, 1964*a,b*), the application of the Fourier convolution theorem method to integral (3.10) immediately reduces the number of the integration dimensions from six to three:

$$C(\mathbf{r}) = (2\pi)^{-3} \int F_a(\mathbf{k})H(\mathbf{k})F_b(\mathbf{k})\exp(-i\mathbf{k}\cdot\mathbf{r})\mathrm{d}\mathbf{k} \tag{3.11}$$

where $\mathbf{k}$ is the transform variable, $\mathbf{k} \equiv (k, \bar{\theta}, \bar{\phi})$, $F_a(\mathbf{k})$ and $F_b(\mathbf{k})$ are the Fourier transforms of functions $\chi_a(\mathbf{r}_a)$ and $\chi_b(\mathbf{r}_b - \mathbf{r})$, respectively, and $H(\mathbf{k})$ is the Fourier transform of $\frac{1}{|\mathbf{r}_b - \mathbf{r}_a|}$, which is simply (Geller, 1964a)

$$H(\mathbf{k}) = 4\pi k^{-2} \tag{3.12}$$

Inserting the expression for $H(\mathbf{k})$ into equation (3.11) gives

$$C(\mathbf{r}) = 4\pi(2\pi)^{-3} \int k^{-2} F_a(\mathbf{k})F_b(\mathbf{k})\exp(-i\mathbf{k}\cdot\mathbf{r})\mathrm{d}\mathbf{k} \tag{3.13}$$

Following Silverstone (1966, 1967 *a,b,c*, 2014) and Todd *et al.* (1970) it is advantageous to define the integral $I^{(N)}(\mathbf{r})$:

$$I^{(N)}(\mathbf{r}) = (2\pi)^{-3} \int k^{2N} F_a(\mathbf{k})F_b(\mathbf{k})\exp(-i\mathbf{k}\cdot\mathbf{r})\, d\mathbf{k} \tag{3.14}$$

so that both the Coulomb and overlap,

$$S(\mathbf{r}) = (2\pi)^{-3} \int F_a(\mathbf{k})F_b(\mathbf{k})\exp(-i\mathbf{k}\cdot\mathbf{r})\, d\mathbf{k} \tag{3.15}$$

integrals can be conveniently represented as a simple function of the integral $I^{(N)}(\mathbf{r})$ (Todd *et al.*, 1970; Silverstone, 2014):

$$C(\mathbf{r}) = 4\pi I^{(-1)}(\mathbf{r}) \tag{3.16}$$

$$S(\mathbf{r}) = I^{(0)}(\mathbf{r}) \tag{3.17}$$

Thus, using the Fourier transform method both the Coulomb and overlap integrals can be obtained *simultaneously.* We note that, according to Wheatley (1998), the overlap integral over charge distributions can be used to approximate the exchange energy.

The plane-wave (Rayleigh) expansion for $\exp(\pm i\mathbf{k}\cdot\mathbf{r})$ in terms of complex spherical harmonics $Y_l^m(\theta,\phi)$ (Weissbluth, 1978) is given by (Geller, 1962; Weissbluth, 1978)

$$\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = 4\pi\sum_{\lambda=0}^{\infty}\left(\sum_{\mu=-\lambda}^{\lambda}\left[(\pm i)^{\lambda} j_{\lambda}(kr)Y_{\lambda}^{\mu}(\theta,\phi)Y_{\lambda}^{\mu*}(\bar{\theta},\bar{\phi})\right]\right) \tag{3.18}$$

where symbol * denotes the complex-conjugate, and $j_{\lambda}(z)$ is the spherical Bessel function of the first kind (Arfken, 1985; Olver *et al.*, 2018). In Section S3.1 of the supporting information (Appendix B) , we repeat derivation of the plane-wave expansion in terms of the associated Legendre polynomials $P_l^m(x)$ (Morse & Feshbach, 1953; Geller, 1963*a*; Coppens, 1997):

$$\begin{aligned}\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = \sum_{\lambda=0}^{\infty}(2\lambda+1)(\pm i)^{\lambda} j_{\lambda}(kr)\sum_{\mu=0}^{\lambda}\epsilon_{\mu}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}\\ \times P_{\lambda}^{|\mu|}(\cos\theta)P_{\lambda}^{|\mu|}(\cos\bar{\theta})\cos[|\mu|(\phi-\bar{\phi})]\end{aligned} \tag{3.19}$$

where ϵ_{μ} is defined as in Morse and Feshbach (1953) and Geller (1963a): $\epsilon_0 = 1$ for $\mu = 0$, and $\epsilon_{\mu} = 2$ for $\mu > 0$, or simply $\epsilon_{\mu} = 2 - \delta_{\mu,0}$. Expression (3.19) also shows that the complex conjugation can be freely exchanged between the two spherical harmonic functions.

The general form of the Fourier transform, $F(\mathbf{k})$, of an *unnormalized* STF $\chi(\mathbf{r})$ with real spherical harmonics $y_l^m(\theta,\phi)$,

$$\chi(\mathbf{r}) = \chi(r,\theta,\phi) = r^{n-1}e^{-\zeta r}y_l^m(\theta,\phi) \tag{3.20}$$

is (Geller, 1963*a*, 1964*b*; Silverstone, 1966; Coppens, 1997; Section S3.2 in Appendix B)

$$F(\mathbf{k}) = f_{nl\zeta}(k)y_l^m(\bar{\theta},\bar{\phi}) \tag{3.21}$$

where

$$f_{nl\zeta}(k) = 4\pi i^l \int\limits_0^\infty r^{n+1}e^{-\zeta r}j_l(kr)dr \tag{3.22}$$

Substituting the expansions for $\exp(-i\mathbf{k}\cdot\mathbf{r})$, $F_a(\mathbf{k})$ and $F_b(\mathbf{k})$ into the expression for $I^{(N)}(\mathbf{r})$ gives

$$\begin{aligned} I^{(N)}(\mathbf{r}) = (2\pi)^{-3}\int k^{2N}f_a(k)y_{l_a}^{m_a}(\bar{\theta},\bar{\phi})f_b(k)y_{l_b}^{m_b}(\bar{\theta},\bar{\phi})\sum_{\lambda=0}^{\infty}(2\lambda+1)(-i)^{\lambda}j_{\lambda}(kr) \\ \times\sum_{\mu=0}^{\lambda}\epsilon_{\mu}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_{\lambda}^{|\mu|}(\cos\theta)P_{\lambda}^{|\mu|}(\cos\bar{\theta})\cos[|\mu|(\phi-\bar{\phi})]\,d\mathbf{k} \end{aligned} \tag{3.23}$$

Rearranging the sums and the integral, introducing the spherical coordinate system, and separating the radial and angular parts of the integral we get

$$\begin{aligned} I^{(N)}(\mathbf{r}) = (2\pi)^{-3}\sum_{\lambda=0}^{\infty}(2\lambda+1)(-i)^{\lambda} \\ \times\int\limits_0^\infty k^{2N+2}f_a(k)f_b(k)j_{\lambda}(kr)dk\sum_{\mu=0}^{\lambda}\epsilon_{\mu}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_{\lambda}^{|\mu|}(\cos\theta) \\ \times\int\limits_0^{2\pi}\int\limits_0^{\pi} y_{l_a}^{m_a}(\bar{\theta},\bar{\phi})y_{l_b}^{m_b}(\bar{\theta},\bar{\phi})P_{\lambda}^{|\mu|}(\cos\bar{\theta})\cos[|\mu|(\phi-\bar{\phi})]\sin\bar{\theta}\,d\bar{\theta}d\bar{\phi} \end{aligned} \tag{3.24}$$

Following Todd *et al.* (1970) and Silverstone (2014), we introduce the radial integral $J_{ab\lambda}^{(N)}(r)$:

$$J_{ab\lambda}^{(N)}(r) = \frac{i^{\lambda}}{2\pi^2}\int_0^{\infty} k^{2N+2} f_a(k) f_b(k) j_{\lambda}(kr)dk, \tag{3.25}$$

which simplifies the expression for the integral $I^{(N)}(\mathbf{r})$ to

$$\begin{aligned} I^{(N)}(\boldsymbol{r}) &= \sum_{\lambda=0}^{\infty} J_{ab\lambda}^{(N)}(r) \frac{(-1)^{\lambda}(2\lambda+1)}{4\pi} \sum_{\mu=0}^{\lambda} \epsilon_{\mu} \frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!} P_{\lambda}^{|\mu|}(\cos\theta) \\ &\times \int_0^{2\pi}\int_0^{\pi} y_{l_a}^{m_a}(\bar{\theta},\bar{\phi}) y_{l_b}^{m_b}(\bar{\theta},\bar{\phi}) P_{\lambda}^{|\mu|}(\cos\bar{\theta}) \cos[|\mu|(\phi-\bar{\phi})] \sin\bar{\theta}\, d\bar{\theta} d\bar{\phi} \end{aligned} \tag{3.26}$$

After defining the angular integral $d_{ab\lambda}(\theta,\phi)$ as

$$\begin{aligned} d_{ab\lambda}(\theta,\phi) &= \frac{(-1)^{\lambda}(2\lambda+1)}{4\pi} \sum_{\mu=0}^{\lambda} \epsilon_{\mu} \frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!} P_{\lambda}^{|\mu|}(\cos\theta) \\ &\times \int_0^{2\pi}\int_0^{\pi} y_{l_a}^{m_a}(\bar{\theta},\bar{\phi}) y_{l_b}^{m_b}(\bar{\theta},\bar{\phi}) P_{\lambda}^{|\mu|}(\cos\bar{\theta}) \cos[|\mu|(\phi-\bar{\phi})] \sin\bar{\theta}\, d\bar{\theta} d\bar{\phi} \end{aligned} \tag{3.27}$$

the expression for the integral $I^{(N)}(\mathbf{r})$ can be written simply as (Silverstone, 2014)

$$I^{(N)}(\mathbf{r}) = \sum_{\lambda=0}^{\infty} d_{ab\lambda}(\theta,\phi) J_{ab\lambda}^{(N)}(r) \tag{3.28}$$

Note an unfortunate similarity in the notation for the angular integral $d_{ab\lambda}(\theta,\phi)$ [equation (3.27) and below] and the density-normalized real spherical harmonic function $d_{l,m}(\theta,\phi)$ [equations (3.7) and (3.8)].

3.2.2 The angular integral $d_{ab\lambda}(\theta,\phi)$

In Section S3.3 of the supporting information (Appendix B) we show that the infinite sum in equation (3.28) is replaced by a finite sum

$$I^{(N)}(\mathbf{r}) = \sum_{\lambda=|l_a-l_b|}^{l_a+l_b} d_{ab\lambda}(\theta,\phi) J_{ab\lambda}^{(N)}(r) \tag{3.29}$$

where

$$\lambda \in \{l_a + l_b, l_a + l_b - 2, \dots, |l_a - l_b|\} \tag{3.30}$$

We also show that the angular integral $d_{ab\lambda}(\theta, \phi)$ can be written as either:

(*a*)

$$\begin{aligned} d_{ab\lambda}(\theta, \phi) = (-1)^{|m_a|+|m_b|+\lambda}(2\lambda + 1)\frac{1}{8\pi^2}\left[\frac{(2l_a + 1)}{(1 + \delta_{m_a,0})}\frac{(2l_b + 1)}{(1 + \delta_{m_b,0})}\right]^{1/2} \\ \times \sum_{\mu} \epsilon_\mu \left[\frac{(\lambda - |\mu|)!}{(\lambda + |\mu|)!}\right]^{1/2} P_\lambda^{|\mu|}(\cos\theta)\Phi_{m_a,m_b,|\mu|}(\phi)\Theta'_{l_a,m_a,l_b,m_b,\lambda,|\mu|} \end{aligned} \tag{3.31}$$

where

$$\mu \in \{\left||m_a| - |m_b|\right|, |m_a| + |m_b|\} \text{ and } \mu \le \lambda \tag{3.32}$$

$$\Phi_{m_a,m_b,|\mu|}(\phi) = \cos(|\mu|\phi)\,\Phi'_{m_a,m_b,|\mu|} + \sin(|\mu|\phi)\,\Phi'_{m_a,m_b,-|\mu|} \tag{3.33}$$

$$\Phi'_{m_a,m_b,m_c} = \int_0^{2\pi} \begin{bmatrix} \cos(|m_a|\bar{\phi}) \\ \sin(|m_a|\bar{\phi}) \\ 1 \end{bmatrix} \begin{bmatrix} \cos(|m_b|\bar{\phi}) \\ \sin(|m_b|\bar{\phi}) \\ 1 \end{bmatrix} \begin{bmatrix} \cos(|m_c|\bar{\phi}) \\ \sin(|m_c|\bar{\phi}) \\ 1 \end{bmatrix} d\bar{\phi} \tag{3.34}$$

and functions Φ'_{m_a,m_b,m_c} are related to the overlap integral over three associated Legendre polynomials (Gaunt, 1929; Racah, 1942; Slater, 1960) which in this study is evaluated as (Dong & Lemus, 2002; Section S3.3.2 of the supporting information), if $\mu = |m_a| + |m_b|$:

$$\Theta'_{l_a,m_a,l_b,m_b,\lambda,|\mu|} = 2(-1)^{|\mu|}\begin{pmatrix} l_a & l_b & \lambda \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} l_a & l_b & \lambda \\ |m_a| & |m_b| & -|\mu| \end{pmatrix} \tag{3.35}$$

while if $\mu = \left||m_a| - |m_b|\right|$:

$$\Theta'_{l_a,m_a,l_b,m_b,\lambda,|\mu|} = 2(-1)^{\xi-|m_a|+|m_b|}\begin{pmatrix} l_a & l_b & \lambda \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} l_a & l_b & \lambda \\ -|m_a| & |m_b| & |m_a| - |m_b| \end{pmatrix} \tag{3.36}$$

where quantities $\begin{pmatrix} a & b & c \\ \alpha & \beta & \gamma \end{pmatrix}$ are the 3-*j* symbols (Edmonds, 1957) and the parameter ξ is defined as (Dong & Lemus, 2002)

$$\xi = \begin{cases} |m_a| & \text{if } |m_b| \geq |m_a| \\ |m_b| & \text{if } |m_b| < |m_a| \end{cases} \tag{3.37}$$

Or (*b*)

$$\begin{aligned} d_{ab\lambda}(\theta, \phi) = (-1)^{\lambda} \sum_{\mu} \Big\{ & y_{\lambda}^{|\mu|}(\theta, \phi) RG_{l_a, m_a, l_b, m_b, \lambda, |\mu|} \\ & + y_{\lambda}^{-|\mu|}(\theta, \phi)\, RG_{l_a, m_a, l_b, m_b, \lambda, -|\mu|} \Big\} \end{aligned} \tag{3.38}$$

where the R-Gaunt coefficients $RG_{l_a, m_a, l_b, m_b, \lambda, |\mu|}$ and $RG_{l_a, m_a, l_b, m_b, \lambda, -|\mu|}$ are defined as an overlap integral over three real spherical harmonics (Homeier & Steinborn, 1996; Coppens 1997):

$$RG_{l_a, m_a, l_b, m_b, l_c, m_c} = \int_0^{2\pi} \int_0^{\pi} y_{l_a}^{m_a}(\bar{\theta}, \bar{\phi}) y_{l_b}^{m_b}(\bar{\theta}, \bar{\phi}) y_{l_c}^{m_c}(\bar{\theta}, \bar{\phi}) \sin\bar{\theta}\, d\bar{\theta} d\bar{\phi} \tag{3.39}$$

While equation (3.39) for $d_{ab\lambda}(\theta, \phi)$ is more aesthetically pleasing, equation (3.31) is more computationally efficient.

3.2.3 The radial integral $J_{ab\lambda}^{(N)}(r)$

The main challenge in calculation of the integral $I^{(N)}(\mathbf{r})$ is the radial integral $J_{ab\lambda}^{(N)}(r)$. As shown by Silverstone (2014), it can be very accurately evaluated via numerical contour integration using an arbitrary-precision arithmetic implemented in computer algebra software, such as *Mathematica* (Wolfram Research, 2018). Alternatively, Todd *et al.* (1970) obtained analytical formulas for $J_{ab\lambda}^{(N)}(r)$ in terms of several auxiliary functions including the exponential-type integral and definite integrals of spherical Bessel functions. We note that expression similar to $J_{ab\lambda}^{(N)}(r)$ also appears in the integral over a product of *B* functions (Filter and Steinborn, 1978) on different centers which was discussed in detail by Grotendorst & Steinborn (1985) and Weniger *et al.* (1986). More recently, analytical expressions for this type of integral were also given by Shestakov (1992) and Özay & Öztekin (2013).

In this work, however, we have chosen to evaluate integral $J_{ab\lambda}^{(N)}(r)$ using the recurrence procedure originally proposed by Geller (1964*a*,*b*), and later refined by Harris & Michels (1967). In Appendix A we show that the radial integral $J_{ab\lambda}^{(N)}(r)$ can be expressed as a finite sum of integrals $W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r)$ defined by Geller & Griffith (1964) and Geller (1964*a*,*b*), and discussed in detail by Harris & Michels (1967) and later by Mar'yashkin & Zimont (1975):

$$J_{ab\lambda}^{(N)}(r) = A \sum_{s=0}^{\left\lfloor \frac{1}{2}(n_a-l_a) \right\rfloor} \sum_{t=0}^{\left\lfloor \frac{1}{2}(n_b-l_b) \right\rfloor} B(s,t) W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r) \tag{3.40}$$

where $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a (*i.e.* the result of the floor function), and

$$A = \pm 4\pi 2^{l_a+l_b}(n_a-l_a)!\,(n_b-l_b)!\,{\zeta_a}^{n_a-l_a}{\zeta_b}^{n_b-l_b} \tag{3.41}$$

$$B(s,t) = (-1)^{s+t}\binom{n_a+l_a+1}{2s+2l_a+1}\binom{n_b+l_b+1}{2t+2l_b+1}\frac{(s+l_a)!\,(t+l_b)!}{s!\,t!\,{\zeta_a}^{2s}{\zeta_b}^{2t}} \tag{3.42}$$

$$W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r) = \frac{2}{\pi}\int_0^\infty \frac{k^{\lambda+2j} j_\lambda(kr)}{(\zeta_a^2+k^2)^i(\zeta_b^2+k^2)^{i'}}dk \tag{3.43}$$

where $i = n_a + 1$, $i' = n_b + 1$, and $j = (2N + 2 + l_a + l_b + 2s + 2t - \lambda)/2$, $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is the binomial coefficient, and the minus sign in equation (3.41) is applied when $(l_a + l_b + \lambda)/2 =$ odd. We note that our equation (3.40) is very similar (as it should be) to those given by Geller (1964*a*,*b*), Harris & Michels (1967), and Mar'yashkin & Zimont (1975).

Geller (1964*b*) derived expressions for a number of Coulomb integrals in terms of the auxiliary integrals $W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r)$, and gave basic recurrence relations for $W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r)$.

While Geller (1964*a*,*b*) advocated the use of numerical integration for evaluation of $W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r)$, which should in principle simplify the treatment of cases when $\zeta_a = \zeta_b$ and $\zeta_a \approx \zeta_b$, Harris & Michels (1967) and Mar'yashkin & Zimont (1975) offered simple and efficient recurrence schemes for evaluation of these auxiliary functions.

3.3 Implementations of the Fourier transform method

3.3.1 Fortran

3.3.1.1 General

In this study, we follow closely the recurrence-based method described by Harris & Michels (1967) who suggested the use of special treatments of the three distinct cases: $\zeta_a \neq \zeta_b$, $\zeta_a \approx \zeta_b$, and $\zeta_a = \zeta_b$. The only minor difference with respect to their work is that we use their equation (73) to advance λ instead of the suggested equation (75). We also note a small typo in their equation (80) – the last term should be $W_{i,0}^{-1,1}$.

The derived equations for evaluation of integrals $C(\mathbf{r})$, $S(\mathbf{r})$, $I^{(N)}(\mathbf{r})$, $J_{ab\lambda}^{(N)}(r)$, $d_{ab\lambda}(\theta,\phi)$,[4] Φ'_{m_a,m_b,m_c}, and $\Theta'_{l_a,m_a,l_b,m_b,\lambda,|\mu|}$, and the Harris & Michels (1967) recurrence scheme for evaluation of integrals $W_{i,i'}^{\lambda,j}(\zeta_a\zeta_b r)$ have been coded using a standard Fortran90 and incorporated in the in-house version of the *XDPROP* program (Volkov *et al.*, 2016).

The 3-j symbols $\begin{pmatrix} a & b & c \\ \alpha & \beta & \gamma \end{pmatrix}$ are evaluated as discussed in Thompson (1994, 1997), while for $\begin{pmatrix} a & b & c \\ 0 & 0 & 0 \end{pmatrix}$ we used a simplified formula given in Edmonds (1957; equation 3.7.17).

[4] In order to match results produced by the Löwdin α-function method in *XDPROP*, we had to apply the factor $(-1)^{l_b}$ to the angular integral $d_{ab\lambda}(\theta,\phi)$.

The modified spherical Bessel function of the second kind, $k_n(x)$, (Olver *et al.*, 2018),

$$k_n(x) = \sqrt{\frac{2}{\pi x}} K_{n+\frac{1}{2}}(x) \tag{3.44}$$

where $K_m(x)$ is the modified Bessel function of the second kind (Olver *et al.*, 2018), needed to calculate the starting values for integrals $W_{i,i'}^{\lambda,j}(\zeta_a \zeta_b r)$, is evaluated fast and reliably using the forward recurrence relations as discussed in Harris & Michels (1967; equations 163, 168 and 169).

For the $\zeta_a = \zeta_b$ case, the function $K_m(x)$ is evaluated using the forward recursion relations as given in Arfken (1985), Olver *et al.* (2018), and Abramowitz & Stegun (1972), though in the latter $K_m(x)$ is called the "modified spherical Bessel function of the *third* kind".

The associated Legendre polynomials in equation (3.31) are evaluated recursively using the approach presented in the work of Press *et al.* (1992; pp. 246-248). We note that the "plgndr" function described in Press *et al.* (1992) includes the Condon-Shortley phase (Condon & Shortley, 1959).

The factorials (that enter a number of functions) are precalculated up to the desired order at the same precision as the Coulomb integrals (see discussion below) at the start of the program, and are stored in computer memory throughout the program execution.

3.3.1.2 Coordinate system

The Löwdin α-function technique (Lα) as described in Paper I works only in the "local" coordinate system defined for a pair of centers/pseudoatoms a and b in such a way that the local coordinate axis Z is directed from center a to center b, with the other two local axes forming an orthogonal coordinate system. That is, the atom/center a is assigned a set of coordinates {0,0,0}, while the spherical coordinates of atom/center b are $\{r, \theta, \phi\} = \{r, 0{,}0\}$, where r is the distance between a and b. In the molecular integral literature such coordinate system is often referred to as “lined-up” - see for example, Steinborn & Ruedenberg (1972), Guseinov & Mamedov (1999) and Yakar *et al.* (2007).

Naturally, the spherical harmonics, or to be more precise, pseudoatom population parameters need to be transformed (rotated) from the global to the local coordinate system for each pair of atoms. In *XDPROP* this is done using the methods described by Cromer, *et al.* (1976) and Su & Coppens (1994, 1995). The advantage of working in the local coordinate system is that the Coulomb and overlap integrals vanish unless $m_a = m_b$ (Roothaan’s Theorems II and I, respectively; Roothaan, 1951).

The Fourier transform (FT) method can evaluate integrals in both the local (constructed as discussed above) and global coordinate systems. Indeed, with the origin of the coordinate system shifted to center a, the spherical coordinates of the center b become $\{r, \theta, \phi\}$, and the directional dependence of a Coulomb or overlap integral is accounted for by the angular function $d_{ab\lambda}(\theta, \phi)$. We note that in the global coordinate system Roothaan’s theorems I and II (Roothaan, 1951) no longer hold, which increases (often, significantly) the number of integrals. This also means that a direct comparison of the individual Coulomb integrals between Lα/Local and FT/Global is not possible. However, one can still compare the *total* Coulomb integrals between any two given pseudoatoms (as these are linear combinations of the individual Coulomb integrals) and of course the overall Coulomb integral and/or the total electrostatic interaction energy E_{es}.

When using the FT method in the local coordinate system, the expression for $d_{ab\lambda}(\theta,\phi)$ is greatly simplified because $\{r,\theta,\phi\} \rightarrow \{r,0,0\}$, and $m_a = m_b$, and one can directly compare the Lα and FT results for each individual Coulomb integral.

3.3.1.3 Numerical precision

In our previous study, the Fortran implementation of the Löwdin α-function technique (Lα) was programmed in *XDPROP* using the 64-bit fixed numerical precision (*aka* double precision, DP, and binary64) arithmetic which allows for representation of numbers in the range $\cong 10^{-308} \dots 10^{308}$ with a precision of $\cong 15-17$ digits (Oracle Corporation, 2017*b*). A typical Fortran90/95 declaration option for a 64-bit parameter is SELECTED_REAL_KIND(15, 307) which requests 15 significant digits and the exponent range at least $-307 \dots 307$ (Lemmon & Schafer, 2005). Pretty much all modern 64-bit computing processors (*i.e.*, those that are based on one of the following architectures: x86_64, IA-64 or AArch64) make use of the hardware acceleration when working with real-valued numbers. The 64-bit numerical format is supported by essentially all modern Fortran compilers.

The double-extended-precision (Oracle Corporation, 2017*b*) or simply the extended-precision (EP) floating point number format available for some processors, including those based on the x86 architecture, uses 80 bits to represent numbers which increases both the number of significant digits ($\cong 18-21$) and the exponent range ($\cong -4932 \dots 4932$) (Oracle Corporation, 2017*b*) albeit at a higher computational cost. To the best of our knowledge, the only two modern Fortran compilers that support the 80-bit precision are GNU Fortran (GFortran; Free Software Foundation, 2018) and DragonEgg (2018), the latter actually being not a standalone compiler but rather a plugin for GFortran.

In these compilers, the extended precision can be requested using the SELECTED_REAL_KIND(p = 18) option.

In the future, the 128-bit computer architecture will allow for hardware-accelerated mathematical operations with real-valued numbers in the quadruple/quad precision (QP, also called binary128) representation. This format extends the number of significant digits to $\cong 33 \dots 36$ with the exponent range matching that of the EP format ($\cong -4932 \dots 4932$) (Oracle Corporation, 2017*b*). Currently, however, only few Fortran compilers support QP for the x86_64 architecture, and those that do, rely on the software implementation of the 128-bit floating-point mathematical operations. The QP format in Fortran can be requested using the SELECTED_REAL_KIND(33, 4391) option.

The current version of *XDPROP* can perform the Lα and FT calculations in any precision listed above, though the desired precision is selected during the program compilation, and thus can not be changed dynamically (to switch to a different precision, the code needs to be recompiled).

3.3.1.4 Fortran compilers and the computer hardware

The Fortran implementations of the Lα and FT methods have been tested on an x86_64 computer platform. The code has been compiled and run on a 64-bit OpenSUSE Leap 15.0 Linux computer equipped with a 2012 4.0 GHz AMD FX-8350 processor (CPU) with 4 × "Piledriver" modules (making it more of 4-core/8-thread CPU), and 32 GB of DDR3 memory running at 1600 MHz. The single-thread Passmark score (PassMark Software, 2018) of this processor (1509) is only slightly higher than that of the AMD Opteron 6348 CPU (1266) used in the previous study. For comparison, the recently released 3.6 GHz Intel Core i9-9900K CPU that can reach 5 GHz in the so-called “Turbo Mode” has the Passmark score of 2912 (PassMark Software, 2018).

In the following we list Fortran compilers used to compile the code (we note that the code was generated and run on the same machine) and the corresponding compiler optimization options:

(i) Lahey/Fujitsu Linux64 (LF64) Fortran Express compiler (Lahey Computer Systems, 2011), version L8.10a; we used the recommended set of optimization flags: `-nap -nchk -ng -fast -npca -nsav -ntrace`; this is the only commercial compiler used in this study.

(ii) Oracle Developer Studio 12.6 2017/05/30 (Oracle Corporation, 2017*a*); optimization flags: `-fstore -O1`; higher optimization options (even usually safe `-O2`) produced unacceptable numerical results.

(iii) GNU Fortran 8.2.1 2018-08-31 (Free Software Foundation, 2018); optimization flags: `-O3 -march=bdver2`; higher optimization options (such as `-Ofast`) produced inadequate numerical results.

(iv) AMD Optimizing Compiler C/C++ (AOCC) (Advanced Micro Devices, 2018) with the DragonEgg (2018) Fortran compiler plugin that uses code optimizers from the LLVM project (Lattner & Adve, 2004); optimization flags: `-Ofast -march=bdver2`.

The Löwdin α-function calculations were performed using the 64-bit (DP), 80-bit (EP) and 128-bit (QP) precision formats, while the Fourier transform benchmarks were done in the 128-bit precision (QP) only as results from the preliminary FT calculations in DP and EP were judged to be too imprecise.

3.3.2 Mathematica

The numerical performance of the Löwdin α-function and Fourier transform methods in *XDPROP* as a function of the Fortran compiler and numerical precision used, was evaluated by comparing values of the individual Coulomb integrals (between each pair of Slater functions) with those from the numerical contour integration performed in *Mathematica* (Wolfram Research, Inc., 2018; Silverstone, 2014). The *Mathematica* program described in section 4.2 of Paper I was modified to work in both the local and global coordinate systems.

In all benchmark calculations, the `WorkingPrecision` option in our *Mathematica* code was set at 90, which made the program maintain 90 digits in all internal computations, and the `AccuracyGoal` option was set at 80, which requested 80 digits in the final numerically evaluated integrals. Since the Fortran quad precision is limited, at best, to 36 digits, requesting 80 digits in *Mathematica* may seem excessive (it also significantly slowed down our calculations). However, this approach ensured that every Fortran digit was verified correctly.

3.4 Benchmark systems

In this study we used the same benchmark systems as in Paper I. In Section 3.5 we focus on the individual integrals in the Gly1 dimer calculated in both the local (Lα and FT) and global (FT only) coordinate systems (Table 3.1). Results for the remaining dimers are analysed in Section 3.6 in terms of the total E_{es} values and speed only (without the comparison of individual Coulomb integrals).

Table 3.1 Number of individual integrals in the Gly1 dimer.

The internuclear distance range is 1.75 … 9.57 Å.

Number of individual Coulomb integrals	Coordinate System	
	Local	Global
total	89 785	180 625
core/valence density ↔ core/valence density	70 225	70 225
core/valence density ↔ deformation density	15 900	84 800
deformation density ↔ deformation density	3 750	25 600

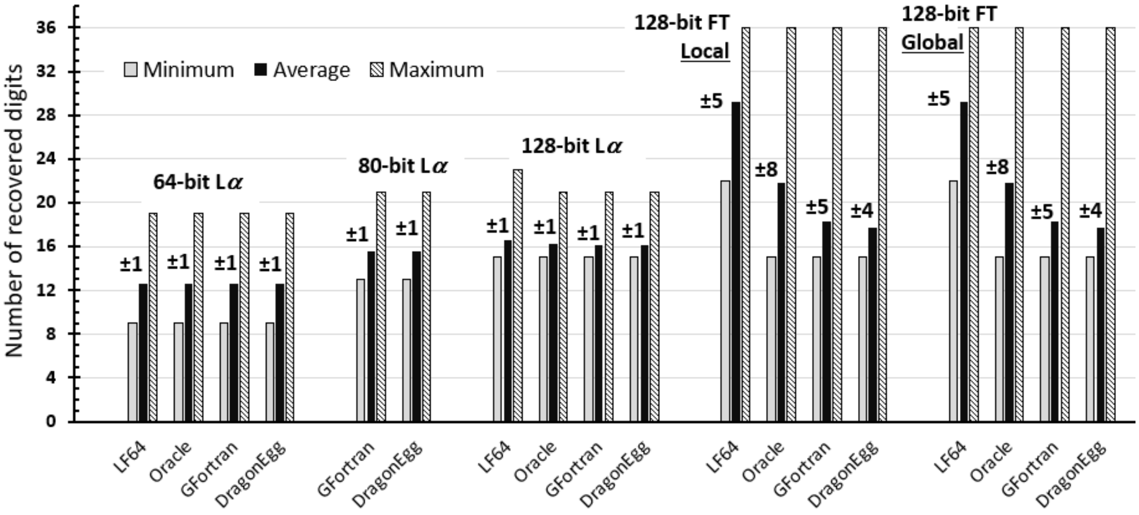
3.5 Precision and speed of the Löwdin α-function and Fourier transform methods for the Gly1 dimer

Since numerical results of our Fortran implementations of the Lα and FT methods appear to be affected by the Fortran compiler (and the compiler optimization flags) used, we shall consider the two phenomena (method and compiler) together at some expense of clarity and structure of the presentation. In Figure 3.1, we summarize the numerical precision results obtained for the Gly1 dimer.

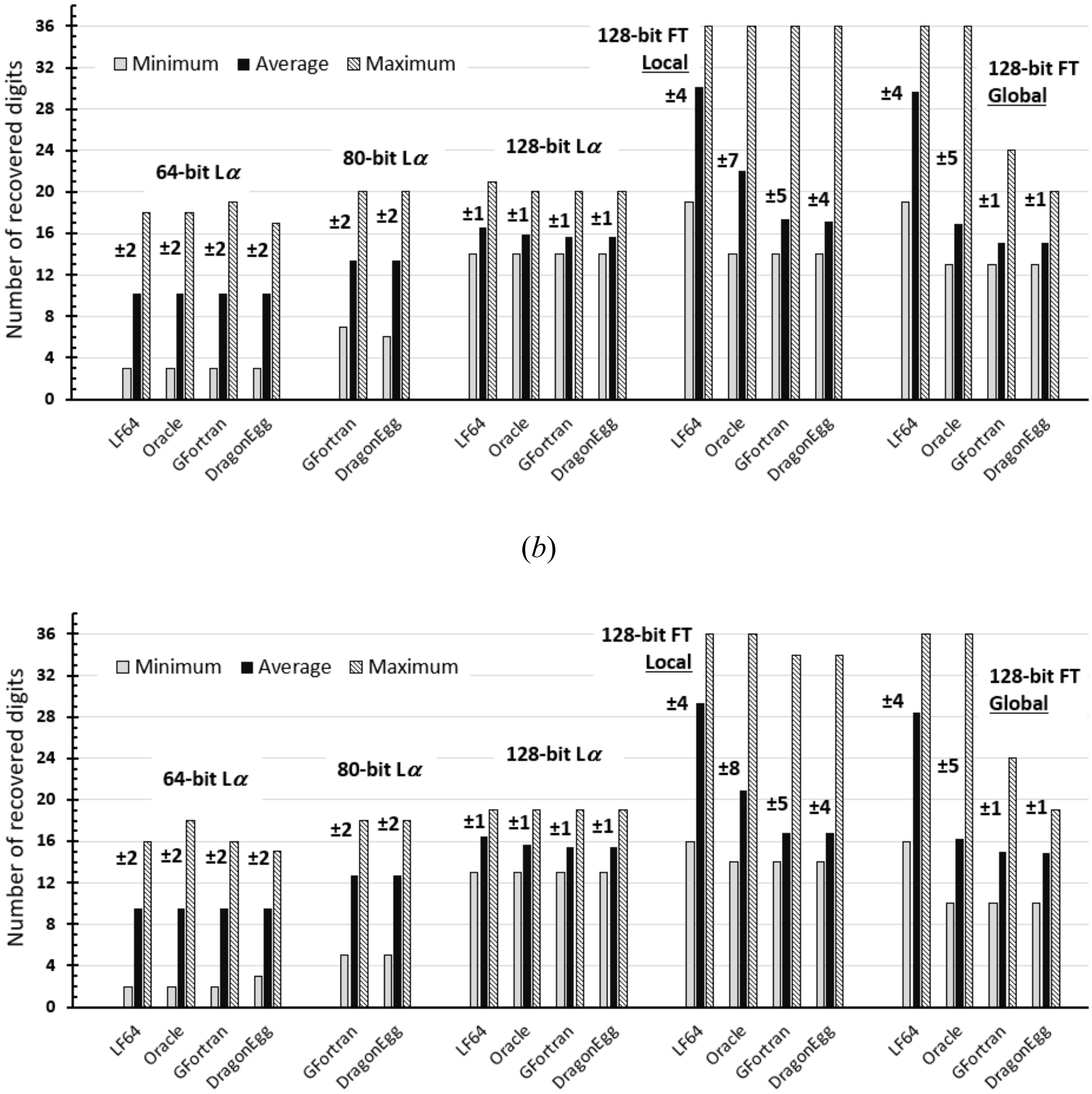
3.5.1 Precision of the Fortran implementation of the Löwdin α-function method

Figure 3.1 shows that the 64-bit precision (DP) Lα runs for all tested compilers produce essentially identical numerical results. This is to be expected as the 64-bit floating point operations are hardware-accelerated, and no unsafe code optimizations were used when compiling the code. That said, a very slight drop in precision from the DragonEgg compiler runs can be explained by the use of the `-Ofast` optimization option. According to the AOCC DragonEgg manual (Advanced Micro Devices, 2018), the `-Ofast` option enables "*aggressive optimizations that may violate strict compliance with language* *standards*", but does not enable automatically the `-fast-math` option which turns on additional unsafe optimizations that may lead to a significant loss of precision. However, in GFortran the `-Ofast` option does turn on such optimizations which explains why the `-Ofast` option in our preliminary GFortran calculations produced results with inadequate precision, and thus had to be excluded from this analysis.

