RICE UNIVERSITY

Linear Scaling Density Functional Theory with Gaussian Orbitals and Periodic Boundary Conditions

by

Konstantin N. Kudin

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Gustavo Scuseria, Chairman
Professor of Chemistry

Bruce Johnson
Senior Faculty Fellow,
Department of Chemistry

Boris Yakobson
Associate Professor of Mechanical
Engineering and Materials Science

Houston, Texas

November, 2000

Abstract

Linear Scaling Density Functional Theory with Gaussian Orbitals and Periodic Boundary Conditions

by

Konstantin N. Kudin

We report methodological and computational details of our Kohn-Sham density functional method with Gaussian orbitals for systems with periodic boundary conditions (PBC). When solving iterative self-consistent field (SCF) equations of density functional theory (DFT), the most computationally demanding tasks are Kohn-Sham (or Fock) matrix formation and the density matrix update step. The former requires evaluation of the Coulomb interactions and the exchange-correlation quadrature, and in our code both of them are computed via $\mathcal{O}(N)$ techniques. An $\mathcal{O}(N)$ approach for the Coulomb problem in electronic structure calculations with PBC is developed here and is based on the direct space fast multipole method (FMM). The FMM achieves not only linear scaling of computational time with system size but also high accuracy, which is pivotal for avoiding numerical instabilities that have previously plagued calculations with large bases, especially those containing diffuse functions. The density matrix update step is carried out via the conven-

tional $\mathcal{O}(N^3)$ diagonalization of the Fock matrix, which for systems with less than ≈ 3000 basis functions is cheaper than the recently developed $\mathcal{O}(N)$ algorithms. In addition to evaluating energy, our code also computes analytic energy gradients with respect to atomic positions and cell dimensions (forces). Combining the latter with the developed in this work redundant internal coordinate algorithm for optimization of periodic systems, it becomes possible to optimize geometries of periodic structures with great efficiency and accuracy. We demostrate the capabilities of our method with benchmark calculations on polyacetylene, poly(*p*-phenylenevinylene) (PPV), and a series of carbon and boron-nitride single wall nanotubes employing basis sets of double zeta plus polarization quality, in conjunction with generalized gradient approximation and kinetic energy density dependent functionals. We also present vibrational frequencies for PPV obtained from finite differences of forces. The largest calculation reported in this work contains 244 atoms and 1344 contracted Gaussians in the unit cell.

Acknowledgments

I thank Dr. Gustavo Scusera for his advice and knowledge that guided me through my thesis work. I also thank Dr. H. Bernhard Schlegel for his help with the development of a redundant internal coordinate algorithm for optimization of periodic systems.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research focused on developing and implementing modern linear scaling density functional theory (DFT) techniques for calculations of systems periodic in one, two or three dimensions [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. One of the most important objectives of this work was to ensure that the developed methodology permitted simulations of periodic systems with the same high accuracy that current molecular DFT calculations enjoy. This makes periodic DFT applicable to problems which require the so called "chemical accuracy", or errors for calculated heats of formation not larger than 2-3 kcal/mol. The attractiveness of the DFT for such applications stems from the fact that the best DFT functionals typically provide results comparable in quality with those of more elaborate *ab initio* methods at a fraction of the computational cost [11, 12]. The computational expense of DFT is substantially less than that of second order Moller-Plesset perturbation theory (MP2) or coupled cluster (CC) methods. One of the very remarkable features of DFT methods is their proven capability of achieving $\mathcal{O}(N)$ scaling of CPU time with respect to system size, making it possible to model molecules with thousands of atoms [13]. Recent developments in MP2 and CC methodologies have reduced the scaling of these methods to near-linear as well [14, 15, 16]. However, their cost

prefactors are still significantly larger than that of state-of-the-art implementations of DFT methods.

The combination of DFT methods with Gaussian type orbitals (GTO) is very popular in calculations of molecular systems. There are also several periodic DFT programs described in the literature that employ GTO basis sets [17, 18, 19, 20, 21, 22, 23]. Some of these DFT codes are based on previous implementations of the periodic Hartree-Fock (HF) method [17, 18, 23], while others have been written as pure DFT programs [19, 20, 21, 22] and lack the hybrid functionals which require exact HF exchange. Although many techniques have been employed in solid state calculations [18], plane wave (PW) basis sets in combination with effective core potentials have traditionally played a very important role in this field. Because the quality of the PW basis set is uniform everywhere in space, one needs a large number of PWs to properly describe "sparsely" packed systems, such as polymers, surfaces, or zeolites where the valence density is very nonuniform and depends significantly on the chemical environment. This limitation makes Gaussian bases better suited to model such typically covalently bonded systems, especially those containing first row atoms (B-F), where the shortcomings of pseudopotentials are well documented [24]. Due to the larger size of the basis set, large scale calculations with PWs are affected by $\mathcal{O}(N^3)$ computational bottlenecks considerably earlier than GTO calculations. Consequently, some groups [25] have recently advocated GTOs as a way to overcome prohibitive computational expenses in very large systems.

Due to the broader availability of high quality molecular codes such as the Gaussian package [26], it is not uncommon to find studies where periodic systems

are approximated by cluster models. Such an approach works fine in many cases where chemical bonds are well localized, as for example, zeolites [27]. On the other hand, there are many problems that do require true periodicity because the interactions in the system are long ranged and the results of cluster calculations converge very slowly. Zero band gap metals and other systems with small band gaps, such as conjugated polymers, are typical examples. Among the latter, there are derivatives of polyacetylene that have conjugation lengths of about 100 monomers [28]. Oligomer calculations of such large size would be very demanding. The technologically important derivatives of poly(*p*-phenylenevinylene) (PPV) have much smaller conjugation length, about 5-10 units. However, their large unit cells also make cluster calculations quite expensive. Other examples of slowly converging cluster calculations are systems containing long-ranged electrostatic interactions, such as molecular crystals of sugars and carboxilic acids.

One of the most important components of our accurate and efficient periodic DFT code is the Fast Multipole Method (FMM) [29]. Recently, an extention of the FMM, the Gaussian very Fast Multipole Method (GvFMM), was shown to be a very efficient solution for the Coulomb problem in large electronic structure calculations with Gaussians [30]. During the course of this work, the GvFMM method was further optimized for accuracy [3], and extended to include evaluation of energy, forces [1, 2] and the electrostatic stress tensor in periodic systems [4]. In the periodic FMM [31, 32, 33, 34, 1], all lattice summations are performed entirely in direct space, and therefore, the algorithm treats equally well systems with periodicity in one, two, or three dimensions. Another important property of

the periodic GvFMM is its ability to achieve arbitrary accuracy via adjustment of two computational parameters. The high accuracy in the Coulomb problem is pivotal in avoiding numerical instabilities that have previously plagued calculations with large bases, especially those containing diffuse functions. Furthermore, the FMM requires CPU time that scales only linearly with respect to system size, thus permitting simulations of very large systems [2, 5].

In addition to developing algorithms for evaluation of the unit cell energy, we have also implemented into our DFT PBC program the techniques for computation of first analytic energy derivatives with respect to both atomic positions and cell dimensions [5]. The latter makes full geometry optimizations for systems containing a large number of independent degrees of freedom practical. To ensure the highest optimization efficiency, a redundant internal coordinate algorithm for optimization of periodic systems was developed in this work [6]. The main feature of this method is implicit adjustment of lattice dimensions via a combination of chemically meaningful internal coordinates such as bonds, angles and dihedrals. This coordinate system was shown to yield the fastest convergence among other coordinate systems in optimizations where the exact curvature of the energy surface is not known [35], such as in our case. The overall efficiency is such that our optimizer is able to locate an energy minimum for a wide range of structures in a single digit number of optimization steps.

The discussion of this work will proceed as follows. Chapter 2 presents the theory behind the DFT PBC code. Chapter 3 mentions basic features of the FMM, and describes our implementation of the method for the Coulomb problem in electronic

structure calculations with PBC. Chapter 4 discusses the non-locality of Gaussian charge distributions and introduces improved estimates for the Gaussian ranges employed in the FMM. Chapter 5 describes a few other details of the DFT PBC program which are important for computational efficiency. Chapter 6 discusses a redundant internal coordinate algorithm for optimization of periodic systems. Chapter 7 presents several systems studied by the DFT PBC code, which include isomers of polyacetylene, poly(*p*-phenylenevinylene) (PPV), and a series of carbon and boron-nitride nanotubes. Finally, Chapter 8 provides some concluding remarks.

Chapter 2

Theoretical framework

Energy and atomic forces

Our formulation of the PBC DFT method is based on Gaussian orbitals of the form [36]

$$\phi(\mathbf{r}) = (x - R_x)^l (y - R_y)^m (z - R_z)^n e^{-\alpha(r-R)^2} \tag{2.1}$$

where $R = (R_x, R_y, R_z)$ is the Gaussian center, l, m, n are integers determining the orbital angular momentum, and α is the Gaussian exponent. Calculations with periodic boundary conditions (PBC) require basis functions with proper translational symmetry. Therefore, GTOs are transformed into "crystalline orbitals" (also referred to as Bloch sums) that have the form [18]

$$\Psi_{\mathbf{k}} = \sum_{\mathbf{g}} \left[\frac{1}{\sqrt{N}} e^{i\mathbf{k}\cdot\mathbf{g}} \right] \psi_{\mathbf{g}}, \tag{2.2}$$

where $\mathbf{k} = (k_x, k_y, k_z)$ is the reciprocal lattice vector, $\psi_{\mathbf{g}}$ is a GTO ψ centered in cell $\mathbf{g}$, and i is the imaginary unit. Vector $\mathbf{k}$ classifies periodic orbitals by the irreducible representations (irreps) of the infinite translation group. Orbitals belonging to different irreps do not interact directly with each other (although they are coupled through the density matrix, see discussion below) and this allows one to

solve conventional self consistent field (SCF) equations separately for each $\mathbf{k}$ point

$$F^{\mathbf{k}}C^{\mathbf{k}} = S^{\mathbf{k}}C^{\mathbf{k}}E^{\mathbf{k}}. \tag{2.3}$$

We note that equation (2.3) is valid both for HF and DFT methods. The exponent in the Bloch orbital definition (2.2) introduces complex factors and therefore all matrices in equation (2.3) are, in general, complex. Matrix elements between periodic orbitals defined in equation (2.2) can be easily computed from matrix elements for localized GTOs

$$\langle \Phi_{\mathbf{k}} \mid A \mid \Psi_{\mathbf{k}} \rangle = \sum_{\mathbf{g}} \langle \phi_{\mathbf{0}} \mid A \mid \psi_{\mathbf{g}} \rangle e^{i\mathbf{k}\cdot\mathbf{g}} = \sum_{\mathbf{g}} A^{\mathbf{0g}}_{\phi\psi} e^{i\mathbf{k}\cdot\mathbf{g}} \quad . \tag{2.4}$$

In this equation, $A^{\mathbf{0g}}_{\phi\psi}$ is a matrix element of operator A between the Gaussian atomic orbitals, ϕ, located in the central cell $\mathbf{0}$ and ψ located in cell $\mathbf{g}$. The Kohn-Sham Hamiltonian matrix elements (or Fock matrix elements in the HF case), $F^{\mathbf{0g}}_{\mu\nu}$, include several contributions:

$$F^{\mathbf{0g}}_{\mu\nu} = T^{\mathbf{0g}}_{\mu\nu} + U^{\mathbf{0g}}_{\mu\nu} + J^{\mathbf{0g}}_{\mu\nu} + V^{\mathbf{0g}}_{\mu\nu} \tag{2.5}$$

where $T^{\mathbf{0g}}_{\mu\nu}$ is the electronic kinetic energy term, $U^{\mathbf{0g}}_{\mu\nu}$ is the electron-nuclear attraction term, $J^{\mathbf{0g}}_{\mu\nu}$ is the electron-electron repulsion term, and $V^{\mathbf{0g}}_{\mu\nu}$ is the contribution from the DFT exchange-correlation potential. $T^{\mathbf{0g}}_{\mu\nu}$ and $U^{\mathbf{0g}}_{\mu\nu}$ terms do not depend on the density matrix, while $J^{\mathbf{0g}}_{\mu\nu}$ and $V^{\mathbf{0g}}_{\mu\nu}$ do. An important feature of the Kohn-Sham Hamiltonian matrix elements , $F^{\mathbf{0g}}_{\mu\nu}$, is their exponential decay with respect to the increasing separation between the μ and ν GTOs. Such behavior arises from the individual decay of the kinetic energy term, the exchange-correlation potential term, and the exponential decay of the combined electrostatic terms. Overall, all

terms in equation (2.5) are quite similar to analogous terms in molecular calculations. The electrostatic terms ($U_{\mu\nu}^{\mathbf{0g}}$ and $J_{\mu\nu}^{\mathbf{0g}}$) include interactions of a given pair of basis functions with all the other charges (or charge distributions) in the system. The number of such interactions is infinite, and this is indeed different from the molecular case. The infinite sums can be handled using the Ewald summation techniques [18, 21, 22] or by the periodic fast multipole method [2].

The real space density matrix elements $P_{\lambda\sigma}^{\mathbf{0g}}$ required for the construction of the Coulomb, exchange, and correlation contributions can be obtained by integrating the complex density $P_{\lambda\sigma}^{\mathbf{k}}$ in reciprocal space

$$P_{\lambda\sigma}^{\mathbf{0g}} = \frac{1}{V_{\mathbf{k}}} \int P_{\lambda\sigma}^{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{g}} d\mathbf{k}, \tag{2.6}$$

where $V_{\mathbf{k}}$ is the volume of the unit cell in $\mathbf{k}$ space. The matrix $P^{\mathbf{k}}$ is obtained from the orbital coefficients $C^{\mathbf{k}}$, which are solutions to the eigenvalue equation (2.3). The transformation described by equation (2.6) is the only coupling of different $\mathbf{k}$ points during the SCF procedure. In practice, the integration is replaced by a weighted sum and the reader is referred to Ref. [18] for detailed discussions on this topic. The energy per unit cell can be computed as

$$E = \sum_{\mu\in\mathbf{0}} \sum_{\mathbf{g}} \sum_{\nu\in\mathbf{g}} P_{\mu\nu}^{\mathbf{0g}} \left(T_{\mu\nu}^{\mathbf{0g}} + U_{\mu\nu}^{\mathbf{0g}} + \frac{1}{2} J_{\mu\nu}^{\mathbf{0g}} \right) + E_{xc} + E_{NR}, \tag{2.7}$$

where E_{xc} is the exchange-correlation energy and E_{NR} is the nuclear repulsion energy. In the following, triple sums like the one in equation (2.7) will be abbreviated by $\sum_{\mu\nu\mathbf{g}}$. In order to avoid convergence problems and to maximize accuracy, it is important that electrostatic terms be grouped together into electronic (E_e) and

nuclear (E_n) terms,

$$\begin{aligned} E_e &= \frac{1}{2}\sum_{\mu\nu\mathbf{g}}(U^{\mathbf{0g}}_{\mu\nu} + J^{\mathbf{0g}}_{\mu\nu})\, P^{\mathbf{0g}}_{\mu\nu}, \\ E_n &= \frac{1}{2}\sum_{\mu\nu\mathbf{g}} U^{\mathbf{0g}}_{\mu\nu} P^{\mathbf{0g}}_{\mu\nu} + E_{NR}. \end{aligned} \tag{2.8}$$

Once the converged density is available, it is possible to compute gradients of the total energy with respect to nuclear displacements (forces). Several authors have described in the literature the required theory [37] and implementation for the HF [38, 39] and DFT [23] methods in the 1D case, as well as DFT implementations for 2D and 3D systems [22, 40]. Quite recently, Hirata and Iwata have extended the analytic formalism even further and reported HF second derivatives [41] and MP2 first derivatives [42] for 1D systems.

We note here that the formulation of analytic energy first derivatives in periodic systems is quite similar to the molecular case. We refer the reader to details in Refs. [43, 44]. The final result relevant to our present discussion is

$$\frac{dE}{dx} = \sum_{\mu\nu\mathbf{g}} P^{\mathbf{0g}}_{\mu\nu}\left(\frac{dT^{\mathbf{0g}}_{\mu\nu}}{dx} + \frac{dU^{\mathbf{0g}}_{\mu\nu}}{dx} + \frac{1}{2}\frac{dJ^{\mathbf{0g}}_{\mu\nu}}{dx}\right) - \sum_{\mu\nu\mathbf{g}} W^{\mathbf{0g}}_{\mu\nu}\frac{dS^{\mathbf{0g}}_{\mu\nu}}{dx} + \frac{dE_{xc}}{dx} + \frac{dE_{NR}}{dx}, \tag{2.9}$$

where x is the nuclear displacement under consideration, $S^{\mathbf{0g}}_{\mu\nu}$ is an overlap matrix term, and $W^{\mathbf{0g}}_{\mu\nu}$ is the real space energy-weighted density matrix computed by integrating $W^{\mathbf{k}}$. The latter is evaluated as

$$W^{\mathbf{k}} = P^{\mathbf{k}} F^{\mathbf{k}} P^{\mathbf{k}}, \tag{2.10}$$

The overlap derivative $dS^{\mathbf{0g}}_{\mu\nu}/dx$ enters the force equation due to the incompleteness of the Gaussian basis set. This term is usually referred to as "Pulay force" [43].

Stress tensor

In periodic systems, unlike molecules, there is another derivative of the energy related to the geometry of the system, namely, the stress tensor, which describes the change in the system energy due to elastic strain [45]. The stress tensor is related to the derivative of the cell energy with respect to cell dimensions. Teramae *et al.* [38] were first to describe the required equations in the HF framework for the 1D case. Later, Feibelman [40] presented a GTO LSDA-based implementation of stress for 1D and 2D periodic systems. These quantities can also be computed in calculations with plane wave (PW) basis sets [46]. Previous formulations [40, 46] rely on calculations in reciprocal space. We, however, prefer to evaluate stress contributions entirely in real space using the FMM, and we present below the required equations.