The comparison of the 64-bit Lα-based runs for different groups of integrals clearly shows the loss of precision for integrals involving functions with $l > 0$, something that was also pointed out in Paper I. On the average, two digits are lost in the core/valence ↔ deformation density group of integrals as compared to the core/valence ↔ core/valence integrals, and one more digit is lost in the deformation ↔ deformation density integral group. This is especially evident when considering the minimum number of digits recovered in these three integral groups: 9, 3 and 2, respectively.



(*a*)



(*b*)

(*c*)

Figure 3.1 The minimum, maximum, and average number of digits for all individual (*a*) core/valence ↔ core/valence density, (*b*) core/valence ↔ deformation density, and (*c*) deformation ↔ deformation density integrals in the Gly1 dimer recovered using each implementation. The ±x symbol denotes the standard deviation of the average. The Löwdin α-function (Lα) calculations were performed in the local coordinate system only, while the Fourier transform (FT) calculations were done in both the local and global coordinate systems.

Results of the Lα calculations improve when using the 80-bit (EP) precision, though only two out of four compilers included in this study support this numerical format. The 80-bit calculations recover on average three more digits in each integral category as compared with the 64-bit results, albeit at some computational overhead (see Section 3.5.3 and Figure 3.2). This is especially evident in the two groups of integrals with $l > 0$: the minimum number of digits improves from three to six-seven for the core/valence ↔ deformation density integrals, and from two-three to five for the deformation ↔ deformation density integral group. It is also satisfying to see that the GFortran and DragonEgg values are essentially indistinguishable despite somewhat different levels of the code optimization.

Switching to the 128-bit precision (QP) allows the Lα method to recover on average two-four more digits, and drastically improve the minimum number of reproducible digits (by up to eight!) in the two categories of integrals with $l > 0$. Indeed, the minimum number of the recovered digits in the 128-bit runs as compared to the 80-bit calculations increases from 6-7 to 14 in the core/valence ↔ deformation density group of integrals, and from 5 to 13 in the deformation ↔ deformation density integral group. For these integrals, the Lahey LF64 compiler recovers approximately one more digit when compared with the other three compilers.

That said, even the most precise integrals evaluated using the 128-bit Lα method do not reach the desired precision level of 30 digits (recall that the binary128 format extends the numerical precision to $\cong 33 \dots 36$ digits): the maxim number of digits recovered is $\cong 23$.

The overall picture for our implementation of the Lα method is however rather gruesome as in order to calculate the individual Coulomb integrals at the 64-bit level of precision (≅ 15 digits), the calculations must be performed in the 128-bit precision which (as discussed in section 3.5.3) incurs a severe performance penalty.

3.5.2 Precision of the Fortran implementation of the Löwdin α-function method

We have not included in the paper the results of the 64-bit and 80-bit runs for the FT method. As mentioned above, the precision was inadequate regardless of the compiler and the code optimization options used. It is not unexpected. Our implementation of the FT method relies heavily on recursions especially when constructing $W_{i,i'}^{0,0}$ and $W_{i,i'}^{1,0}$ [equation (74) in Harris & Michels, 1967],

$$(\zeta_b^2 - \zeta_a^2)W_{i,i'}^{\lambda,j} = W_{i,i'-1}^{\lambda,j} - W_{i-1,i'}^{\lambda,j} \tag{3.45}$$

and advancing λ and j to generate the final set of $W_{i,i'}^{\lambda,j}$ integrals [equations (71)-(73) in Harris & Michels, 1967]:

$$\zeta_a^2 W_{i,i'}^{\lambda,j} = W_{i-1,i'}^{\lambda,j} - W_{i,i'}^{\lambda,j+1} \tag{3.46}$$

$$\zeta_b^2 W_{i,i'}^{\lambda,j} = W_{i,i'-1}^{\lambda,j} - W_{i,i'}^{\lambda,j+1} \tag{3.47}$$

$$\left(\frac{2\lambda+1}{r}\right) W_{i,i'}^{\lambda,j} = W_{i,i'}^{\lambda+1,j} + W_{i,i'}^{\lambda-1,j+1} \tag{3.48}$$

While the case when ζ_a/ζ_b is near unity is treated separately as described in Harris & Michels (1967), a large value of $(\zeta_b^2 - \zeta_a^2)$ may introduce numerical instability in equation (3.45). By the same token, when $\zeta_{a/b}^2$ is large, equations (3.46), (3.47) may be subject to the loss of precision. Finally, for large values of r, equation (3.48) will likely result in the loss of digits as well. This phenomenon, called the "*digital erosion*", was described in detail by Barnett (2002). The challenges of high-precision computations were also discussed in Bailey & Borwein (2015). The computer algebra software that employs the arbitrary-precision arithmetic, such as *Mathematica*, do not suffer from this setback as long as the appropriate working precision is requested. However, the speed with which such calculations are performed is usually inadequate even when using the modern computer hardware. That said, libraries for multiple- and arbitrary-precision floating-point computations with correct rounding are now available for programming languages such as C/C++ and Fortran, for example: GNU MPFR (Fousse *et al*., 2007), GNU MPC (Enge *et al*, 2018), GNU GMP (Granlund, 2016), ARPREC (Bailey *et al.*, 2002), MPFUN2015 (Bailey, 2015), FMLIB (Smith, 1991) *etc*. The work to explore these options is in progress.

The last two sets of columns in Figure 3.1(a) – 3.1(c) show the results for our Fourier transform recurrence-based calculations performed using the 128-bit (QP) arithmetic in both the local and global coordinate systems.

The compiler effect is much more significant for the 128-bit FT method than for the 128-bit Lα computations. The Lahey/Fujitsu 64-bit compiler is able to recover significantly more digits than its nearest competitor, Oracle Studio 12.6, despite being much faster (recall that we had to lower the optimization options in the Oracle Studio compiler to `-O1` as any higher level of optimization lead to inadequate numerical results). The GFortran and DragonEgg compilers trail the pack in terms of precision but lead in terms of speed (see Section 3.6). Lowering the code optimization level and enabling additional options to improve the IEEE compliance, unfortunately, did not improve the precision yet significantly lowered the performance.

The Lahey/Fujitsu LF64 compiler recovered on average 29-30 digits in all three integral groups in both the local and global coordinate systems with a minimum of 16 reproducible digits in the deformation ↔ deformation density integral group. Its performance, especially for the core/valence ↔ core/valence integrals, is truly amazing: an average of 29 digits with a minimum of 22 digits. While the minimum number of reproducible digits drops to 19 for the core/valence ↔ deformation density set of integrals, and further to 16 digits for the deformation ↔ deformation density integral group, the average number of recovered digits is consistently around 29-30.

We also note that the Fourier transform method with the LF64 compiler gives by far the most precise integrals regardless of whether the calculation is performed in the local or global frame. For the other three compilers, the situation is not as clear. In fact, while in the local frame the remaining three FT-based runs are ahead of Lα in terms of precision, the situation flips for the integrals with $l > 0$ when FT calculations are performed in the global frame, especially when using the GFortran and DragonEgg compilers. Currently, we do not have a reasonable explanation for this behaviour.

In general, one should not be too much surprised by the numerical differences between compilers as the 128-bit floating point operations on a 64-bit platform are done primarily in software rather than in hardware [this is why some Fortran compilers, such as PGI (2018) and Open64 (Advanced Micro Devices, 2012), do not even support the binary128 calculations]. When performing the 128-bit mathematical operations, compilers use different software libraries so the differences in the numerical results are to be expected (albeit not to the degree that we have found).

To conclude this section, we note that while the precision of the implemented Lα method does improve significantly when switching to the 80-bit and especially 128-bit precision, the recurrence schemes in the Fourier transform method benefits enormously from the QP mathematical libraries included in the Lahey/Fujitsu LF64 compiler.

3.5.3 The computational efficiency of the 64-, 80- and 128-bit calculations

As discussed in the previous section, the use of the 80- and 128-bit arithmetic significantly improves precision of the results, but the computational overhead may be too severe since these calculations are not fully hardware-accelerated. In Figure 3.2, we show the so-called "user" time (in seconds), *i.e.* the time spent exclusively by the code, for calculation of E_{es} in the Gly1 dimer as a function of the method, numerical precision, and compiler used.

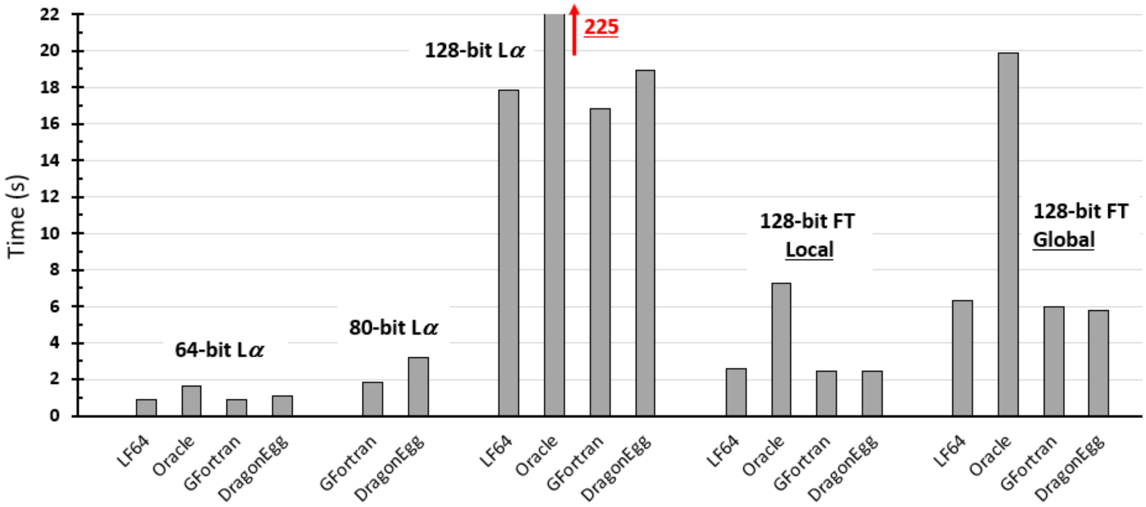


Figure 3.2 Time (s) used by the selected implementations (the "user" time) to calculate E_{es} in the Gly1 dimer. Each plotted time is an average over five runs. The Löwdin α-function (Lα) calculations were performed in the local coordinate system only, while the Fourier transform (FT) calculations were done in both the local and global coordinate systems.

As expected, the 64-bit Lα method undisputedly dominates the performance picture with both the Lahey LF64 and GFortran compilers producing slightly faster executables than the DragonEgg compiler, and being twice as fast as that produced by the Oracle Studio Fortran compiler. This is to be expected as we had to use a very low code optimization level (`-O1`) in the Oracle Studio compiler as all higher optimization levels produced inadequate numerical results.

Switching the Lα method to the extended 80-bit number format improves numerical precision of the results (see Section 3.5.1) at the additional computational cost. The GFortran executable in the 80-bit precision is about 1 s (≅ 100%) slower than its 64-bit counterpart, while the performance drop for the DragonEgg compiler is more significant (≅ 2 s, ≅ 300%).

The Lα method suffers a much more severe performance penalty when coded in the 128-bit precision. The user time increases to 18-19 s for three out of four compilers with GFortran being ahead of LF64 and DragonEgg by about 1 and 2 s, respectively. This is not unexpected. The Lα method relies on the continuous evaluation of the exponential function, which is hardware-accelerated in the 64-bit precision but likely done primarily in software when using the 128-bit precision, which naturally leads to a significant performance drop. The unreasonably long execution time for the Oracle Studio-based executable is undoubtedly the result of using the most basic optimization level (recall that using the optimization level beyond `-O1` in the Oracle Studio compiler produced inadequate numerical results).

The most interesting feature of Figure 3.2 is, of course, the performance of the 128-bit recurrence-based FT method. The user time for the local coordinate system-based FT code compiled with LF64, GFortran and DragonEgg is around 2.4-2.6 s, which makes it very competitive with both the 64- and 80-bit implementations of the Lα method (0.9-1.1 and 1.9-3.2 s, respectively). It is also pleasing to see the Oracle Studio Fortran compiler-based 128-bit FT executable being not too far in terms of performance from the other three compilers, which was a serious issue with the 128-bit Lα method.

As expected, the computational efficiency of the FT method deteriorates when switching to the global coordinate system. Indeed Table 3.1 shows that the total number of integrals in the Gly1 dimer in the global frame is essentially twice that in the local frame. However, this overall picture is somewhat deceiving. A closer inspection of Table 3.1 reveals that while the number of the core/valence ↔ core/valence integrals stays the same, recall that both the core and valence functions in the pseudoatom model (Stewart,1976; Hansen & Coppens, 1978; Coppens, 1997) are spherical and thus direction-independent, the number of integrals in the core/valence ↔ deformation density integral group increases by a factor of 5 (15 900 → 84 800), while the number of the deformation ↔ deformation density integrals increases by a factor of ≅ 7 (3750 → 25 600). Since the integrals with $l > 0$ are more computationally demanding than those with $l = 0$, the slowdown of the FT code in the global frame is quite expected. That said, the calculation of ~181 000 integrals (in the global coordinate system) in just over 6 s is still an impressive performance.

The fact that in the 128-bit precision the Fourier transform code is much faster than the Lα method is not surprising. Our FT implementation relies heavily on recursion, which includes simple, not too computationally demanding mathematical operations such as addition/subtraction and multiplication/division, as compared to the continuous uses of the exponentiation in the Lα method. This opens additional possibilities to improve (if necessary) numerical results of the recurrence-based Fourier transform implementation when switching to the so-called “quad-double” (“octupole”; 256-bit) precision, implemented for example in the *QD* package (Hida *et al.*, 2012), or employing any of the arbitrary-precision libraries (Fousse *et al.*, 2007; Enge *et al.*, 2018; Granlund, 2016); Bailey *et al.*, 2002; Bailey, 2015; Smith, 1991), all without a significant performance penalty. These options are currently being explored.

3.6 Performance of the Löwdin α-function and Fourier transform-based aEP/MM methods for all benchmark systems

In this section, we discuss the performance, in terms of both precision and speed, of the hybrid analytical exact potential/multipole moment method (aEP/MM) in which the analytical exact potential part (aEP) is calculated using the selected LF64- and GFortran-based implementations of the Löwdin α-function (64/80/128-bit) and Fourier transform (128-bit only) techniques. We have excluded the results from the Oracle Studio and DragonEgg compilers as these do not seem to offer any advantage over LF64 and GFortran. The resulting intermolecular electrostatic interaction energies are listed (in kJ/mol) in Table 3.2, while the program (“user”) times are given (in seconds) in Table 3.3.

Table 3.2 Total intermolecular electrostatic interaction energies E_{es} (kJ/mol) evaluated using the aEP/MM method (R_{cutoff} = 5 Å) with selected implementations for calculation of the two-center Coulomb integrals.

Differences are highlighted in **bold**. The Löwdin α-function (Lα) calculations were performed in the local coordinate system only, while the Fourier Transform (FT) calculations were done in both the local and global coordinate systems.

Dimers	64-bit		80-bit	128-bit			
	Lα	Lα	Lα	Lα	Lα	FT/Local	FT/Global
	LF64	GFortran	GFortran	LF64	GFortran	LF64	LF64
Gly1	-115.7709**1228**	-115.7709**0193**	-115.7709**0182**	-115.7709**1238**	-115.7709**0182**	-115.7709**1238**	-115.7709**0917**
Gly2	-29.0698**3082**	-29.0698**2446**	-29.0698**2435**	-29.0698**3075**	-29.0698**2435**	-29.0698**3075**	-29.0698**4486**
Gly3	-87.3811**7315**	-87.3811**7646**	-87.3811**7647**	-87.3811**7323**	-87.3811**7647**	-87.3811**7323**	-87.3811**6890**
Gly4	-163.6679**2266**	-163.6679**0882**	-163.6679**0876**	-163.6679**2274**	-163.6679**0876**	-163.6679**2274**	-163.6678**8523**
Gly5	46.5180**4121**	46.5180**3817**	46.5180**3814**	46.5180**4126**	46.5180**3814**	46.5180**4126**	46.5180**5258**
Gly6	-23.2447**5935**	-23.2447**6034**	-23.2447**6048**	-23.2447**5951**	-23.2447**6048**	-23.2447**5951**	-23.2447**5874**
Lenk1	-312.862**81532**	-312.862**81289**	-312.862**81288**	-312.862**81531**	-312.862**81288**	-312.862**81531**	-312.862**78574**
Lenk2	-156.7309**5907**	-156.7309**6574**	-156.7309**6608**	-156.7309**5903**	-156.7309**6608**	-156.7309**5903**	-156.7309**8223**
Lenk3	-15.5899**8727**	-15.5899**8273**	-15.5899**8279**	-15.5899**8729**	-15.5899**8279**	-15.5899**8729**	-15.5899**3531**
Lenk4	-64.1962**2869**	-64.1962**2967**	-64.1962**2978**	-64.1962**2871**	-64.1962**2978**	-64.1962**2871**	-64.1962**3783**
Lenk5	-62.2858**6009**	-62.2858**5929**	-62.2858**5928**	-62.2858**6009**	-62.2858**5928**	-62.2858**6009**	-62.2858**6689**
Lenk6	-44.5656**4595**	-44.5656**4482**	-44.5656**4484**	-44.5656**4591**	-44.5656**4484**	-44.5656**4591**	-44.5656**5326**
Lenk7	-53.8027**8235**	-53.8027**7910**	-53.8027**7910**	-53.8027**8235**	-53.8027**7910**	-53.8027**8235**	-53.8027**8772**
Lenk8	-35.3689**3957**	-35.3689**4041**	-35.3689**4039**	-35.3689**3957**	-35.3689**4039**	-35.3689**3957**	-35.3689**4892**
Lenk9	-23.7565**7922**	-23.7565**7899**	-23.7565**7907**	-23.7565**7943**	-23.7565**7907**	-23.7565**7943**	-23.7565**6525**
Lenk10	-29.9161**7940**	-29.9161**7957**	-29.9161**7939**	-29.9161**7929**	-29.9161**7939**	-29.9161**7929**	-29.9161**7811**
Lenk11	-42.8526**2905**	-42.8526**2892**	-42.8526**2892**	-42.8526**2906**	-42.8526**2892**	-42.8526**2906**	-42.8526**3293**
Nonapep1	-134.887**27400**	-134.887**27655**	-134.887**27677**	-134.887**27408**	-134.887**27677**	-134.887**27408**	-134.887**33407**
Nonapep2	-11.465**69483**	-11.465**69597**	-11.465**69601**	-11.465**69491**	-11.465**69601**	-11.465**69491**	-11.465**70153**
Nonapep3	-40.13266**263**	-40.13266**291**	-40.13266**311**	-40.13266**278**	-40.13266**311**	-40.13266**278**	-40.13266**271**
Decapep1	-198.1320**3554**	-198.1320**4301**	-198.1320**4307**	-198.1320**3528**	-198.1320**4307**	-198.1320**3528**	-198.1320**5315**
Decapep2	-58.6551**3507**	-58.6551**3590**	-58.6551**3594**	-58.6551**3515**	-58.6551**3594**	-58.6551**3515**	-58.6551**5725**
Decapep3	-29.3274**5874**	-29.3274**6473**	-29.3274**6491**	-29.3274**5910**	-29.3274**6491**	-29.3274**5910**	-29.3274**6355**
Decapep4	-44.7324**3177**	-44.7324**3193**	-44.7324**3165**	-44.7324**3112**	-44.7324**3165**	-44.7324**3112**	-44.7324**4485**
Decapep5	-12.4321**0805**	-12.4321**1523**	-12.4321**1517**	-12.4321**0794**	-12.4321**1517**	-12.4321**0794**	-12.4321**0480**
Dodecapep1	-202.5720**2949**	-202.5720**2899**	-202.5720**2862**	-202.5720**2953**	-202.5720**2862**	-202.5720**2953**	-202.5720**8202**
Dodecapep2	-14.53178**441**	-14.53178**332**	-14.53178**338**	-14.53178**437**	-14.53178**338**	-14.53178**437**	-14.53178**834**
Dodecapep3	-12.96945**881**	-12.96946**209**	-12.96946**204**	-12.96945**865**	-12.96946**204**	-12.96945**865**	-12.96947**647**

Table 3.3 Time (s) used (the "user" time) to evaluate E_{es} for each benchmark dimer via the aEP/MM method (R_{cutoff} = 5 Å) with the selected implementations for calculation of the two-center Coulomb integrals.

The times listed are the average over five runs. Times above 10 s are shown in the **bold**. The Löwdin α-function (Lα) calculations were performed in the local coordinate system only, while the Fourier Transform (FT) calculations were done in both the local and global coordinate systems.

Dimers	64-bit		80-bit	128-bit			
	Lα	Lα	Lα	Lα	Lα	FT/Local	FT/Global
	LF64	GFortran	GFortran	LF64	GFortran	LF64	LF64
Gly1	0.33	0.30	0.62	5.9	5.5	0.84	2.1
Gly2	0.36	0.33	0.69	6.6	6.2	0.97	2.3
Gly3	0.55	0.52	1.1	10	9.6	1.5	3.7
Gly4	0.84	0.82	1.7	**17**	**16**	2.4	5.8
Gly5	0.64	0.61	1.3	**13**	**12**	1.8	4.3
Gly6	0.36	0.34	0.69	6.7	6.2	0.96	2.3
Lenk1	3.3	3.3	6.8	**65**	**61**	9.3	**24**
Lenk2	1.6	1.5	3.1	**29**	**27**	4.3	10
Lenk3	1.6	1.6	3.2	**30**	**28**	4.5	**11**
Lenk4	0.29	0.27	0.51	4.4	4.2	0.71	1.7
Lenk5	0.20	0.19	0.32	2.6	2.4	0.45	1.1
Lenk6	0.31	0.28	0.54	4.7	4.4	0.73	1.7
Lenk7	0.24	0.22	0.40	3.3	3.1	0.53	1.3
Lenk8	0.23	0.20	0.38	3.1	2.9	0.53	1.3
Lenk9	0.61	0.57	1.2	**11**	10	1.6	4.1
Lenk10	0.33	0.32	0.60	5.4	5.1	0.84	2.1
Lenk11	0.11	0.09	0.13	0.92	0.83	0.21	0.40
Nonapep1	2.9	3.0	5.8	**52**	**49**	7.8	**20**
Nonapep2	1.7	1.8	3.2	**27**	**25**	4.2	10
Nonapep3	1.2	1.2	2.4	**22**	**22**	3.5	7.9
Decapep1	3.1	3.2	6.2	**56**	**53**	8.6	**20**
Decapep2	0.47	0.43	0.85	7.8	7.3	1.2	3.0
Decapep3	0.83	0.81	1.7	**16**	**15**	2.4	5.7
Decapep4	0.72	0.69	1.3	**12**	**12**	1.9	4.6
Decapep5	0.28	0.24	0.44	3.7	3.4	0.58	1.4
Dodecapep1	3.1	3.3	6.0	**50**	**47**	7.9	**18**
Dodecapep2	1.0	1.1	2.1	**18**	**19**	3.1	6.0
Dodecapep3	1.2	1.1	2.4	**21**	**21**	3.5	7.8

3.6.1 Accuracy and precision

The inspection of the E_{es} energies listed in Table 3.2 shows that the overall agreement is at least 5×10^{-5} kJ/mol, regardless of the method, numerical precision, and compiler used.

For a given method, for example Lα, the results obtained with the same compiler (for example, GFortran) agree within 10^{-7} kJ/mol or better for all tested numerical precisions. Equally satisfying is the perfect (within the printed precision of 10^{-8} kJ/mol) agreement between the 128-bit Lα and FT/Local results obtained with the LF64 compiler.

Based on the extended analysis of individual integrals in the Gly1 dimer in *Mathematica* (see Section 3.5), the most precise values should be provided by the 128-bit precision Fourier transform method. The reason for the small differences ($\leq10^{-5}$ kJ/mol) between the FT-based E_{es} energies calculated in the local and global coordinate systems is almost certainly due to the fact that the rotation of multipoles is done in *XDPROP* using the 64-bit precision, which inevitably leads to some small changes introduced in the multipolar population parameters for each pair of atoms as the functions are rotated from the global to the local frame. In principle, this inconsistency can be removed by performing the rotation in higher precision (for example, 80- or 128-bit), but this would require additional conversions between numerical formats which will likely slow down the code. There are no plans to modify *XDPROP* to include such changes as we consider the precision of the current results to be more than satisfactory.

While the FT-based E_{es} values calculated in the global frame with the 128-bit precision should be considered as the target/benchmark values, the results from all calculations in the local frame listed in Table 3.2 are more than acceptable as they agree with the benchmark values within 5×10^{-5} kJ/mol (usually, much better). Thus, the recommended algorithm (out of those listed in Table 3.2) should be chosen based on computational efficiency, which is discussed in the next section.

3.6.2 Speed

Table 3.3 gives the program (“user”) time (in seconds) for each method listed in Table 3.2. First, we note that the longest calculation (Enk1, Lα, 128-bit, Lahey LF64 compiler) took only 65 s. However, there is no reason to use the 128-bit Lα calculations as the resulting E_{es} values are essentially indistinguishable from those performed using the extended (80-bit) precision (see Table 3.2), which are *significantly* faster (approximately by a factor of 10). Indeed, the *slowest* 80-bit Lα-based calculation (Enk1) took only 6.8 s.

That said, the 128-bit precision Fourier transform-based method seems to offer the best combination of (*a*) flexibility (it can work in both the global and local coordinate systems), (*b*) precision (it outperforms Lα in determining individual Coulomb integrals, though the level of precision it provides may seem “unnecessary” for the current purpose), and (*c*) computational efficiency (while it somewhat trails the 64- and 80-bit Lα calculations in terms of speed, it is still able to complete the most time-consuming calculation under 10 s).

We also note that the FT method has much greater potential if precision beyond 128-bit, for example, the “quadruple-double” (256-bit), is desired. As discussed in Section 3.5, the computational overhead for the recurrence-based implementation of the FT method when switching to a higher precision should be significantly lower than that for the Lα method. This is because the former relies on recursion making use of the simple mathematical operations such as addition/subtraction and multiplication/division, while the latter requires continuous evaluation of a very computationally costly exponential function.

Thus, for the *XDPROP* users who desire the higher speed we recommend the 80-bit precision-based Löwdin α-function method, while those users who value precision and accuracy over speed will find the 128-bit Fourier transform technique more suitable for their needs. The work is currently underway to have *XDPROP* use either of the two methods without recompiling the code.

3.7 Summary and concluding remarks

The formulation of the FT method for evaluation of the two-center Coulomb integrals over STFs as proposed by Silverstone (1966), Geller (1963*a*,*b*, 1964*a*,*b*) and Harris & Michels (1967) has been applied to calculation of intermolecular electrostatic interaction energies in molecular crystals where charge distributions are defined using a pseudoatom model (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). The numerical Fortran90 implementation included in the in-house version of *XDPROP*, part of the *XD* software suite (Volkov *et al*., 2016), is based on the recurrence scheme developed by Harris & Michels (1967).

The previously reported Löwdin α-function technique (Nguyen *et al*., 2018) and the present Fourier transform approach have been thoroughly tested in terms of both the speed and precision by comparing the values of *all* individual two-center Coulomb integrals in one of the benchmark glycine dimers (Gly1; Volkov, Koritsanszky *et al*., 2004; Nguyen *et al*., 2018) with those calculated using the numerical contour integration technique (Silverstone, 2014) implemented in the computer algebra-based software

Mathematica (Wolfram Research, 2018) which was required to maintain 90 digits in all internal computations and evaluate numerical integrals to 80 digits.

The Fortran-based Löwdin α-function calculations were performed in the local (*aka* the “lined-up”) coordinate system (Steinborn & Ruedenberg, 1972; Guseinov & Mamedov, 1999; Yakar *et al*., 2007) for each pair of pseudoatoms using the 64-, 80- and 128-bit numerical precision that are expected to provide on average 15, 18 and 33 digits, respectively. The Fortran-based Fourier transform calculations employed the 128-bit precision only but made use of both the local and global Cartesian coordinate systems.

All calculations were performed using a 64-bit 2012 4.0 GHz AMD FX-8350 processor (CPU) and a 64-bit Linux operating system. Unlike the 64-bit fixed floating-point calculations that are performed primarily in the 64-bit CPU hardware, the higher precision calculations need to be performed partially (80-bit) or almost entirely (128-bit) in software. Thus, we also had to investigate the extent to which the Fortran compiler and the code optimization options affect numerical values of the resulting Coulomb integrals. In this work we have included results from the latest versions of the three freely distributed compilers: Oracle Developer Studio (Oracle Corporation, 2017*a*), GFortran (Free Software Foundation, 2018), and DragonEgg (2018) as included in the AMD Optimizing C/C++ compiler suite (Advanced Micro Devices, 2018), and an aging commercial (yet very affordable) Lahey/Fujitsu LF64 Express 8.1 compiler (Lahey Computer Systems, 2011).

In the first part of the study we have performed a detailed analysis of *all* two-center Coulomb integrals in the benchmark glycine dimer Gly1 (Table 3.1, and Figure 3.1 and 3.2) with the pseudoatom-based electron density constructed using the University at

Buffalo Aspherical Atom Databank (Volkov, Li, *et al*., 2004; Dominiak *et al*., 2007). The results can be summarized as follows:

(a) The 64-bit Löwdin α-function calculations are extremely fast (something that we also pointed out in our first paper) and produce results at essentially the same numerical precision regardless of the compiler used. This is to be expected as many 64-bit mathematical operations are hardware-accelerated. However, as discovered in the first paper, a significant loss of precision occurs for integrals that involve non-spherical ($l > 0$) functions, which results in some integrals being recovered to as few as 2 and 3 digits.

(b) Increasing precision of the mathematical operations to 80-bit allows the Löwdin α-function method to reproduce noticeably more digits with a relatively small computational performance overhead, especially when using the GFortran compiler.

(c) The Coulomb integrals from the 128-bit precision Löwdin α-function calculations, while much more accurate than those obtained with the 80- and especially 64-bit precision, are evaluated much slower though *not* to the extent when they can be considered computationally prohibitive. As in the case of the 64- and 80-bit calculations, numerical results and timing are essentially compiler-independent for three out of four tested compilers (GFortran, DragonEgg, and Lahey). The Oracle Studio Fortran compiler had to be limited to the very basic level of the code optimization, and thus produced rather slow 128-bit precision-based executable.

(d) The 128-bit recurrence-based Fourier transform implementation combined with the Lahey/Fujitsu LF64 compiler is able to reproduce on average 29-30 digits, which is rather close to the 128-bit precision limit (≅ 33-36 digits), at a moderate computational cost overhead. While its speed is slightly below that of the 80-bit Löwdin α-function method, it is able to recover almost twice as many digits. The *lowest* number of digits it recovered was 16 in the deformation ↔ deformation density group of integrals. The remaining three Fortran compilers significantly trail LF64 in terms of the precision, though both GFortran and DragonEgg are very slightly faster than LF64.

In the second part of the study we have combined the selected two-center Coulomb integral implementations with the multipole moment approximation for interatomic interactions beyond 5 Å forming different variations of our aEP/MM method. Using the same set of 28 benchmark molecular dimers employed in the previous study (Nguyen, *et al.*, 2018), the resulting implementations have been compared in terms of the accuracy of the calculated values of E_{es} (Table 3.2) and the speed (Table 3.3). The benchmark dimers, ranging from water-water to dodecapeptide-dodecapeptide size, are composed of the H, C, N and O atoms whose aspherical molecular electron densities were constructed using the University at Buffalo Aspherical Atom Databank (Volkov, Li, *et al.*, 2004; Dominiak *et al.*, 2007). The results are summarized as follows:

(a) Regardless of the Coulomb integral evaluation method applied, the numerical precision used, and the compiler employed (though in this part of the study our tests have been limited to LF64 and GFortran), the resulting E_{es} values agree to within 5×10^{-5} kJ/mol (usually, better) which is more than satisfactory considering the typical uncertainties of the determined theoretical and experimental pseudoatom parameters (we note that the method can equally well be applied to calculation of E_{es} from the experimental X-ray pseudoatom-based charge densities).

(b) In terms of speed, the clear winner is the 64-bit Löwdin α-function implementation (regardless whether LF64 or GFortran are used), followed closely by the 80-bit Löwdin α-function code compiled with GFortran (the only x86_64 Fortran compiler known to us that supports the extended precision), and the 128-bit Fourier transform method compiled with LF64. For example, the most time-consuming calculation in our study (first enkepahlin dimer, Enk1) took 2.9, 6.8 and 9.3 s using the three methods listed above, respectively. We note that all these calculations were performed using the local ("lined-up") coordinate system for each pair of pseudoatoms.

(c) The 128-bit Fourier transform-based calculations in the global Cartesian coordinate system require evaluation of a significantly larger number of integrals with $l > 0$, and thus on average are two-three times slower than calculations using the same method in the local interatomic frames. The small differences between the 128-bit Fourier transform-based calculations in two different coordinate systems are attributed to the loss of precision when rotating multipolar populations between the frames which in *XDPROP* is done in 64-bit precision. Currently, we do not see any practical application for the calculations in the global frame except perhaps for debugging a code that rotates multipolar functions (though much easier techniques can be used for that purpose).

(d) The 128-bit Löwdin α-function implementation gives essentially the same results at the 128-bit Fourier transform approach, but at significant computational overhead, and thus is not recommended for production runs.

In conclusion, the intermolecular electrostatic interaction energies between pseudoatom-based molecular densities can be accurately, reliably and rapidly computed via the aEP/MM method that uses either the 80-bit precision implementation of the Löwdin α-function approach or the 128-bit precision Fourier transform technique for evaluation of two-center Coulomb integrals. The latter is the preferred choice when the Lahey/Fujitsu LF64 compiler (Lahey Computer Systems, 2011) is available, while the former works well with the freely available GFortran compiler (Free Software Foundation, 2018).

Since our code development is done entirely under Linux, we are unable to provide benchmark values for the Windows and MacOs platforms, but because the GFortran compiler binaries are available for both of those systems, it makes sense to recommend the GFortran-based 80-bit precision implementation of the Löwdin α-function method for production runs on these platforms.

Currently, our future plans include: (*i*) exploration of the performance of the Fourier transform implementation using the so-called quad double (256-bit) precision, albeit for a slightly different purpose as the E_{es} precision beyond 5×10^{-5} kJ/mol is likely not needed for routine charge density studies, and (*ii*) extension of the aEP/MM method to calculation of the electrostatic component of the lattice energy in molecular crystals with charge distributions defined via the pseudoatom formalism (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996); a very rudimentary approach already included in *XDPROP* will be refined and extended using the combination of the analytical integration and the Ewald summation technique (Sagui *et al*., 2004).

References

Abergel, R. & Moisan, L. *Fast and accurate evaluation of a generalized incomplete gamma function*. MAP5 2016-14. 2016. 〈hal-01329669〉 *http://hal.archives-ouvertes.fr/hal-01329669*

Abramowitz, M. & Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th printing. Dover, New York.

Advanced Micro Devices, Inc. (2012). *Using the x86 Open64 Compiler Suite*. *For x86 Open64 version 4.5.2*. http://developer.amd.com/x86-open64-compiler-suite (accessed on 23 September, 2018).