In order to derive an expression for the stress tensor, we consider a uniform lattice deformation defined by

$$R_a \rightarrow \sum_b (\delta_{ab} + \epsilon_{ab}) R_b, \tag{2.11}$$

where a and b are Cartesian indices, and δ_{ab} is the Kronecker delta. For a given strain component, ϵ_{ab}, the stress can be calculated as

$$\frac{\delta E}{\delta \epsilon_{ab}} = \sum_{\mathbf{g}} \sum_{I \in \mathbf{g}} r^b_{I\mathbf{g}} \frac{dE}{dr^a_{I\mathbf{g}}}, \tag{2.12}$$

where $r^a = x, y$, or z, and the second sum is over all atoms I in cell $\mathbf{g}$. Let us apply such differentiation to the energy expression (2.7). It is convenient to classify contributions to the total energy by the number of atomic centers participating in

the interaction. In general, overlap, kinetic, and exchange-correlation contributions are effectively two center terms, nuclear attraction integrals are three center terms, and the electron repulsion contributions are four center terms. Overlap derivatives occurring in equation (2.9) are also two center terms. Let us examine the two center contributions using as a particular example the overlap derivative term in equation (2.9) (Pulay force), denoted with a superscript "S" in the following. For the sake of simplicity, we assume that the a and b axes are both along x, and the system is periodic only in one dimension, with translational vector t, and a single *integer* cell index g. Then, using the identity $x_{\nu g} = x_{\nu 0} + gt$ one obtains

$$\begin{aligned}
\frac{\delta E^S}{\delta \epsilon_{xx}} &= \sum_{\mu\nu g} W_{\mu\nu}^{0g} \left[\left(\frac{d\mu_0}{dx_{\mu 0}} \middle| \nu_g \right) x_{\mu 0} + \left(\mu_0 \middle| \frac{d\nu_g}{dx_{\nu g}} \right) x_{\nu g} \right] \\
&= \sum_{\mu\nu g} W_{\mu\nu}^{0g} \left[\left(\frac{d\mu_0}{dx_{\mu 0}} \middle| \nu_g \right) x_{\mu 0} + \left(\mu_0 \middle| \frac{d\nu_g}{dx_{\nu g}} \right) (x_{\nu 0} + gt) \right] \\
&= \sum_{Ig} F_{Ig}^S x_{Io} + t \sum_{Ig} g F_{Ig}^S, \qquad (2.13)
\end{aligned}$$

where I_g is some atomic center I in cell g, and F_{Ig}^S is the Pulay force due to the displacement of this center

$$F_{Ig}^S = \frac{dE^S}{dx_{Ig}} = \sum_{\mu\nu h} W_{\mu\nu}^{0h} \frac{d}{dx_{Ig}} (\mu_0 | \nu_h) \,. \qquad (2.14)$$

We note that the overlap integrals $(\mu_0|\nu_h)$ and their derivatives decay very rapidly with increasing distance between basis functions, resulting in only a small number of non-zero F_{Ig}^S terms. The usual atomic force due to an identical displacement of the atom I and all its replicas I_g can be written in terms of F_{Ig}^S as

$$F_I^S = \frac{dE^S}{dx_I} = \sum_g F_{Ig}^S. \qquad (2.15)$$

Consequently, we can simplify equation (2.13) into

$$\frac{\delta E^S}{\delta \epsilon_{xx}} = \sum_I F_I^S x_{Io} + t \sum_{Ig} g F_{Ig}^S. \tag{2.16}$$

The first part of equation (2.16) contains atomic gradients multiplied by atomic positions. The second part can be interpreted as the change in the system energy due to the change of the lattice vector t, but in each cell the atoms remain fixed with respect to each other. We will refer to the this second part of the stress as the "short ranged solid cell stress."

Four center (and three center) terms are more complicated than the example discussed above. For illustrative purposes, we describe below the electronic repulsion part of the stress. By itself, this contribution is divergent and must be considered together with the electron-nuclear attraction terms. For clarity, we neglect this fact for a moment and examine how the equations look like

$$\begin{aligned}
\frac{\delta E^{ee}}{\delta \epsilon_{xx}} &= \frac{1}{2} \sum_{\mu\nu g\sigma\lambda hn} P_{\mu\nu}^{0g} \; P_{\sigma\lambda}^{n,n+h} \left[\left(\frac{d\mu_0}{dx_{\mu 0}} \nu_g \middle| \sigma_n \lambda_{n+h} \right) x_{\mu 0} \right. \\
&+ \left(\mu_0 \frac{d\nu_g}{dx_{\nu g}} \middle| \sigma_n \lambda_{n+h} \right) (x_{\nu 0} + gt) + \left(\mu_0 \nu_g \middle| \frac{d\sigma_n}{dx_{\sigma n}} \lambda_{n+h} \right) (x_{\sigma 0} + nt) \\
&+ \left. \left(\mu_0 \nu_g \middle| \sigma_n \frac{d\lambda_{n+h}}{dx_{\lambda,n+h}} \right) (x_{\lambda 0} + ht + nt) \right] \\
&= \sum_{\mu\nu g\sigma\lambda hn} P_{\mu\nu}^{0g} \; P_{\sigma\lambda}^{n,n+h} \left[\left(\frac{d\mu_0}{dx_{\mu 0}} \nu_g \middle| \sigma_n \lambda_{n+h} \right) x_{\mu 0} \right. \\
&+ \left. \left(\mu_0 \frac{d\nu_g}{dx_{\nu g}} \middle| \sigma_n \lambda_{n+h} \right) (x_{\nu 0} + gt) \right] \\
&+ \frac{1}{2} \sum_{\mu\nu g\sigma\lambda hn} P_{\mu\nu}^{0g} \left(\mu_0 \nu_g \middle| \frac{d\{\sigma_n \lambda_{n+h}\}}{dx_n} \right) nt \; P_{\sigma\lambda}^{n,n+h}.
\end{aligned} \tag{2.17}$$

In equation (2.17), one finds terms similar to the overlap derivatives encountered before in (2.13). At the same time, there is a new term representing the change in

the system energy due to the expansion of the lattice of charges. Effectively, during such deformation, $\mu_n\nu_{n+h}$ pairs in cell n are kept fixed, while the cell n is displaced with respect to the cell 0 by nt. This extra term is very similar to the Coulomb contribution to the stress tensor in systems with point charges recently discussed in Ref. [4]. Finally, the short form of equation (2.17) is

$$\frac{\delta E^{ee}}{\delta \epsilon_{xx}} = \sum_I F_I^{ee} x_{Io} + t\sum_{Ig} g F_{Ig}^{ee} + \frac{1}{2}t \sum_{\mu\nu g\sigma\lambda hn} P_{\mu\nu}^{0g} \left(\mu_0 \nu_g \left| \frac{d\{\sigma_n \lambda_{n+h}\}}{dx_n} \right. \right) n P_{\sigma\lambda}^{n,n+h} \quad (2.18)$$

Here, we emphasize again that there is only a finite number of F_{Ig}^{ee} terms contributing to this equation because of the fast decaying nature of the $\mu_0\nu_g$ overlap. As mentioned above, the electron-electron and electron-nuclear interactions in infinite systems should be treated together, and in practice, we compute F_{Ig}^{ee} together with F_{Ig}^{en}.

The full equation for the stress tensor with all contributions included is

$$\begin{aligned} \frac{\delta E}{\delta \epsilon_{xx}} &= \sum_I \frac{dE}{dx_I} x_I + t \sum_{Ig} g \frac{dE}{dx_{Ig}} \\ &+ \frac{1}{2}t \left\{ \sum_{\mu\nu g\sigma\lambda hn} n P_{\mu\nu}^{0g} \left(\mu_0 \nu_g \left| \frac{d\{\sigma_n \lambda_{n+h}\}}{dx_n} \right. \right) P_{\sigma\lambda}^{n,n+h} + \sum_{\mu\nu gIn} n \frac{d}{dx_n} \left(\mu_0 \left| \frac{m_I}{r_{In}} \right| \nu_g \right) \right\} \\ &+ \frac{1}{2}t \left\{ \sum_{I\mu\nu hn} n \frac{d}{dx_n} \left(\mu_n \left| \frac{m_I}{r_{Io}} \right| \nu_{n+h} \right) + \sum_{IJn} n \frac{d}{dx_n} \frac{m_I m_J}{|r_{Io} - r_{Jn}|} \right\}, \end{aligned} \quad (2.19)$$

where I and J are nuclei with m_I and m_J charges, respectively. The electrostatic terms are grouped together such that each sum in curly brackets is convergent.

To summarize, the stress tensor can be obtained as follows. First, we differentiate the energy expression (2.7) with respect to atomic positions I_g and accumulate forces F_{Ig} separately for each I_g. These terms allow us to compute atomic gradients and the short ranged part of the solid cell force. At that point, the only part

of the stress tensor which has not been computed yet is the four- and three-center terms contributing to the last part of equation (2.19). At large separations between the interacting 0 and g cells, this becomes a point multipole problem with point multipoles being nuclei and basis function pairs $\{\mu_0\nu_g P^{0g}_{\mu\nu}\}$, and such problem was addressed in Ref. [4].