Advanced Micro Devices, Inc. (2018). *AMD Optimizing C/C++ Compiler*, version 1.2.1. http://developer.amd.com/amd-aocc (accessed on 23 September, 2018).

Antone, A. A. (1985). *J. Math. Phys.* **26**, 940–942.

Arfken, G. (1985). *Mathematical Methods for Physicists*, 3rd edition. Academic Press, Inc., San Diego.

Bailey, D. H. (2015). *MPFUN2015: A thread-safe arbitrary precision computation package*, 2015. http://www.davidhbailey.com/dhbpapers/mpfun2015.pdf (accessed on 23 September, 2018).

Bailey, D. H. & Borwein, J. M. (2015). *Mathematics* **3**, 337–367.

Bailey, D. H., Hida, Y., Li, X. S. & Thompson, B. (2002). *ARPREC: An Arbitrary Precision Computation Package*, 2002. http://www.davidhbailey.com/dhbpapers/arprec.pdf (accessed on 23 September, 2018).

Barnett, M. P. (2002). *Theor. Chem. Acc.* **107**, 241–245.

Belkić, D. (1989). *Physica Scripta* **39**, 226–229.

Bouferguene, A. (2005). *J. Phys. A: Math. Gen.* **38**, 2899–2916.

Bouferguene, A. & Safouhi, H. (2006). *Int J Quantum Chem*. **106**, 2398–2407.

Berlu, L. (2004). *J. Theor. Comput. Chem*. **2**, 257–267.

Blanco, M. A., Flórez, M. & Bermejo, M. (1997). *J. Mol. Struct. (Theochem)* **419**, 19–27.

Chisholm, C. D. H. (1976). *Group Theoretical Techniques in Quantum Chemistry*. Academic Press, London, England.

Condon, E. U. & Shortley, G. (1959). *The Theory of Atomic Spectra*. Cambridge University Press, Cambridge, England:

Coppens, P. (1997). *X-ray Charge Densities and Chemical Bonding.* Oxford University Press, New York.

Cromer, D. T., Larson, A. C. & Stewart, R. F. (1976). *J. Chem. Phys.* **65**, 336–349.

Demizu, Y, Yamashita, H., Doi, M., Misawa, T., Oba, M., Tanaka, M. & Kurihara, M. (2015). *J. Org. Chem.* **80**, 8597–8603.

Dominiak, P. M., Volkov, A., Li, X., Messerschmidt, M. & Coppens, P. (2007). *J. Chem. Theory Comput.* **3**, 232–247.

Dong, S.-H. & Lemus, R. (2002). *Appl. Math. Lett.* **15**, 541–546.

DragonEgg (2018). *A GCC plugin that replaces GCC's optimizers and code generators with those from the LLVM project*. http://dragonegg.llvm.org (accessed on 23 September, 2018).

Edmonds, A. R. (1957). *Angular Momentum In Quantum Mechanics*. Princeton University Press, Princeton, New Jersey.

Enge, A., Théveny, P. & Zimmermann, P. (2018). *The GNU Multiple Precision Complex Library,* Edition 1.1.0, January 2018, INRIA. http://www.multiprecision.org/mpc (accessed on 23 September, 2018).

Filter, E. & Steinborn, E. O. (1978). *J. Math. Phys.* **19**, 79–84.

Fousse, L., Hanrot, G., Lefèvre, V., Pélissier, P. & Zimmermann, P. *ACM Trans. Math. Soft. (TOMS)*, 33, 13-1 – 13-14.

Free Software Foundation, Inc. (2018). *GCC 8.2 GNU Fortran Manual.* http://gcc.gnu.org (accessed on 23 September, 2018).

Gaunt, J. A. (1929). *Phil Trans Roy Soc. A* **228**, 151–196.

Geller, M. (1962). *J. Chem. Phys.* **36**, 2424–2428.

Geller, M. (1963a). *J. Chem. Phys.* **39**, 84–89.

Geller, M. (1963b). *J. Chem. Phys.* **39**, 853–854.

Geller, M. (1964a). *J. Chem. Phys.* **41**, 4006–4007.

Geller, M. (1964b). *Technical Report No. 32-673*, Jet Propulsion Laboratory, Pasadena, California.

Geller, M. & Griffith, R. W. (1964) *J. Chem. Phys.* **40**, 2309–2325.

Granlund, T. (2016). *GNU MP - The GNU Multiple Precision Arithmetic Library*, Edition 6.1.2, 16 December 2016, Free Software Foundation, Inc. http://gmplib.org (accessed on 23 September, 2018).

Grotendorst J. & Steinborn, E. O. (1985). *J. Comput. Phys.* **61**, 195–217.

Guseinov, I. I. & Mamedov, B. A. (1999). *J. Mol. Struct. (Theochem)* **465**, 1–6.

Guseinov, I. I. & Mamedov, B. A. (2002). *J. Math. Chem.* **32**, 309–322.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Harris, F. E. (2002). *Int J Quantum Chem.* **88**, 701–734. http://www.physics.utah.edu/~harris/home.html (accessed on 23 September, 2018).

Harris, F. E. (2003). *Int J Quantum Chem.* **93**, 332–334.

Harris, F. E. & Michels, H. H. (1967). *Adv. Chem. Phys.* **13**, 205–266.

Hida, Y., Li, X. S. & Bailey, D. H. (2012). *Quad Double computation package*. Revision date: 13 March 2012 http://crd-legacy.lbl.gov/~dhbailey/mpdist (accessed on 23 September, 2018).

Hirshfeld, F. L. (1971) *Acta Cryst.* B**27**, 769–781.

Homeier, H. H. H. & Steinborn, E. O. (1996). *J. Mol. Struct. (Theochem)*. **38**, 31–37.

Huzinaga, S. (1967). *Suppl. Progress Theor. Phys.* **40**, 52–77.

Ivanic, J. & Ruedenberg, K. (1996). *J. Phys. Chem.* **100**, 6342–6347.

Jones, H. W. (1980). *Int. J. Quantum Chem.* **18**, 709–713.

Jones, H. W. (1981). *Int. J. Quantum Chem.* **20**, 1217–1224.

Jones, H. W. (1984). *Phys. Rev. A* **30**, 1–4.

Jones, H. W. (1991). *J. Comp. Chem.* **12**, 1217–1222.

Jones, H. W. (1992). *Int. J. Quantum Chem.* **41**, 749–754.

Jones, H. W. (1993). *Int. J. Quantum Chem.* **45**, 21–30.

Jones, H. W. & Weatherford, C. A. (1978). *Int. J. Quantum Chem. Symp.* **12**, 483–488.

Jones, H. W. & Weatherford, C. A. (1989). *J. Mol. Struct. (Theochem)* **199**, 233–243.

Lahey Computer Systems, Inc. (2011). *LF Fortran Express User's Guide*, Revision D.05. http://www.lahey.com/doc.htm (accessed on 23 September, 2018).

Lemmon, D. R. & Schafer, J. L. (2005). *Developing Statistical Software in Fortran 95 (Statistics and Computing).* Springer Science+Business Media, Inc.

Lattner, C. & Adve, V. (2004). *Proc. 2004 Int. Symp. on Code Gen. and Opt. (CGO'04)*, Palo Alto, California.

Löwdin, P. O. (1956). *Adv. Phys.* **5**, 1–171.

Mamedov, B. A. & Çopuroğlu. E. (2011). *J. Math. Chem.* **49**, 201–207.

Mar'yashkin, N. Ya. & Zimont, S. L. (1975). in *Matem. Problem. Khimii*, 204–210 (in Russian).

Morse, P. M. & Feshbach, H. (1953) *Methods of Theoretical Physics*. McGraw-Hill Book Company, New York.

Niukkanen, A. W. (1984). *Int. J. Quantum Chem.* **25**, 941–955.

Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536.

Niehaus, T. A., López & Rico, J. F. (2008). *J. Phys. A: Math. Theor.* **41**, 485205 1–14.

O-Ohata, K. & Ruedenberg, K. (1966). *J. Math. Phys.* **7**, 547–559.

Olver, F. W. J., Olde Daalhuis, A. B., Lozier, D. W., Schneider, B. I., Boisvert, R. F., Clark, C. W., Miller, B. R. & Saunders, B. V. *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov, Release 1.0.20 of 2018-09-15 (accessed on 23 September, 2018).

Oracle Corporation (2017a). *Oracle Developer Studio 12.6: Fortran User's Guide*, Part No: E77790. http://docs.oracle.com/en (accessed on 23 September, 2018).

Oracle Corporation (2017b). *Oracle Developer Studio 12.6: Numerical Computation Guide*. Part No: E77791. http://docs.oracle.com/en (accessed on 23 September, 2018).

Özay, S. & Öztekin, E. *Adv. Quantum Chem.* **67**, 245–265.

Özcan, S. & Öztekin, E. (2009). *J. Math. Chem.* **45**, 1153–1165.

Ozdogan T. & Nalcaci, A. *Int. J. Phys. Sci.* **7**, 5378–5390.

Öztekin, E. (2004). *Int. J. Quantum Chem.* **100**, 236–243.

Öztekin, E., Yavuz, M. & Atalay, S. (2001). *Theor. Chem. Acc.* **106**, 264–270.

Öztekin, E. & Özcan, S. (2006). *J. Math. Chem.* **42**, 337–351.

Paturle, A. & Coppens, P. (1988). *Acta. Cryst.* A**44**, 6-7.

PassMark Software Pty Ltd, Redwood City, California, USA. http://www.cpubenchmark.net (accessed on 28 October, 2018).

PGI Compilers and Tools (2018). *PGI Release Notes for x86 CPUs*. Version 2018, NVIDIA Corporation. http://www.pgroup.com/resources/docs/18.7/pdf/pgirn187-x86.pdf (accessed on 23 September, 2018).

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1992). *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd edition. Cambridge University Press, Cambridge, England:

Prosser, F. P. & Blanchard, C. H. (1962). *J. Phys. Chem*. **36**, 1112.

Roothaan, C. C. J. (1951). *J. Chem. Phys*. **19**, 1445–1458.

Sagui, C., Pedersen, L. G. & Darden, T. A. (2004). *J. Chem. Phys*. **120**, 73–87.

Sharma, R. R. (1976). *Phys. Rev. A* **13**, 517–527.

Shestakov, A. F. (1992). *J. Struct. Chem.* **33**, 131–133.

Silverstone, H. J. (1966). *J. Phys. Chem.* **45**, 4337–4341.

Silverstone, H. J. (1967a). *J. Phys. Chem.* **46**, 4368–4376.

Silverstone, H. J. (1967b). *J. Phys. Chem.* **46**, 4377–4380.

Silverstone, H. J. (1967c). *J. Phys. Chem.* **47**, 537–540.

Silverstone, H. J. (2014). *J. Phys. Chem. A* **118**, 11971–11974.

Silverstone, H. J. & Moats, R. K. (1977). *Phys. Rev. A* **16**, 1731–1732.

Smith, D.M. (1991). *ACM Trans. Math. Soft. (TOMS)* **17**, 273–283.

Slater, J. C. (1932). *Phys. Rev.* **42**, 33–43.

Steinborn, E. O. & Ruedenberg, K. (1972). *Int. J. Quantum Chem*. **6**, 413–438.

Stewart, R. F. (1969). *J. Chem. Phys.* **51**, 4569–4577.

Stewart, R. F. (1976). *Acta Cryst.* A**32**, 565–574.

Stewart, R. F. (1978). In *Electron and Magnetization Densities in Molecules and Crystals*, edited by P. Becker, p. 439. Plenum , New York.

Su, Z. & Coppens, P. (1990). *J. Appl. Cryst.* **23**, 71–73.

Su, Z. & Coppens, P. (1994). *Acta Cryst.* A**50**, 636–643.

Su, Z. & Coppens, P. (1995). *Acta Cryst.* A**51**, 198.

Suzuki, N. (1984). *J. Math. Phys.* **25**, 1133–1138.

Suzuki, N. (1985). *J. Math. Phys.* **26**, 3193–3198.

Suzuki, N. (1987). *J. Math. Phys.* **28**, 769–773.

Suzuki, N. (1992). *J. Math. Phys.* **33**, 4288–4299.

Tam, P. T. (2008). *A Physicist's Guide to Mathematica*, 2nd edition. Academic Press, Elsevier Inc.

Thompson, W. J. (1994). *Angular Momentum*, 1st edition. John Wiley & Sons, Inc. New York.

Thompson, W. J. (1997). *Atlas for Computing Mathematical Functions: An Illustrated Guide for Practitioners with Programs in FORTRAN and Mathematica*. John Wiley & Sons, Inc. New York.

Todd, H. D., Kay, K. G. & Silverstone, H. J. (1970). *J. Chem. Phys.* **53**, 3951–3956.

Trivedi, H. P. & Steinborn, E. O. (1983). *Phys. Rev. A* **27**, 670–679.

Tsirelson, V. G. & Ozerov, R. P. (1996). *Electron Density and Bonding in Crystals*. Bristol, England / Philadelphia, USA: Institute of Physics Publishing.

Volkov, A., Koritsanszky T. S. & Coppens, P. (2004). *Chem. Phys. Lett.* **391**, 170–175.

Volkov, A., Li, X., Koritsanszky, T. S. & Coppens, P. (2004). *J. Phys. Chem. A* **108**, 4283–4300.

Volkov, A., Macchi, P., Farrugia, L. J., Gatti, C., Mallinson, P., Richter, T. & Koritsanszky, T. (2016) *XD2016 - A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors*.

Weissbluth, M. (1978). *Atoms and Molecules*. Academic Press, Inc., New York / London.

Weniger, E. J. & Steinborn, E. O. (1983) *J. Chem. Phys.* **78**, 6121–6132.

Weniger, E. J., Grotendorst J. & Steinborn, E. O. (1986). *Phys. Rev. A.* **33**, 3688–3705.

Wheatley, R. J. (1998). *Chem. Phys. Lett*. **294**, 487–492.

Wolfram Research, Inc. (2018). *Mathematica 11.3*. Champaign, Illinois, USA.

Yakar, Y., Özmen, A. & Atav, Ü. (2006). *Chin. J. Chem.* **24**, 603–608.

Yakar, Y., Özmen, A., Çakir, B. & Yüksel, H. (2007). *J. Chin. Chem. Soc.* **54**, 1139–1144.

APPENDICES

APPENDIX A: The Radial Integral $J_{ab\lambda}^{(N)}(r)$

The radial integral $J_{ab\lambda}^{(N)}(r)$ is defined as (Todd *et al.*, 1970; Silverstone, 2014)

$$J_{ab\lambda}^{(N)}(r) = \frac{i^{\lambda}}{2\pi^2}\int_0^{\infty} k^{2N+2} f_a(k) f_b(k) j_\lambda(kr) dk) \tag{3.49}$$

where $j_\lambda(z)$ is the spherical Bessel function of the first kind (Arfken, 1985; Olver *et al.*, 2018), and $f(k)$ is the radial part of the Fourier transform of an STF (Slater, 1932) given by (Geller, 1963*a*; Silverstone, 1966):

$$f(k) = 4\pi i^l \int_0^{\infty} r^{n+1} e^{-\zeta r} j_l(kr) dr \tag{3.50}$$

The closed-form expression for the radial integral $f(k)$ has been given in a variety of forms by many authors, for example: Geller (1963*a*), Silverstone (1966), Stewart (1969, 1978), Weniger & Steinborn (1983), Niukkanen (1984), Belkić & Taylor(1989), Su & Coppens (1990). In the present study we use Geller's (1963*a*) representation obtained via the mathematical induction

$$\int_0^{\infty} r^{n+1} e^{-\zeta r} j_l(kr) dr$$
$$= \frac{(2k)^l (n-l)!\, \zeta^{n-l}}{(\zeta^2 + k^2)^{n+1}} \sum_{s=0}^{\left\lfloor \frac{1}{2}(n-l) \right\rfloor} (-1)^s \binom{n+l+1}{2s+2l+1} \frac{(s+l)!}{s!} \left(\frac{k}{\zeta}\right)^{2s} \tag{3.51}$$

where $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a (*i.e.* the result of the floor function) and $\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is the binomial coefficient (Geller, 1963*a*). Applying Geller's formula to $f_a(k)$ and $f_b(k)$,

$$f_a(k) = 4\pi i^{l_a} \frac{(2k)^{l_a}(n_a - l_a)!\,\zeta_a{}^{n_a - l_a}}{(\zeta_a^2 + k^2)^{n_a+1}}$$
$$\times \sum_{s=0}^{\left\lfloor \frac{1}{2}(n_a - l_a) \right\rfloor} (-1)^s \binom{n_a + l_a + 1}{2s + 2l_a + 1} \frac{(s + l_a)!}{s!} \left(\frac{k}{\zeta_a}\right)^{2s} \tag{3.52}$$

$$f_b(k) = 4\pi i^{l_b} \frac{(2k)^{l_b}(n_b - l_b)!\,\zeta_b{}^{n_b - l_b}}{(\zeta_b^2 + k^2)^{n_b+1}}$$
$$\times \sum_{t=0}^{\left\lfloor \frac{1}{2}(n_b - l_b) \right\rfloor} (-1)^t \binom{n_b + l_b + 1}{2t + 2l_b + 1} \frac{(t + l_b)!}{t!} \left(\frac{k}{\zeta_b}\right)^{2t} \tag{3.53}$$

and substituting these into equation (3.49) we obtain

$$J_{ab\lambda}^{(N)}(r) = \frac{i^\lambda}{2\pi^2} \int_0^\infty \left[\{k^{2N+2} j_\lambda(kr) 4\pi i^{l_a} 4\pi i^{l_b} \frac{(2k)^{l_a}(n_a - l_a)!\,\zeta_a{}^{n_a - l_a}}{(\zeta_a^2 + k^2)^{n_a+1}} \right.$$
$$\times \frac{(2k)^{l_b}(n_b - l_b)!\,\zeta_b{}^{n_b - l_b}}{(\zeta_b^2 + k^2)^{n_b+1}} \sum_{s=0}^{\left\lfloor \frac{1}{2}(n_a - l_a) \right\rfloor} (-1)^s \binom{n_a + l_a + 1}{2s + 2l_a + 1} \frac{(s + l_a)!}{s!} \left(\frac{k}{\zeta_a}\right)^{2s} \tag{3.54}$$
$$\left. \times \sum_{t=0}^{\left\lfloor \frac{1}{2}(n_b - l_b) \right\rfloor} (-1)^t \binom{n_b + l_b + 1}{2t + 2l_b + 1} \frac{(t + l_b)!}{t!} \left(\frac{k}{\zeta_b}\right)^{2t} \right] dk$$

Moving constants out of the integral, and rearranging the order of the integration and summations gives

$$J_{ab\lambda}^{(N)}(r) = A \sum_{s=0}^{\left\lfloor \frac{1}{2}(n_a - l_a) \right\rfloor} \sum_{t=0}^{\left\lfloor \frac{1}{2}(n_b - l_b) \right\rfloor} B(s,t) \left[\frac{2}{\pi} \int_0^\infty \frac{k^{2N+2+l_a+l_b+2s+2t} j_\lambda(kr)}{(\zeta_a^2 + k^2)^{n_a+1}(\zeta_b^2 + k^2)^{n_b+1}} dk \right] \tag{3.55}$$

where

$$A = 4\pi i^{l_a + l_b + \lambda} 2^{l_a + l_b} (n_a - l_a)!\,(n_b - l_b)!\,\zeta_a{}^{n_a - l_a} \zeta_b{}^{n_b - l_b} \tag{3.56}$$

$$B(s,t) = (-1)^{s+t} \binom{n_a + l_a + 1}{2s + 2l_a + 1} \binom{n_b + l_b + 1}{2t + 2l_b + 1} \frac{(s + l_a)!\,(t + l_b)!}{s!\,t!\,\zeta_a{}^{2s} \zeta_b{}^{2t}} \tag{3.57}$$

Note that since the angular integral $d_{ab\lambda}(\theta, \phi)$ is non-zero only if $l_a + l_b + \lambda =$ even, then $i^{l_a+l_b+\lambda} = 1$ if $(l_a + l_b + \lambda)/2 =$ even and $i^{l_a+l_b+\lambda} = -1$ if $(l_a + l_b + \lambda)/2 =$ odd, which simplifies the expression for A to

$$A = \pm 4\pi 2^{l_a+l_b}(n_a - l_a)!\,(n_b - l_b)!\,{\zeta_a}^{n_a-l_a}{\zeta_b}^{n_b-l_b} \tag{3.58}$$

By comparing the term in curly brackets in equation (3.55) with the integral $W_{i,i'}^{L,j}(\delta_1\delta_2 R)$ originally defined in Geller & Griffith (1964) and Geller (1964*a,b*), discussed in detail by Harris & Michels (1967), and commented on by Mar'yashkin & Zimont (1975),

$$W_{i,i'}^{L,j}(\delta_1\delta_2 R) = \frac{2}{\pi}\int_0^\infty \frac{k^{L+2j} j_L(kr)}{(\delta_1^2 + k^2)^i(\delta_2^2 + k^2)^{i'}}\,dk \tag{3.59}$$

it follows that

$$\delta_1 = \zeta_a \tag{3.60}$$

$$\delta_2 = \zeta_b \tag{3.61}$$

$$R = r \tag{3.62}$$

$$i = n_a + 1 \tag{3.63}$$

$$i' = n_b + 1 \tag{3.64}$$

$$L = \lambda \text{ [because } j_\lambda(kr) \leftrightarrow j_L(kr)\text{]} \tag{3.65}$$

$$j = (2N + 2 + l_a + l_b + 2s + 2t - \lambda)/2 \tag{3.66}$$

The last relation is obtained from

$$2j = 2N + 2 + l_a + l_b + 2s + 2t - \lambda, \tag{3.67}$$

so that

$$\begin{aligned} L + 2j &= \lambda + (2N + 2 + l_a + l_b + 2s + 2t - \lambda) \\ &= 2N + 2 + l_a + l_b + 2s + 2t \end{aligned} \tag{3.68}$$

APPENDIX B: Supporting Information

S1. Plane wave expansion of $\exp(\pm i\mathbf{k}\cdot\mathbf{r})$ in terms of the associated Legendre polynomials

Define vectors **r** and **k** in the spherical coordinate system:

$$\mathbf{r} \equiv (r, \theta, \phi) \tag{1}$$

$$\mathbf{k} \equiv (k, \bar{\theta}, \bar{\phi}) \tag{2}$$

The plane wave (Rayleigh) expansion for $\exp(\pm i\mathbf{k}\cdot\mathbf{r})$ in terms of complex spherical harmonics $Y_l^m(\theta,\phi)$ is given by (Geller, 1962; Weissbluth, 1978):

$$\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = 4\pi \sum_{\lambda=0}^{\infty}\left(\sum_{\mu=-\lambda}^{\lambda}\left[(\pm i)^{\lambda} j_{\lambda}(kr) Y_{\lambda}^{\mu}(\theta,\phi) Y_{\lambda}^{\mu *}(\bar{\theta},\bar{\phi})\right]\right) \tag{3}$$

where symbol * denotes the complex-conjugate, and $j_\lambda(z)$ is the spherical Bessel function of the first kind (Arfken, 1985; Olver *et al.*, 2018). The complex spherical harmonic function $Y_l^m(\theta,\phi)$ for $l \geq 0$ and $-l \leq m \leq l$ is defined as (Weissbluth, 1978; Weniger & Steinborn, 1982; Homeier & Steinborn, 1996)

$$Y_l^m(\theta,\phi) = (-1)^{m+|m|}\sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}P_l^m(\cos\theta)e^{im\phi} \tag{4}$$

where functions $P_l^m(x)$ are the associated Legendre polynomials (Press *et al.*, 1992; Tam, 2008; Wolfram Research, 2018),

$$P_l^{|m|}(x) = (-1)^{|m|}(1-x^2)^{|m|/2}\frac{d^{|m|}P_l(x)}{dx^{|m|}} \tag{5}$$

$$P_l^{-|m|}(x) = (-1)^{|m|}\frac{(l-|m|)!}{(l+|m|)!}P_l^{|m|}(x) \tag{6}$$

and functions $P_l(x)$ are the ordinary (unassociated) Legendre polynomials (Press *et al.*, 1992; Tam, 2008; Wolfram Research, 2018). In this study we follow the notation used in

Mathematica (Press *et al.*, 1992; Tam, 2008; Wolfram Research, 2018) according to which the associated Legendre polynomials include the Condon-Shortley phase $(-1)^{|m|} = (-1)^m$ (Condon & Shortley, 1959). Note that equation (4) is valid for both the positive and negative values of m, and is consistent with *Mathematica*'s function $\mathrm{SphericalHarmonicY}[l, m, \theta, \phi]$ (Wolfram Research, 2018).

The associated Legendre polynomials for the positive and negative values of m, equations (5) and (6), are related to each other via the following relationship

$$P_l^m(x) = (-1)^{|m|}\frac{(l+m)!}{(l-m)!}P_l^{-m}(x) \tag{7}$$

which is valid regardless whether m is positive or negative. Equations (5), (6) and (7) are consistent with *Mathematica*'s function $\mathrm{LegendreP}[l, m, x]$ (Wolfram Research, 2018).

Expanding the complex-conjugate spherical harmonic function $Y_\lambda^{\mu*}(\bar{\theta}, \bar{\phi})$ in terms of $P_\lambda^{-\mu}(\cos\theta)$,

$$Y_\lambda^{\mu*}(\bar{\theta}, \bar{\phi}) = (-1)^{\mu+|\mu|}\sqrt{\frac{(2\lambda+1)}{4\pi}\frac{(\lambda-\mu)!}{(\lambda+\mu)!}}(-1)^{|\mu|}\frac{(\lambda+\mu)!}{(\lambda-\mu)!}P_\lambda^{-\mu}(\cos\bar{\theta})e^{-i\mu\bar{\phi}} \tag{8}$$

and substituting it into the product $Y_\lambda^\mu(\theta, \phi)Y_\lambda^{\mu*}(\bar{\theta}, \bar{\phi})$ we get

$$\begin{aligned} Y_\lambda^\mu(\theta, \phi)Y_\lambda^{\mu*}(\bar{\theta}, \bar{\phi}) &= (-1)^{\mu+|\mu|}\sqrt{\frac{(2\lambda+1)}{4\pi}\frac{(\lambda-\mu)!}{(\lambda+\mu)!}}P_\lambda^\mu(\cos\theta)e^{i\mu\phi} \\ &\quad\times(-1)^{\mu+|\mu|}\sqrt{\frac{(2\lambda+1)}{4\pi}\frac{(\lambda-\mu)!}{(\lambda+\mu)!}}(-1)^{|\mu|}\frac{(\lambda+\mu)!}{(\lambda-\mu)!}P_\lambda^{-\mu}(\cos\bar{\theta})e^{-i\mu\bar{\phi}} \\ &= (-1)^{|\mu|}\frac{(2\lambda+1)}{4\pi}\frac{(\lambda-\mu)!}{(\lambda+\mu)!}\frac{(\lambda+\mu)!}{(\lambda-\mu)!}P_\lambda^\mu(\cos\theta)P_\lambda^{-\mu}(\cos\bar{\theta})e^{i\mu\phi}e^{-i\mu\bar{\phi}} \\ &= (-1)^{|\mu|}\frac{(2\lambda+1)}{4\pi}P_\lambda^\mu(\cos\theta)P_\lambda^{-\mu}(\cos\bar{\theta})e^{i\mu(\phi-\overline{\phi})} \end{aligned} \tag{9}$$

Inserting this result into the expansion for $\exp(\pm i\mathbf{k}\cdot\mathbf{r})$ gives

$$\exp(\pm i\mathbf{k}\cdot\mathbf{r})$$
$$= 4\pi \sum_{\lambda=0}^{\infty}\left(\sum_{\mu=-\lambda}^{\lambda}\left[(\pm i)^{\lambda} j_{\lambda}(kr)(-1)^{|\mu|}\frac{(2\lambda+1)}{4\pi}P_{\lambda}^{\mu}(\cos\theta)P_{\lambda}^{-\mu}(\cos\bar{\theta})e^{i\mu(\phi-\bar{\phi})}\right]\right) \tag{10}$$

Cancelling out the two 4π factors, moving the μ-independent terms out of the inner sum, and applying Euler's formula $e^{ix} = \cos x + i\sin x$ results in

$$\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = \sum_{\lambda=0}^{\infty}(2\lambda+1)(\pm i)^{\lambda} j_{\lambda}(kr)$$
$$\times\sum_{\mu=-\lambda}^{\lambda}\left[(-1)^{|\mu|}P_{\lambda}^{\mu}(\cos\theta)P_{\lambda}^{-\mu}(\cos\bar{\theta})\{\cos(\mu(\phi-\bar{\phi}))+i\sin(\mu(\phi-\bar{\phi}))\}\right] \tag{11}$$

Consider the expanded inner sum, and note that terms with both the positive and negative μ are present:

$$\sum_{\mu=-\lambda}^{\lambda}\left[(-1)^{|\mu|}\{P_{\lambda}^{\mu}(\cos\theta)P_{\lambda}^{-\mu}(\cos\bar{\theta})\cos(\mu(\phi-\bar{\phi}))\right.$$
$$\left. + iP_{\lambda}^{\mu}(\cos\theta)P_{\lambda}^{-\mu}(\cos\bar{\theta})\sin(\mu(\phi-\bar{\phi}))\}\right] \tag{12}$$

Using the well-known trigonometric identities $\sin(-x) = -\sin(x)$ and $\cos(-x) = \cos(x)$, and expressing the associated Legendre polynomial with $\mu < 0$ in terms of that with $\mu > 0$ via equation (6),

$$P_{\lambda}^{-|\mu|}(\cos\bar{\theta}) = (-1)^{|\mu|}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_{\lambda}^{|\mu|}(\cos\bar{\theta}) \tag{13}$$

we get for the $\sin(\dots)$ terms:

$\mu > 0$: $iP_{\lambda}^{|\mu|}(\cos\theta)P_{\lambda}^{-|\mu|}(\cos\bar{\theta})\sin(|\mu|(\phi-\bar{\phi})) = i(-1)^{|\mu|}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_{\lambda}^{|\mu|}(\cos\theta)P_{\lambda}^{|\mu|}(\cos\bar{\theta})\sin(|\mu|(\phi-\bar{\phi}))$

$\mu < 0$: $iP_{\lambda}^{-|\mu|}(\cos\theta)P_{\lambda}^{|\mu|}(\cos\bar{\theta})\sin(-|\mu|(\phi-\bar{\phi})) = -i(-1)^{|\mu|}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_{\lambda}^{|\mu|}(\cos\theta)P_{\lambda}^{|\mu|}(\cos\bar{\theta})\sin(|\mu|(\phi-\bar{\phi}))$

$\mu = 0$: $iP_{\lambda}^{0}(\cos\theta)P_{\lambda}^{0}(\cos\bar{\theta})\sin(0(\phi-\bar{\phi})) = 0$

It follows that the sum of all the $\sin(\ldots)$ terms in equation (12) is zero, and thus they can be safely eliminated from the summation. For the $\cos(\ldots)$ terms,

$$\mu > 0\colon P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{-|\mu|}(\cos\bar{\theta})\cos(\;\;|\mu|(\phi-\bar{\phi})) = (-1)^{|\mu|}\tfrac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(|\mu|(\phi-\bar{\phi}))$$

$$\mu < 0\colon P_\lambda^{-|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(-|\mu|(\phi-\bar{\phi})) = (-1)^{|\mu|}\tfrac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(|\mu|(\phi-\bar{\phi}))$$

$$\mu = 0\colon P_\lambda^{0}(\cos\theta)P_\lambda^{0}(\cos\bar{\theta})\cos(0(\phi-\bar{\phi})) = P_\lambda^{0}(\cos\theta)P_\lambda^{0}(\cos\bar{\theta})$$

Because terms for the positive and negative μ are the same, we can limit the summation over μ to the non-negative values only:

$$\begin{aligned}\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = \sum_{\lambda=0}^{\infty}&(2\lambda+1)(\pm i)^{\lambda} j_\lambda(kr)\\ &\times\sum_{\mu=0}^{\lambda}\epsilon_\mu(-1)^{|\mu|}(-1)^{|\mu|}\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(|\mu|(\phi-\bar{\phi}))\end{aligned} \tag{14}$$

where ϵ_μ is defined as in Morse & Feshbach (1953) and Geller (1963)

$$\epsilon_0 = 1 \text{ for } \mu = 0, \text{ and } \epsilon_\mu = 2 \text{ for } \mu > 0, \text{ or simply } \epsilon_\mu = 2 - \delta_{\mu,0}$$

Note that there is only one $\mu = 0$ term, while all the $\mu > 0$ terms need to be counted twice. Finally, since $(-1)^{|\mu|}(-1)^{|\mu|} = 1$, the desired expression for $\exp(\pm i\mathbf{k}\cdot\mathbf{r})$ becomes

$$\begin{aligned}\exp(\pm i\mathbf{k}\cdot\mathbf{r}) = \sum_{\lambda=0}^{\infty}&(2\lambda+1)(\pm i)^{\lambda} j_\lambda(kr)\\ &\times\sum_{\mu=0}^{\lambda}\epsilon_\mu\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(|\mu|(\phi-\bar{\phi}))\end{aligned} \tag{15}$$

which agrees with the expression given for example in Morse & Feshbach (1953), Geller (1963) and Coppens (1997). Note that since $\cos(|\mu|(\phi-\bar{\phi})) = \cos(-|\mu|(\phi-\bar{\phi})) = \cos(|\mu|(\bar{\phi}-\phi))$, it is clear that the complex conjugation can be freely exchanged between the two spherical harmonic functions.

S2. Fourier transform of Slater-type function with real spherical harmonics

The *unnormalized* Slater-type function (Slater, 1932) is given by

$$\chi(\mathbf{r}) = \chi(r,\theta,\phi) = r^{n-1}e^{-\zeta r}y_l^m(\theta,\phi) \tag{16}$$

where all the terms are as defined in the main body of the manuscript. The Fourier transform of $\chi(\mathbf{r})$, $F(\mathbf{k})$, is (Geller, 1963)

$$F(\mathbf{k}) = \int \exp(i\mathbf{k}\cdot\mathbf{r})\chi(\mathbf{r})\,d\mathbf{r} \tag{17}$$

Expanding the $\exp(i\mathbf{k}\cdot\mathbf{r})$ term as described in Section S1, defining $y_l^m(\theta,\phi)$ in terms of the associated Legendre polynomials (see equations (3) and (4) in the main body of the manuscript), and switching to the spherical coordinate system we get:

$$\begin{gathered} F(\mathbf{k}) = \int_0^\infty \int_0^{2\pi} \int_0^\pi \sum_{\lambda=0}^{\infty}(2\lambda+1)i^\lambda j_\lambda(kr)\sum_{\mu=0}^{\lambda}\epsilon_\mu \frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\theta)P_\lambda^{|\mu|}(\cos\bar{\theta})\cos(|\mu|(\phi-\bar{\phi})) \\ \times r^{n-1}e^{-\zeta r}(-1)^{|m|}N_{l,m}P_l^{|m|}(\cos\theta)\begin{Bmatrix}\cos(|m|\phi), m>0 \\ \sin(|m|\phi), m<0 \\ 1, m=0\end{Bmatrix} r^2\sin\theta\,d\theta d\phi dr \end{gathered} \tag{18}$$

Rearranging the integrals and the sums gives

$$\begin{gathered} F(\mathbf{k}) = (-1)^{|m|}N_{l,m}\sum_{\lambda=0}^{\infty}(2\lambda+1)i^\lambda\sum_{\mu=0}^{\lambda}\epsilon_\mu\frac{(\lambda-|\mu|)!}{(\lambda+|\mu|)!}P_\lambda^{|\mu|}(\cos\bar{\theta})\int_0^\infty r^{n-1}r^2e^{-\zeta r}j_\lambda(kr)dr \\ \times\int_0^\pi P_\lambda^{|\mu|}(\cos\theta)P_l^{|m|}(\cos\theta)\sin\theta\,d\theta\int_0^{2\pi}\cos(|\mu|(\phi-\bar{\phi}))\begin{Bmatrix}\cos(|m|\phi) \\ \sin(|m|\phi) \\ 1\end{Bmatrix}d\phi \end{gathered} \tag{19}$$

The radial integral can be simplified to

$$\int_0^\infty r^{n-1}r^2e^{-\zeta r}j_\lambda(kr)dr = \int_0^\infty r^{n+1}e^{-\zeta r}j_\lambda(kr)dr \tag{20}$$

The angular integral over ϕ is (Geller, 1963)

$$\int_0^{2\pi} \cos(|\mu|(\phi - \bar{\phi})) \begin{Bmatrix} \cos(|m|\phi) \\ \sin(|m|\phi) \\ 1 \end{Bmatrix} d\phi = \frac{2\pi}{\epsilon_\mu} \begin{Bmatrix} \cos(|m|\bar{\phi}) \\ \sin(|m|\bar{\phi}) \\ 1 \end{Bmatrix} \delta_{|\mu|,|m|} \tag{21}$$

where ϵ_μ is defined as in Morse & Feshbach (1953) and Geller (1963)

$\epsilon_0 = 1$ for $\mu = 0$, and $\epsilon_\mu = 2$ for $\mu > 0$, or simply $\epsilon_\mu = 2 - \delta_{\mu,0}$

Since the ϕ integral is non-zero only if $m = |\mu|$, the θ integral simplifies to (Geller, 1963):

$$\int_0^{\pi} P_\lambda^{|m|}(\cos\theta) P_l^{|m|}(\cos\theta) \sin\theta \, d\theta = \frac{2}{2l+1} \frac{(l+|m|)!}{(l-|m|)!} \delta_{\lambda,l} \tag{22}$$

Thus, the only term that survives in the double sum is the one with $\lambda = l$ and $|\mu| = |m|$. As such, the integral $F_{nlm}(\mathbf{k})$ becomes:

$$F(\mathbf{k}) = (-1)^{|m|} N_{l,m} P_l^{|m|}(\cos\bar{\theta}) \begin{Bmatrix} \cos(|m|\bar{\phi}) \\ \sin(|m|\bar{\phi}) \\ 1 \end{Bmatrix} 4\pi i^l \int_0^\infty r^{n+1} e^{-\zeta r} j_l(kr) dr \tag{23}$$

Because

$$y_l^m(\bar{\theta}, \bar{\phi}) = (-1)^{|m|} N_{l,m} P_l^{|m|}(\cos\bar{\theta}) \begin{Bmatrix} \cos(|m|\bar{\phi}) \\ \sin(|m|\bar{\phi}) \\ 1 \end{Bmatrix} \tag{24}$$

the final expression for $F(\mathbf{k})$ is

$$F(\mathbf{k}) = y_l^m(\bar{\theta}, \bar{\phi}) 4\pi i^l \int_0^\infty r^{n+1} e^{-\zeta r} j_l(kr) dr = f_{nl\zeta}(k) y_l^m(\bar{\theta}, \bar{\phi}) \tag{25}$$

where

$$f_{nl\zeta}(k) = 4\pi i^l \int_0^\infty r^{n+1} e^{-\zeta r} j_l(kr) dr \tag{26}$$

which agrees with expressions given by Geller (1963), Silverstone (1966), Coppens (1997) and others.

CHAPTER IV: *ARTICLE 3* - FAST ANALYTICAL EVALUATION OF INTERMOLECULAR ELECTROSTATIC INTERACTION ENERGIES USING THE PSEUDOATOM REPRESENTATION OF THE ELECTRON DENSITY. III. APPLICATION OF THE EP/MM METHOD TO CALCULATION OF THE ELECTROSTATIC INTERACTION ENERGIES IN MOLECULAR CRYSTALS VIA THE EWALD AND DIRECT SUMMATION TECHNIQUES

Abstract

The previously reported exact potential and multipole moment (EP/MM) method for fast and accurate evaluation of the intermolecular interaction energies using the pseudoatom representation of the electron density (Volkov, A., Koritsanszky T. S. & Coppens, P. (2004). *Chem. Phys. Lett.* **391**, 170–175; Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536; Nguyen, D., & Volkov, A. (2019) *Acta Cryst.* A**75**, 448-464) is extended to calculation of the electrostatic interaction energies in molecular crystals using the two newly developed implementations: i) the Ewald summation (ES) which includes interactions up to the hexadecapolar level and the exact potential correction to account for the short-range electron density penetration effects, and ii) the enhanced EP/MM-based direct summation (DS) which at sufficiently large intermolecular separations replaces the atomic multipole moment approximation to the electrostatic energy with that based on the molecular multipole moments. As in the previous study (Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536), the electron repulsion integral in the exact potential is evaluated analytically using the Löwdin α-function approach. The resulting techniques, incorporated in the XDPROP module of the

software package XD2016, have been tested on several small molecule crystal systems (benzene, L-dopa, paracetamol, amino acids, etc.) and a crystal structure of a 181-atom decapeptide molecule (Z=4) using electron densities constructed via the University at Buffalo Aspherical Pseudoatom Databank (Volkov, A., Li, X., Koritsanszky, T. S. & Coppens, P. (2004). *J. Phys. Chem. A* **108**, 4283-4300). Using a 2015 2.8 GHz Intel Xeon E3-1505M v5 computer processor, a 64-bit implementation of the Löwdin α-function, and one of the higher optimization levels in the GNU Fortran compiler, the Ewald summation method evaluates the electrostatic interaction energy with a numerical precision of at least 10^{-5} kJ/mol under 6 seconds for any of the tested small molecule crystal structures, and in 48.5 seconds for the decapeptide structure. The direct summation approach is competitive in terms of precision and speed with the Ewald summation technique only for crystal structures of small molecules that do not carry a large molecular dipole moment. The electron density penetration effects, correctly accounted for by the two described methods, contribute 28-64% to the total electrostatic energy in the examined systems, and thus cannot be neglected.

4.1 Introduction

Evaluation of electrostatic properties of molecules in crystals is an important component of modeling of crystal structures (Gavezzotti, 2002a; Dunitz & Gavezzotti, 2005). One of the main challenges is to adequately describe the directionality, or anisotropy, of electrostatic forces with a simple and efficient model (Day *et al.*, 2005). By far, the most reliable technique is a fully-periodic quantum mechanical treatment of crystal structures (Pisani & Dovesi, 1980; Saunders, 1984; Causa *et al.*, 1987; Towler, Zupan & Causa, 1996; Blaha *et al.*, 1990; Schwarz, Blaha & Madsen, 2002; Singh, 1994; Dovesi *et al.*, 2005; Clark *et al.*, 2005). Unfortunately, it is somewhat limited by the size of the systems that can be studied and subject to approximations inherent to the level of theory (*i.e.*, Hartree-Fock method, Density Functional Theory, *etc.*) and the basis set expansion (i.e., atomic functions, plane waves, pseudopotentials, *etc.*).

To reduce the computational effort, an accurate quantum mechanical treatment can be applied to the "central" molecule only, while the effects of the surrounding crystal field are approximated by the external potential which adds a perturbation term $\widehat{\mathcal{H}'}$ to the Hamiltonian $\widehat{\mathcal{H}}$ of the "central" molecule (Swerts *et al.*, 2002). The external potential is often represented by a number of point charges positioned around the central molecule (Whitten, McKinnon & Spackman, 2006; Bjornsson & Bühl, 2012). These are related to a class of combined quantum mechanics and molecular mechanics (Warshel & Levitt, 1976; Singh & Kollman, 1986; Field, Bash & Karplus, 1990; Bakowies & Thiel, 1996) or ONIOM (Chung *et al.*, 2015) methods. In more advanced implementations, higher-order atomic multipole moments, *i.e.*, dipoles, quadrupoles, *etc.*, are also included. The success of this approach relies heavily on the accuracy of representation of the external potential, which is intimately related to the electron density (ED) distribution in the crystal.

Electrostatic interactions in a crystal can also be computed from the knowledge of the molecular ED distribution, or its description in terms of molecular/atomic multipole moments (Jensen, 1958; Buckingham, 1959; Stogryn & Stogryn, 1966). The electrostatic interaction energy can then be evaluated via pairwise-type summation of interactions of the "central" molecule with all symmetry-equivalent molecules in the crystal until energy convergence is reached. The method used for computation of pairwise electrostatic energies in the crystal may include either a multipole moment approximation (Buckingham, 1967, 1978; Buckingham, Fowler & Hutson, 1988; Stone, 1996) or a grid-based numerical integration (Gavezzotti, 2002b, 2003, 2005; Ma & Politzer, 2004). In addition, very efficient computational methods, for example the Ewald-type summation (Cummins *et al.*, 1976; de Leeuw, Pertain, & Smith, 1980; Heyes, 1981; Allen & Tildesley, 1987; Williams, 1989; Smith, 1982, 1998; Su & Coppens, 1995; Toukmaji & Board, 1996; Challacombe, White & Head-Gordon, 1997; Abramov *et al.*, 2000; Nymand & Linse, 2000; Frenkel & Smit 2002; Aguado & Madden, 2003; Sagui *et al.*, 2004; Arnold & Holm, 2005; Stenhammar, Trulsson & Linse, 2011; Giese *et al.*, 2015) and a clever workaround known as the Wolf formalism (Wolf *et al.*, 1999; Zahn, Schilling & Kast, 2002; Fennell & Gezelter, 2006; Lamichhane, Gezelter & Newman, 2014; Stenqvist *et al.*, 2015) have been developed. Nevertheless, the major drawbacks of these methods are well known: the multipole approximation breaks down for overlapping charge distributions, while a numerical approach is rather computationally demanding and subject to grid limitations. From our point of view, an ideal method should provide an accurate description of the electron density distribution at any point in space, and should be able to faithfully and efficiently retrieve the electrostatic properties, even when using approximations. In this study, we focus on the pseudoatom representation of the electron density (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996).

In the field of experimental X-ray charge density determination, the electron density (ED) at each point in space $\rho(\mathbf{r})$ is represented by a superposition of atomic-like densities $\rho_a(\mathbf{r})$, called pseudoatoms (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996):

$$f(\mathbf{r}) = \sum_{a} \rho_a(\mathbf{r} - \mathbf{R}_a) \tag{4.1}$$

where $\mathbf{R}_a$ denotes the location of the nucleus of pseudoatom a (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). By far the most widely used aspherical pseudoatom formalism is based on the Hansen-Coppens multipolar model of ED (Hansen & Coppens, 1978; Coppens, 1997) in which each pseudoatom is modelled using the modified Laplace series:

$$\rho_a(\mathbf{r}) = P_{\mathrm{c}}\rho_{\mathrm{c}}(r) + P_{\mathrm{v}}\kappa^3\rho_{\mathrm{v}}(\kappa r) + \sum_{l=0}^{l_{\max}} \kappa'^3 R_l(\kappa' r) \sum_{m=0}^{l} P_{lm\pm} d_{lm\pm}(\theta, \varphi) \tag{4.2}$$

In this representation, $\rho_{\mathrm{c}}(r)$ and $\rho_{\mathrm{v}}(r)$ are the spherically averaged core and valence densities, respectively, precalculated at the Hartree-Fock (Clementi, & Roetti, 1974; Bunge *et al.*, 1992, 1993), Density Functional (Volkov & Macchi, 2006), or Dirac-Slater (Su & Coppens, 1998; Macchi & Coppens, 2001) level of theory. The P_{c} parameter is the core electron population (which is usually kept frozen), while the P_{v} and κ parameters represent the population and the degree of expansion/contraction of the valence shell (Hansen &

Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). The spherical harmonics expansion in the third term describes the aspherical deformation density, $\Delta\rho(\mathbf{r})$, where coefficients $P_{lm\pm}$ are the population parameters, and κ' are the dimensionless adjustment coefficients of the radial functions $R_l(r)$ (Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996). In equation (2), $r = |\mathbf{r} - \mathbf{R}_a|$ represents the radial distance from the pseudoatom center, while the parameters (θ, φ) are the corresponding angular coordinates, *i.e.*, $\mathbf{r} - \mathbf{R}_a \equiv (r, \theta, \varphi)$. The angular factors $d_{lm\pm}(\theta, \varphi)$ are the real density-normalized spherical harmonics (Hansen & Coppens, 1978; Paturle & Coppens, 1988; Coppens, 1997; Michael & Volkov, 2015).

Calculation of the electrostatic intermolecular interaction energies based on the pseudoatom model was pioneered by Spackman (Spackman, 1986a; Spackman, Weber & Craven, 1988) and later extended to crystals in the XDINTER program by Abramov *et al.* (2000). The program XDINTER evaluated the molecular electrostatic energy in an infinite crystal using solely the pseudoatom atomic moments (aMM) (Buckingham, 1967, 1978; Buckingham *et al.*, 1988; Stone, 1996), and allowed evaluation of the exchange-repulsion and dispersion interactions via the atom-atom potentials (Spackman, 1986b; Williams & Cox, 1984; Coombes *et al.* 1996). The electrostatic component evaluation employed the Ewald-type lattice summation for the terms up to the dipole-dipole (Smith, 1982, 1998), while the higher-order contributions were computed by summation of the pairwise interactions over an increasing number of neighbouring unit cells.

The major issue with the XDINTER implementation is the complete neglect of the short-range penetration effects originating from overlap of the pseudoatom charge distributions of the interacting molecules as explained in detail by Spackman (2006) who also suggested an elegant correction with the help of the promolecular (Hirshfeld & Rzotkiewicz, 1974; Spackman & Maslen, 1986) charge densities.

We have solved the problem using the so-called EP/MM approach (Volkov, Koritsanszky & Coppens, 2004), in which the electrostatic interaction energy between pseudoatoms a and b, $E(ab)$[5] located at $R_{ab} < R_{\text{cutoff}}$, with $R_{ab} = |\mathbf{R}_b - \mathbf{R}_a|$ being the distance between the two pseudoatoms and R_{cutoff} is a user-defined parameter (for the second-row atoms, an optimal R_{cutoff} value was found to be 4-5 Å), is evaluated via integration of the exact Coulomb potential (EP) conveniently expanded into the nuclear-nuclear (E_{nn}), electron-nuclear (E_{en}), and electron-electron (E_{ee}) interaction terms:

$$\begin{aligned} E^{\text{EP}}(ab) &= E_{\text{nn}} + E_{\text{en}}(ab) + E_{\text{en}}(ba) + E_{\text{ee}} \\ &= \frac{Z_a Z_b}{R_{ab}} - \int \rho_a(\mathbf{r}_a) V_b^{\text{nuc}}(\mathbf{r}_a) d\mathbf{r}_a - \int \rho_b(\mathbf{r}_b) V_a^{\text{nuc}}(\mathbf{r}_b) d\mathbf{r}_b \\ &\quad + \int\int \frac{\rho_a(\mathbf{r}_a)\rho_b(\mathbf{r}_b)}{|\mathbf{r}_a - \mathbf{r}_b|} d\mathbf{r}_a d\mathbf{r}_b \end{aligned} \tag{4.3}$$

where Z_J and V_J^{nuc} are the nuclear charge and nuclear potential of pseudoatom *J*, and each $E_{\text{en}}(JK)$ term represents the interaction of the electron charge of pseudoatom *J* with the nuclear charge of pseudoatom *K*. In this expression, evaluation of only the last term known as the Coulomb or electron-repulsion integral, ERI (Huzinaga, 1967; Gill, 1994; Helgaker, Jørgensen & Olsen, 2000; Reine, Helgaker & Lindh, 2012), presents a significant challenge. For the long-range interatomic interactions, the integral (3) is approximated by its well-known expansion in terms of the pseudoatom atomic moments (aMM)

[5] While in our previous papers we always denoted the electrostatic interaction energy as E_{es}, in this paper we are using symbol E in order to simplify the notation as we discuss the electrostatic energy only, and also to avoid confusion with the Ewald summation-based (ES) electrostatic energy.

(Buckingham, 1967, 1978; Buckingham, Fowler & Hutson, 1988; Stone, 1996; Coppens, 1997; Kisiel, 2001; Kisiel, 2006):

$$
\begin{aligned}
E^{\mathrm{aMM}}(ab) \ &= T_{ab}\, q_a q_b + T_{ab}^{\alpha}\left(q_a \mu_{\alpha,b} - q_b \mu_{\alpha,a}\right) \\
&\quad + T_{ab}^{\alpha\beta}\left(\frac{1}{3} q_a \theta_{\alpha\beta,b} + \frac{1}{3} q_b \theta_{\alpha\beta,a} - \mu_{\alpha,a}\mu_{\beta,b}\right) \\
&\quad + \ T_{ab}^{\alpha\beta\gamma}\left(\frac{1}{15} q_a \omega_{\alpha\beta\gamma,b} - \frac{1}{15} q_b \omega_{\alpha\beta\gamma,a} - \frac{1}{3}\mu_{\alpha,a}\theta_{\beta\gamma,b} + \frac{1}{3}\mu_{\alpha,b}\theta_{\beta\gamma,a}\right) \qquad (4.4) \\
&\quad + T_{ab}^{\alpha\beta\gamma\delta}\left(\frac{1}{9}\theta_{\alpha\beta,a}\theta_{\gamma\delta,b} + \cdots\right) + \cdots \\
&\quad + T_{ab}^{\alpha\beta\gamma\delta,\varepsilon\zeta\eta\lambda}\left(\frac{1}{11025}\phi_{\alpha\beta\gamma\delta,a}\phi_{\varepsilon\zeta\eta\lambda,b} + \cdots\right)
\end{aligned}
$$

where the summation, according to the Einstein notation, is carried out over the repeated indices, and $T_{ab}^{\alpha\beta\gamma\ldots\omega}$are the interaction tensors (Buckingham, 1967, 1978; Buckingham, Fowler & Hutson, 1988; Stone, 1996; Coppens, 1997) for the pseudoatoms a and b,

$$T_{ab}^{\alpha\beta\gamma\ldots\omega} = \nabla_\alpha \nabla_\beta \nabla_\gamma \cdots \nabla_\omega \frac{1}{\mathbf{R}_{ab}} \qquad (4.5)$$

with $\alpha, b, \gamma \ldots \omega$ representing the Cartesian coordinates $\{x, y, z\}$, and q_i, $\mu_{\alpha,i}$, $\theta_{\alpha\beta,i}$, $\omega_{\alpha\beta\gamma,i}$ and $\phi_{\alpha\beta\gamma\delta,i}$ are the atomic multipole moments of pseudoatom i, $i \in \{a, b\}$ (Coppens, 1997). By convention, q is called either a charge or a monopole (M), μ is a dipole (D), θ – quadrupole (Q), ω – octupole (O), and ϕ – hexadecapole (H). Note a slight deviation in our notation from that used in Buckingham (1967, 1978) and Coppens (1997) as we reserve the upper-case letters for the *molecular* multipole moments (mMM).

In the original EPMM implementation (Volkov, Koritsanszky & Coppens, 2004), the exact potential integral (3) was evaluated numerically (this approach is called the *numerical* exact potential, nEP) on a nuclei-centered grids of points via a three-dimensional (3D) quadrature integration (Becke, 1988a) which hindered the method in terms of speed and accuracy as both are directly related to the number of radial and angular grid points, something that was not fully investigated in Volkov, Koritsanszky & Coppens (2004). Indeed, decreasing the number of grid points speeds up the calculation but may lead to a significant loss of accuracy, and *vice versa*.

In our recent papers (Nguyen, Kisiel & Volkov, 2018; Nguyen & Volkov, 2019), we described two fully-analytical, fast and precise methods for evaluation of the ERI term in integral (3) (these are called the *analytical* exact potential, aEP, methods) using the Löwdin α-function (Lα) (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993) and the Fourier transform (FT) technique (Silverstone, 1966-2014; Geller, 1962-1964; Geller & Griffith, 1964; Harris & Michels, 1967). The exhaustive numerical analysis showed that in order to reach the desired precision of 5×10^{-5} kJ/mol in the electrostatic intermolecular interaction energy in a reasonable amount of computational time, the Löwdin α-function (Lα) method should be evaluated using either 64- or 80-bit floating point arithmetic, while the Fourier transform (FT) must be run in 128-bit precision mode.

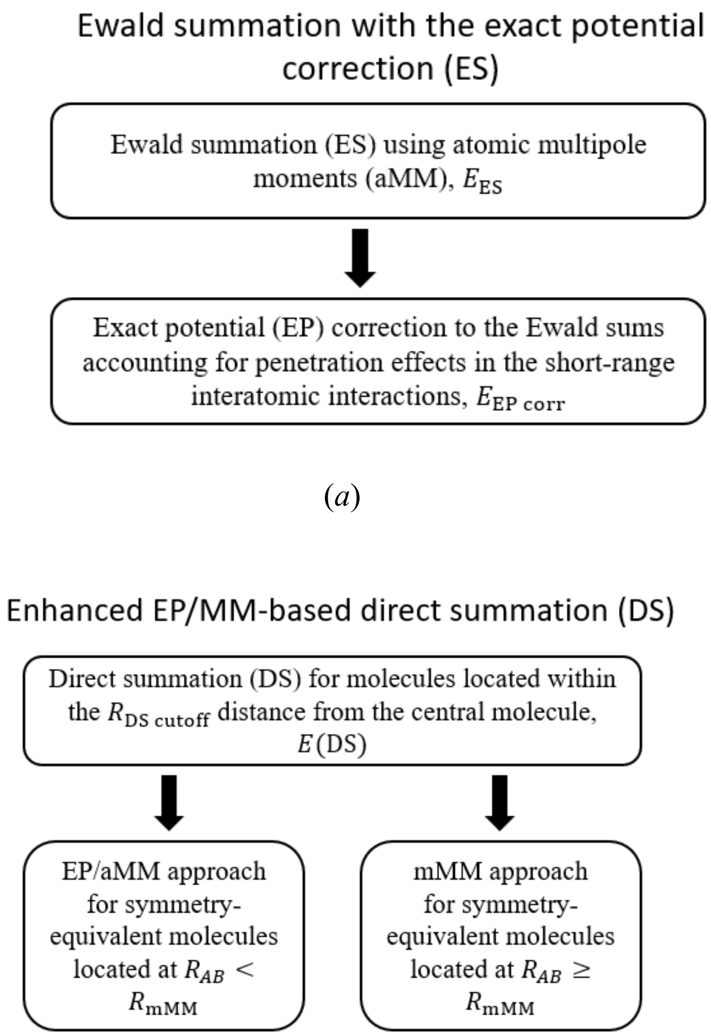
In the following, we present an extension of the analytical exact potential and multipole moment method (EP/MM) to evaluation of the molecular electrostatic energy in an infinite crystal, thus correcting deficiencies in the atomic multipole moment-only XDINTER approach (Abramov *et al.*, 2000). While the electron densities of benchmark compounds used in this study were generated using the University at Buffalo theoretical databank of transferable pseudoatoms, UBDB (Volkov, Li *et al.*, 2004; Dominiak *et al.*, 2007), the approach is directly applicable to the experimentally determined electron densities.

4.2 Methods

In this study, we develop and perform a comprehensive numerical analysis of the Ewald (ES) and direct (DS) summation methods (Figure 4.1) which use the exact potential method (Volkov, Koritsanszky & Coppens, 2004; Nguyen, Kisiel & Volkov, 2018; Nguyen & Volkov, 2019) to correct for the short-range penetration effects originating from the overlap of neighbouring molecular charge densities.

4.2.1 Ewald summation method with the exact potential correction (ES)

Two distinct features of the proposed Ewald summation (ES) method (Figure 4.1a) are (i) the inclusion of multipolar interactions up to the hexadecapolar level, and (ii) the exact potential (EP) correction which accounts for the short-range electron density penetration effects. In the following, we list all the individual multipole moment contributions to the total interaction energy and their abbreviations throughout the paper: M-M (monopole-monopole), M-D (monopole-dipole), M-Q (monopole-quadrupole), M-O (monopole-octupole), M-H (monopole-hexadecapole), D-D (dipole-dipole), D-Q (dipole-quadrupole), D-O (dipole-octupole), D-H (dipole-hexadecapole), Q-Q (quadrupole-quadrupole), Q-O (quadrupole-octupole), Q-H (quadrupole-hexadecapole), O-O (octupole-octupole), O-H (octupole-hexadecapole), and H-H (hexadecapole-hexadecapole) interactions. The resulting energy is denoted as E_{ES}, where ES stands for "Ewald summation".


Ewald summation with the exact potential correction (ES)
Ewald summation (ES) using atomic multipole moments (aMM), E_{ES}
Exact potential (EP) correction to the Ewald sums accounting for penetration effects in the short-range interatomic interactions, $E_{\mathrm{EP\ corr}}$
Enhanced EP/MM-based direct summation (DS)
Direct summation (DS) for molecules located within the $R_{\mathrm{DS\ cutoff}}$ distance from the central molecule, $E(\mathrm{DS})$
EP/aMM approach for symmetry-equivalent molecules located at $R_{AB} < R_{\mathrm{mMM}}$
mMM approach for symmetry-equivalent molecules located at $R_{AB} \geq R_{\mathrm{mMM}}$

(*a*)

(*b*)

Figure 4.1 Schematic diagrams of the implemented (*a*) Ewald (ES) and (*b*) direct (DS) summation methods. In the direct summation method, R_{mMM} and $R_{\mathrm{DS\ cutoff}}$ are user-specified parameters (see Figure 4.2)

The E_{ES} value is calculated following the approach described by Aguado & Madden (2003) who presented explicit expressions for all multipole contributions up to the quadrupolar level. As in most of the Ewald-based summation techniques (Cummins *et al.*, 1976; de Leeuw, Pertain, & Smith, 1980; Heyes, 1981; Allen & Tildesley, 1987; Smith, 1982, 1998; Toukmaji & Board, 1996; Challacombe, White & Head-Gordon, 1997; Nymand & Linse, 2000; Frenkel & Smit 2002; Aguado & Madden, 2003; Sagui, Pedersen & Darden, 2004; Arnold & Holm, 2005; Stenhammar, Trulsson & Linse, 2011; Giese *et al.*, 2015) the total electrostatic energy E_{ES} is represented by a combination of the following contributions: the direct (real) space energy (E_{direct}), the reciprocal (imaginary, or Fourier) space energy (E_{recip}), the self-energy correction (E_{self}), and the surface energy (E_{surf}).

$$E_{\text{ES}} = E_{\text{direct}} + E_{\text{recip}} + E_{\text{self}} + E_{\text{surf}} \tag{4.6}$$

Note that in the conventional Ewald-type summation, the E_{ES} value in equation (4.6) includes interactions between all molecules in the "central" unit cell with all other molecules in the crystal, as well all interatomic interactions in the "central" unit cell. Since we are only interested in the intermolecular electrostatic interaction energy of a symmetry-unique molecule in the "central" unit cell, equation (4.6) is modified as follows:

$$E_{\text{ES}} = \frac{E_{\text{direct}} + E_{\text{recip}} + E_{\text{self}} + E_{\text{surf}}}{M} - E_{\text{intra}} \tag{4.7}$$

where M is the number of molecules in the unit cell, and E_{intra} is the *intramolecular* electrostatic interaction energy of one of the M molecules in "central" unit cell which is evaluated as:

$$E_{\text{intra}} = \frac{1}{2}\sum_{a\in \text{A}}\sum_{b\in \text{A}, b\neq a} E^{\text{aMM}}(ab) \tag{4.8}$$

where a, b are atoms of the symmetry-unique molecule A.

4.2.1.1 The direct (real) space energy term, E_{direct}

The direct (real) space energy term E_{direct} in equations (4.6) and (4.7) accounts for the short-range multipolar interactions screened by a Gaussian distribution (Allen & Tildesley, 1987; Toukmaji & Board, 1996; Nymand & Linse, 2000; Frenkel & Smit 2002; Stenhammar, Trulsson & Linse, 2011) which is needed to achieve rapid convergence. While the E_{direct} formulas for up to Q-Q are given in Aguado & Madden (2003), we chose a more direct approach proposed by Nymand & Linse (2000). The E_{direct} term for any multipolar contribution can be obtained by replacing the interaction tensors $T_{ab}^{\alpha\beta\gamma\ldots\omega}$ in equation (4.4) with their modified/screened counterparts $\hat{T}_{ab}^{\alpha\beta\gamma\ldots\omega}$ (Nymand & Linse, 2000; Aguado & Madden 2003; Stenhammar, Trulsson & Linse, 2011) defined as

$$\hat{T}_{ab}^{\alpha\beta\gamma\ldots\omega} = \nabla_\alpha\nabla_\beta\nabla_\gamma\cdots\nabla_\omega \frac{\operatorname{erfc}({\mathchar'26\mkern-9mu k}R)}{\mathbf{R}_{ab}} = \nabla_\alpha\nabla_\beta\nabla_\gamma\cdots\nabla_\omega \frac{\widehat{1}}{\mathbf{R}_{ab}} \tag{4.9}$$

where ${\mathchar'26\mkern-9mu k}$ is the Ewald smearing / convergence parameter (Nymand & Linse, 2000; Aguado & Madden 2003; Stenhammar, Trulsson & Linse, 2011) and $\operatorname{erfc}(x)$ is the complimentary error function defined by $\operatorname{erfc}(z) \equiv 1 - \operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_z^\infty e^{-t^2}dt$ (equation 7.2.2 in Olver *et al.*, 2019), where the error function $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt$ (equation 7.2.1 in Olver *et al.*, 2019). We note that because ${\mathchar'26\mkern-9mu k}R > 0$, the complimentary error function in equation (6) can be replaced by the incomplete gamma function $\Gamma\left(\frac{1}{2}, {\mathchar'26\mkern-9mu k}^2R^2\right)/\sqrt{\pi}$ (equation 8.4.6 in Olver *et al.*, 2019).

Considering the periodic nature of the Ewald summation method, each direct space energy term must include the translation-related images of the unit cell. For example, the monopole-monopole and monopole-dipole terms become:

$$E_{\text{direct}}^{\text{M-M}} = \frac{1}{2} \sum_{|\mathbf{r}_{ab,\boldsymbol{\mathcal{R}}}|=0}^{r_{\text{cut}}} \sum_{a=1}^{N} \sum_{b=1}^{N}{}^{\dagger} \hat{T}_{ab} q_a q_b \tag{4.10}$$

$$E_{\text{direct}}^{\text{M-D}} = -\frac{1}{2} \sum_{|\mathbf{r}_{ab,\boldsymbol{\mathcal{R}}}|=0}^{r_{\text{cut}}} \sum_{a=1}^{N} \sum_{b=1}^{N}{}^{\dagger} \hat{T}_{ab}^{\alpha} \left(q_a \mu_{\alpha,b} - q_b \mu_{\alpha,a}\right) \tag{4.11}$$

where *N* is the number of atoms in the unit cell, $\mathbf{r}_{ab,\boldsymbol{\mathcal{R}}} = \mathbf{r}_a - (\mathbf{r}_b + \boldsymbol{\mathcal{R}})$, denotes the displacement vector of atom a in the unit cell and the image copy of atom b in the lattice system, where the lattice vector $\boldsymbol{\mathcal{R}}$ is the position of the image copy of the unit cell defined by $\boldsymbol{\mathcal{R}} = n_x \boldsymbol{L}_x + n_x \boldsymbol{L}_y + n_x \boldsymbol{L}_z$ with $(\boldsymbol{L}_x, \boldsymbol{L}_y, \boldsymbol{L}_z)$ being the basis vectors of the unit cell. The symbol † in the summation indicates the exclusion of the self-interaction in the unit cell (*i.e.*, $\mathbf{r}_{ab,\boldsymbol{\mathcal{R}}} = 0$). Finally, in order to avoid numerical issues, the direct space contributions need to be terminated at some distance r_{cut}, the choice of which is discussed below.

4.2.1.2 The reciprocal space energy term, $\boldsymbol{E}_{\textbf{recip}}$

The reciprocal space energy term arises from the need to cancel the Gaussian charge distributions used to dampen interactions when calculating E_{real} (Allen & Tildesley, 1987; Nymand & Linse, 2000; Frenkel & Smit 2002; Aguado & Madden, 2003; Stenhammar, Trulsson & Linse, 2011). Aguado & Madden (2003) described a simple and intuitive approach for the derivation of these terms, and included explicit expressions for up to the quadrupolar level. We have rederived their equations and extended the formulas up to the hexadecapolar level. Consider a unit cell that includes N atoms. Following Aguado & Madden (2003), we use vector $\mathbf{r}_a$ to denote the Cartesian coordinates of atom a in the unit cell ($a = 1 \ldots N$), and vector **h** denotes the reciprocal space vector defined in the original paper as:

$$\mathbf{h} = 2\pi(\bar{\mathbf{H}}^{-1})^{\mathrm{T}}\mathbf{n} \tag{4.12}$$

where $\bar{\mathbf{H}}$ is the orthonormalization matrix ($V = \det(\bar{\mathbf{H}})$, V is the unit cell volume), and **n** is a vector of integers, $\mathbf{n} = (n_x, n_y, n_z)$ (Aguado & Madden, 2003). The reciprocal term for interaction of multipole X with multipole Y is then given by (Aguado & Madden, 2003):

$$E_{\mathrm{recip}}^{\mathrm{X-Y}} = \frac{4\pi}{V}\sum_{h>0}^{h_{\mathrm{cut}}}\left\{\frac{e^{-h^2/4\mathscr{k}^2}}{h^2}(\mathcal{A}_{\mathrm{X-Y}} + \mathcal{B}_{\mathrm{X-Y}})\right\} \tag{4.13}$$

where $h = |\mathbf{h}|$, $\mathscr{k}$ is the Ewald smearing parameter, h_{cut} is the reciprocal-space cutoff parameter (the choice of which is discussed below), and $\mathcal{A}_{\mathrm{X-Y}}$ and $\mathcal{B}_{\mathrm{X-Y}}$ are the auxiliary terms listed in Table 4.1. The derived expressions up to the quadrupolar level agree perfectly with those given in Aguado & Madden (2003). Expressions for the higher-order terms are not too complicated considering that they all are based on the tensor contraction operations.

4.2.1.3 The self-energy term, E_{self}

The self-energy energy term E_{self} corrects for counting the interaction of each particle with itself in reciprocal space. Aguado & Madden (2003) derived simple expressions for the M-M, D-D, M-Q, and Q-Q contributions while also stating that the M-D and D-Q terms are zero “due to symmetry properties of these terms”. Since they mentioned that the derivation of the Q-Q term was “very lengthy”, we abandoned the idea to derive by hand equations for the octupole and hexadecapole terms, and turned to Mathematica (Wolfram Research, 2019).

Table 4.1 Auxiliary terms to be used in equation (4.13) for calculation of the reciprocal space energy $E_{\mathrm{recip}}^{\mathrm{X-Y}}$ between multipoles X and Y. In these expressions, $h_{\alpha\beta} = h_\alpha h_\beta$, $h_{\alpha\beta\gamma} = h_\alpha h_\beta h_\gamma$, and $h_{\alpha\beta\gamma\delta} = h_\alpha h_\beta h_\gamma h_\delta$.