In general, a three dimensional periodic solid will have three translational vectors, and equation (2.19) will have terms for each g_i. For example, the short ranged solid cell force will look like

$$X_1 \sum_{Ig} g_1 \frac{dE}{dx_{I\mathbf{g}}} + X_2 \sum_{Ig} g_2 \frac{dE}{dx_{I\mathbf{g}}} + X_3 \sum_{Ig} g_3 \frac{dE}{dx_{I\mathbf{g}}} \tag{2.20}$$

where $\mathbf{g} = t_1 g_1 + t_2 g_2 + t_3 g_3$, and the periodic vectors are $t_i = (X_i, Y_i, Z_i)$. Also, now one has to compute all other components of the stress tensor, such as $dE/d\epsilon_{xy}$, $dE/d\epsilon_{yy}$, $dE/d\epsilon_{xz}$, $dE/d\epsilon_{yz}$ and $dE/d\epsilon_{zz}$.

Chapter 3

Coulomb problem

Introduction

The computation of Coulomb interactions between Gaussian type orbitals (GTO) and charge distributions required in electronic structure methods such as Hartree-Fock (HF) and Density Functional Theory (DFT) is a computationally demanding task. Formally, the number of four-center two-electron integrals scales as N^4 with system size N. It is also well known that in the limit of large systems, the number of significant pairs of basis functions (charge distributions) making a non-negligible contribution to the Coulomb problem becomes proportional to the system size N. Therefore, the CPU time for computing these integrals asymptotically scales as $\mathcal{O}(N^2)$ [47]. Recent extensions of the Fast Multipole Method (FMM) [29, 48] to Gaussian charge distributions, such as GvFMM [30] and similar methods [49], reduce this scaling to $\mathcal{O}(N)$.

A very important characteristic of any linear scaling method is the crossover point when it becomes less expensive than the conventional algorithm. The crossover point obviously relates to the "prefactor" of the $\mathcal{O}(N)$ method. Extensive benchmarks have shown that GvFMM is faster than state-of-the-art $\mathcal{O}(N^2)$ analytic

integration of Gaussians for systems as small as ∼20 atoms or ∼200 basis functions, depending on the accuracy and system's dimensionality [30].

In Refs. [31, 32, 33, 34, 1] it has been shown that the point charge FMM can be applied to the Coulomb problem in simulations of systems with periodic boundary conditions (PBC) to provide a viable alternative to the Ewald summation method [50]. Similar to the original FMM, the CPU time required in the periodic FMM also scales linearly with the number of charges in the simulation cell. Thus, we seek in this work the advantage of merging the periodic point charge FMM with the non-periodic GvFMM to achieve linear scaling for the Coulomb problem in electronic structure calculations with PBC.

A complicating factor present only in the FMM for electronic structure calculations is the non-locality of Gaussian charge distributions, which has to be taken into account by the introduction of spatial ranges [51, 49, 30, 3]. These ranges control the relative error of approximating charge distributions by charge multipoles. When the error is larger than a particular threshold, the integrals need to be calculated analytically regardless of the distance between charges. Controlling such error is an extremely important issue for the GvFMM efficiency, thus it will be investigated in greater detail in the following chapter.

Because the point charge FMM for periodic systems is a subset of the periodic GvFMM, we first discuss some important features of the point charge FMM. Then we mention how non-locality of Gaussian charge distributions complicates the picture. Having given some description of the FMM ideas, we proceed to describe the details of our periodic GvFMM implementation.

The basics of the Fast Multipole Method

The point charge FMM is an $\mathcal{O}(N)$ method developed by Greengard and Rokhlin for efficient simulations of large systems [29, 48]. To achieve these favorable scaling properties the FMM uses two kinds of truncated electrostatic moments. Multipole moments represent a group of charges located in some region of space as a single entity. Local moments, on the other hand, contain the information about the field induced in a given region of space by some charges located relatively far from this region. In a system with N particles, instead of always computing $\mathcal{O}(N^2)$ of separate $1/r$ terms, the FMM on many occasions evaluates interactions between multipole and local moments, which can contain contributions from a very large number of particles and thus greatly reduce the computational expense. The truncation of the expansion length at l_{max} introduces errors that can be controlled by not using moments to approximate interactions for charges or groups of charges located too close to each other. As a result all the interactions in the system are divided into the near-field (NF) and far-field (FF) depending on whether they are computed via exact or approximate multipole expressions. There is only a limited number of particles located within a certain distance from a given particle, so it is clear that the NF can be easily kept more or less constant per particle regardless of the system size. To make the FF work also constant per particle, the FMM employs a hierarchy of cubic boxes. The system under study is embedded into a cubic box (level 0), which is then divided in half along each Cartesian axis to form "children" of the parent box. The division is carried out recursively n times, yielding a tree of

depth n. The n is usually chosen such as to approximately have a constant number of particles per box at the finest level regardless of the total number of particles in the system. The number of boxes at the ith level is $(2 \times 2 \times 2)^i = 8^i$, while the total number of boxes at all levels is $8^n + 8^{n-1} + ... + 8^1 + 1 \approx (8/7) \cdot 8^n$. Consequently, the total number of boxes in the FMM tree is proportional to the number of boxes on the finest mesh level. By keeping the work per box at each level constant, the FF work can be made proportional to the number of particles. As a consequence, both the NF and the FF computations require $\mathcal{O}(N)$ operations, leading to the desired linear scaling.

Let us briefly outline how the FMM keeps the FF work per box at each level constant. First, all particles are sorted and assigned to boxes at the finest mesh level, and the multipole expansions are formed for each box at the finest mesh level. These expansions are propagated to all coarser levels of the tree. The FMM tools permit one to obtain local moments in one box from multipole moments in another box at the same level. The task for a given box is to collect the local moments (field) from boxes which are located not too close not too far from this box. The too close located boxes cannot be interacted due to large errors, while the too far located boxes could be handled with smaller computational expense via their parents located not too close to each other on their length scale. As a result, in such multipole to local moment translations, the number of boxes interacting with the given box at any level is constant. At the consequent stage the local moments from coarse mesh levels are propagated to finer levels and ultimately down to the finest level boxes, thus assembling for each such box electrostatic field contributions

from all regions of space but the closest ones. After adding the FF potential to the near-field terms, one obtains the full Coulomb potential for every particle.

The version of the point charge FMM for periodic systems [31, 32, 33, 34, 1] is similar to the non-periodic FMM, with few additions. At the multipole to local moment translation stage, the boxes in the central cell get contributions from boxes in neighboring cells as well. Moreover, at level 0 it is not possible to consider the too far located boxes (entire cells!) on the coarser levels simply because there are no such levels. At this stage one needs to compute local moment contributions from the infinite lattice of cells. Conveniently, all such cells are exact replicas of the central cell, and the whole system can be viewed as a lattice of identical multipole moments. Very efficient methods [31, 32, 33, 34, 1] were developed for performing direct space evaluation of the electrostatic potential inside such a lattice, and the computational expense of these algorithms in insignificant compared to other FMM stages. The absolute convergence of such direct space infinite summation depends on the leading terms of the total multipole moment, and the next section is devoted to the discussion of this important issue.

Convergence of the infinite summation

In the periodic FMM the infinite summations take place completely in direct space, therefore it is necessary to make sure that they are convergent. Three-dimensional periodicity represents the most difficult case since the summation is performed over R^3 space. Charge-charge interactions have a R^{-1} dependence and therefore are divergent ($R^3 \cdot R^{-1} = R^2$, $R \to \infty$). To avoid this problem, the total charge of the

cell must be zero. In electronic structure calculations this objective is easy to fulfill because the electron-nuclear mixture is neutral. Dipole-dipole terms are on the edge of convergence (R^{-3}) and lead to a shift in the total cell energy. The infinite part of the periodic FMM corresponds to the calculation of the potential inside a simulation cell that is surrounded by layers of its replicas. A non-zero dipole moment in this situation effectively leads to the accumulation of charges on two opposite sides of the bulk and an electric field in the direction of the dipole moment is created. This is unphysical because if one redefines the cell boundaries, the dipole moment may change and yield a different field. Yet, the new system is equivalent to the original one and therefore should have the same potential. To avoid this problem one may introduce an external electric field that would cancel this artificial dipole dependent field. We do this in our method by employing fictitious charges [1, 2]. Fictitious charges are placed at the corners of the lattice such that they cancel the dipole moment of the simulation cell. When the cells are arranged in a lattice, their corners upon which the fictitious charges reside coincide. As a result, the fictitious charges cancel each other everywhere except on the surface, where they counteract the charges that appeared due to the non-zero cell dipole moment. The overall effect of this procedure is that we effectively reduced the dipole-dipole R^{-3} interaction to higher order terms. The latter have no worse than R^{-4} dependence and are absolutely convergent.

Another feature of the infinite FMM summation is that the spatial average of the potential in the simulation cell will in general be some constant different from zero. This constant does not affect the energy of the system because the total charge

in the simulation cell is always zero. Quite often, however, one wants to know the absolute position of the one electron eigenvalues, and in such cases the constant needs to be set to zero. It was shown in Ref. [52] that the spatial average of the potential can be calculated from the spheropole moment, Q_o, of the simulation cell

$$V_{av} = -\frac{2\pi}{3V} Q_o = -\frac{2\pi}{3V} \int \rho(\mathbf{r}) |\mathbf{r} - \mathbf{r_0}|^2 d\mathbf{r} \tag{3.1}$$

Essentially, the spheropole moment is the trace of the Cartesian quadrupole tensor. This tensor is origin independent when the monopole and dipole of the cell are zero. Overall, when one accounts for the dipole and spheropole moments of the simulation cell, the infinite FMM potential is exactly the same as the widely used Ewald potential.