X − Y	$\mathcal{A}_{\mathbf{X-Y}}$	$\mathcal{B}_{\mathbf{X-Y}}$
M-M	$\sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$\sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
M-D	$-2 \sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$2 \sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
D-D	$\sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$\sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
M-Q	$-\frac{2}{3} \sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{3} \sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
D-Q	$\frac{2}{3} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{3} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
Q-Q	$\frac{1}{9} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$\frac{1}{9} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
M-O	$\frac{2}{15} \sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{15} \sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
D-O	$-\frac{2}{15} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{15} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
Q-O	$-\frac{2}{45} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$\frac{2}{45} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
O-O	$\frac{1}{225} \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$	$\frac{1}{225} \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$
M-H	$\frac{2}{105} \sum_a q_a \,\mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$\frac{2}{105} \sum_a q_a \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
D-H	$-\frac{2}{105} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$\frac{2}{105} \sum_a h_\alpha \mu_{\alpha,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
Q-H	$-\frac{2}{315} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{315} \sum_a h_{\alpha\beta} \theta_{\alpha\beta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
O-H	$\frac{2}{1575} \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$-\frac{2}{1575} \sum_a h_{\alpha\beta\gamma} \omega_{\alpha\beta\gamma,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$
H-H	$\frac{1}{11025} \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Cos}[\mathbf{h}.\mathbf{r}_a]$	$\frac{1}{11025} \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a] \sum_a h_{\alpha\beta\gamma\delta} \phi_{\alpha\beta\gamma\delta,a} \mathrm{Sin}[\mathbf{h}.\mathbf{r}_a]$

In the following, we illustrate our derivations for the quadrupole-hexadecapole (Q-H) and octupole-hexadecapole (O-H) and terms only (it is easier to go down the complexity level than up). Following Aguado & Madden (2003), we write the reciprocal space contribution for the interaction between multipoles X and Y as the difference between the total and the direct (real) space energies:

$$E_{\text{self}} = E_{\text{tot}} - E_{\text{direct}} \tag{4.14}$$

We first consider an interaction between an octupole of atom a and hexadecapole on atom b (*i.e.*, the O-H interaction) for which the total and direct space energy contributions are:

$$E_{\text{tot}}^{\text{O-H}} = -\frac{1}{1575}\sum_{a}\sum_{b} \omega_{\alpha\beta\gamma,a} T_{ab}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,b} \tag{4.15}$$

$$E_{\text{real}}^{\text{O-H}} = -\frac{1}{1575}\sum_{a}\sum_{b} \omega_{\alpha\beta\gamma,a} \hat{T}_{ab}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,b} \tag{4.16}$$

where $T_{ab}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta}$ and $\hat{T}_{ab}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta}$ are defined via equation (4.5) and (4.9), respectively. Since we need to calculate the interaction of an atom with itself, we set $a = b$

$$E_{\text{tot}}^{\text{O-H}} = -\frac{1}{1575}\sum_{a} \omega_{\alpha\beta\gamma,a} T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,a} \tag{4.17}$$

$$E_{\text{real}}^{\text{O-H}} = -\frac{1}{1575}\sum_{a} \omega_{\alpha\beta\gamma,a} \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,a} \tag{4.18}$$

and take the difference:

$$
\begin{aligned}
E_{\text{self}}^{\text{O-H}} = E_{\text{tot}}^{\text{O-H}} - E_{\text{real}}^{\text{O-H}} \\
&= \frac{1}{1575}\sum_a \omega_{\alpha\beta\gamma,a} T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,a} \\
&- \frac{1}{1575}\sum_a \omega_{\alpha\beta\gamma,a} \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,a} \\
&= \frac{1}{1575}\sum_a \omega_{\alpha\beta\gamma,a} \left(T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} - \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta}\right) \phi_{\delta\varepsilon\zeta\eta,a} \\
&= \frac{1}{1575}\sum_a \omega_{\alpha\beta\gamma,a} \Delta T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} \phi_{\delta\varepsilon\zeta\eta,a}
\end{aligned}
\tag{4.19}
$$

where we have introduced the "difference" tensor $\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} = T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} - \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta}$. Evaluation of $\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta\eta}$ in the limit of $x \to 0$, $y \to 0$, and $z \to 0$ in Mathematica (Wolfram Research, 2019) gives $\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta\eta} = 0$ which means that the $E_{\text{self}}^{\text{O-H}}$ term vanishes. In fact, $\Delta T^{\alpha\beta\gamma\ldots\omega} = 0$ whenever the order of the tensors is an odd number, which means that $E_{\text{self}}^{\text{X-Y}} = 0$ whenever the sum of the orders of multipoles X and Y is an odd number (M-D, D-Q, M-O, Q-O, D-H, and O-H).

Applying the same approach to a quadrupole-hexadecapole (Q-H) interaction leads to

$$
\begin{aligned}
E_{\text{self}}^{\text{Q-H}} = E_{\text{tot}}^{\text{Q-H}} - E_{\text{real}}^{\text{Q-H}} \\
&= \frac{1}{315}\sum_a \theta_{\alpha\beta,a} T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta} \phi_{\gamma\delta\varepsilon\zeta,a} - \frac{1}{315}\sum_a \theta_{\alpha\beta,a} \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta} \phi_{\gamma\delta\varepsilon\zeta,a} \\
&= \frac{1}{315}\sum_a \theta_{\alpha\beta,a} \Delta T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta} \phi_{\gamma\delta\varepsilon\zeta,a}
\end{aligned}
\tag{4.20}
$$

where $\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta} = T_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta} - \hat{T}_{aa}^{\alpha\beta\gamma\delta\varepsilon\zeta}$. Evaluation of $\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta}$ in the limit of $x \to 0$, $y \to 0$, and $z \to 0$ Mathematica (Wolfram Research, 2019) gives only a few non-zero elements:

$$
\Delta T^{\alpha\alpha\alpha\alpha\alpha\alpha} = -\frac{240k^7}{7\sqrt{\pi}} \qquad \Delta T^{\alpha\alpha\alpha\alpha\beta\beta} = -\frac{48k^7}{7\sqrt{\pi}} \qquad \Delta T^{\alpha\alpha\beta\beta\gamma\gamma} = -\frac{16k^7}{7\sqrt{\pi}} \tag{4.21}
$$

We now define tensor $\breve{T}^{\alpha\beta\gamma\delta\varepsilon\zeta} = -\frac{1}{315}\Delta T^{\alpha\beta\gamma\delta\varepsilon\zeta}$ where the factor $-\frac{1}{315}$ comes from equation (4.20), and write the $E_{\text{self}}^{\text{Q−H}}$ contribution simply as:

$$E_{\text{self}}^{\text{Q−H}} = \sum_{a} \theta_{\alpha\beta,a} \breve{T}^{\alpha\beta\gamma\delta\varepsilon\zeta} \phi_{\gamma\delta\varepsilon\zeta,a} \tag{4.22}$$

The expressions derived using the described approach are listed in Table 4.2. Note that the sign of our expressions is different from that used in Aguado & Madden (2003). The sign convention adopted here is consistent with equations (4.6) and (4.7) and, for example, in Stenhammar, Trulsson & Linse (2011) according to which the E_{self} contribution is *added* to the sum of the other terms, while Aguado & Madden (2003) *subtract* theirs. Our results agree perfectly with the previously reported values for the M-M, D-D, and M-Q terms (Aguado & Madden, 2003; Stenhammar, Trulsson & Linse, 2011). For the Q-Q self-energy correction term, our equation is slightly different and more complicated than that given in Aguado & Madden (2003), though several numerical tests showed that the two equations give the same result. Note that the $E_{\text{self}}^{\text{M−Q}} = \breve{T} \sum_{a} q_a \text{Tr}(\theta_{\alpha\beta,a})$ term includes a trace of the quadrupole moment, which means it is always zero when using the pseudoatom moments as they are traceless by definition. Likely for the same reason, the dipole-octupole, monopole-hexadecapole, and quadrupole-hexadecapole terms in our benchmark systems have always evaluated to zero as well. Thus, the only important E_{self} contributions are M-M, D-D, Q-Q, O-O, and H-H.

Table 4.2 The self-energy correction expressions up to the hexadecapolar (H-H) level. $\mathscr{k}$ is the Ewald smearing / convergence parameter. The sign convention adopted here is consistent with equations (4.6) and (4.7) for the Ewald summation-based energy. The M-D, D-Q, M-O, Q-O, D-H, and O-H terms are zero.

$\mathbf{X-Y}$	$E_{\text{es}}^{\text{X}-\text{Y}}(\text{self})$	**Non-zero elements of tensors** $\breve{T}^{\alpha\beta\gamma\ldots\omega}$	
M-M	$\breve{T}\sum_a q_a^2$	$\breve{T}=-\frac{\mathscr{k}}{\sqrt{\pi}}$	
D-D	$\breve{T}\sum_a \mu_a^2$	$\breve{T}=-\frac{2}{3}\frac{\mathscr{k}^3}{\sqrt{\pi}}$	
M-Q	$\breve{T}\sum_a q_a\mathrm{Tr}(\theta_{\alpha\beta,a})$	$\breve{T}=\frac{4}{9}\frac{\mathscr{k}^3}{\sqrt{\pi}}$	
Q-Q	$\sum_a \theta_{\alpha\beta,a}\breve{T}^{\alpha\beta\gamma\delta}\theta_{\gamma\delta,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha}=-\frac{12}{45}\frac{\mathscr{k}^5}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\beta\beta}=-\frac{4}{45}\frac{\mathscr{k}^5}{\sqrt{\pi}}$
D-O	$\sum_a \mu_{\alpha,a}\breve{T}^{\alpha\beta\gamma\delta}\omega_{\beta\gamma\delta,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha}=\frac{24}{75}\frac{\mathscr{k}^5}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\beta\beta}=\frac{8}{75}\frac{\mathscr{k}^5}{\sqrt{\pi}}$
O-O	$\sum_a \omega_{\alpha\beta\gamma,a}\breve{T}^{\alpha\beta\gamma\delta\varepsilon\zeta}\omega_{\delta\varepsilon\zeta,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha\alpha\alpha}=-\frac{120}{1575}\frac{\mathscr{k}^7}{\sqrt{\pi}}$ $\breve{T}^{\alpha\alpha\beta\beta\gamma\gamma}=-\frac{8}{1575}\frac{\mathscr{k}^7}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\alpha\alpha\beta\beta}=-\frac{24}{1575}\frac{\mathscr{k}^7}{\sqrt{\pi}}$
M-H	$\sum_a q_a\breve{T}^{\alpha\beta\gamma\delta}\phi_{\alpha\beta\gamma\delta,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha}=-\frac{24}{525}\frac{\mathscr{k}^5}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\beta\beta}=-\frac{8}{525}\frac{\mathscr{k}^5}{\sqrt{\pi}}$
Q-H	$\sum_a \theta_{\alpha\beta,a}\breve{T}^{\alpha\beta\gamma\delta\varepsilon\zeta}\phi_{\gamma\delta\varepsilon\zeta,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha\alpha\alpha}=\frac{240}{2205}\frac{\mathscr{k}^7}{\sqrt{\pi}}$ $\breve{T}^{\alpha\alpha\beta\beta\gamma\gamma}=\frac{16}{2205}\frac{\mathscr{k}^7}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\alpha\alpha\beta\beta}=\frac{48}{2205}\frac{\mathscr{k}^7}{\sqrt{\pi}}$
H-H	$\sum_a \phi_{\alpha\beta\gamma\delta,a}\breve{T}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta\lambda}\phi_{\varepsilon\zeta\eta\lambda,a}$	$\breve{T}^{\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha}=-\frac{560}{33075}\frac{\mathscr{k}^9}{\sqrt{\pi}}$ $\breve{T}^{\alpha\alpha\alpha\alpha\beta\beta\beta\beta}=-\frac{48}{33075}\frac{\mathscr{k}^9}{\sqrt{\pi}}$	$\breve{T}^{\alpha\alpha\alpha\alpha\alpha\alpha\beta\beta}=-\frac{80}{33075}\frac{\mathscr{k}^9}{\sqrt{\pi}}$ $\breve{T}^{\alpha\alpha\alpha\alpha\beta\beta\gamma\gamma}=-\frac{16}{33075}\frac{\mathscr{k}^9}{\sqrt{\pi}}$

4.2.1.4 The surface term, E_{surf}

The surface energy term E_{surf} accounts for the nature of the boundary of an infinite periodic lattice at infinity (Frenkel & Smit, 2002; Sala *et al.*, 2010; Stenhammar, Trulsson & Linse, 2011; Allen & Tidesley, 2017). In the Ewald summation techniques, the two extreme cases are usually considered (Frenkel & Smit, 2002; Sala, Guàrdia & Masia, 2010; Stenhammar, Trulsson & Linse, 2011; Allen & Tidesley, 2017): vacuum conditions (dielectric constant / relative permittivity ϵ of the surrounding medium is 1) and tinfoil conditions (when the sample is surrounded by a good conductor the dielectric constant ϵ is infinite). The fluctuating dipole moment of the unit cell gives rise to a surface charge at the boundary of the sphere, which, in turn, is responsible for a homogeneous depolarizing field (Frenkel & Smit, 2002). The surface energy term E_{surf} is related to work that must be performed against this depolarizing field to recover the net polarization of the sample (Frenkel & Smit, 2002). Thus, the correction is important for samples with a large dipole moment in the unit cell embedded in a vacuum condition, and can be ignored if the periodic system is surrounded by a conductor since the depolarizing field vanishes (Frenkel & Smit, 2002; Allen & Tidesley, 2017). The surface term implemented in our code is based on the formulas given by Stenhammar, Trulsson & Linse (2011; see equation 26 in their paper). We note that since the E_{surf} is related to the dipole moment of the unit cell, only monopolar and dipolar functions of the pseudoatoms contribute to this correction. The default value of the dielectric constant in our implementation is 1, but the user has an option to specify his/her own value. Finally, we point out that the E_{surf} term with the dielectric constant of the exterior surface $\epsilon = 1$ (vacuum condition) was shown by Nymand & Linse (2000) to yield identical results to the direct summation method.

4.2.1.5 The exact potential correction, $E_{\text{EP corr}}$

The Ewald summation method described above neglects the short-range penetration effects originating from overlap of the neighbouring charge distributions. The issue is solved by applying a correction, $E_{\text{EP corr}}$, calculated using the EP/MM method (Volkov, Koritsanszky & Coppens, 2004; Nguyen, Kisiel & Volkov, 2018; Nguyen & Volkov, 2019). The interactions between atoms of the "central" molecule and all those of the first coordination sphere that are located below the user-specified distance R_{aMM} (*i.e.*, $R_{ab} < R_{\text{aMM}}$; note that R_{aMM} was called R_{cutoff} in our previous publications, and was estimated to be 4-5 Å for the first- and second-row atoms) use *both* the exact potential (EP, equation 4.3) and the multipole approximation using the atomic moments (aMM, equation 4.4). The value of $E_{\text{EP corr}}$ is simply the sum of differences between E^{EP} and E^{aMM} over all short-range interatomic interactions.

The final value of the electrostatic interaction energy for the EP-corrected Ewald summation method, $E(\text{ES})$, is calculated by adding the EP-based correction, $E_{\text{EP corr}}$, to the Ewald sum, E_{ES}:

$$E(\text{ES}) = E_{\text{ES}} + E_{\text{EP corr}} = \frac{E_{\text{direct}} + E_{\text{recip}} + E_{\text{self}} + E_{\text{surf}}}{M} - E_{\text{intra}} + E_{\text{EP corr}} \qquad (4.23)$$

4.2.1.6 Selection of the Ewald summation parameters

Accuracy/precision and performance of the Ewald summation-based calculations, in practice, depend on the choice of three parameters:

(1) the Ewald smearing/convergence parameter $\mathscr{k}$,

(2) the direct (real) space cut-off r_{cut}, and

(3) the reciprocal space cut-off h_{cut}.

The Ewald convergence parameter $\mathscr{k}$ defines the relative rate of convergence between the direct and reciprocal sums (Toukmaji & Board, 1996; Smith, 1982, 1998). The two cut-off terms r_{cut} and h_{cut} are used to avoid unnecessary calculations since only terms satisfying $|\boldsymbol{r}_{ab\boldsymbol{\mathcal{R}}}| < r_{\mathrm{cut}}$ in the direct space part and $h = |\mathbf{h}| < h_{\mathrm{cut}}$ for the reciprocal space part need to be evaluated. As $\mathscr{k}$ is increased, the direct space energy converges more quickly, and thus needs a smaller value of r_{cut}, whereas the reciprocal space part converges more slowly and needs a larger h_{cut} value. Conversely, as $\mathscr{k}$ is decreased, the converge will occur more slowly in the direct space part and more quickly in the reciprocal space part. In this case, r_{cut} should be larger, but h_{cut} should be smaller. Many studies (Smith, 1986; Allen & Tildesley, 1987; Rycerz & Jacobs, 1992; Kolafa & Perram, 1992; Toukmaji & Board, 1996) have shown that it is possible to find an optimal value for $\mathscr{k}$ (and hence, optimal values for r_{cut} and h_{cut}) for the desired precision of the resulting energy, ε. However, it turns out that many of those published values are either (*i*) too specific, *e.g.*, just applied for systems with a cubic unit cell, (*ii*) inconsistent, and (*iii*) too empirical. To circumvent the problem, we used the expressions proposed by Jackson and Catlow (1982) and Leslie (1987) to determine the optimal values for $\mathscr{k}$, r_{cut}, h_{cut}:

$$\mathscr{k} = \sqrt{\pi}\left(\frac{N}{V^2}\right)^{1/6} \qquad (4.24)$$

$$r_{\text{cut}} = \frac{\sqrt{-\ln\varepsilon}}{\mathscr{k}} \qquad (4.25)$$

$$h_{\text{cut}} = 2\mathscr{k}\sqrt{-\ln\varepsilon} \qquad (4.26)$$

where N is the number of atoms in the unit cell, V is the unit cell volume, and ε is the desired precision of the resulting energy. For the sake of achieving sufficiently accurate results, we set $\varepsilon = 10^{-18}$ in the implementation due to the 64-bit fixed numerical precision (*aka* double-precision) being on the order of 10^{-17} (Oracle Corporation, 2017).

In practice, $h_{\max}$, an integer defining the summation range and the maximum number of vectors in the reciprocal space, is commonly used for simplicity instead of h_{cut}. The parameter $h_{\max}$ is determined by:

$$\boxed{h_{\max} = \left\lfloor \frac{h_{\text{cut}}}{\pi} \times \max\left(|\boldsymbol{L}_x|, |\boldsymbol{L}_y|, |\boldsymbol{L}_z|\right) \right\rfloor + 1 \qquad (4.22)}$$

where $\lfloor s \rfloor$ denotes the greatest integer less than or equal to s (*i.e.*, the result of the floor function).

More complex versions of expression (4.24) with an extra factor w governing the computational expense of direct and reciprocal space summations have also been presented (Gale, 1997; Fincham, 1994; Matthey, 2005). Since the numerical value of w may vary between different platforms and implementations, and because the computational performance of the Ewald summation in small systems was found to be relatively insensitive to the value of w (Gale, 1997), we have elected not to include it in our equation for $\mathscr{k}$.

4.2.2 Enhanced EP/MM-based direct summation method (DS)

The direct summation EP/MM-based algorithm (Figures 4.1 and 4.2) extends the "standard" EP/MM method for molecules at larger separations by replacing the atomic (pseudoatom) multipole moments (aMM) $q, \mu, \theta, \omega, \phi$ in equation (4.4) with molecular multipole moments (mMM) $Q, M, \Theta, \Omega, \Phi$:

$$\begin{aligned} E^{\mathrm{mMM}}(AB) = {} & T_{AB}\, Q_A Q_B + T_{AB}^{\alpha}\left(Q_A M_{\alpha,B} - Q_B M_{\alpha,A}\right) \\ & + T_{AB}^{\alpha\beta}\left(\frac{1}{3} Q_A \Theta_{\alpha\beta,B} + \frac{1}{3} Q_B \Theta_{\alpha\beta,A} - M_{\alpha,A} M_{\beta,B}\right) \\ & + T_{AB}^{\alpha\beta\gamma}\left(\frac{1}{15} Q_A \Omega_{\alpha\beta\gamma,B} - \frac{1}{15} Q_B \Omega_{\alpha\beta\gamma,A} - \frac{1}{3} M_{\alpha,A} \Theta_{\beta\gamma,B}\right. \\ & \left. + \frac{1}{3} M_{\alpha,B} \Theta_{\beta\gamma,A}\right) + T_{AB}^{\alpha\beta\gamma\delta}\left(\frac{1}{9} \Theta_{\alpha\beta,A} \Theta_{\gamma\delta,B} + \cdots\right) + \cdots \\ & + T_{AB}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta\lambda}\left(\frac{1}{11025} \Phi_{\alpha\beta\gamma\delta,A} \Phi_{\varepsilon\zeta\eta\lambda,B} + \cdots\right) \end{aligned} \tag{4.28}$$

where $T_{AB}^{\alpha\beta\gamma\ldots\omega}$ are the interaction tensors (Buckingham, 1967, 1978; Buckingham, Fowler & Hutson, 1988; Stone, 1996; Coppens, 1997) for the molecules A and B $\left(\nabla_\alpha \nabla_\beta \nabla_\gamma \cdots \nabla_\omega \mathbf{R}_{AB}^{-1}\right)$, where $\mathbf{R}_{AB} = \mathbf{R}_B - \mathbf{R}_A$, and $\mathbf{R}_A$ and $\mathbf{R}_B$ are the locations of the geometrical centers of molecules A and B, respectively. When a generated symmetry-equivalent molecule is located below a user-specified threshold (*i.e.*, $R_{AB} < R_{\mathrm{mMM}}$, where R_{mMM} is a user-specified parameter), the energy is evaluated via the standard EP/aMM method: interactions between atoms separated at $R_{ab} \geq R_{\mathrm{aMM}}$ are evaluated via equation (4.4), otherwise either the Löwdin α-function (Lα) or the Fourier transform (FT) method is used to analytically evaluate the electron repulsion integral in the exact potential (equation 4.3). At larger intermolecular separations ($R_{AB} \geq R_{\mathrm{mMM}}$) the numerous atom-atom interactions can be approximated by a single evaluation of E^{mMM} between a set of molecular multipoles moments of the “central” molecule and that of the symmetry-equivalent one (Figure 4.2). Provided that the selected value of R_{mMM} is large enough, no precision in the resulting energy value should be lost. At the same time, replacing E^{aMM} with E^{mMM} is expected to significantly speed up the calculations, especially for molecules that contain many atoms.

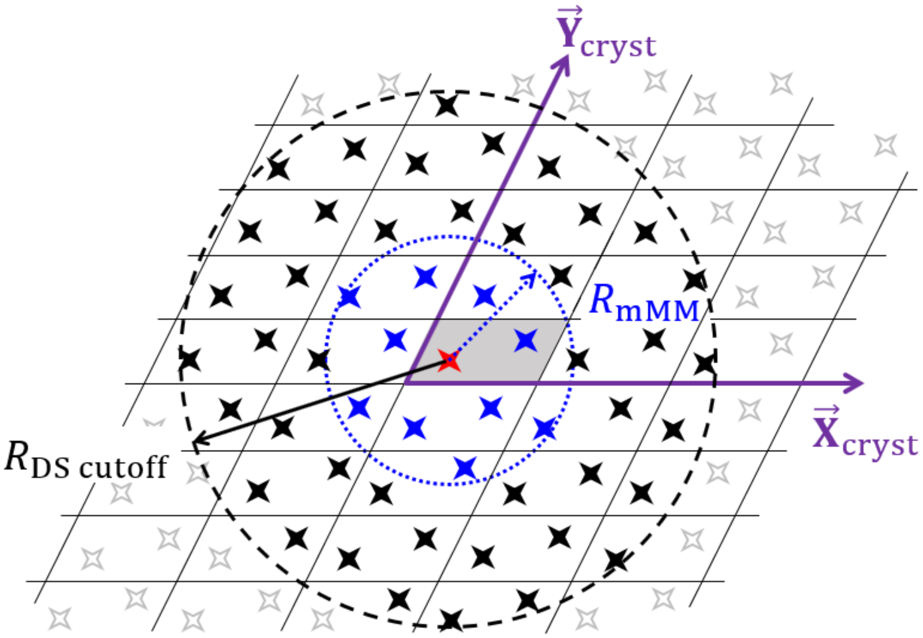

$\vec{\mathbf{Y}}_{\mathrm{cryst}}$
R_{mMM}
$\vec{\mathbf{X}}_{\mathrm{cryst}}$
$R_{\mathrm{DS\ cutoff}}$

Figure 4.2 A schematic schematic diagram explaining the meaning of the R_{mMM} and $R_{\mathrm{DS\ cutoff}}$ parameters in the EP/MM-based direct summation (DS) algorithm. Each paralepidid represents a unit cell in the crystal lattice with the axes $\vec{\mathbf{X}}_{\mathrm{cryst}}$ and $\vec{\mathbf{Y}}_{\mathrm{cryst}}$. The shaded paralepidid represents the “central” unit cell. The symmetry-independent molecule A in the “central” unit cell is identified with ✦. The symmetry-equivalent molecules B whose centers are located below the R_{mMM} distance from the center of molecule A, i.e. $R_{AB} < R_{\mathrm{mMM}}$, are identified with ✦. Interactions to those molecules are evaluated using either the EP or aMM approach – it depends on the inter*atomic* distance. Interactions to molecules located at $R_{\mathrm{mMM}} \leq R_{AB} \leq R_{\mathrm{DS\ cutoff}}$, shown with ✦, are evaluated using the mMM method. Molecules located at $R_{AB} > R_{\mathrm{DS\ cutoff}}$ (✧) are excluded from the analysis.

We note that for electroneutral molecules with a relatively small net dipole moment the direct sum evaluated using the molecular multipole moment (mMM) approach is expected to converge well as all interaction terms that include monopoles will be zero, those with the dipoles will be small, and the remaining terms converge quickly. In such cases, we expect to see an excellent agreement between the energy values calculated via the Ewald summation, $E(\mathrm{ES})$, and the direct summation, $E(\mathrm{DS})$.

4.3 Implementation details

4.3.1 General

The Ewald and direct summation algorithms described in the previous sections have been implemented in the 64-bit precision (*aka* double-precision, DP, or binary64) in-house version of program XDPROP, a part of the XD2016 package (Volkov, Macchi *et al.*, 2016). As discussed in Nguyen, Kiesel & Volkov (2018), the interaction tensors $T^{\alpha\beta\gamma\ldots\omega}$ (equation 4.5) up to order eight (which are needed to account for interactions up to the hexadecapole-hexadecapole level) are calculated using a highly-optimized subroutine from the program MIN16 (Kisiel, 2001; Kisiel, 2006).

While Nymand & Linse (2000) derived a recursive formula for evaluation of $\widehat{\frac{1}{\mathbf{R}_{ab}}}$ for any order [beware, there are typos in that expression in Nymand & Linse (2000), Aguado & Madden (2003), and Laino & Hutter (2008) – the correct equation is given in Stenhammar, Trulsson & Linse (2011)], we used Mathematica (Wolfram Research, 2019) to derive an expression for each component of all the $\hat{T}_{ab}^{\alpha\beta\gamma\ldots\omega}$ tensors (equation 4.9) up to order eight ($\hat{T}^{\alpha\beta\gamma\delta\varepsilon\zeta\eta\lambda}$) and convert them to the Fortran code (using Mathematica's function `FortranForm[]`), and then manually optimized each formula in the resulting Fortran code. As discussed below, the Fortran-based Ewald summation code is extremely fast, which makes the loss of generality the only drawback of this approach. That said, if we ever need to perform the Ewald summation for interactions beyond the hexadecapolar level, we can easily replace the existing procedure in our code with the recursion approach.

The molecular multipole moments up to the hexadecapole level are calculated from the atomic (pseudoatom) moments as described in Coppens (1997) and Tsirelson & Ozerov (1996). The direct summation (DS) method convergence is monitored by printing out the energy values calculated within a concentric shell of a user-specified radius (in this study, a value of 10 Å was used in all calculations).

4.3.2 Numerical precision

It is well known that the numerical precision used in calculations may significantly affect the final result. In our previous study (Nguyen & Volkov, 2019), we carefully analyzed the effect of the fixed-precision computer arithmetic on the values of the resulting electrostatic interaction energies in molecular dimers, and have concluded that both the 80-bit implementation of the Löwdin α-function (Lα) and the 128-bit Fourier transform (FT) method provide an excellent balance between the precision (10^{-8} kJ/mol or better) and speed. However, even the 64-bit Löwdin α-function was able to reproduce the energy values of dimers within 10^{-7} kJ/mol all the while being noticeably faster.

In this study, we have analysed the performance of only the 64- and 80-bit precision implementations of the Löwdin α-function for evaluation of the electron repulsion integrals in the exact potential component of the EP/MM method. A single 128-bit precision Fourier transform-based calculation was performed for each compound to check the 80-bit Löwdin α-function energy value, and no differences were found between energies from the two calculations printed out in kJ/mol to eight decimal points.

Finally, a single test run for each benchmark compound included the calculation of the interaction tensors $T^{\alpha\beta\gamma\dots\omega}$ using the 80-bit numerical precision. It was done in order to check the numerical performance of the 64-bit calculations of $T^{\alpha\beta\gamma\dots\omega}$ in all other runs. No differences were found in the energy values printed out in kJ/mol to eight decimal digits.

4.3.3 The computer hardware and Fortran compiler details

The described Fortran implementation has been tested on an x86_64 computer platform. The code was compiled and run on a 64-bit OpenSUSE Leap 15.1 Linux computer equipped with a 2015 quad-core / eight-thread Intel Xeon E3-1505M v5 (Skylake microarchitecture) central processing unit (CPU) and 64 GB of the DDR4 2133 memory. While the base frequency of the Xeon E3-1505M v5 is only 2.8 GHz, all calculations were performed in the "Turbo Boost" frequency mode of approximately 3.7 GHz. The single-thread Passmark score (PassMark Software, 2020) of this processor (1915) is somewhat higher than that of the AMD FX-8350 (1510) and Opteron 6348 CPU (1266) used in our previous studies (Nguyen, Kiesel & Volkov, 2018; Nguyen & Volkov, 2019). At the same time, it is well below that of the recently released AMD Ryzen 7 3700X (2906) and Intel Core i9-9900KS (3009) processors (PassMark Software, 2020).

The Fortran code was compiled using GNU Fortran 9.2.1 (Free Software Foundation, 2019) with the three code optimization flags explored: $-\mathrm{O2}$, $-\mathrm{O3}$, and $-\mathrm{Ofast}$. The $-\mathrm{O2}$ flag is usually the safest level of optimization that improves the speed of the generated code while maintaining the desired precision (Free Software Foundation, 2019). The $-\mathrm{O3}$ flag is usually also safe when it comes to numerical precision though the compilation time may increase in order to produce a faster code via rearrangement and vectorization of the loops. In the previous publication, we discarded the $-\mathrm{Ofast}$ compilation flag, one of the highest optimization levels in GFortran which turns off strict IEEE/ISO standard compliance (Free Software Foundation, 2019), as it reduced precision of individual ERI integrals. However, in this study, we have included this option to gauge the precision-speed trade-off.

4.4 Benchmark systems

The algorithms have been tested on ten molecular crystal structures:

a) amino-acids L-serine , SER (Kistenmacher, Rand & Marsh, 1974), α-glycine, GLY (Destro *et al.*, 2000), L-valine , VAL (Dalhus & Görbitz, 1996), L-alanine, ALA (Destro, Marsh & Bianchi, 1988) in their zwitterionic forms, and the closely related structure of γ-aminobutyric acid (ABA) (Weber, Craven & McMullan, 1983);

b) structures of N-acetylglycine, ACG (Mackay, 1975), L-dopa, LDOPA (Howard *et al.*, 1995), L-(+)-lactic acid, LAC (Schouten, Kanters & Krieken, 1994), paracetamol, PARA (Wilson, 2000), and benzene, BENZ (Bacon, Curry & Wilson, 1964).

The main criteria for selection of the test systems were the molecular size, accuracy of the crystal structure determination (neutron data preferred), and availability of the atom types in the University at Buffalo pseudoatom databank (Volkov, Koritsanszky & Coppens, 2004; Dominiak *et al.*, 2007). As in our previous studies, in crystal structures determined using the spherical-atom refinement of X-ray data, the X−H bonds (where X = C, N, O) were extended to standard neutron distances (Allen *et al.*, 1995). Crystallographic data for the test systems are given in Table S4.1 of the Supplementary material (Appendix A).

4.5 Theoretical calculations

Theoretical calculations were performed using the experimental geometries in order to estimate molecular dipole moments of the benchmark compounds in a gas phase and the solid state. Theoretical dipole moments were then compared to the values obtained from the University at Buffalo pseudoatom databank (Volkov, Koritsanszky & Coppens, 2004; Dominiak *et al.*, 2007).

Gas-phase wavefunctions of molecular structures were calculated with the Gaussian09 suite of programs (Frisch *et al.*, 2004) at the B3LYP (Becke, 1988b, 1993, Lee, Yang & Parr, 1988) and MP2 (Møller & Plesset, 1934, Head-Gordon, Pople & Frisch, 1988) levels of theory using the 6-31G** (Hariharan & Pople, 1973) basis set. All Gaussian09 calculations were performed with the tight SCF convergence criteria.

Theoretical fully-periodic calculations were performed in the CRYSTAL98 program (Saunders *et al.*, 1999) at the B3LYP/6-31G** level of theory. Molecular dipole moments in the crystalline phase were evaluated using the TOPOND98 program (Gatti, 1999) via the Quantum Theory of Atoms in Molecules (Bader, 1990).

4.6 Results and Discussion

In all of the following sections except for Section 4.6.6 (in which the effects of precision of evaluation of the exact potential (EP) correction, and the compiler optimization options are discussed) all numerical values were obtained with the EP correction evaluated using the 80-bit implementation of the Löwdin α-function, and the code compiled using the $-\mathrm{O2}$ flag, the safest optimizing GFortran compiler option (Free Software Foundation, 2019).

In the first section, we discuss the convergence issues of the direct summation (DS) method (which has been available in XDPROP since 2007, albeit with the numerical evaluation of the exact potential) that eventually forced us to implement the Ewald summation (ES).

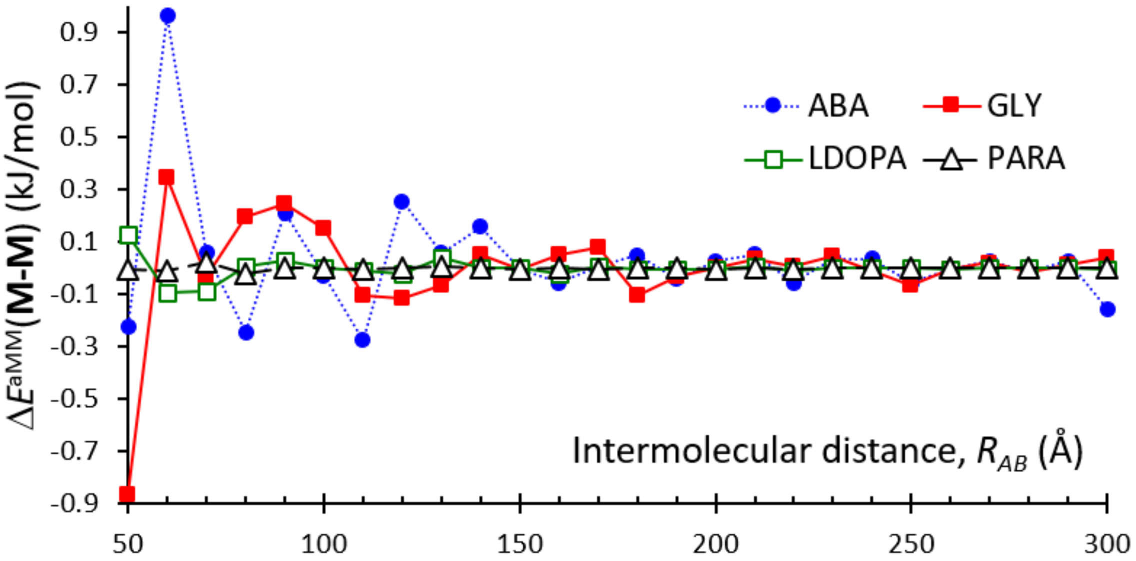
4.6.1 Energy convergence in the direct summation (DS) method

Based on the rate of convergence of the DS energies as a function of the upper spherical summation limit $R_{\mathrm{DS\,cutoff}}$ the benchmark compounds can be roughly separated into two groups. The group with the "fast" energy convergence includes ACG, BENZ, LAC, LDOPA, PARA, while ABA, ALA, GLY, SER and VAL can be classified as "slow" convergences.

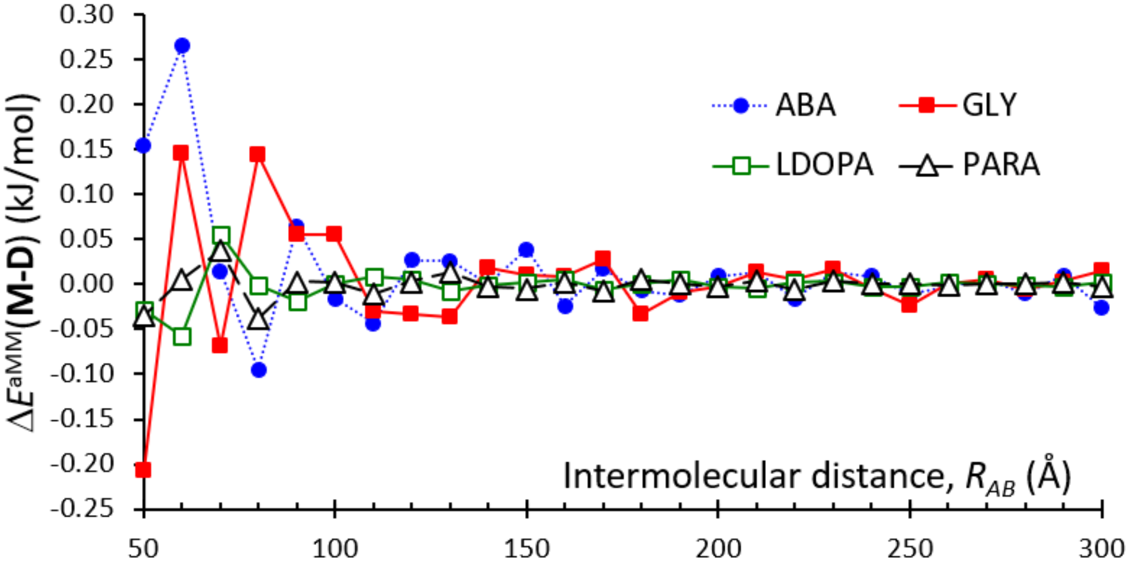
The convergence of the standalone E^{aMM}-based direct summation (DS) method, DS/aMM (equation 4.4), is shown in Figure 4.3 for two compounds from each group (slow: ABA and GLY, fast: LDOPA and PARA). For LDOPA, PARA and the other compounds in their group, all contributions, including the notorious monopole-monopole term converge well within 200 Å. For the slow convergence compounds, relatively small (< 0.2 kJ/mol) fluctuations persist beyond 200 Å for the monopole-monopole (M-M, ~R^{-1}) term. The largest one with a magnitude of 0.16 kJ/mol is observed for ABA in the 290-300 Å shell. As expected, the monopole-dipole and dipole-dipole interactions show a better convergence of 0.05 kJ/mol and 0.01 kJ/mol, respectively. However, the difference in the rate of energy convergence for the two groups is still evident.

These results are somewhat unexpected since it is well accepted that "*the tail correction to the potential energy diverges, unless the potential energy function ... decays faster than r^{-3}*" (Frenkel & Smit, 2002). While the monopole-dipole and dipole-dipole terms decay as ~R^{-2} and ~R^{-3}, respectively, their convergence in our benchmark compounds is reasonable. This likely because the pseudoatoms in the University at Buffalo databank are not too highly polarized.

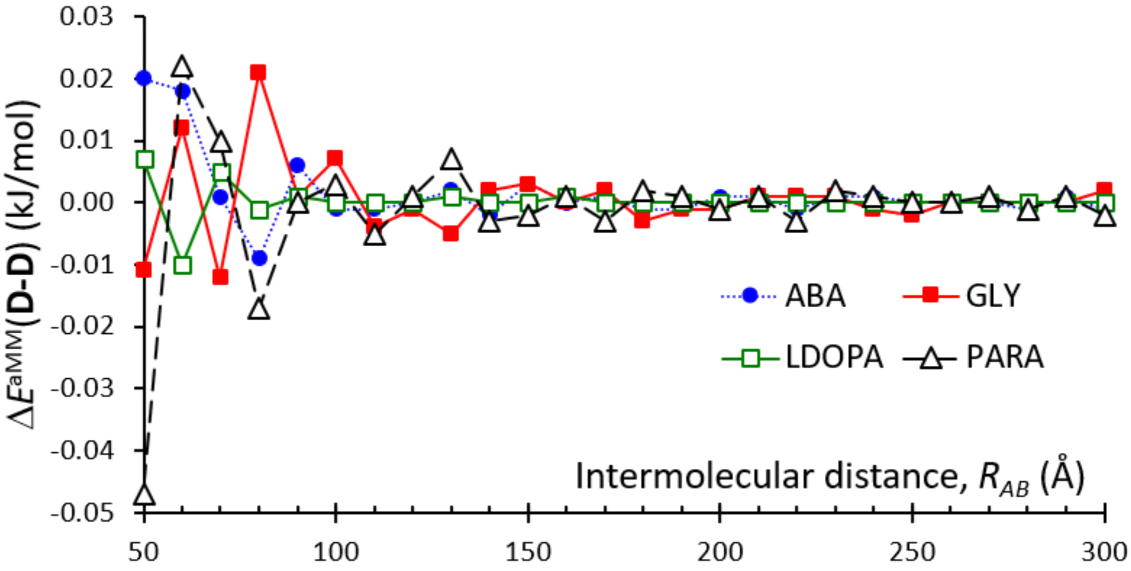
While the DS/aMM calculations show a very decent (certainly exceeding our expectations) convergence, these calculations are extremely time-consuming. For example, the DS/aMM calculation for ABA included ~2.1×10^8 atom-atom interactions and took **50 minutes** on a Xeon E3-1505M v5 processor while the LDOPA calculation took even longer: evaluation of ~3.3×10^8 atom-atom interactions used **76 minutes**. It is clear that the DS/aMM calculations quickly become cost-prohibitive for larger systems.


ΔE^{aMM}(**M-M**) (kJ/mol)
0.9
0.7
0.5
0.3
0.1
-0.1
-0.3
-0.5
-0.7
-0.9
50
100
150
200
250
300
Intermolecular distance, R_{AB} (Å)
ABA
GLY
LDOPA
PARA

(*a*)


ΔE^{aMM}(**M-D**) (kJ/mol)
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15
-0.20
-0.25
50
100
150
200
250
300
Intermolecular distance, R_{AB} (Å)
ABA
GLY
LDOPA
PARA

(*b*)

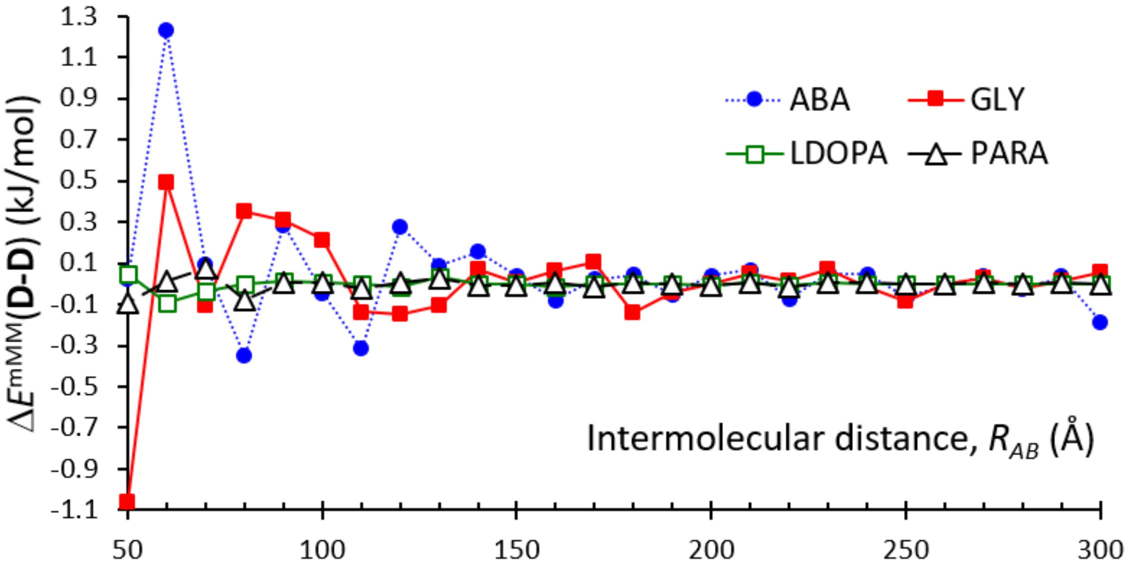


(*c*)

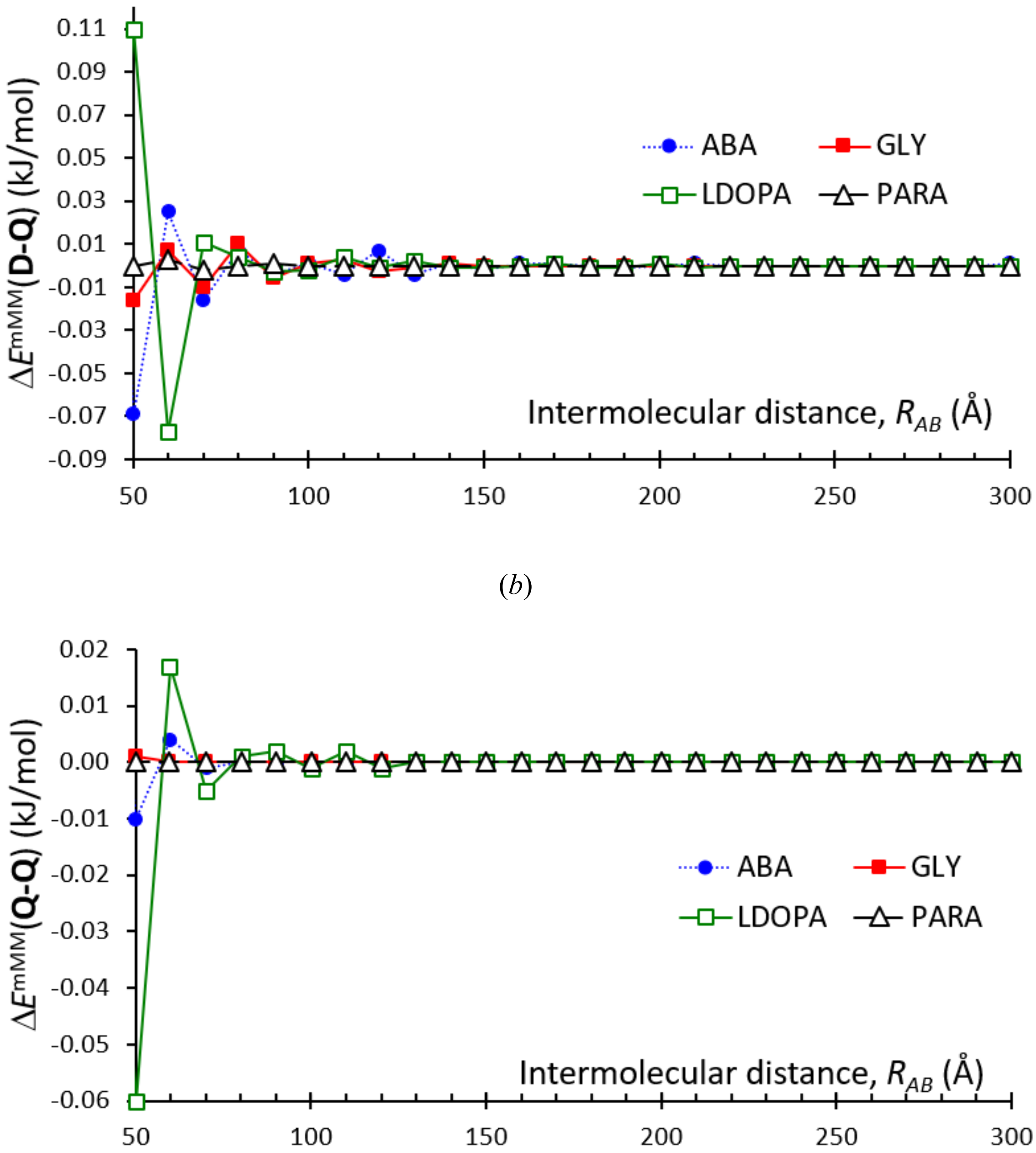
Figure 4.3 Convergence of the **(a)** M-M, **(b)** M-D and **(c)** D-D terms in the direct summation when the atomic multipole moment approximation (DS/aMM).

As mentioned in Section 4.2.2, for larger intermolecular separations it is possible to replace the atomic multipole moment approximation (E^{aMM}, equation 4.4) with the one that uses the total molecular multipole moments (E^{mMM}, equation 4.28). Since the molecules considered in this study are electroneutral, all interaction terms that include molecular monopoles should vanish, which makes the dipole-dipole (D-D) interaction (~R^{-3}) the slowest converging term. The convergence of the DS/mMM approximation in the range of 50 – 300 Å for the D-D, D-Q, and Q-Q terms is shown in Figure 4 for the same four compounds (ABA, GLY, LDOPA, and PARA). The D-Q, Q-Q and all higher-order terms for all compounds, as well as the D-D interactions in LDOPA and PARA, converge below 0.01 kJ/mol at about 100-120 Å. The D-D contributions in ABA and GLY show a very slow convergence with oscillations of 0.1-0.2 kJ/mol. In the case of ABA, both the DS/aMM and DS/mMM method show a significant energy change of 0.2 kJ/mol in the 290-300 Å range though. In the former, it is due to the M-M term while in the latter, it originates from the D-D term.

The DS/mMM calculations are several orders of magnitude faster than DS/aMM. For example, the DS/mMM calculation for ABA required only ~8.2×10^5 E^{mMM} evaluations which took about 9.3 seconds. The most time-consuming E^{mMM} calculation was for GLY: 1.5×10^6 evaluations were completed in 16.4 seconds.



(*a*)



(*b*)

(*c*)

Figure 4.4 Convergence of the **(a)** D-D, **(b)** D-Q and **(c)** Q-Q terms in the direct summation when using the molecular multipole moment approximation (DS/mMM).

The convergence of energy in the direct summation method is intimately related to the polarity of molecules. The fast energy convergence group includes species (BENZ, ACG, LAC, LDOPA, and PARA) with the pseudoatom databank-based molecular dipole moments below 2.5 atomic units (au, Table 4.3), while in the other group the molecular dipole moments are at or above 4.5 au. As expected, in more polar systems, such as ABA, and even more so in ionic crystals, the convergence of the energy is inherently much slower.

4.6.2 Multipole contributions in the Ewald summation (ES) method

While both the atomic (aMM) and molecular (mMM) multipole moments-based direct summation methods exhibit some (though for our systems, not too outrageous) convergence issues, the Ewald summation technique does not have this drawback. In Table 4.4, we show the difference between the direct (DS/aMM) and Ewald summation (ES) contributions for each multipolar term. The direct summation upper limit ($R_{\mathrm{DS\ cutoff}}$) was set at 300 Å.

Figure 4.3 Magnitudes of the molecular dipole moments (atomic units) in the benchmark compounds. The benchmark compounds are separated into two groups: those with a fast direct summation energy convergence (ACG, BENZ, LDOPA, PARA), and those with a slow energy convergence (ABA, ALA, GLY, SER, VAL)

	Gas phase			**Crystal**	
	UB Databank	**B3LYP 6-31G****	**MP2 6-31G****	**B3LYP 6-31G****	**X-ray experiment**
ACG	0.8	1.2	1.2	1.5	—
BENZ	0	0	0	0	0
LAC	1.6	1.2	1.2	1.8	—
LDOPA	2.4	4.1	4.2	4.6	4.7[a]
PARA	2.3	1.9	1.9	3.0	—
ABA	6.4	7.3	7.6	8.0	5.1[b]
ALA	4.5	4.4	4.5	5.3	5.0-5.2[c]
GLY	4.5	4.5	4.7	5.4	5.9[d]
SER	5.2	4.9	5.1	6.1	6.5[e]
VAL	4.6	4.2	4.3	5.3	—

[a] Howard et al., 1995
[b] Craven & Weber, 1983
[c] Destro, Bianchi & Morosi, 1989
[d] Destro et al., 2000
[e] in DL-serine (Coppens *et al.*, 1999)

Table 4.4 Absolute differences (kJ/mol) between electrostatic multipole interactions evaluated via the Ewald (ES) and direct (DS) summations, $E(\mathrm{ES}) - E(\mathrm{DS})$ using the atomic multipole moments (aMM). In the direct summation calculations, the parameter $R_{\mathrm{DS\ cutoff}}$ was set at 300 Å.

	~*R*$^{\mathrm{n}}$	**ACG**	**BENZ**	**LAC**	**LDOPA**	**PARA**	**ABA**	**ALA**	**GLY**	**SER**	**VAL**
M-M	$\sim R^{-1}$	-5×10^{-5}	5×10^{-8}	4×10^{-4}	2×10^{-3}	-7×10^{-4}	2×10^{-1}	-2×10^{-2}	-3×10^{-2}	1×10^{-2}	-2×10^{-3}
M-D	$\sim R^{-2}$	-7×10^{-5}	-4×10^{-7}	9×10^{-4}	5×10^{-5}	-8×10^{-4}	3×10^{-2}	-6×10^{-3}	-9×10^{-3}	8×10^{-3}	-1×10^{-4}
D-D	$\sim R^{-3}$	-5×10^{-6}	7×10^{-7}	3×10^{-4}	-4×10^{-5}	8×10^{-5}	7×10^{-4}	-6×10^{-4}	-9×10^{-4}	1×10^{-3}	1×10^{-4}
M-Q	$\sim R^{-3}$	1×10^{-6}	1×10^{-7}	-5×10^{-7}	9×10^{-6}	-3×10^{-5}	3×10^{-4}	3×10^{-6}	2×10^{-4}	9×10^{-6}	-3×10^{-5}
D-Q	$\sim R^{-4}$	2×10^{-7}	-5×10^{-7}	-2×10^{-6}	3×10^{-6}	-2×10^{-5}	3×10^{-5}	-6×10^{-7}	4×10^{-5}	4×10^{-6}	-8×10^{-6}
M-O	$\sim R^{-4}$	*	*	-3×10^{-8}	7×10^{-8}	-2×10^{-8}	1×10^{-6}	1×10^{-8}	-1×10^{-7}	3×10^{-8}	-1×10^{-7}
Q-Q	$\sim R^{-5}$	-1×10^{-8}	1×10^{-7}	-1×10^{-8}	-2×10^{-7}	-2×10^{-6}	1×10^{-7}	3×10^{-8}	9×10^{-8}	*	1×10^{-7}
D-O	$\sim R^{-5}$	*	*	-2×10^{-8}	*	-5×10^{-8}	8×10^{-8}	-1×10^{-8}	-5×10^{-8}	*	-4×10^{-8}
Q-O	$\sim R^{-6}$	*	*	*	*	1×10^{-8}	*	*	*	*	*
O-O	$\sim R^{-7}$	*	*	*	*	*	*	*	*	*	*
M-H	$\sim R^{-5}$	*	*	*	*	*	-1×10^{-8}	*	*	*	*
D-H	$\sim R^{-6}$	*	*	*	*	*	*	*	1×10^{-8}	*	*
Q-H	$\sim R^{-7}$	*	*	*	*	*	*	*	*	*	*
O-H	$\sim R^{-8}$	*	*	*	*	*	*	*	*	*	*
H-H	$\sim R^{-9}$	*	*	*	*	*	*	*	*	*	*
Total		-1×10^{-4}	1×10^{-7}	2×10^{-3}	2×10^{-3}	-1×10^{-3}	2×10^{-1}	-2×10^{-2}	-4×10^{-2}	2×10^{-2}	-2×10^{-3}

* the difference $E(\mathrm{ES}) - E(\mathrm{DS})$ is less than 1×10^{-8} kJ/mol

The results in Table 4.4 show an excellent agreement of less than 0.03 kJ/mol between the direct and Ewald sums in all compounds for all multipolar contributions above the monopole-monopole (M-M) level. The M-M term (~R^{-1}) shows a very large variation: between 5×10^{-5} kJ/mol for ACG (BENZ is discussed below) and 0.2 kJ/mol in ABA, though even in the latter case the relative error of the DS/aMM calculation is only about 0.1%. As expected, the average absolute deviations for the DS/aMM energies decrease significantly as the order of the interacting multipoles increases, which is especially evident for the molecules in the second ("slow" energy convergence) group. Note that above the Q-Q level (~R^{-5}), the DS and ES results are essentially indistinguishable when comparing the energy values printed out in kJ/mol to eight decimal digits.

While our results agree with the conventional wisdom that the Ewald summation is important for the low-order ~R^{-1} interactions, a very good agreement is observed for the M-D and D-D interactions (~R^{-2} and ~R^{-3}, respectively) with the differences between the ES and DS energies being consistently below 0.1 kJ/mol. For interactions involving the quadrupole moments, the agreement is even better, never exceeding 10^{-3} kJ/mol, which is not unexpected.

An excellent test for consistency between the Ewald and direct summations is provided by the benzene molecule (BENZ) that lacks not only the net charge but also the symmetry forbidden net dipole and octupole moments. In addition, the atoms in benzene are only slightly polarized which results in small net atomic charges. Thus, the direct summation method applied to BENZ is expected to recover all the individual multipolar energy contributions within the numerical precision used. The benzene entries in Table 4.4, indeed, show an essentially perfect agreement between the ES and DS methods for all terms with the differences being ~$10^{-7} – 10^{-8}$ kJ/mol. Such an agreement is also a strong indication that the Ewald summation implementation in our code is correct for all multipolar terms.

The results in Table 4.4 may suggest that the Ewald summation is likely overkill for the quadrupoles and higher-order terms, and may not even be necessary for the monopoles and dipoles. However, we note that

(i) the DS-based energies listed in Table 4.4 were obtained using a very large upper summation limit of 300 Å which resulted in most calculations running for 30-50 minutes (76 minutes for LDOPA),

(ii) in the absence of the target (benchmark) values from the Ewald summation, it is not easy to judge the quality of the resulting M-M and perhaps M-D and D-D energies, and

(iii) the Ewald summation is extremely fast: for the test systems, it never exceeded two seconds (see the discussion of timings below).

4.6.3 Exact potential (EP) correction to the Ewald sums

While the Ewald summation is extremely fast (see discussion below) and not hindered by convergence issues, its main drawback is the inability to correctly account for the penetration terms arising due to overlap of neighboring charge densities, which is always the case for molecules in crystals. Of course, neither does the DS/aMM method as it uses the same atomic multipole moments as the Ewald summation. As discussed in Section 4.2.1.5, we rectify this deficiency by applying the exact potential correction, $E_{\mathrm{EP\ corr}}$. The importance of the exact potential (EP) correction to the atomic multipole moments-based energies (E^{aMM}) emphasized in our original EP/MM study (Volkov, Koritsanszky, & Coppens, 2004) is also confirmed by the results presented in Table 4.5. Indeed, for the four molecules (ACG, BENZ, LAC, and PARA, all with a small dipole moment), the EP correction exceeds 50% of the EP-corrected Ewald sum. The smallest absolute value of the EP correction (-12.1 kJ/mol) is observed for benzene though on a relative scale, it accounts for 62% of total energy. The smallest percentage value (28%, -84.2 kJ/mol) is observed for SER in which the uncorrected ES energy is -220.8 kJ/mol. The largest absolute EP correction (-140.7 kJ/mol) is found in LDOPA (E_{ES} = -198.4 kJ/mol), while the most significant percent-wise EP contribution is observed for ACG (64%, -94.8 kJ/mol).

We also note that the exact potential correction to the Ewald sum in the benchmark systems is always negative, which means that the Ewald sum and direct summation methods without the EP correction grossly underestimate the magnitude of the electrostatic interactions in these crystal structures as both are based on the same atomic multipole moment (aMM) aproximation.

Table 4.5 The exact potential (EP) correction ($E_{\text{EP corr}}$) to the Ewald sum E_{ES} for all multipolar terms up to hexadecapole-hexadecapole (kJ/mol). For clarity, the energy values are rounded off to only one decimal digit.

	Ewald sum, E_{ES}	**EP correction,** $E_{\text{EP corr}}$	**EP-corrected Ewald sum,** $E(\text{ES}) = E_{\text{ES}} + E_{\text{EP corr}}$	**EP correction contribution** (percent of the total value)
ACG	-54.3	-94.8	-149.0	64
BENZ	-7.5	-12.1	-19.6	62
LAC	-68.5	-79.9	-148.3	54
LDOPA	-198.4	-140.7	-339.1	41
PARA	-64.2	-87.0	-151.2	58
ABA	-213.3	-114.4	-327.7	35
ALA	-180.0	-90.0	-270.0	33
GLY	-179.8	-88.4	-268.1	33
SER	-220.8	-84.2	-305.0	28
VAL	-173.1	-92.5	-265.6	35

4.6.4 Total electrostatic molecular interaction energies in the crystal structures and timing

Table 4.6 summarizes the total electrostatic interaction energies E (in kJ/mol) and timing (in seconds) for the benchmark structures calculated using the EP-corrected Ewald summation (ES) method and the EP/MM-based direct summation (DS) algorithm. In the DS method the $R_{\text{DS cutoff}}$ parameter was set at 300 Å, and the following unified EP/MM strategy was utilized:

- the R_{mMM} parameter which specifies the intermolecular distance R_{AB} (distance between the geometrical centers of the central molecule A and a symmetry-equivalent molecule B) at which the electrostatic energy evaluation switches to the molecular multipole moment (mMM) approximation (equation 4.9) was set to 20 Å;
- interactions between molecules separated at $R_{AB} < R_{\mathrm{mMM}} = 20\,\text{Å}$ were evaluated using the "standard" EP/aMM method in which the interatomic interactions below 5 Å were calculated via the 80-bit Löwdin α-function technique, otherwise the atomic multipole moments (aMM) approximation (equation 4.4) was used.

The agreement between the electrostatic energy values calculated via the ES and DS methods is very good (< 0.05 kJ/mol) for all compounds except ABA for which the difference is ~0.2 kJ/mol. The absolute deviation of the EP/MM-based DS energies averaged over all compounds except BENZ (which was excluded in order to prevent bias as it is expected to show essentially a perfect agreement) is just over 0.03 kJ/mol. Even for relatively polar molecules such as alanine, glycine, serine, and valine, the agreement is excellent. To some extent, this is to be expected as beyond the intermolecular separation of 20 Å, the EP/MM-based DS method switches to the molecular multipole moment (mMM) approximation in which the slowest converging term for our systems is dipole-dipole (recall, that all the benchmark molecules are electroneutral). As such, its convergence should follow the predictions shown in Figure 4.4 which identifies the D-D term as the only one with a challenging convergence.

.

Table 4.6 The total electrostatic interaction energies E (kJ/mol) in the benchmark crystal structures, and the computation times (seconds) for the Ewald (ES) and direct (DS) summation methods. The DS R_{mMM} and $R_{\mathrm{DS\ cutoff}}$ parameters were set at 20 Å and 300 Å, respectively. Δ is the difference for a given value (either in kJ/mol or seconds) between the two methods (ES-DS). The numerical values have been rounded off in such a way as to clearly show the differences. $\overline{|\Delta|}$ is the average absolute difference calculated over all compounds except BENZ

	Method	***E* (kJ/mol)**	**Time used by a given component of the method (seconds)**				
			Total	**EP**	**ES**	**aMM**	**mMM**
ACG	ES	-149.0309	7	5.9	1.2	0.009	
	DS	-149.0287	17	5.9	—	0.6	9
	Δ	-0.0022	-9	0.0	—	—	—
BENZ	ES	-19.6186220	4	2.8	0.8	0.005	
	DS	-19.6186234	15	2.9	—	0.4	10
	Δ	0.0000014	-11	0.0	—	—	—
LAC	ES	-148.3401	5	4.0	0.8	0.007	
	DS	-148.3412	17	4.0	—	0.5	11
	Δ	0.0011	-12	-0.1	—	—	—
LDOPA	ES	-339.0551	10	9.0	0.9	0.013	
	DS	-339.0585	17	9.2	—	1.0	6
	Δ	0.0034	-7	-0.1	—	—	—
PARA	ES	-151.2099	9	6.8	1.8	0.010	
	DS	-151.2061	15	6.9	—	0.8	7
	Δ	-0.0037	-7	-0.1	—	—	—
ABA	ES	-327.71	6	4.5	1.3	0.009	
	DS	-327.89	15	4.4	—	0.6	9
	Δ	0.18	-9	0.1	—	—	—
ALA	ES	-270.035	5	4.3	1.0	0.008	
	DS	-270.011	18	4.3	—	0.6	12
	Δ	-0.024	-13	0.0	—	—	—
GLY	ES	-268.133	5	4.4	0.7	0.007	
	DS	-268.103	23	4.4	—	0.5	16
	Δ	-0.030	-18	-0.1	—	—	—
SER	ES	-305.003	7	5.6	1.1	0.010	
	DS	-305.028	18	5.4	—	0.6	11
	Δ	0.026	-12	0.2	—	—	—
VAL	ES	-265.6221	6	5.6	0.7	0.014	
	DS	-265.6191	17	5.3	—	1.0	9
	Δ	-0.0030	-10	0.3	—	—	—
	$\overline{|\Delta|}$	**0.034**	**11**				

While the DS-based calculations employed a fairly large upper summation limit of 300 Å, its performance is on the same time scale (seconds) as that of the EP-corrected Ewald summation method, not exceeding 25 seconds for any of the benchmark compounds. That said, for these systems, the Ewald summation is about 2-3 times faster than the DS procedure. While the *fastest* DS calculations (BENZ, PARA, and ABA) took 15 seconds, it is still 5 seconds above the *slowest* ES calculation (LDOPA, 10 seconds). The ES performance gain over the DS method varies from about 7 seconds for LDOPA and PARA (a factor of 1.7-1.8 speedup) up to 18 seconds in GLY (a factor of 4.6 speedups) all the while providing more significant digits in the determined energies. In fact, without the target ES values, it would have been difficult to estimate the precision of the DS-based energies.

In addition to listing the total execution times, Table 4.6 also shows timings for the most important calculation steps in each of the ES and DS runs.

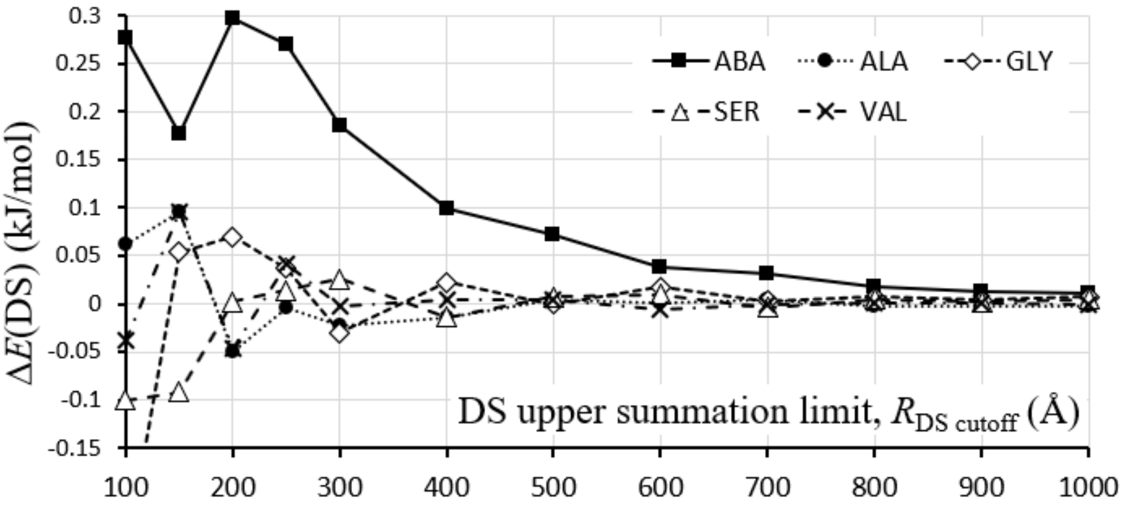
There are essentially two steps in the ES method: Ewald summation itself and the EP correction which includes evaluation of the short-range interatomic interaction using both the EP and aMM methods. Despite including terms up to the hexadecapolar level, the Ewald summation is fast. In the benchmark structures, it took on average 1.0±0.3 seconds with the longest calculation of 1.8 seconds in LDOPA. Evaluation of the EP correction is about five times slower than the Ewald summation with the integration of the exact potential significantly dominating over the aMM evaluations. Since the calculation of the EP correction requires an equal number of the aMM and EP evaluations, and because each aMM calculation is several orders of magnitude faster than the EP one, the contribution of the aMM evaluations to the EP correction calculation time is negligible. For the benchmark structures, integration of the exact potential needed for calculation of the EP correction to the Ewald sum accounts for 80-90% of the total computation time, with evaluation of the Ewald sums using the remaining 10-20%.

The direct summation (DS) approach includes three distinct types of calculations: (i) integration of the exact potential for the short-range inter*atomic* interactions, (ii) evaluation of E^{aMM} (equation 4.4) for the intermediate-range inter*atomic* interactions, and (iii) computation of E^{mMM} (equation 4.28) for large interm*olecular* separations. As expected, the calculation time for the EP component of the DS method is essentially identical to that in the ES approach. However, its relative contribution to the total computation time is much smaller in the DS calculations: the largest percent value is 55% in LDOPA while the smallest is 19-20% in BENZ, GLY, and SER. The aMM component of the DS method does not use much time (at most 1 second in LDOPA) as it complements EP when dealing with inter*atomic* interactions. Its contribution to the total DS computation time is 2-6%. The remaining time is used by the E^{mMM} evaluations that account for anywhere between 35% (LDOPA) and 60-70% (BENZ, LAC, ABA, ALA, GLY and SER). This is expected as mMM is the only method that evaluates energies for intermolecular separations between 20 Å and 300 Å. As such, the computational weight of the mMM component should increase when dealing with more polar molecules which will likely require a larger spherical summation range.

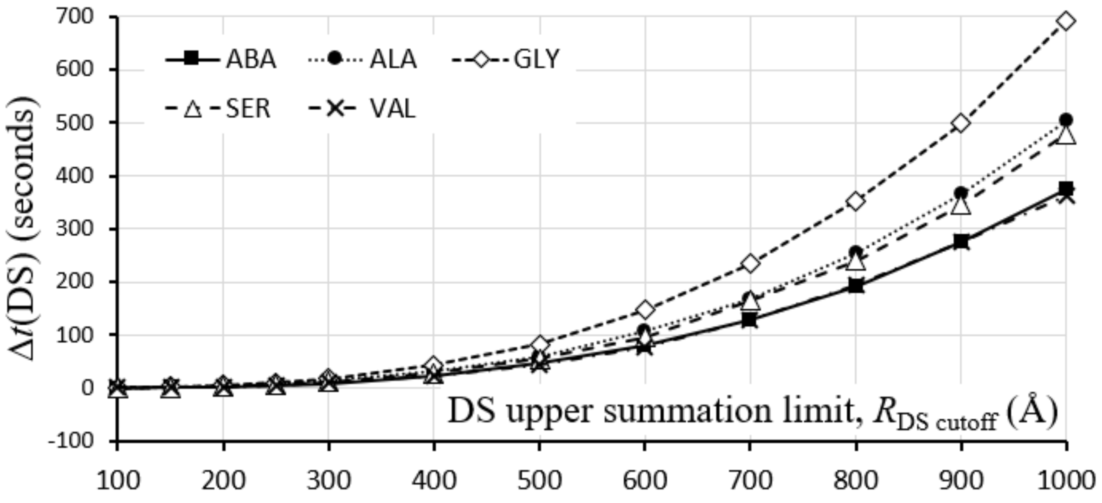
4.6.5 Precision and speed of the DS method as a function of the summation limit

Since the small amino acids and, especially, ABA can be considered a prototype for difficult convergence cases for the direct summation (DS) method, we have performed a series of additional DS calculations for these systems with the upper summation limit values of 400, 500, …, 1000 Å, while also monitoring the computation time.

The results are summarized in Figure 4.5. For the four amino acid structures, an acceptable convergence of 0.05 kJ/mol is achieved at about 250 Å which makes the DS method slower than the Ewald summation calculations by at most 10 seconds. To improve the energy agreement to 0.02 kJ/mol the summation range must be increased to 500 Å which still maintains a reasonable computational overhead of under 90 seconds relative to ES. Finally, at 700 Å the DS energies get within 0.01 kJ/mol from the ES values but at the computational overhead of 2-4 minutes relative to the ES calculation times.