Handling the non-local nature of Gaussians within the FMM

The original FMM deals with point charges whose interactions can be computed by a simple $1/r$ formula. The multipole expansions used in the FMM replace precisely such $1/r$ terms. On the other hand, the exact interactions of the non-local charge distributions, such as Gaussians, need to be computed by a more complicated spatial integral. Therefore, in order to use the FMM methodology for non-local charge distributions, one has to first approximate their electrostatic integrals by the point charge formulae. At large separations interaction of two identical spherically symmetric Gaussian charge distributions, $e^{\alpha(r-r_0)^2}$, is indistinguishable from the interaction of two corresponding point charges. In this limit the $1/r$ formula is fully applicable. When the charge distributions are brought closer to each other, the

classical $1/r$ term will start to deviate more and more from the exact integral, and at some distance the relative difference will exceed a preselected ϵ. Consequently, each charge distribution can be assigned a range which is half of such set by ϵ distance. At distances larger than the sum of the ranges for the interacting charge distributions, the $1/r$ formula is considered to be valid, and one can utilize multipole approximations embedded within the FMM. For smaller distances, the exact integration must be performed.

To introduce the range concept into the point charge FMM, non-local charge distributions are assigned a range index in addition to the three Cartesian indices that specify the box where the given distribution is located. Consequently, when converting multipole to local moments one needs to check not only if the boxes are far enough from each other according to the point charge FMM criteria, but also to check that the distance between them is larger than the sum of their ranges. When the boxes "overlap" according to this range criteria, they will always be treated in the near-field (NF) by exact integration. The optimal choice of ranges allows a good balance between exact NF and approximate terms FF and yields better accuracy for a given computational cost compared to other schemes. Our developments in this area will be described in more detail in the following chapter.

It is worth noting that there are two limiting cases of the FMM for non-local charge distributions. When all ranges are set to 0, one recovers the point charge FMM. In contrast, if all ranges are larger than the size of the system, no multipole approximations can be used and all interactions are computed via exact integration,

yielding $\mathcal{O}(N^2)$ computational cost. Thus, the algorithm possesses sufficient flexibility to work with any intermediate situations between the two described above.

Coulomb problem in DFT calculations with PBC

As was discussed in the previous chapter, the electrostatic interactions evaluated by the periodic GvFMM include the electron-nuclear, $U_{\mu\nu}^{\mathbf{0g}}$, and electron-electron, $J_{\mu\nu}^{\mathbf{0g}}$, terms. The electron nuclear attraction term is calculated as

$$U_{\mu\nu}^{\mathbf{0g}} = \sum_{I\mathbf{n}} \left(\mu_{\mathbf{0}} \left| \frac{m_I}{r_{I\mathbf{n}}} \right| \nu_{\mathbf{g}} \right) \tag{3.2}$$

$I_{\mathbf{n}}$ is nucleus I located in cell $\mathbf{n}$ and m_I is its charge. The two-electron term is expressed as

$$J_{\mu\nu}^{\mathbf{0g}} = \frac{1}{2} \sum_{\mathbf{n}} \sum_{\sigma\lambda\mathbf{h}} P_{\sigma\lambda}^{\mathbf{n},\mathbf{n+h}} \left(\mu_{\mathbf{0}}\nu_{\mathbf{g}} \right| \sigma_{\mathbf{n}}\lambda_{\mathbf{n+h}}) \tag{3.3}$$

where $P_{\sigma\lambda}^{\mathbf{n},\mathbf{n+h}}$ are the real density space matrix elements between functions $\sigma_{\mathbf{n}}$ and $\lambda_{\mathbf{n+h}}$. An important property of such terms is their translational invariance: $P_{\sigma\lambda}^{\mathbf{n},\mathbf{n+h}} = P_{\sigma\lambda}^{\mathbf{0},\mathbf{h}}$. It is convenient to define the pair of basis functions $\mu_{\mathbf{0}}\nu_{\mathbf{g}}$ as belonging to cell 0. Consequently, a pair $\sigma_{\mathbf{n}}\lambda_{\mathbf{n+h}}$ belongs to cell $\mathbf{n}$. Therefore, in equation (3.3) we have 2 infinite sums: the internal one is over an infinite number of basis function pairs belonging to a given cell, and the external one is over infinite number of cells in the system. The total number of electrons in the system is

$$\sum_{\mu\nu\mathbf{g}} S_{\mu\nu}^{\mathbf{0g}} P_{\mu\nu}^{\mathbf{0g}} = N \tag{3.4}$$

where $S_{\mu\nu}^{\mathbf{0g}}$ is the overlap integral between a function $\mu_{\mathbf{0}}$ and $\nu_{\mathbf{g}}$. Given that the overlap between two Gaussians decreases exponentially with the distance between their centers, the sum over real space vector $\mathbf{g}$ in equation (3.4) is absolutely convergent. For large enough distances, the pair $\mu_{\mathbf{0}}\nu_{\mathbf{g}}$ is negligible and can be excluded from further consideration [53]. In this manner, we end up with a finite number of significant basis function pairs $\mu_{\mathbf{0}}\nu_{\mathbf{g}}$ in the central cell. Every other cell in the system is an exact replica of the central cell and has the same set of basis function pairs. Consequently, only one infinite summation —over all cells in the system indexed by $\mathbf{n}$— survives in equation (3.3). This situation is similar to the one in periodic systems with point charges where there is a finite number of charges in each cell but an infinite number of cells in the system. Therefore one can apply the FMM techniques developed for periodic systems with point charges.

Implementation

We have incorporated our periodic GvFMM into the Gaussian suite of programs for electronic structure calculations [26]. Cell coordinates $\mathbf{n}$ are stored in a linear array to achieve a simple loop structure; its entries are sorted according to a distance criterion. This scheme allows us to treat systems of any periodicity in a uniform fashion. For computational efficiency, a list of significant $\mu_{\mathbf{0}}\nu_{\mathbf{g}}$ basis function pairs is generated before the evaluation of electrostatic interactions. At some point the distance between cells $\mathbf{0}$ and $\mathbf{g}$ becomes so large that the overlap between any two functions from these cells becomes negligible and all such pairs are discarded.

In order for the FMM to yield high accuracy at a fixed computational expense, the leading term of the multipole expansions (i.e., the charge term) should be small. In the case of electronic structure calculations, such an objective is fairly easy to achieve by merging electronic and nuclear contributions together, because on average, the electron-nuclear charge mixture is neutral. Therefore, we have chosen in our implementation to replace all nuclei by charge distributions that are products of two very tight Gaussians with very large exponents (10^{30}). For all practical purposes, such pairs are indistinguishable from the nuclear point charges they replace. Consequently, we compute all electron-electron and electron-nuclear terms simultaneously by the same code. As a result, both in the near-field and far-field computations, large individual positive and negative contributions are merged as early as possible. The FMM accuracy for a given l_{max} is improved compared to the earlier implementation where electron-electron and electron-nuclear contributions were evaluated separately [30, 2].

Benchmarks

In order to study the properties of our periodic GvFMM, we run a series of benchmarks on periodic NaCl, (5,5), and (10,10) carbon nanotubes with varying cell sizes (Figures 1, 2, and 3). These benchmarks were carried out on one R10000/195 processor of an SGI Origin 2000. The CPU times shown in the figures are for the calculation of all significant ($U_{\mu\nu}^{\mathbf{0g}} + J_{\mu\nu}^{\mathbf{0g}}$) elements in one SCF iteration. The CPU times for the corresponding cluster calculations (i.e., clusters made out exclusively of the atoms in the unit cell) are shown by dotted lines and were produced by the

same version of our program using periodicity in 0 dimensions. The density matrix for these benchmarks is taken from a model calculation. In the near-field portion of the algorithm, we assume that the density matrix is dense and no integral screening is applied. The symmetry of the system under study is also not exploited. In the FMM part of the code, we use parameters that yield microHartree accuracy in non-periodic calculations (l_{max}=12 and Tol= 10^{-6}). Two basis sets were employed: STO-3G and 3-21G.

Our results demonstrate linear scaling computational cost with simulation cell size. As expected, the cost of the corresponding cluster calculation is lower than that of the periodic system, but only by a modest amount. Their relative difference decreases as the system size increases and strongly depends on the system periodicity. Thus, for the largest unit cell considered here, i.e. for 3-dimensional NaCl (512 atoms), the periodic calculation is about twice as expensive as the cluster calculation (Figure 1). On the other hand, for a comparable 480 atom (5,5) carbon nanotube unit cell, the difference between cluster and solid calculation is minimal (Figure 2).