(*a*)



(*b*)

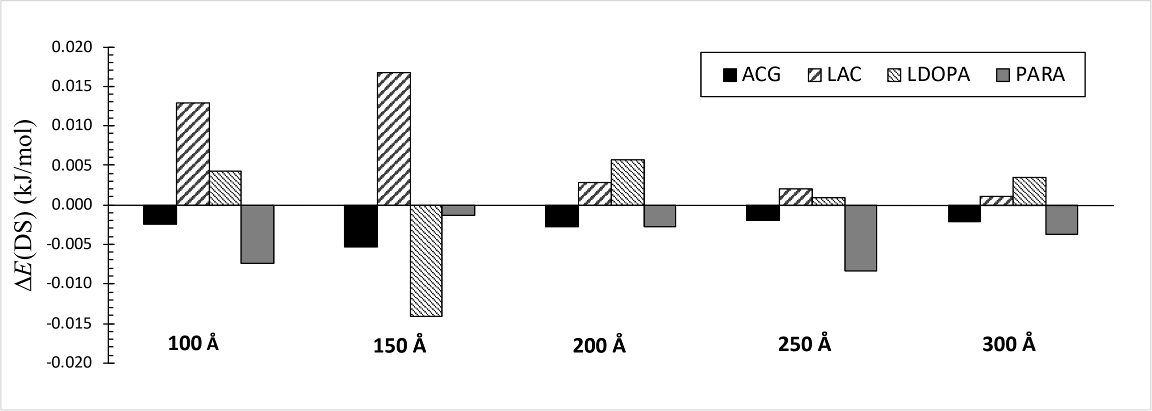
Figure 4.5 (*a*) The electrostatic interaction energy difference, ΔE (kJ/mol) and (*b*) the computational time overhead, Δt (seconds) for the direct summation (DS) calculation for the four amino acids and ABA structures relative to the ES calculation, as a function of the upper summation limit, $R_{\mathrm{DS\ cutoff}}$. The code used the 80-bit Löwdin α-function and was compiled with the GFortran $-\mathrm{O}2$ flag.

The energy convergence in ABA is much slower than in any other tested system. In order to reach the 0.05 kJ/mol energy convergence, the upper summation limit in ABA must be extended to 600 Å which makes it about 90 seconds slower than the Ewald summation. However, in order to get to within 0.02 kJ/mol from the target ES energy, the $R_{\mathrm{DS\ cutoff}}$ parameter must be set to 800 Å, which extends the computation time by over 3 minutes (!).

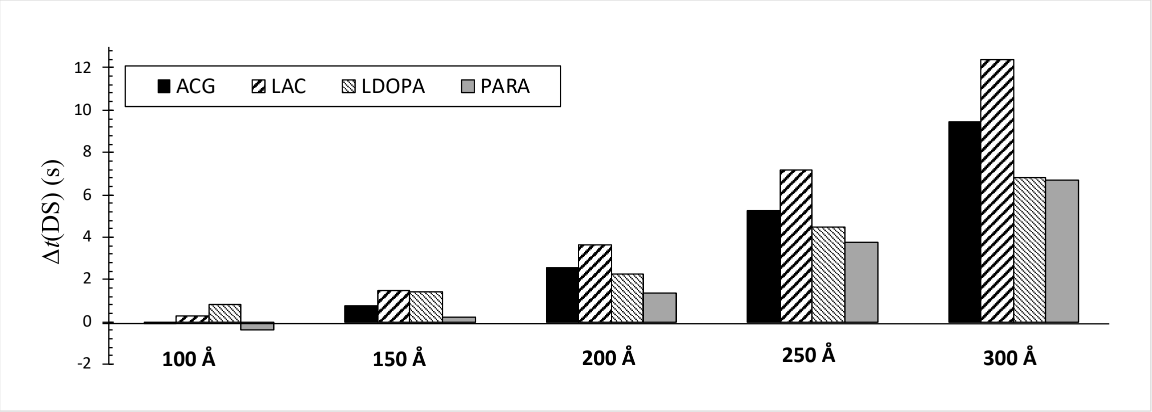
In general, the results confirm the difficulties (albeit not unsurmountable) when using the standalone EP/MM-based direct summation (DS) method for highly polar molecules as it requires a thorough convergence testing which unfortunately is time-consuming. The implemented Ewald summation (ES) technique does not have this drawback.

Figure 4.6a shows the DS energy convergence as a function of the upper summation limit $R_{\mathrm{DS\ cutoff}}$ between 100 Å and 300 Å in the other benchmark systems. BENZ, in which the energy converges within the 10^{-3} kJ/mol at $R_{\mathrm{DS\ cutoff}} = 30$ Å is not shown.

Unlike the amino acids and ABA, the energy convergence of 0.02 kJ/mol in ACG, LAC, LDOPA, and PARA is achieved at about 100-150 Å. Because of the relatively small upper summation limit, these DS calculations are essentially as fast as the Ewald summation. This is illustrated in Figure 4.6b which shows the difference in the computation time between the two methods. A positive value of Δt indicates that the DS is slower than the Ewald summation, and *vice versa*. For most of the cases, the Ewald summation is a clear winner in terms of speed, except for ACG and PARA at $R_{\mathrm{DS\ cutoff}} = 100$ Å when it trails the DS calculation by less than half a second. In order to achieve the desired energy convergence of 0.01 kJ/mol, the direct summation limit must be extended to at least 200 Å, which introduces a computational overhead of up to 4 seconds relative to the Ewald summation. However, for these small systems in which the calculations are very fast, such a difference is not significant. Finally, at 300 Å, the convergence reaches 0.005 kJ/mol or better but at a cost of a higher computational overhead relative to ES: 7 seconds for LDOPA and PARA, 9.5 seconds for ACG, and 12 seconds for LAC.



(*a*)



(*b*)

Figure 4.6 (*a*) The electrostatic interaction energy difference, ΔE (kJ/mol) and (*b*) the computational time overhead, Δt (seconds) for the direct summation (DS) calculation for the ACG, LAC, LDOPA, and PARA structures relative to the ES calculation as a function of the upper summation limit, $R_{\mathrm{DS\ cutoff}}$. The code used the 80-bit Löwdin α-function and was compiled with the GFortran $-\mathrm{O2}$ flag.

In general, the EP-corrected Ewald summation method is clearly superior in terms of both precision and speed to the EP/MM DS approach despite the fact that the latter uses the molecular multipole moment approximation to evaluate interaction energies at larger intermolecular separations. Just like the DS approach, the ES method uses analytical integration of the exact potential for short-range interactions to account for interpenetration of the neighbouring charge density distributions, but unlike DS, it does not suffer from the convergence issues regardless how polar the molecules are, all the while being several times faster.

The only two questions regarding the implemented EP-corrected Ewald summation method that remain to be answered are related to (*i*) its numerical stability, and (*ii*) its performance when applied to larger systems, both of which are discussed below.

4.6.6 Numerical precision and computational performance considerations

The discussion of the precision and computational performance of both the ES and DS methods presented above was limited to a single implementation with both methods using the 80-bit Löwdin α-function (Lα) for evaluation of the electron repulsion integrals (ERI) in the exact potential (EP) (equation 4.3), and the code being compiled with a very conservative −O2 optimization option in the GNU Fortran (GFortran) compiler. Therefore, it was necessary to test the numerical precision of the two methods when using the 64-bit implementation of the Löwdin α-function (Lα), as well as the two higher code optimization levels (−O3 and −Ofast) available in GFortran, with the ultimate goal to determine the fastest yet precise enough configuration.

The results of a thorough analysis of the precision and speed of both the ES and DS methods are summarized in Table 4.7. The −O3 compiler option generated executables that produced energies that were identical within the printed 10^{-8} kJ precision to those from the −O2 executable within a given Löwdin α-function implementations. However, the timings for the −O3 executable were inconsistent: in most of the cases, it was somewhat slower than the $-\mathrm{O}2$ executable. It was especially evident in the DS runs for several of which the $-\mathrm{O}3$-based calculation was 1-2 seconds slower than $-\mathrm{O}2$. For the ES runs, the timings were within 0.5 seconds from each other with the $-\mathrm{O}3$ executable being faster for some cases, while trailing the $-\mathrm{O}2$ runs in others. As such, the results from the $-\mathrm{O}3$ executables are not included in Table 4.7.

Table 4.7 Precision and speed of the Ewald (ES) and direct (DS) summation methods as a function of i) precision of the Löwdin α-function (Lα) when evaluating the exact potential (EP), and ii) GFortran optimization level when compiling the code. In the energy values, identical digits (after rounding off) within a given implementation (ES or DS) are greyed out, differences are highlighted in **bold**. The underlined time values illustrate improvement when switching from the highest-precision (80-bit Lα / −O2 compiler flag) to the lowest-precision (64-bit Lα / −Ofast compiler flag) version of the code.

EP Lα precision	**GFortran compiler option**		**ES**		**DS**			**ES**		**DS**	
			E_{es} (kJ/mol)	**Time (s)**	**E_{es} (kJ/mol)**	**Time (s)**		**E_{es} (kJ/mol)**	**Time (s)**	**E_{es} (kJ/mol)**	**Time (s)**
80-bit	−O2	**ACG**	-149.03087**92**	7.2	-149.02867**85**	16.6	**ABA**	-327.7074**01**	5.9	-327.8920**28**	15.1
	−Ofast		-149.03088**50**	5.5	-149.02868**43**	14.9		-327.7074**17**	4.6	-327.8920**45**	13.8
64-bit	−O2		-149.03087**92**	4.5	-149.02867**85**	14.5		-327.7074**01**	3.8	-327.8920**28**	13.5
	−Ofast		-149.03088**51**	4.2	-149.02868**44**	14.0		-327.7074**18**	3.6	-327.8920**45**	12.1
80-bit	−O2	**BENZ**	-19.61862**20**	3.8	-19.61862**34**	14.5	**ALA**	-270.0345**03**	5.3	-270.0109**62**	18.1
	−Ofast		-19.61862**41**	2.9	-19.61862**55**	14.4		-270.0345**14**	4.1	-270.0109**73**	16.6
64-bit	−O2		-19.61862**21**	2.5	-19.61862**34**	13.5		-270.0345**03**	3.4	-270.0109**62**	16.4
	−Ofast		-19.61862**41**	2.4	-19.61862**55**	12.4		-270.0345**15**	3.3	-270.0109**73**	15.8
80-bit	−O2	**LAC**	-148.340081**1**	4.9	-148.34123**01**	17.2	**GLY**	-268.1326**58**	5.1	-268.1031**11**	23.1
	−Ofast		-148.340080**5**	3.8	-148.34122**95**	16.7		-268.1326**68**	4.0	-268.1031**21**	22.1
64-bit	−O2		-148.340081**3**	3.2	-148.34123**04**	16.9		-268.1326**58**	3.2	-268.1031**11**	22.4
	−Ofast		-148.340080**7**	3.2	-148.34122**98**	15.3		-268.1326**68**	3.2	-268.1031**21**	20.6
80-bit	−O2	**LDOPA**	-339.055143**5**	10.0	-339.05854**52**	16.8	**SER**	-305.00264**9**	6.7	-305.02824**5**	18.4
	−Ofast		-339.055153**7**	7.8	-339.05855**54**	15.4		-305.00265**2**	5.0	-305.02824**9**	17.0
64-bit	−O2		-339.055143**5**	6.1	-339.05854**52**	13.9		-305.00264**9**	4.0	-305.02824**5**	16.6
	−Ofast		-339.055153**9**	5.9	-339.05855**55**	12.1		-305.00265**2**	3.8	-305.02824**9**	15.0
80-bit	−O2	**PARA**	-151.20986**25**	8.7	-151.20614**42**	15.3	**VAL**	-265.6221**00**	6.3	-265.6191**49**	16.5
	−Ofast		-151.20986**51**	6.8	-151.20614**68**	13.5		-265.6221**12**	4.5	-265.6191**60**	15.2
64-bit	−O2		-151.20986**24**	5.8	-151.20614**41**	12.6		-265.6221**00**	3.5	-265.6191**49**	13.7
	−Ofast		-151.20986**52**	5.3	-151.20614**70**	12.1		-265.6221**12**	3.5	-265.6191**60**	12.8

4.6.6.1 Precision of the Löwdin α-function (Lα)

The total energies for a given method (ES or DS) and the same compiler optimization option obtained when using the 64- and 80-precision Löwdin α-function (Lα) implementations, agree with each other well within 10^{-6} kJ/mol, with the 64-bit implementation being faster. This is in agreement with our previous study (Nguyen & Volkov, 2019). On average we see a speed improvement of 1-4 seconds when using the 64-bit Löwdin α-function evaluations, which has a more pronounced overall effect on the ES calculations because they are faster. For example, the ES calculation time for LDOPA dropped from 10 seconds to 6.1 seconds (40% speedup) while in BENZ a decrease in computation time from 3.8 seconds down to 2.5 seconds (1.3 seconds speedup) yielded an improvement of 65%. In general, the code compiled with the −O2 option benefits more from switching the Lα evaluation from 80-bit to 64-bit, than the −Ofast executable. It is understandable as in order to provide a faster executable the −Ofast option enables several additional code optimizations including `-ffast-math`, `-funsafe-math-optimizations`, and `-fexcess-precision=fast` that disregard the IEEE, ANSI and/or ISO compliance (Free Software Foundation, 2019). Average speed improvement when lowering the precision of the Löwdin α-function (Lα) determination from 80-bit to 64-bit, is 2.1 and 1.4 seconds for the −O2 and −Ofast executables, respectively.

4.6.6.2 Compiler optimization options

The analysis of results in Table 4.7 also shows that energies obtained with the same precision of the Löwdin α-function (Lα) but using the two different GFortran compiler optimization levels ($-\mathrm{O2}$ and $-\mathrm{Ofast}$) agree within 10^{-5} kJ/mol, with the $-\mathrm{Ofast}$ executables being faster. It agrees with our expectations as the $-\mathrm{Ofast}$ compiler option is designed to improve the speed at a cost of reducing the numerical precision (Free Software Foundation, 2019). However, the loss of precision is very small: the energies obtained with the $-\mathrm{Ofast}$ executable lose approximately one extra digit when compared to the effect of lowering the precision of the Löwdin α-function determination from 80-bit to 64-bit. The performance improvement of the $-\mathrm{Ofast}$ executable, however, is not as significant as that when lowering the precision of the Löwdin α-function. On average, the $-\mathrm{Ofast}$ executable gains about 1 second relative to the $-\mathrm{O2}$-based calculations. It is interesting to note that the 80-bit implementation of the Löwdin α-function benefits more in terms of speed from using the $-\mathrm{Ofast}$ compiler option than the 64-bit implementation.

4.6.6.3 Compiler optimization options

As expected, the most noticeable performance improvement is found when we simultaneously (*i*) reduced precision of the Löwdin α-function (Lα) determination from 80-bit to 64-bit, and (*ii*) increased the GFortran compiler optimization level from $-\mathrm{O2}$ to $-\mathrm{Ofast}$.

This combination of options produced the fastest executable, which improved the execution time for the benchmark runs on average by 2.7 seconds. That said, the speedup varies among the compounds between 1.6 seconds for the LAC ES run, and 4.7 seconds in the DS run for LDOPA. Note that because the DS calculation for LDOPA is much longer than the ES calculation for LAC, the *relative* speedup for the former is not as pronounced: 28% for the LDOPA DS calculation, and 33% for the LAC ES run.

The loss of precision in the determined energies is however almost the same as when increasing the compiler optimization level alone because, as discussed above, the effect of the decrease in precision of the Löwdin α-function ($\mathrm{L}\alpha$) determination in most cases is much smaller than that due to the increase in the compiler optimization level. As such, the energies produced by the fastest executable (64-bit $\mathrm{L}\alpha$ / $-\mathrm{Ofast}$ GFortran) are well within 10^{-5} kJ/mol from the target values obtained with the 80-bit $\mathrm{L}\alpha$ / $-\mathrm{O2}$ GFortran code.

These results are somewhat surprising as in our previous study (Nguyen & Volkov, 2019) we determined that the "*higher optimization options* [in GFortran] *(such as* $-\mathrm{Ofast}$*) produced inadequate numerical results*." At this time, we are not sure about the origin of this discrepancy as too many computational variables have changed. For example, in this study, we are using a different processor (Intel Xeon E3-1505M v5 vs. AMD FX-8350 used previously), and a newer version of the GFortran compiler (9.2.1 vs. 8.2.1). The compiler version effect is quite possible because while we had some numerical issues with the $-\mathrm{Ofast}$ option in GFortran 8.2.l, the $-\mathrm{Ofast}$ option worked just fine in the AMD version (Advanced Micro Devices, 2018) of the DragonEgg Fortran compiler (DragonEgg, 2018) in which the code optimizers were taken from the LLVM project (Lattner & Adve, 2004). While we do not have access to the GFortran 8.2.1 compiler anymore, we have performed several tests with GFortran 8.3.1 using the Intel Xeon E3-1505M v5 processor, and did not detect any additional precision loss beyond that discussed above. It is likely that in the previous study (Nguyen & Volkov, 2019) we were too cautious when checking the number of recovered digits in the *individual* electron-repulsion integrals, and might have too harshly discarded results from the −Ofast option in GFortran 8.2.l because of low precision of a number of integrals with very small magnitudes.

Whatever the case, a detailed and thorough analysis of numerous calculations performed in this study shows that the −Ofast GFortran compiler optimization option combined with the 64-bit precision evaluation of the Löwdin α-function (Lα) provides the fastest implementation for both the Ewald and direct summation methods with a negligible (< 10^{-5} kJ/mol) loss of precision in the resulting electrostatic intermolecular energies. Finally, we note that the wide availability of the Windows Subsystem for Linux (WSL) (Harsh, 2019) brings a fully featured native GFortran environment to Windows 10 users without using a virtual machine. We have tested the new XDPROP code in the OpenSUSE 15.1 WSL environment under Windows 10 running on the same Intel Xeon E3-1505M v5 computer as used for all calculations performed in this study, and, aside from a slight performance deterioration, did not encounter any numerical issues with either the Ewald or direct summation methods.

4.6.7 Application to large systems

While both the Ewald and direct summation methods perform very well for the selected small benchmark systems, it was imperative to assess their performance for a larger system as well. To this purpose we have computed the electrostatic interaction energy in a crystal structure of a previously used (Nguyen, Kisiel & Volkov, 2018; Nguyen

& Volkov, 2019) decapeptide Boc-(L-Leu-Aib)$_5$-OMe (Demizu et al., 2016; CSD code IPUNAK), where Boc is tert-butyloxycarbonyl, Leu is leucine, and Aib is the α-aminoisobutyric acid. For simplicity, the solvent molecules (methanol and water) were not included in the calculations. While this molecule is smaller (181 atoms) than the dodecapeptide Boc-L-Leu-L-Leu-Aib-(D-Leu-D-Leu-Aib)$_2$-L-Leu-L-Leu-Aib-OMe (Demizu, Yamashita, Doi et al., 2015; CSD code VUQZUE) (225 atoms after removal of the solvent molecules) also used in our previous studies (Nguyen, Kisiel & Volkov, 2018; Nguyen & Volkov, 2019), the decapeptide crystallizes in the $P2_12_12_1$ (Z=4) space group while that of the dodecapeptide is P_1. Thus, there are a total of 724 decapeptide atoms in the unit cell which should present a much more significant challenge to Ewald summation implementation as compared to the dodecapeptide unit cell. Also, since for the decapeptide molecule, the electron density generated by the University at Buffalo pseudoatom databank (Volkov, Koritsanszky & Coppens, 2004; Dominiak *et al.*, 2007) predicts a dipole moment of 27.7 au (67.9 debye), which is about four times greater than the dipole moment in ABA (6.4 au), it is expected to be challenging for the direct summation approach as well.

A preliminary set of the direct space calculations suggested 80 Å as the optimal value for the R_{mMM} parameter as it produced energies within 0.0002 kJ/mol from those when using the R_{mMM} values of 90 Å, 100 Å, and 120 Å, but is less computationally demanding. The upper direct summation limit was set to 1500 Å which, since the molecule itself is large, carried an overhead of only about 80 and 60 seconds as compared to the 500 Å and 1000 Å upper summation limits, respectively, but produced more precise energies. Based on the discussion of the precision and performance (section 4.6.6) we examined (*i*) the 80-bit Löwdin α-function implementation combined with the $-\mathrm{O}2$ GFortran compiler optimization option, and (*ii*) the 64-bit Löwdin α-function code generated with the −Ofast compiler flag. The former calculation provided base values for energy and speed. In addition to the Xeon E3-1505M v5 calculations, the executables were run on the AMD FX-8350 CPU used in the previous study (Nguyen & Volkov, 2019), as well as recently released AMD Ryzen 7 3700X microprocessor (courtesy of Mr. Justin Marsee, Molecular Biosciences Ph.D. Program, Middle Tennessee State University).

	80-bit Löwdin α-function and −O2 GFortran compiler flag		64-bit Löwdin α-function and −Ofast GFortran compiler flag	
	E (kJ/mol)	Time (s)	E (kJ/mol)	Time (s)
ES	-306.135**79**	117.3[†] (+94%) 60.5[*] 45.9[‡] (–24%)	-306.135**80**	77.3[†] (+59%) 48.5[*] 37.1[‡] (–24%)
DS	-306.136**55**	1047.2[†] (+76%) 596.4[*] 486.1[‡] (–18%)	-306.136**56**	744.1[†] (+38%) 538.0[*] 365.7[‡] (–32%)

[†] 2012 4.0 GHz AMD FX-8350 computer processor, PassMark (2020) score = 1510
[*] 2015 2.8 GHz Intel Xeon E3-1505M v5 computer processor, PassMark (2020) score = 1915
[‡] 2019 3.6 GHz AMD AMD Ryzen 7 3700X computer processor, PassMark (2020) score = 2906

The results yet again demonstrate a significant advantage of the Ewald summation (ES) method over the direct summation (DS) approach in terms of both precision and speed, as well as the benefit of using the 64-bit implementation of the Löwdin α-function combined with the −Ofast GFortran compiler optimization option. The energies from all calculations agree to within 0.001 kJ/mol with the Ewald summation soundly dominating the DS method in terms of both precision and especially, speed. Originally, we were not sure about the performance of the Ewald summation when calculating all interactions up to the hexadecapolar level, but the results exceeded our expectations. When using the maximum level of optimization (64-bit Lα method and the $-\mathtt{Ofast}$ compiler optimization level) the Ewald summation was able to complete the calculation well under 1 minute which brings it closer in terms of computation time to precise DS-based calculations for much smaller benchmark systems. Recall that it took the DS method about 20 seconds in the calculation for GLY to get within 0.03 kJ/mol from the ES energy.

The results also show a significant improvement in the floating-point performance of the recently released AMD Ryzen 7 3700X CPU which is based on the Zen2 microarchitecture relative to AMD's old Piledriver core found in the FX-8350 CPU. Despite a 10% lower base clock speed, the AMD Ryzen 7 3700X processor is more than two times faster in both the ES and DS calculations than FX-8350. The four-year-old Intel Xeon E3-1505M v5 CPU holds ground when compared to the AMD Ryzen 7 3700X, but is still 20-30% slower. The upcoming AMD Zen3 processors are rumoured to have an even better (at least 8-10%) floating-point performance than the Zen2 CPUs. As such, we do not currently plan to further optimize the described Ewald summation implementation in XDPROP unless there is a need to study much larger molecular structures.

Thus, it is without reservations that we recommend the Ewald summation (ES) method with the exact potential (EP) correction implemented in the new version of XDPROP for calculation of the electrostatic interaction energies in molecular crystals: it is numerically stable and fast, even for larger systems such as the tested decapeptide structure. That said, for smaller systems, the EP/MM-based direct summation (DS) method is very much competitive in terms of both speed and precision though it requires a careful selection of (a) the R_{mMM} parameter which determines the minimum intermolecular distance at which the method switches to using the molecular multipole moment (mmM) approximation, and (*b*) the upper spherical summation cutoff limit, $R_{\mathrm{DS\ cutoff}}$. A higher value for the former parameter must be chosen for larger systems, while the latter must be increased for systems that carry a large dipole moment.

4.7 Summary and Concluding remarks

We have presented and discussed the details of the implementation of the two methods for evaluation of the electrostatic interaction energies in molecular crystals within the pseudoatom electron density formalism:

1) The Ewald summation technique (ES) extended up to the hexadecapolar interactions, and combined with the analytical exact potential integration correcting for the short-range electron density penetration effects, and
2) The enhanced EP/MM-based direct summation approach (DS) which uses the analytical exact potential (EP) integration for short-range inter*atomic* interactions, the atomic multipole moment (aMM) approximation for the intermediate-range inter*atomic* interactions, and the molecular multipole moment (mMM) approximation for the long-range inter*molecular* interactions.

For the Ewald summation method, we followed the work by Aguado & Madden (2003) to derive the reciprocal space energy and self-energy correction terms up to the hexadecapolar method, and used the Nymand & Linse (2000) recipe to evaluate the direct space energy terms.

The electron-repulsion integrals (ERI) in the exact potential (EP) component of the ES and DS summation methods are evaluated in this study using the Löwdin α-function

(Lα) method (Nguyen, Kiesel & Volkov, 2018) which has been tested using both the 64-bit and 80-bit precision.

The Ewald (ES) and direct (DS) summation methods have been implemented using Fortran in the in-house version of XDPROP, part of the XD2016 software suite (Volkov, Macchi *et al*., 2016), and have been thoroughly examined in terms of speed and precision using a number of small molecular systems (serine, glycine, valine, alanine, aminobutyric acid, N-acetylglycine, L-dopa, lactic acid, paracetamol, and benzene), and a decapeptide molecular crystal structure with 181 atoms in the molecule and a total of four molecules (Z=4) in the unit cell.

All calculations except for a couple of additional decapeptide runs, were performed using a 64-bit 2015 2.8 GHz Intel Xeon E3-1505M v5 processor (CPU) and a 64-bit Linux operating system. The XDPROP executables were created using the 64-bit version of the GNU Fortran (GFortran) 9.2.1 compiler for which we investigated the effect of the $-\mathrm{O2}$ and $-\mathrm{Ofast}$ compiler flags that enable different levels of the code optimization. While the $-\mathrm{O2}$ flag is usually the safest level of optimization that improves the speed of the generated code while maintaining the desired precision (Free Software Foundation, 2019), the $-\mathrm{Ofast}$ compilation flag turns off strict IEEE/ISO standard compliance in order to produce faster executables at the cost of lowering the numerical precision.

The results of the exhaustive series of calculations followed by a detailed analysis of the precision and computational performance of the methods can be summarized as follows:

1) Both the Ewald (ES) and direct (DS) summation methods provide reliable electrostatic interaction energies in molecular crystals, though the DS method

requires extra care when selecting the optimal numerical values for parameters such as (*a*) the smallest intermolecular distance (R_{mMM}) at which the method switches to the molecular multipole moment (mMM) approximation, and (*b*) the upper spherical summation limit ($R_{\mathrm{DS\ cutoff}}$).

2) After the Ewald summation (ES) method has been thoroughly tested, the ES-based energy values corrected for the electron density penetration effects determine using the 80-bit Löwdin α-function (Lα) method, and obtained with the GFortran executable optimized at the $-\mathrm{O}2$ level, have been used to determine the benchmark (target) values for all systems.

3) The penetration effects which arise from the overlap of the neighbouring charge density distributions have been found to be extremely important in all tested systems. Their numerical values range in the studied structures from 12 kJ/mol to 140 kJ/mol, and constitute anywhere between 28 and 64% of the total electrostatic interaction energy. The exact potential correction which accounts for these effects is implicitly included in the EP/MM-based direct summation (DS) approach but had to be added separately to the aMM-based Ewald summation.

4) The convergence of the DS energy as a function of the upper spherical summation limit ($R_{\mathrm{DS\ cutoff}}$) in the studied molecular systems is very much dependent on the dipolar charge separation: calculations for molecules with a relatively small dipole moment converge fairly quickly, while molecules with a large dipole moment require the direct summation to include interactions at very large intermolecular separations. For example, in the γ-aminobutyric acid

structure with a molecular dipole moment of over six atomic units it was necessary to extend the upper summation limit ($R_{\mathrm{DS\ cutoff}}$) to 600 Å in order to bring the electrostatic interaction energy within 0.05 kJ/mol from the target value. By the same token, for the decapeptide structure the $R_{\mathrm{DS\ cutoff}}$ parameter had to be set to 1500 Å.

5) In agreement with the literature (see, for example, Frenkel & Smit, 2002), the convergence of the DS energy is ultimately related to the monopole-monopole interactions (~R^{-1}) when using the atomic multipole moment approximation (aMM), and the dipole-dipole interactions (~R^{-3}) when the molecular multipole moment approximation (mmM) is applied to electroneutral systems.

6) The need to operate with sufficiently large R_{mMM} and $R_{\mathrm{DS\ cutoff}}$ parameters hinders the computational performance of the direct summation (DS) method, especially when dealing with larger molecules carrying a significant dipole moment. For example, while for most of the smaller molecular benchmark systems (glycine, alanine, serine, valine, L-dopa, N-acetylglycine, lactic acid, paracetamol, and of course benzene) an energy convergence of 0.05 kJ/mol was achieved when using $R_{\mathrm{mMM}} = 20$ Å and $R_{\mathrm{DS\ cutoff}} \leq 300$ Å which made the DS method quite competitive to the ES technique in terms of speed (< 20 seconds), for the aminobutyric acid the upper summation limit $R_{\mathrm{DS\ cutoff}}$ had to be extended to 600 Å which increased the computation time to almost 90 seconds. For the 181-atom decapeptide molecule, the DS calculation used almost 10 minutes (when using the 2015 Intel E3-1505M v5 microprocessor) to get to within 0.001 kJ/mol from the target value which is primarily due to the required

large R_{mMM} value of 80 Å: approximately 7 out of 10 minutes were spent on evaluating interatomic interactions via the atomic multipole moment approximation.

7) The Ewald summation (ES) method does not suffer from any of the convergence issues found in the direct summation (DS) technique. Also, unlike the DS method that requires the user to specify numerical values for several input parameters, a set of reliable parameters for the Ewald summation is determined automatically in the code based on the crystal structure information provided.
8) The exact potential-corrected Ewald summation (ES) implementation is fast when compared to the DS method. Depending on the system and specifics of the DS computations, the ES method has been found to be 2-10 times faster than DS all the while providing more precise energies.
9) The computational performance of both the Ewald (ES) and direct (DS) summation implementations in the new version of XPDROP has been further improved by i) reducing precision in evaluation of the Löwdin α-function (Lα) when integrating the exact potential from 80-bit to 64-bit, and ii) increasing the GFortran compiler optimization level from $-\mathrm{O2}$ to $-\mathrm{Ofast}$ which enables several additional code optimizations including `-ffast-math`, `-funsafe-math-optimizations` and `-fexcess-precision=fast` that disregard the IEEE, ANSI and/or ISO compliance (Free Software Foundation, 2019). A combination of these features noticeably reduces the calculation time while having a very minor effect (<10^{-5} kJ/mol) on the precision of the evaluated

electrostatic interaction energies. For example, the ES calculation time for the decapeptide structure decreased from 60.5 seconds to 48.5 seconds, while that for the DS calculation reduced from 596 seconds to 538 seconds, all when using the 2015 Intel E3-1505M v5 microprocessor.

10) An additional set of ES and DS calculations for the decapeptide structure performed using 2012 AMD FX-8350 and 2019 AMD Ryzen 7 3700X computer processors shows a significant improvement in the floating-point performance of the AMD newest Zen2 architecture (Ryzen 7 3700X) relative to the old Piledriver core (FX-8350). Despite a 10% lower base clock speed, Ryzen 7 3700X was not only able to complete the decapeptide ES and DS calculations more than two times faster than FX-8350, but also holds a sound 20-30% lead over the 2015 Intel E3-1505M v5 CPU.

In conclusion, it is without reservation that we recommend using the XDPROP implementation of the proposed Ewald summation (ES) method with the exact potential (EP) correction for calculation of the electrostatic interaction energies in molecular crystals: it is numerically stable and fast even for larger systems. For example, for a crystal structure of the tested decapeptide (Demizu et al., 2016; CSD code IPUNAK) with 181 atoms in a molecule and four symmetry-equivalent molecules in the unit cell (Z=4), the evaluation of the electrostatic interaction energy using the described Ewald summation method took, depending on the level of optimization of the code, 37-46 seconds on a 2019 AMD Ryzen 7 3700X central processing unit, 49-61 seconds on a 2015 Intel E3-1505M v5 microprocessor, and 77-117 seconds on an old 2012 AMD FX-8350 CPU.

References

Abramov, Y. A., Volkov, A., Wu, G. & Coppens, P. (2000). *Acta Cryst.* A**56**, 585-591.

Advanced Micro Devices, Inc. (2018). *AMD Optimizing C/C++ Compiler*, version 1.2.1. http://developer.amd.com/amd-aocc (accessed on 23 September, 2018).

Aguado, A. & Madden, P. A. (2003). *J. Chem. Phys.* **119**, 7471-7483.

Allen, F. H., Kennard, O., Watson, D. G., Brammer, L., Orpen, A. G. & Taylor, R. (1995). *International Tables for Crystallography, Volume C*. Dordrecht, Netherlands: Kluwer Academic Publishers.

Allen, M. P. & Tildesley, J. (1987). Computer Simulation of Liquids, Clarendon, Oxford Press.

Arnold, A. & Holm, C. (2005). Adv. Polym. Sci. **185**, 59–109.

Bacon, G. E., Curry, N. A. & Wilson, S. A. (1964). *Proc. Roy. Soc. (London) Ser. A* **279**, 98-110.

Bader, R. F. (1990). *Atoms in Molecules: A Quantum Theory* Oxford: Claredon Press.

Becke, A. D. (1988a). *J. Chem. Phys.* **88**, 2547-2553.

Becke, A. D. (1988b). *Phys. Rev. A* **38**, 3098-3100.

Becke, A. D. (1993). *J. Chem. Phys.* **98**, 5648-5652.

Blaha, P., Schwarz, K., Sorantin, P. & Trickey, S. B. (1990). *Comp. Phys. Comm.* **59**, 399-415.

Buckingham, A. D. (1959). *Quart. Rev. Chem. Soc.* **13**, 183-214.

Buckingham, A. D. (1967). *Adv. Chem. Phys.* **12**, 107-142.

Buckingham, A. D. (1978). In *Intermolecular Forces: From Diatomics to Biopolymers*; edited by B. Pullman, Wiley: New York.

Buckingham, A. D., Fowler, P. W. & Hutson, J. M. (1988). *Chem. Rev.* **88**, 963-988.

Bunge, C. F., Barrientos, J. A., Bunge, A. V. & Cogordan (1992). *J. A. Phys. Rev.* A **46**, 3691-3696.

Bunge, C. F., Barrientos, J. A. & Bunge, A. V. (1993). *At. Data and Nucl. Data Tables* **53**, 113-162.

Causa, M., Dovesi, R., Pisani, C., Colle, R. & Fortunelli, A. (1987). *Phys. Rev. B* **36**, 891-897.

Challacombe, M., White, C. & Martin Head-Gordon, M. (1997) *J. Chem. Phys.* **107**, 10131-10140

Clark, S. J., Segall, M. D., Pickard, C. J., Hasnip, P. J., Probert, M. J., Refson, K., Payne, M. C. (2005). *Zeitschrift fuer Kristallographie* **220**, 567-570.

Clementi, E. & Roetti, C. (1974). *At. Data and Nucl. Data Tables* **14**, 177-478.

Coombes, D. S., Price, S. L., Willock, D. J., Leslie, M. (1996). *J. Phys. Chem.* **100**, 7352-7360.

Coppens, P. (1997). *X-ray Charge Densities and Chemical Bonding.* Oxford University Press, New York.

Coppens, P., Abramov, Y., Carducci, M., Korjov, B., Novozhilova, I., Alhambra, C. & Pressprich, M. R. (1999). *J. Am. Chem. Soc.* **121**, 2585-2593.

Craven, B. M. & Weber, H. P. (1983). *Acta Cryst.* B**39**, 743-748

Cummins, P. G., Dunmur, D.A., Munn, R.W., Newham, R. J. (1976) *Acta Cryst.* A**32**, 847-853/

de Leeuw, S. W., Pertain, J. W., & Smith, E. R. (1980). *Proc. R. Soc. Lond. A* **373**, 27-56.

DragonEgg (2018). *A GCC plugin that replaces GCC's optimizers and code generators with those from the LLVM project*. http://dragonegg.llvm.org (accessed on 23 September, 2018).

Dalhus, B. & Görbitz, C. H. (1996). *Acta Cryst.* C**52**, 1759-1761.

Demizu, Y., Okitsu, K., Yamashita, K., Doi, M., Misawa, T., Oba, M., Tanaka, M. & Kurihara, M. (2016). *Eur. J. Org. Chem.* **2016**, 2815-2820.

Demizu, Y, Yamashita, H., Doi, M., Misawa, T., Oba, M., Tanaka, M. & Kurihara, M. (2015). *J. Org. Chem.* **80**, 8597-8603.

Destro, R., Bianchi, R. & Morosi, G. (1989). *J. Phys. Chem.* **93**, 4447-4457.

Destro, R., Marsh, R. E. & Bianchi, R. (1988). *J. Phys. Chem.* **92**, 966-973.

Destro, R., Roversi, P., Barzaghi, M. & Marsh, R. E. (2000). *J. Phys. Chem. A.* **104**, 1047-1054.

Dominiak, P. M., Volkov, A., Li, X., Messerschmidt, M. & Coppens, P. (2007). *J. Chem. Theory Comput.* **3**, 232-247.

Dovesi, R., Orlando, R., Civalleri, B., Roetti, C., Saunders, V. R. & Zicovich-Wilson, C. M. (2005). *Zeitschrift fuer Kristallographie* **220**, 571-573.

Fennell, C. J. & Gezelter, J. D. (2006). *J. Chem. Phys.* **124**, 234104-1 - 234104-12.

Field, M. J., Bash, P. A. & Karplus, M. (1990). *J. Comput. Chem.* **11**, 700-733.

Fincham, D. (1994). *Mol. Sim*. **13**, 1–9.

Free Software Foundation, Inc. (2019). *GCC 9.2 GNU Fortran Manual.* http://gcc.gnu.org (accessed on 22 December, 2019).

Frenkel, D. & Smit, B. (2002). Understanding Molecular Simulation, 2nd Ed, Academic Press.

Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., J. A. Montgomery, J., Vreven, T., Kudin, K. N., Burant, J. C., Millam, J. M., Iyengar, S. S., Barone, J. T. V., Mennucci, B., Cossi, M., Scalmani, G., Rega, N., Petersson, G. A., Nakatsuji, H., Hada, M., Ehara, M., Toyota, K., Fukuda, R., Hasegawa, J., Ishida, M., Nakajima, T., Honda, Y., Kitao, O., Nakai, H., Klene, M., Li, X., Knox, J. E., Hratchian, H. P., Cross, J. B., Bakken, V., Adamo, C., Jaramillo, J., Gomperts, R., Stratmann, R. E., Yazyev, O., Austin, A. J., Cammi, R., Pomelli, C., Ochterski, J. W., Ayala, P. Y., Morokuma, K., Voth, G. A., Salvador, P., Dannenberg, J. J., Zakrzewski, V. G., Dapprich, S., Strain, A. D. D. M. C., Farkas, O., Malick, D. K., Rabuck, A. D., Raghavachari, K., Foresman, J. B., Ortiz, J. V., Cui, Q., Baboul, A. G., Clifford, S., Cioslowski, J., Stefanov, B. B., Liu, G., Liashenko, A., Piskorz, P., Komaromi, I., Martin, R. L., Fox, D. J., Keith, T., Al-Laham, M. A., Peng, C. Y., Nanayakkara, A., Challacombe, M., Gill, P. M. W., Johnson, B., Chen, W., Wong, M. W., Gonzalez, C. & Pople, J. A. (2004). *Gaussian 03, Revision C.02*. Wallingford CT: Gaussian, Inc.

Gale, J. D. J. (1997). *Chem. Soc., Faraday Trans.* **93**, 629-637.

Gatti, C. (1999). *TOPOND98 Users' Manual.*

Gavezzotti, A. (2002a). *J. Mol. Struct.* **615**, 5-12.

Gavezzotti, A. (2002b). *J. Phys. Chem. B* **106**, 4145-4154.

Gavezzotti, A. (2003). *J. Phys. Chem. B* **107**.

Gavezzotti, A. (2005). *Zeitschrift fuer Kristallographie* **220**, 499-510.

Geller, M. (1962). *J. Chem. Phys.* **36**, 2424–2428.

Geller, M. (1963a). *J. Chem. Phys.* **39**, 84–89.

Geller, M. (1963b). *J. Chem. Phys.* **39**, 853–854.

Geller, M. (1964a). *J. Chem. Phys.* **41**, 4006–4007.

Geller, M. (1964b). *Technical Report No. 32-673*, Jet Propulsion Laboratory, Pasadena, California.

Geller, M. & Griffith, R. W. (1964) *J. Chem. Phys.* **40**, 2309–2325.

Giese, T. J., Panteva, M. T., Chen, H. & York, D. M. (2015). *J. Chem. Theory Comput.* **2015**, 11, 436−450.

Gill, P. M. W. (1994). *Adv. Quant. Chem.* **25**, 141-205.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Hariharan, P. C. & Pople, J. A. (1973). *Theor. Chim. Acta* **28**, 213-222.

Harris, F. E. & Michels, H. H. (1967). *Adv. Chem. Phys.* **13**, 205–266.

Harsh, M. (2016). *"Run Bash on Ubuntu on Windows"*. Building Apps for Windows. Microsoft. https://blogs.windows.com/windowsdeveloper/2016/03/30/run-bash-on-ubuntu-on-windows/ (accessed on 19 January, 2020).

Head-Gordon, M., Pople, J. A. & Frisch, M. J. (1988). *Chem. Phys. Lett.* **153**, 503-506.

Helgaker, T., Jørgensen, P. & Olsen, J. (2000). *Molecular Electronic Structure Theory*, Wiley.

Heyes, D. M. (1981). J. *Chem. Phys.* **74**, 1924-1929.

Hirshfeld, F. L. (1971) *Acta Cryst.* B**27**, 769-781.

Hirshfeld, F. L. & Rzotkiewicz, S. (1974). *Mol. Phys.* **27**, 1319-1343.

Howard, S. T., Hursthouse, M. B., Lehmann, C. W. & Poyner, E. A. (1995). *Acta Cryst.* B**51**, 328-337.

Huzinaga, S. (1967). *Suppl. Prog. Theor. Phys*. **40**, 52-77.

Jackson, R. A. & Catlow, C. R. A. (1988). *Mol. Simul.* **1**, 207-224.

Jensen, L. (1958). *Phys. Rev.* **110**, 661-669.

Jones, H. W. (1980). *Int. J. Quantum Chem.* **18**, 709–713.

Jones, H. W. (1981). *Int. J. Quantum Chem.* **20**, 1217–1224.

Jones, H. W. (1984). *Phys. Rev. A* **30**, 1–4.

Jones, H. W. (1991). *J. Comp. Chem.* **12**, 1217–1222.

Jones, H. W. (1992). *Int. J. Quantum Chem.* **41**, 749–754.

Jones, H. W. (1993). *Int. J. Quantum Chem.* **45**, 21–30.

Jones, H. W. & Weatherford, C. A. (1978). *Int. J. Quantum Chem. Symp.* **12**, 483–488.

Jones, H. W. & Weatherford, C. A. (1989). *J. Mol. Struct. (Theochem)* **199**, 233–243.

Kistenmacher, T. J., Rand, G. A. & Marsh, R. E. (1974). *Acta Cryst.* B**30**, 2573-2578.

Kisiel, Z. (2001). In *Spectroscopy from Space*, edited by J. Demaison, K. Sarka, E. A. Cohen, Kluwer Academic Publishers, Dordrecht.

Kisiel, Z. (2006). *PROSPE - Programs for ROtational SPEctroscopy,* http://info.ifpan.edu.pl/~kisiel/prospe.htm

Koetzle, T. F., Frey, M. N., Lehmann, M. S. & Hamilton, W. C. (1973). *Acta Cryst.* B**29**, 2571-2575.

Kolafa, J. & Perram, J. (1992). *Mol. Simul*. **9**, 351-368.

Lamichhane, M., Gezelter, J. D. & Newman, K. E. (2014b). *J. Chem. Phys.* **141**, 134110-1 - 134110-12.

Lattner, C. & Adve, V. (2004). *Proc. 2004 Int. Symp. on Code Gen. and Opt. (CGO'04)*, Palo Alto, California.

Lee, C., Yang, W. & Parr, R. G. (1988). *Phys. Rev. B* **37**, 785-789.

Leslie, M. (1987). *Inf. Q. Comput. Simulation Condensed Phases*. **26**, 61-67.

Loferer, M. J., Loeffler, H. H. & Liedl, K. R. (2003). *J. Comput. Chem.* **24**, 1240-1249.

Löwdin, P. O. (1956). *Adv. Phys.* **5**, 1–171.

Mackay, M. F. (1975). *Cryst. Struct. Comm.* **225**, 225-228.

Macchi, P. & Coppens, P. (2001). *Acta Cryst.* A57, 656-662.

Matthey, T. (2005). *Plain Ewald and PME*. http://protomol.sourceforge.net/ewald.pdf (accessed on 10 January 2020).

Michael, J. R. & Volkov, A. (2015). *Acta Cryst.* A**71**, 245–249

Møller, C. & Plesset, M. S. (1934). *Phys. Rev.* **46**, 618-622.

Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536.

Nguyen, D., & Volkov, A. (2019) *Acta Cryst.* A**75**, 448-464.

Nymand, T. M. & Linse, P. (2000). *J. Chem. Phys.* **112**, 6152-6160.

Olver, F. W. J., Olde Daalhuis, A. B., Lozier, D. W., Schneider, B. I., Boisvert, R. F., Clark, C. W., Miller, B. R., Saunders, B. V., Cohl, H. S., & McClain, M. A. (2019). *NIST Digital Library of Mathematical Functions.* http://dlmf.nist.gov/, Release 1.0.25 of 2019-12-15.

Oracle Corporation (2017). *Oracle Developer Studio 12.6: Numerical Computation Guide*. Part No: E77791. http://docs.oracle.com/en (accessed on 23 September, 2018).

PassMark Software Pty Ltd, Redwood City, California, USA. http://www.cpubenchmark.net (accessed on 19 January, 2020).

Paturle, A. & Coppens, P. (1988). *Acta Cryst.* A**44**, 6-7.

Pisani, C. & Dovesi, R. (1980). *Int. J. Quantum. Chem.* **17**, 501-516.

Reine, S., Helgaker, T. & Lindh, R. (2012) *WIREs Comput. Mol. Sci.*, 2 290–303.

Rycerz, Z. & Jacobs, P. (1992). *Mol. Simul.* **8**, 197-213.

Saunders, V. R. (1984). *Faraday Symp. Chem. Soc.* **19**, 79-84.

Saunders, V. R., Dovesi, R., Roetti, C., Causa, M., Harrison, N. M., Orlando, R. & Zicovich-Wilson, C. M. (1999). *CRYSTAL98 User's Manual*

Sagui, C., Pedersen, L. G. & Darden, T. A. (2004). *J. Chem. Phys.* **120**, 73-87.

Sala, J., Guàrdia, E. & Masia, M. *J. Chem. Phys.* **133**, 234101-1 - 234101-14.

Schwarz, K., Blaha, P. & Madsen, G. K. H. (2002). *Comp. Phys. Comm.* **147**, 71-76.

Sharma, R. R. (1976). *Phys. Rev. A* **13**, 517–527.

Schouten, A., Kanters, J. A. & Krieken, J. v. (1994). *J. Mol. Struct.* **323**, 165-168.

Silverstone, H. J. (1966). *J. Phys. Chem.* **45**, 4337–4341.

Silverstone, H. J. (1967a). *J. Phys. Chem.* **46**, 4368–4376.

Silverstone, H. J. (1967b). *J. Phys. Chem.* **46**, 4377–4380.

Silverstone, H. J. (1967c). *J. Phys. Chem.* **47**, 537–540.

Silverstone, H. J. (2014). *J. Phys. Chem. A* **118**, 11971–11974.

Silverstone, H. J. & Moats, R. K. (1977). *Phys. Rev. A* **16**, 1731–1732.

Singh, D. (1994). *Plane waves, pseudopotentials and the LAPW method*. Kluwer Academic.

Smith, W. (1982) CCP5 Info. Quart. 4, 13.

Smith, W. (1987). *Inf. Q. Comput. Simulation Condensed Phases*. **21**, 43-50.

Smith, W. (1998) CCP5 Info. Quart. 4, 46, 18-30.

Spackman, M. A. (1986a). *J. Chem. Phys.* **85**, 6587-6601.

Spackman, M. A. (1986b). *J. Chem. Phys.* **85,** 6579-6586.

Spackman, M. A. (2006). *Chem. Phys. Lett.* **418**, 158-162.

Spackman, M. A. & Maslen, E. N. (1986). *J. Phys. Chem.* **90**, 2020-2027.

Spackman, M. A., Weber, H. P. & Craven, B. M. (1988). *J. Am. Chem. Soc.* **110**, 775-782.

Stenhammar, J., Trulsson, M. & Linse, P. (2011). *J. Chem. Phys.* **134**, 224104-1 - 224104-5.

Stenqvist, B., Trulsson, M., Abrikosov, A. I. & Lund, M. (2015). *J. Chem. Phys.* **143**, 014109-1 - 014109-9.

Stewart, R. F. (1976). *Acta Cryst.* A**32**, 565-574.

Stogryn, D. E. & Stogryn, A. P. (1966). *Mol. Phys.* **11**, 371-393.

Stone, A. J. (1996). *The Theory of Intermolecular Forces* Oxford: Clarendon Press.

Su, Z. & Coppens, P. (1998). *Acta Cryst.* A**54**, 646-652

Swerts, B., Van Droogenbroeck, J., Peeters, A. & Van Alsenoy, C. (2002). *J. Phys. Chem. A* **106**, 4245-4250.

Toukmaji, A. Y. & Board, J. A. J. (1996). *Comp. Phys. Comm.* **95**, 73-92.

Towler, M. D., Zupan, A. & Causa, M. (1996). *Comp. Phys. Comm.* **98**, 181-205.

Tsirelson, V. G. & Ozerov, R. P. (1996). *Electron Density and Bonding in Crystals*. Bristol, England / Philadelphia, USA: Institute of Physics Publishing.

Volkov, A., Koritsanszky T. S. & Coppens, P. (2004a). *Chem. Phys. Lett.* **391**, 170–175.

Volkov, A., Li, X., Koritsanszky, T. S. & Coppens, P. (2004b). *J. Phys. Chem. A* **108**, 4283-4300.

Volkov, A. & Macchi, P. (2005). Unpublished results.

Volkov, A., Macchi, P., Farrugia, L. J., Gatti, C., Mallinson, P., Richter, T. & Koritsanszky, T. (2016) *XD2016 - A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors*.

Williams, D. E. & Cox, S. R. (1984). *Acta Cryst.* B**40**, 404-417.

Williams, D. E. (1989) *Cryst. Rev.* **2**, 3-25.

Wilson, C. C. (2000). *Zeitschrift fuer Kristallographie* **215**, 693-701.

Wolf, D., Keblinski, P., Phillpot, S. R. & Eggebrecht, J. (1999). *J. Chem. Phys.* **110**, 8254 - 8282.

Wolfram Research, Inc., (2019) *Mathematica, Version 12.0*, Champaign, IL.

Zahn, D., Schilling, B. & Kast, S. M. (2002). *J. Phys. Chem. B* **106**, 10725-10732.

APPENDIX A: Supplementary Material

Table S4.1 Crystallographic data for the benchmark systems

	ACG[1]	**BENZ**[2]	**LAC**[3]	**LDOPA**[4]	**PARA**[5]
CSD code	ACYGLY11	BENZEN	YILLAG	LDOPAS03	HXACAN13
Formula	$C_4H_7N_1O_3$	C_6H_6	$C_3H_6O_3$	$C_9H_{11}N_1O_4$	$C_8H_9N_1O_2$
Space group	$P2_1/c$	Pbca	$P2_12_12_1$	$P2_1$	$P2_1/a$
a	4.859	7.440	5.490	13.619	12.667
b (Å)	11.546	9.550	8.422	5.232	9.166
c	9.811	6.920	9.345	6.062	7.073
α	90	90	90	90	90
β (°)	97.060	90	90	97.56	115.51
γ	90	90	90	90	90
V (Å^3)	546.2	491.7	432.1	428.2	741.2
Z	4	4	4	2	4

	ABA[6]	**ALA**[7]	**GLY**[8]	**SER**[9]	**VAL**[10]
CSD code	GAMBUT02	LALNIN03	GLYCIN85	LSERIN01	VALIDL02
Formula	$C_4H_9NO_2$	$C_3H_7N_1O_2$	$C_2H_5N_1O_2$	$C_3H_7N_1O_3$	$C_2H_{11}N_1O_2$
Space group	$P2_1/a$	$P2_12_12_1$	$P2_1/n$	$P2_12_12_1$	$P\bar{1}$
a	8.214	5.928	5.087	8.599	5.222
b (Å)	10.000	12.260	11.773	9.348	5.406
c	7.208	5.794	5.460	5.618	10.838
α	90	90	90	90	90.89
β (°)	110.59	90	111.99	90	92.34
γ	90	90	90	90	110.02
V (Å^3)	554.2	421.1	303.2	451.6	287.1
Z	4	4	4	4	2

[1] Mackay, 1975; standard crystallographic settings, originally reported unit cell is 4.859 Å, 11.546 Å, 14.633 Å, 90°, 138.29°, 90°.

[2] Bacon, Curry & Wilson, 1964

[3] Schouten, Kanters & Krieken, 1994

[4] Howard *et al.*, 1995

[5] Wilson, 2000

[6] Weber, Craven & McMullan, 1983

[7] Destro, Marsh & Bianchi, 1988

[8] Destro *et al.*, 2000

[9] Kistenmacher, Rand & Marsh, 1974

[10] Dalhus & Görbitz, 1996

CHAPTER V: CONCLUDING REMARKS AND OUTLOOK[6]

This thesis includes our three studies that describe fully-analytical, precise, reliable, and computationally efficient methods for the evaluation of the electrostatic interaction energies in molecular systems of various sizes within the pseudoatom representation of the electron density (Hirshfeld, 1971; Stewart, 1976; Hansen & Coppens, 1978; Coppens, 1997; Tsirelson & Ozerov, 1996).

The main points/achievements of the studies can be summarized as follows:

1) In the first study (Chapter II; Nguyen, Kisiel & Volkov, 2018), we introduce an accurate, numerically stable, and fast analytical approach that successfully replaced the numerical (quadrature) evaluation of the electron–nuclear attraction (NAI) and electron– electron repulsion integrals (ERI) of the exact potential integrals in the original EPMM implementation (Volkov, Koritsanszky & Coppens, 2004). In this approach, the electron-nuclear attraction integrals (NAI) are evaluated using the electronic potential, $V^{\text{elec}}(\mathbf{r})$, formulas derived in (Volkov, King, Coppens & Farrugia, 2006); whereas the electron-electron repulsion integrals (ERI) are evaluated with the help of the Löwdin α-function (Löwdin, 1956; Sharma, 1976; Silverstone & Moats, 1977; Jones & Weatherford, 1978) and *C*-matrix of Jones and Weatherford (Jones & Weatherford, 1978, 1989; Jones, 1980, 1981, 1984, 1991, 1992, 1993). This analytical approach (aEP) and its combination with the multipole moment (MM) approximation for interatomic interactions beyond 5 Å have been

[6] Portions of this section also appear in *Acta Crystallographica Section A: Foundations and Advances* (Nguyen *et al.*, 2018, 2019).

programmed using standard Fortran90 with 64-bit fixed numerical precision (*aka.* double-precision, DP, and binary64) arithmetic in the in-house version of the *XDPROP* program of the *XD* package (Volkov *et al*., 2016), and tested on a set of 28 benchmark molecular dimers composed of H, C, N, and O atoms, ranging in size from water-water to dodecapeptide-dodecapeptide dimers using electron densities constructed via the University at Buffalo Aspherical Atom Databank (Volkov, Koritsanszky *et al*., 2004; Dominiak *et al*., 2007). For all benchmark systems, this aEP/MM method provides electrostatic interaction energies within uncertainty ≤ 0.2 kJ/mol. Also, the running time for all benchmark aEP/MM calculations never exceeded 4 seconds on a 2012 central processing unit (2.8 GHz AMD Opteron 6348) and 3 seconds on a relatively modern processor (2.8 GHz Intel Xeon E3-1505M v5).

2) In the second study (Chapter III; Nguyen & Volkov, 2019), we present another comparably accurate and fast analytical approach to calculate the two- electron–electron repulsion integrals (ERI) needed for evaluation of the electrostatic interaction energies using the Fourier transform (FT) technique proposed by Silverstone (1966), Geller (1963*a*,*b*, 1964*a*,*b*) and Harris & Michels (1967). In the aim to explore the computational aspects of both the Löwdin α -function and Fourier transform methods and to determine which method is better suited for evaluation of ERI, we have implemented the two analytical methods in Fortran 90 in the in-house version of the *XDPROP* program of the *XD* package (Volkov *et al*., 2016) with different levels of numerical precision arithmetic (*i.e.*, the 64-, 80- and 128-bit) and have tested various types of Fortran compilers such as Lahey/Fujitsu

LF64 Express 8.1 compiler (Lahey Computer Systems, 2011), Oracle Developer Studio 12.6 (Oracle Corporation, 2017*a*), GFortran8.2.1 (Free Software Foundation, 2018), and AMD Optimizing C/C++ compiler suite (Advanced Micro Devices, 2018) with the DragonEgg (2018) Fortran compiler plugin that uses code optimizers from the LLVM project (Lattner & Adve, 2004), with specified code optimization options. From results of the primarily tests, the 64-, 80-bit Löwdin α-function, and 128-bit Fourier transform have been found to be the most effective and reliable implementations. The three implementations are combined with the multipole moment (MM) approximation to form different variations of our new EP/MM method that have been tested on the same set of benchmark molecular dimers employed in the study I (Nguyen, Kisiel & Volkov, 2018). The final results show that (*i*) regardless of the Coulomb integral evaluation method applied, the numerical precision used, and the compiler employed (LF64 and GFortran), the resulting electrostatic interaction energies E_{es} values agree to within 5×10^{-5} kJ/mol (usually, better) which is more than satisfactory considering the typical uncertainties of the determined theoretical and experimental pseudoatom parameters, (*ii*) in terms of speed, the clear winner is the 64-bit Löwdin α-function implementation (regardless whether LF64 or GFortran are used), followed closely by the 80-bit Löwdin α-function code compiled with GFortran, and the 128-bit Fourier transform method compiled with LF64. For example, the most time-consuming calculation in our study (first enkepahlin dimer, Enk1) takes 2.9, 6.8 and 9.3 seconds using the three methods listed above, respectively. These three implementations are highly recommended for calculation of intermolecular electrostatic interaction energies in practice.

3) In the third study (Chapter V), we present an extension of the analytical exact potential and multipole moment method (EP/MM) to the evaluation of the molecular electrostatic energy in infinite molecular crystals using the two newly developed implementations: (*i*) the Ewald summation (ES) which includes interactions up to the hexadecapolar level and the exact potential correction for the short-range electron density penetration effects, and (*ii*) the enhanced EP/MM-based direct summation (DS) which uses the analytical exact potential integration for short-range interatomic interactions, the atomic multipole moment (aMM) approximation for the intermediate-range interatomic interactions, and the molecular multipole moment (mMM) approximation for the long-range intermolecular interactions. Both methods have been also implemented using Fortran90 in the in-house version of XDPROP, part of the XD software suite (Volkov, Macchi *et al.*, 2016), and have been thoroughly examined in terms of speed and precision using a number of small molecular systems (serine, glycine, valine, alanine, aminobutyric acid, N-acetylglycine, L-dopa, lactic acid, paracetamol, and benzene), and a decapeptide molecular crystal structure (Demizu et al., 2016; CSD code IPUNAK) with 181 atoms in the molecule and a total of four molecules (Z=4) in the unit cell. The final results of the exhaustive series of calculations performed by the two methods show that (*i*) both methods provide reliable electrostatic interaction energies in molecular crystals though the DS method requires extra care when selecting the optimal sets of parameters, (*ii*) the

electron density penetration effects, correctly accounted for by the two methods, contribute 28-64% to the total electrostatic energy in the examined systems, and thus can not be neglected, (*iii*) ES method is fast when compared to the DS method. Depending on the system and specifics of the DS computations, the ES method has been found to be 2-10 times faster than the DS calculations all the while providing more precise energies. Consequently, we recommend using the Ewald summation method (ES) with the exact potential correction implemented in the new version of XDPROP for calculation of electrostatic interaction energies in molecular crystals: it is numerically stable and fast, even for larger systems such as the tested decapeptide structure. For example, using a 2015 2.8 GHz Intel Xeon E3-1505M v5 computer processor, a 64-bit implementation of the Löwdin α-function combined with the `-Ofast` GFortran compiler optimization option, the Ewald summation method evaluates the electrostatic interaction energies with a numerical precision of at least 10^{-5} kJ/mol under 6 seconds for any of the tested small molecular crystal structures, and in 48.5 seconds for the decapeptide structure.

To conclude, the newly developed hybrid aEP/MM scheme is highly recommended for calculation of electrostatic interaction energies in molecular systems of various sizes as long as their electron densities are defined within the pseudoatom electron density formalism. For the XDPROP users who desire the higher speed, we recommend the 60- or 80-bit-precision-based Löwdin α-function method, while those users who value precision and accuracy over speed will find the 128-bit Fourier transform technique more suitable for their needs. Additionally, the XDPROP users are highly recommended to use the Ewald summation (ES) with the exact potential (EP) correction for evaluation of electrostatic interaction energies in infinite molecular crystal structures.

Currently, our future plans include: (i) exploration of the performance of the described methods to larger macromolecular systems, (ii) extension of the Ewald summation method to calculation of other important electrostatic properties in molecular crystal structures such as the electrostatic potential, electric field, and electric field gradient, and (iii) formulation and implementation of the interaction tensors $T_{ab}^{\alpha\beta\gamma\ldots\omega}$ [in equation (1.9)] and their modified/screened counterparts $\hat{T}_{ab}^{\alpha\beta\gamma\ldots\omega}$ [equation (4.7)] in the spherical presentation for many advances as suggested by Stone (2013).

The new techniques described in this thesis are available in the latest version of the XDPROP module of the XD2016 software suite (Volkov, Macchi *et al.*, 2016).

REFERENCES

Abramov, Y. A., Volkov, A., Wu, G. & Coppens, P. (2000). *Acta Cryst. A* **56**, 585-591.

Barnett, M. P. & Coulson C. A. (1951). *Phil. Trans. Royal Soc. London. Series A, Math. Phys. Sci.* **243**, 221-249.

Becke, A. D. (1988). *J. Chem. Phys.* **88**, 2547-2553.

Berlu, L. (2004). *J. Theor. Comput. Chem*. **2**, 257–267.

Bickelhaupt, F. M.; Baerends, E. J. (2000). *Rev. Comput. Chem*. **15**, 1-86.

Buckingham, A. D. (1967). *Adv. Chem. Phys.* **12**, 107-142.

Buckingham, A. D. (1978). In *Intermolecular Forces: From Diatomics to Biopolymers*; edited by B. Pullman, New York: Wiley.

Buckingham, A. D., Fowler, P. W. & Hutson, J. M. (1988). *Chem. Rev*. **88**, 963–988.

Bunge, C. F., Barrientos, J. A. & Bunge, A. V. (1993). *At. Data and Nucl. Data Tables* **53**, 113-162.

Bunge, C. F., Barrientos, J. A., Bunge, A. V. & Cogordan (1992). *J. A. Phys. Rev.* A **46**, 3691-3696.

Clementi, E. & Roetti, C. (1974). *At. Data and Nucl. Data Tables* **14**, 177-478.

Coppens, P. (1992) *The structure factor. International Tables for Crystallography, Vol. B: Reciprocal space*, 1st ed., edited by U. Shmueli, ch. 1.2. Dordrecht: Kluwer Academic Publishers.

Coppens, P. (1997). *X-ray Charge Densities and Chemical Bonding.* Oxford University Press, New York.

Dykstra, C. E. (1993). *Chem. Rev*. **93**, 2339-2353.

Fernandez Rico, J., Fernandez, J. J., Lopez, R. & Ramirez, G. (2000). *Int. J. Quantum. Chem.* **78**,137-145.

Filter, E. & Steinborn, E. O. (1978). *Phys. Rev. A.* **18**, 1-11.

Flierler, U. & Stalke, D. (Eds.) (2012). *Electron Density and Chemical Bonding I: Experimental Charge Density Studies.* Berlin, Heidelberg: Springer-Verlag.

Freitag, M. A., Gordon, M. S., Jensen, J. H. & Stevens, W. J. (2000). *J. Chem. Phys*. **112**, 7300–7306.

Gatti, C. & Macchi, P. (2012). Editors. *Modern Charge-Density Analysis*. Netherlands: Springer.

Gavezzotti, A. (2002). *J. Mol. Struct.* **615**, 5-12.

Geller, M. (1962). *J. Chem. Phys.* **36**, 2424–2428.

Geller, M. (1963a). *J. Chem. Phys.* **39**, 84–89.

Geller, M. (1963b). *J. Chem. Phys.* **39**, 853–854.

Geller, M. (1964a). *J. Chem. Phys.* **41**, 4006–4007.

Geller, M. (1964b). *Technical Report No. 32-673*, Jet Propulsion Laboratory, Pasadena, California.

Geller, M. & Griffith, R. W. (1964) *J. Chem. Phys.* **40**, 2309–2325.

Guseinov, I. I. (1970). *J. Phys. B. At. Mol. Phys.* **3**, 1399-1412.

Guseinov, I. I. & Mamedov, B. A. (2002). *J. Math. Chem.* **32**, 309–322.

Hansen, N. K. & Coppens, P. (1978). *Acta Cryst.* A**34**, 909–921.

Harris, F. E. & Michels, H. H. (1967). *Adv. Chem. Phys.* **13**, 205-266.

Hirshfeld, F. L. (1971) *Acta Cryst.* B**27**, 769-781.

Hirshfeld, F. L. & Rzotkiewicz, S. (1974). *Mol. Phys.* **27**, 1319-1343.

Huzinaga, S. (1967). *Suppl. Prog. Theor. Phys.* **40**, 52-77.

Jeziorski, B., Moszynski, R., Szalewicz, K. (1994). *Chem. Rev.* **94**, 1887-193.

Jones, H. W. (1980). *Int. J. Quantum Chem.* **18**, 709-713.

Jones, H. W. (1981). *Int. J. Quantum Chem.* **20**, 1217-1224.

Jones, H. W. (1984). *Phys. Rev. A* **30**, 1-4.

Jones, H. W. (1991). *J. Comp. Chem.* **12**, 1217-1222.

Jones, H. W. (1992). *Int. J. Quantum Chem.* **41**, 749-754.

Jones, H. W. (1993). *Int. J. Quantum Chem.* **45**, 21-30.

Jones, H. W. & Weatherford, C. A. (1978). *Int. J. Quantum Chem. Symp.* **12**, 483-488.

Jones, H. W. & Weatherford, C. A. (1989). *J. Mol. Struct. (Theochem)* **199**, 233-243.

Kairys V., Jensen J. (1999). *Chem. Phys. Lett.* **315**, 140-144.

Kisiel, Z. (2001). In *Spectroscopy from Space*, edited by J. Demaison, K. Sarka, E. A. Cohen. Dordrecht: Kluwer Academic Publishers.

Kisiel, Z. (2006). *PROSPE - Programs for ROtational SPEctroscopy,* Kitaigorodsky, A. I. (1973). *Molecular Crystals and Molecules.* New York: Academic Press.

Koritsanszky, T. S. (2001). *Chem. Rev*. **101**, 1583-1628.

Lesiuk, M. & Moszynski, R. (2014). *Phys. Rev. E.* **90**, 063318.

Löwdin, P. O. (1956). *Adv. Phys.* **5**, 1-171.

Macchi, P. & Coppens, P. (2001). *Acta Cryst.* A57, 656-662.

McLean, A. D. & Yoshimine, M. (1968). *IBM J. Res. Dev.* **12**, 206-233.

Michael, J. R. & Volkov, A. (2015). *Acta Cryst.* A**71**, 245–249.

Murray, C. W., Handy, N. C. & Laming, G. J. (1993). *Mol. Phys.* **78**, 997-1014.

Nguyen, D., Kisiel, Z. & Volkov, A. (2018) *Acta Cryst.* A**74**, 524–536.

Nguyen, D. & Volkov, A. (2019) *Acta Cryst.* A**75**, 448-464.

O-Ohata, K. & Ruedenberg, K. (1966). *J. Math. Phys.* **7**, 547–559.

Öztekin, E. (2004). *Int. J. Quantum Chem.* **100**, 236–243.

Öztekin, E. & Özcan, S. (2007). *J. Math. Chem.* **42**, 337–351.

Öztekin, E., Yavuz, M. & Atalay, S. (2001). *Theor. Chem. Acc.* **106**, 264–270.

Paturle, A. & Coppens, P. (1988). *Acta Cryst.* A**44**, 6-7.

Piquemal, J.P., Gresh, N. & Giessner-Prettre, C. (2003). *J. Phys. Chem*. **107**, 10353–10359.

Qian, W. & Krimm, S. (2006). *J. Mol. Struct. Theochem*. **766**, 93-104.

Ruedenberg, K., Roothaan, C. C. J. & Jaunzemis, W. (1956). *J. Chem. Phys.* **24**, 201-220.

Sharma, R. R. (1976). *Phys. Rev. A* **13**, 517–527.

Shestakov, A. F. (1992). *J. Struct. Chem.* **33**, 131-133.

Silverstone, H. J. (1966). *J. Phys. Chem.* **45**, 4337-4341.

Silverstone, H. J. (1967a). *J. Phys. Chem.* **46**, 4368–4376.

Silverstone, H. J. (1967b). *J. Phys. Chem.* **46**, 4377–4380.

Silverstone, H. J. (1967c). *J. Phys. Chem.* **47**, 537–540.

Silverstone, H. J. (2014). *J. Phys. Chem. A* **118**, 11971-11974

Silverstone, H. J. & Moats, R. K. (1977). *Phys. Rev. A* **16**, 1731–1732.

Slater, J. C. (1932). *Phys. Rev.* **42**, 33-43.

Spackman, M. A. (1986). *J. Chem. Phys.* **85**, 6587-6601.

Spackman, M. A. (2006). *Chem. Phys. Lett.* **418**, 158-162.

Spackman, M. A. (2007). *Acta Cryst.* A**63**, 198-200.

Spackman, M. A. & Maslen, E. N. (1986). *J. Phys. Chem.* **90**, 2020-2027.

Stewart, R. F. (1976). *Acta Cryst.* A**32**, 565-574.

Stone, A. (1981). *J. Chem. Phys. Lett*. **83**. 233-239.

Stone, A. (1996). *The Theory of Intermolecular Forces* Oxford: Clarendon Press.

Stone, A. (2013). *The Theory of Intermolecular Forces*, 2nd ed. Oxford University Press.

Su, Z. & Coppens, P. (1998). *Acta Cryst.* A**54**, 646-652.

Todd, H. D., Kay, K. G. & Silverstone, H. J. (1970). *J. Chem. Phys.* **53**, 3951-3956.

Tsirelson, V. G. & Ozerov, R. P. (1996). *Electron Density and Bonding in Crystals*. Bristol, England / Philadelphia, USA: Institute of Physics Publishing.

Volkov, A. & Coppens, P. (2004). *J. Comput. Chem*. **25**, 921-934.

Volkov, A. & Macchi, P. (2005). Unpublished results.

Volkov, A., King, H. F., Coppens, P. & Farrugia, L. J. (2006). *Acta Cryst.* A**62**, 400–408.

Volkov, A., Koritsanszky T. S. & Coppens, P. (2004). *Chem. Phys. Lett.* **391**, 170–175.

Volkov, A., Macchi, P., Farrugia, L. J., Gatti, C., Mallinson, P., Richter, T. & Koritsanszky, T. (2016). *XD2016 - A Computer Program Package for Multipole Refinement, Topological Analysis of Charge Densities and Evaluation of Intermolecular Energies from Experimental and Theoretical Structure Factors*.

Wahl, A. C., Cade, P. E. & Roothaan, C. C. J. (1964). *J. Chem. Phys.* **41**, 2578-2599.

Weatherford, C. A. & Jones, H. W. (1982). Editors. *ETO Multicenter Molecular Integrals Proceedings of the First International Conference* held at Florida A&M University, Tallahassee, Florida, U.S.A., August 3–6, 1981. Netherlands: Springer.

Weniger, E. J. & Steinborn, E. O. (1983) *J. Chem. Phys.* **78**, 6121–6132.

Zimont, S. L. & Mar’yaskin, N. Ya. (1972). *Dokl. Akad. Nauk SSSR,* **205**, 1059-1062.