The large difference between cluster and solid computational demands in NaCl may be rationalized by two factors. First, due to the periodicity, many extra $\mu_{\mathbf{0}}\nu_{\mathbf{g}}(\mathbf{g} \neq \mathbf{0})$ pairs are added to the unit cell in the solid case compared to the cluster calculation. Incidentally, one should point out that basis set requirements in periodic calculations may be significantly different than those in molecules precisely because of this reason. In other words, small basis sets that may not be deemed adequate for high-accuracy cluster calculations may yield acceptable representations

in solids. Secondly, there are many additional near-field integrals between basis function pairs belonging to neighboring cells.

In order to demonstrate the basis set effect on the periodic calculations, we compare timings for the (10,10) carbon nanotube with STO-3G and 3-21G bases sets in Figure 3. The 3-21G calculations are roughly 2 times slower than the STO-3G calculations. At the same time the relative differences for periodic and non-periodic calculations are quite small for both basis sets considered here.

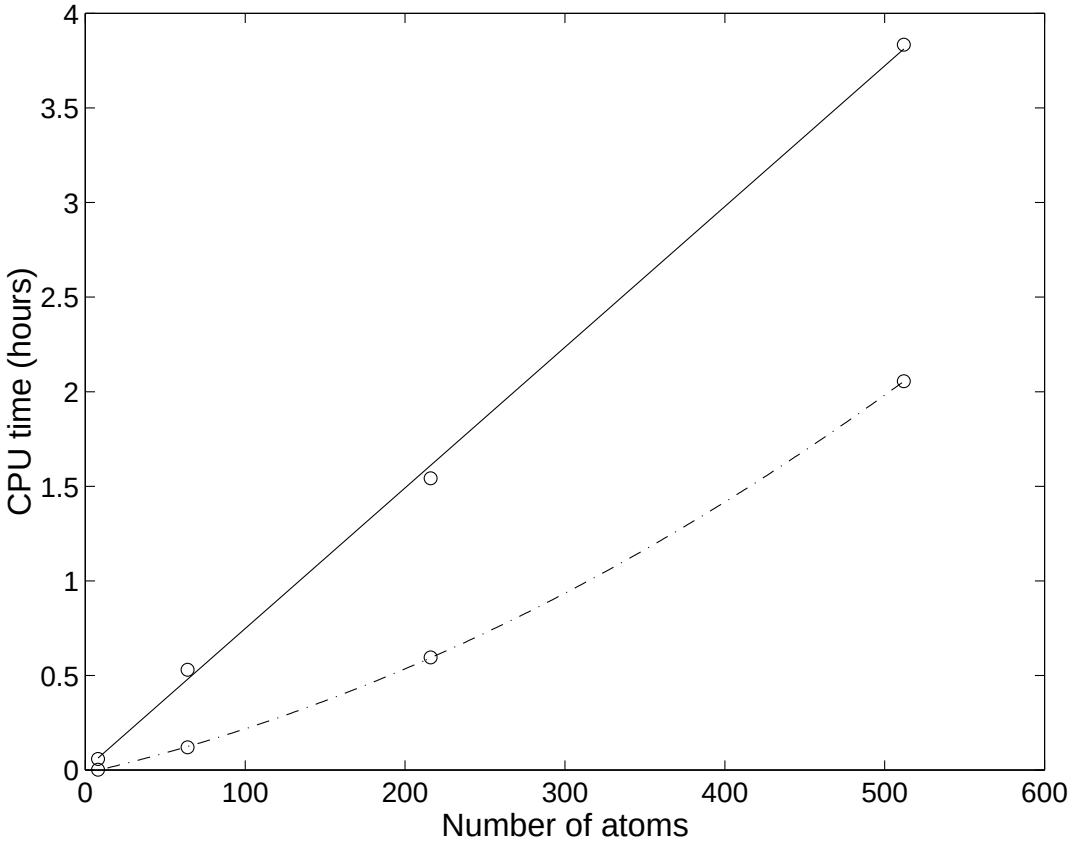

Figure 1 Coulomb matrix formation CPU times for NaCl with an STO-3G basis set. $2n$ x $2n$ x $2n$ unit cells are shown (n=1-4). The corresponding cluster calculations are indicated by a dash-dotted line.

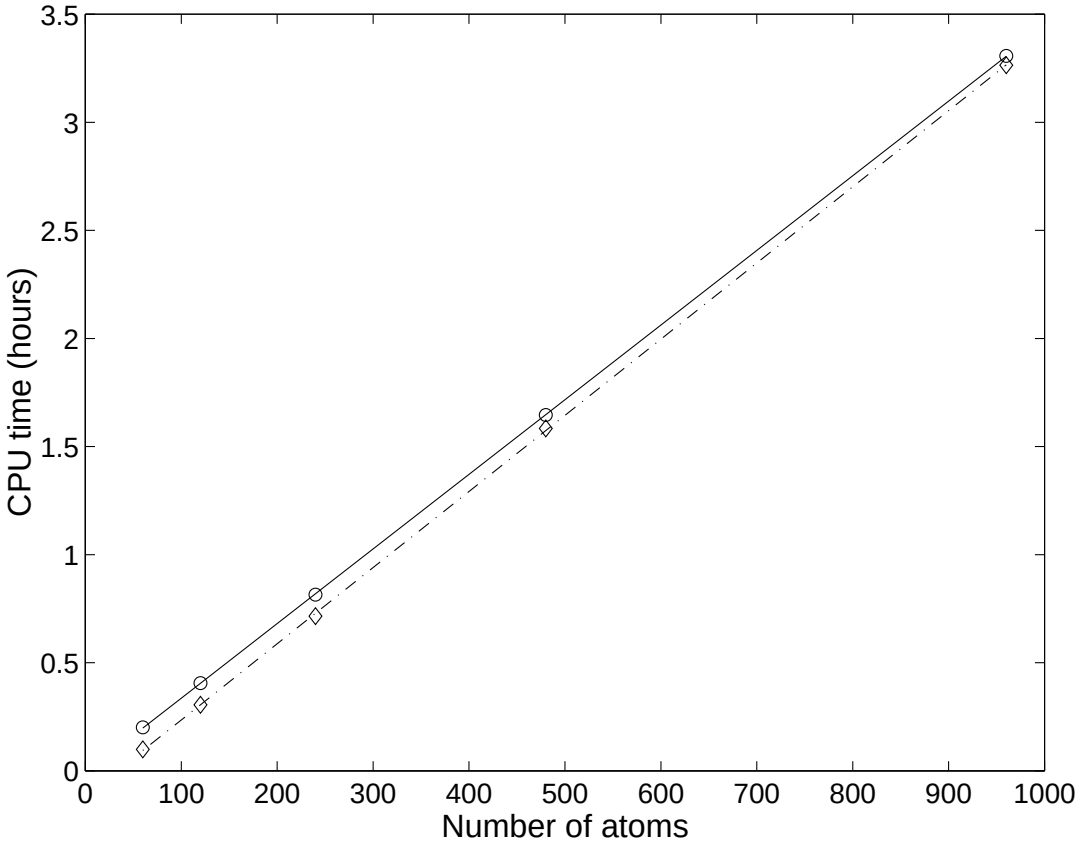

Figure 2 Coulomb matrix formation CPU times for an infinite (5,5) carbon nanotube with an STO-3G basis set. 60 x 2^n unit cells are shown (n=0-4). The corresponding cluster calculations are indicated by a dash-dotted line.

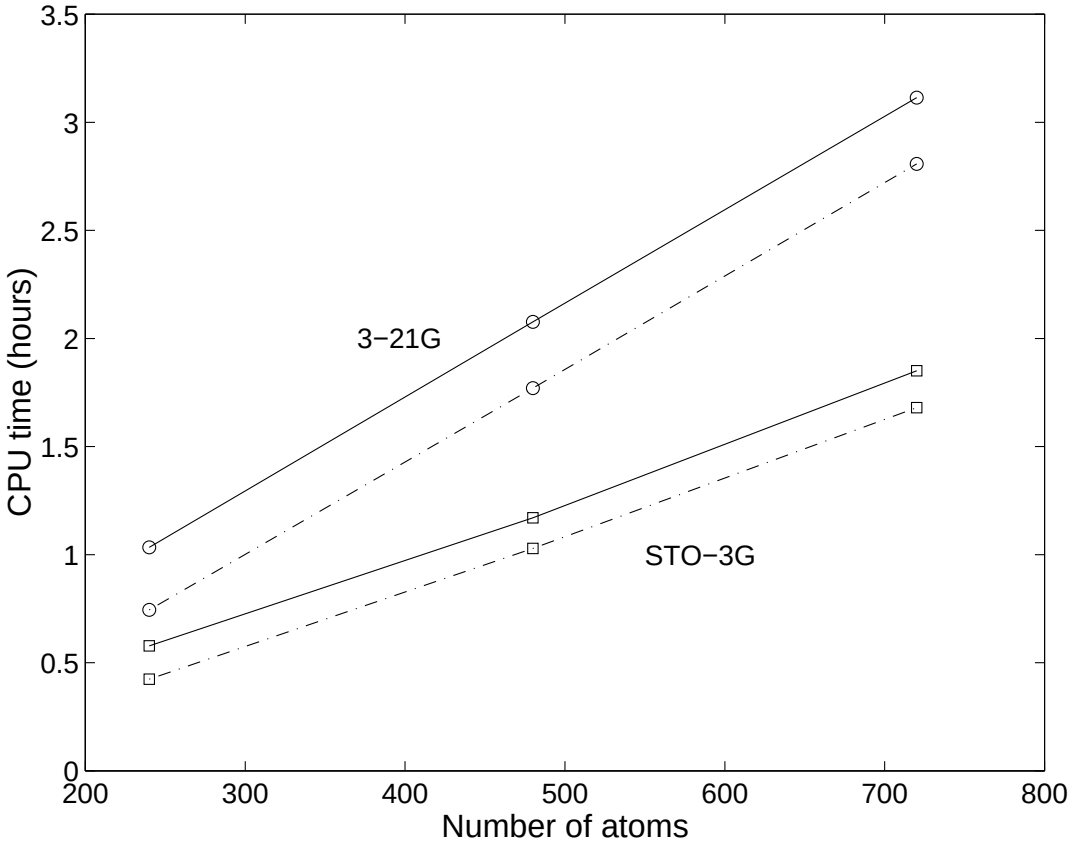

Figure 3 Coulomb matrix formation CPU times for an infinite (10,10) carbon nanotube with STO-3G and 3-21G basis sets. 240 x n unit cells are shown (n=1-3). The corresponding cluster calculations are indicated by a dash-dotted line.

Chapter 4

Range definitions for Gaussian-type charge distributions

Introduction

In the Fast Multipole Method (FMM), some fraction of the exact integrals is approximated by multipole-multipole interactions. Therefore, it is very important to estimate the distance where it is possible to replace interacting charge distributions by point multipoles while keeping the error in the total energy at some predetermined level. In the following, we refer to this error as *discretization* error. Truncating the length of multipole expansions to a finite size (l_{max}) yields a different error which we will refer to as *truncation* error. The truncation error has recently been addressed in Ref. [54]. It is worth noting that controlling the multipole truncation error has only limited success because even infinite order multipole expansions will not yield the exact energy when continuous distributions are too close to each other.

Some of the multipole based algorithms, such as tree codes [55, 56], can use information about both interacting charge distributions to determine the minimum separation for their discretization. In contrast, the Gaussian very Fast Multipole Method (GvFMM) [30] or the Continuous Fast Multipole Method (CFMM) [51],

do not have such an opportunity. In these algorithms, ranges must be defined for each charge distribution in advance without using any data about other distributions. A simple scheme to compute such ranges has been proposed by White and Head-Gordon [51, 49] and adopted with modifications in earlier GvFMM implementations [30, 57, 58]. This method does not distinguish charge distributions with different angular momentum and assigns ranges solely on the basis of the Gaussian exponent. The main purpose of the ranges is to divide all the interactions in a system between a near-field (NF) and a far-field (FF). The near-field portion of the Coulomb potential is computed via exact analytic integration. The far-field contributions are calculated with the help of multipole expansions. Better estimates of discretization error allow one to increase the accuracy of the calculation that takes a given amount of CPU time simply by moving interactions treated with higher than needed accuracy from NF to FF while simultaneously moving interactions treated with lower than needed accuracy from FF to NF. Such redistribution is possible not only for the FMM based algorithms but for any code that replaces exact integrals by multipole interactions. We have explored the features of non-zero angular momentum charge distributions and obtained a range scheme that offers a better balance between NF and FF compared to the previous scheme mentioned in the literature [51, 30, 49, 57, 58].

Derivations

Atom-centered Cartesian Gaussian type functions (CGTF) introduced by Boys [36] are used in many quantum chemistry programs and defined by

$$\phi_a(\mathbf{r}) = (x - R_x)^l (y - R_y)^m (z - R_z)^n e^{-\alpha(r-R)^2} \tag{4.1}$$

where $R = (R_x, R_y, R_z)$ is the atomic position, and α is referred to as the Gaussian exponent. The total angular momentum of the CGTF is the sum of Cartesian angular momenta $j = l + m + n$. Functions with $j = 0, 1, 2, 3$ are usually referred to as s, p, d, f orbitals, respectively. The convenience of CGTF stems from the fact that their product is a sum of several other CGTF with angular momentum ranging from 0 up to $j_1 + j_2$. Therefore, to analyze the spatial extent of the Gaussian charge distributions one only needs to look at the properties of the usual Gaussian, such as the one in Equation (4.1).

An electrostatic repulsion integral between two normalized s type gaussian charge distributions with exponents α and β, centered at P and Q is computed as follows

$$\left(s_P^\alpha | s_Q^\beta\right) = \frac{1}{|P-Q|}\, erf(\sqrt{\frac{\alpha\beta}{\alpha+\beta}|P-Q|^2}) = \frac{1}{|P-Q|}\,(1-\epsilon) \tag{4.2}$$

The error function erf quickly approaches 1 as its argument grows, which makes $(1 - \epsilon)$ a good representation of the erf. The originally proposed definition of Gaussian ranges [51, 30] is based on the estimate of the error caused by replacing the exact $(s|s)$ integral by the approximate expression similar to equation (4.2),

yielding a range definition

$$R_0 = \sqrt{\frac{2}{\alpha}} erf^{-1}(1-\epsilon) \tag{4.3}$$

where ϵ is the required absolute accuracy of the approximation. We will also refer to ϵ as tolerance. Then the integer ranges used in the FMM are computed as

$$R_i = \text{int}[R_0/L_{box}] + 1 \tag{4.4}$$

where L_{box} is the FMM box length. The s ranges also were used for higher angular momentum functions though this was not rigorously justified. It is worth noting that when one uses the integer ranges defined in equation (4.4), then even if the estimate for R_0 is not good, in cases when the ratio R_0/L_{box} is small (< 3) the rounding operation on average will increase the ranges quite substantially percentage-wise, and to some extent will hide the errors in R_0. Therefore, cases where R_0/L_{box} ratios are large such as calculations with very small FMM box sizes (in practice, 0.8-1.2 Bohrs) or diffuse basis sets (6-31+G) should be very good indicators of whether the estimates for higher angular momentum functions need to be adjusted. And indeed, benchmarks have demonstrated that in calculations where there was a substantial fraction of distributions with large ratios R_0/L_{box}, the absolute accuracy of the Coulomb energy was much lower than for lower R_0/L_{box} ratios with the same discretization tolerance ϵ. Such observations prompted us to look at higher angular momentum functions in more detail.

Let us consider a three dimensional (3D) s Gaussian. We wish to examine it in one dimension (1D) and obtain the same expectation value of $< r >$ as in three dimensions. Therefore, we also include in this 1D Gaussian the Jacobian of the

transformation from the Cartesian to polar coordinates, r^2

$$f_0(r) = N_0 e^{-\alpha r^2} r^2 \tag{4.5}$$

where N_0 is a normalization constant such that the integral over dr is 1. By analogy with s charge distributions, we represent higher angular momentum distributions as

$$f_l(r) = N_l e^{-\alpha r^2} r^{(2+l)} \tag{4.6}$$

We note that all non-zero angular momentum charge distributions have regions with positive and negative charge so that the total charge of the distribution is zero. However, here we are more concerned with the extent of the distribution. The function $f_l(r)$, defined by equation (4.6), roughly describes the ability of a non-zero angular momentum charge distribution to overlap with other distributions. We compute the expectation value of $< r >_l$ for several different l

$$< r >_0 = \frac{\int_0^\infty e^{-\alpha r^2} r^2 r dr}{\int_0^\infty e^{-\alpha r^2} r^2 dr} = \frac{2}{\sqrt{\pi\alpha}}$$

$$< r >_1 = \frac{3}{4}\sqrt{\frac{\pi}{\alpha}}; \quad < r >_2 = \frac{8}{3}\frac{1}{\sqrt{\pi\alpha}}; \quad < r >_3 = \frac{15}{16}\sqrt{\frac{\pi}{\alpha}} \tag{4.7}$$

and generalize the results into recursion relations

$$< r >_k = < r >_{k-2} \frac{k+2}{k+1}; \quad < r >_k = \frac{k+2}{2\alpha < r >_{k-1}} \tag{4.8}$$

Note that the expectation value of $< r >_0$ has the same $\alpha^{-1/2}$ dependence as the original definition of range given by equation (4.3). So, the idea central to the

new range definition is to take the range of an s charge distribution with the same exponent α and multiply it by the ratio of $< r >_l$ to $< r >_0$

$$\frac{< r >_1}{< r >_0} = \frac{3\pi}{8}; \quad \frac{< r >_2}{< r >_0} = \frac{4}{3}; \quad \frac{< r >_3}{< r >_0} = \frac{15\pi}{32} \quad ... \tag{4.9}$$

Table 1 Multiplication factors for the new ranges as a function of angular momentum l.

0	1	2	3	4	5	6
1.0000	1.1781	1.3333	1.4726	1.6000	1.7181	1.8286

Numerical values of the ratios for several different l are shown in Table 1. As one can see, in the new scheme the ranges increase with the angular momentum of the charge distribution.

Another way to improve the balance between the NF and FF computations is to require the absolute approximation error rather than the relative error to be below a certain threshold. Many charge distributions present in quantum chemistry calculations have quite a small overlap prefactor U. Defining the accuracy (Acc) being sought as

$$Acc = -log_{10}\,(\epsilon) \tag{4.10}$$

and requiring that the absolute rather than the relative error of a given interaction be less than a given $Tol = -log_{10}\,(\epsilon)$ we obtain

$$Acc = Tol + log_{10}\,U \tag{4.11}$$

The net effect of such an adjustment is an overall decrease of the ranges allowing the use of multipole approximations for a larger number of integrals.

Benchmarks

In this section we focus our attention on the discretization errors. To remove the effect of the multipole truncation errors, we use high $l_{max} = 25$ in all our benchmarks. Higher l_{max} values are not necessary because discretization errors are always substantially larger than multipole truncation errors for all tests carried out.

The ranges of all charge distributions are controlled through a single parameter Tol. It is quite clear that for a given Tol the new range scheme increases the number of the exact integrals and leads to smaller discretization errors than the previous scheme with the same Tol. Therefore, a meaningful approach to compare these ranges is to analyze the dependence of their accuracy on the number of analytic integrals computed. The latter can be roughly gauged by the CPU time required for the near-field portion of the FMM. Evidently, a better range scheme should yield higher accuracy at a fixed NF computational cost. We note that FF timings are very similar for calculations that treat the same number of interactions in the FF, and we therefore omit FF timings from our discussion completely.

In order to demonstrate the benefits of the new ranges defined here and implemented in the *Gaussian* program [26], we present benchmarks for a 10-glycine chain (1D), a $C_{54}H_{18}$ graphene sheet (2D), and a $C_{35}H_{36}$ diamond chunk (3D). Diffuse 6-31+G basis sets were used in all calculations. We also employ a rather small FMM box length of 1.2 Bohrs in order to make the discretization errors introduced by the ranges more pronounced. The near-field timings obtained on an IBM 3CT workstation for one SCF iteration are shown in Figures 4, 5 and 6. Each

curve was produced by varying Tol from 3 to 9 and then connecting the points that correspond to consecutive Tol values. It is also worth mentioning that for a fixed Tol, the previous scheme usually produces an error in the total energy about 100 times larger than the error of the new scheme, but NF CPU times are also smaller. In order to achieve an error of about 10^{-6} a.u., new ranges require $Tol = 6$, whereas the previous ranges require a somewhat larger value.

The data in all figures follows the same trend after taking into account usual error fluctuations, especially noticeable for the new ranges in Figure 5. The new ranges practically always yield higher accuracy at fixed computational expense than the original ranges. For a given NF CPU time, new range scheme usually leads to 1.5-2 orders of magnitude smaller total errors than the previous scheme. Thus, selective increase of the ranges depending on the angular momentum of the charge distributions does help achieve better balance between the near and the far-field portions of the FMM and improve the accuracy significantly.

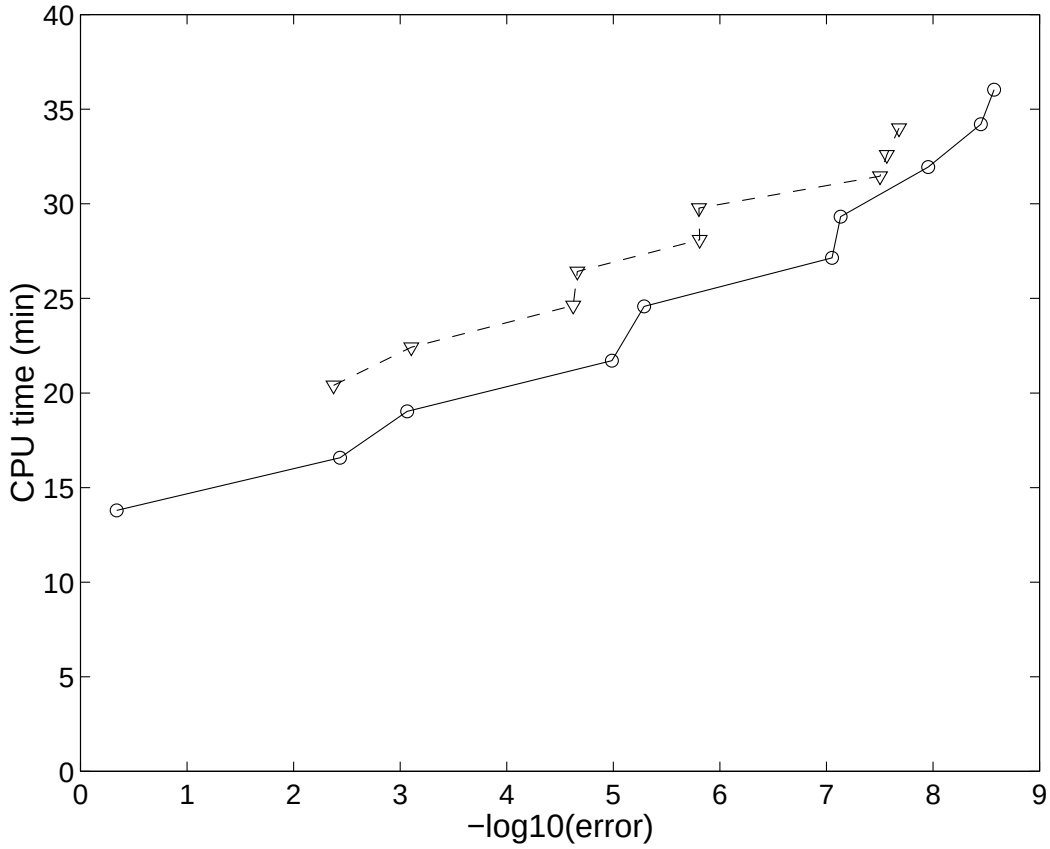

Figure 4 Near field CPU time versus error in the Coulomb energy for 10-glycine. $\bigtriangledown$ and dashed line represent the original ranges used in the previous work [30, 57, 58]; $\bigcirc$ and solid line represent new ranges.

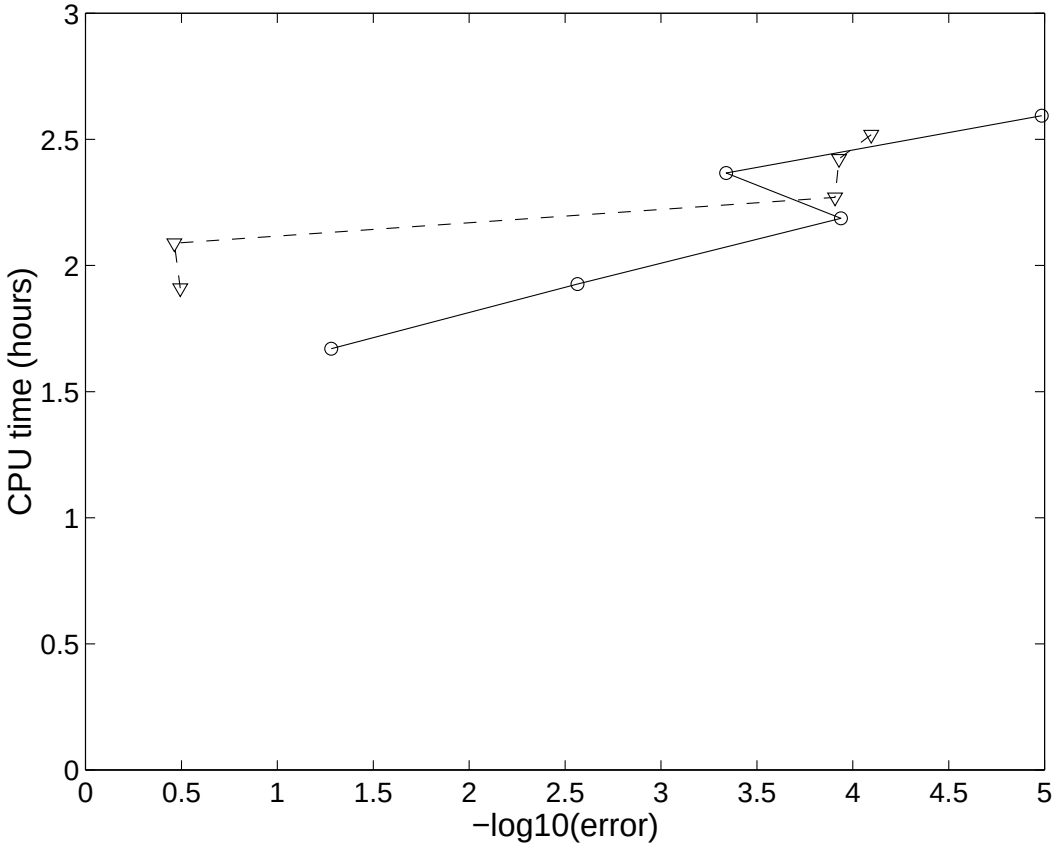

Figure 5 Near field CPU time versus error in the Coulomb energy for $C_{54}H_{18}$ graphene sheet. $\bigtriangledown$ and dashed line represent the original ranges used in the previous work [30, 57, 58]; $\bigcirc$ and solid line represent new ranges.

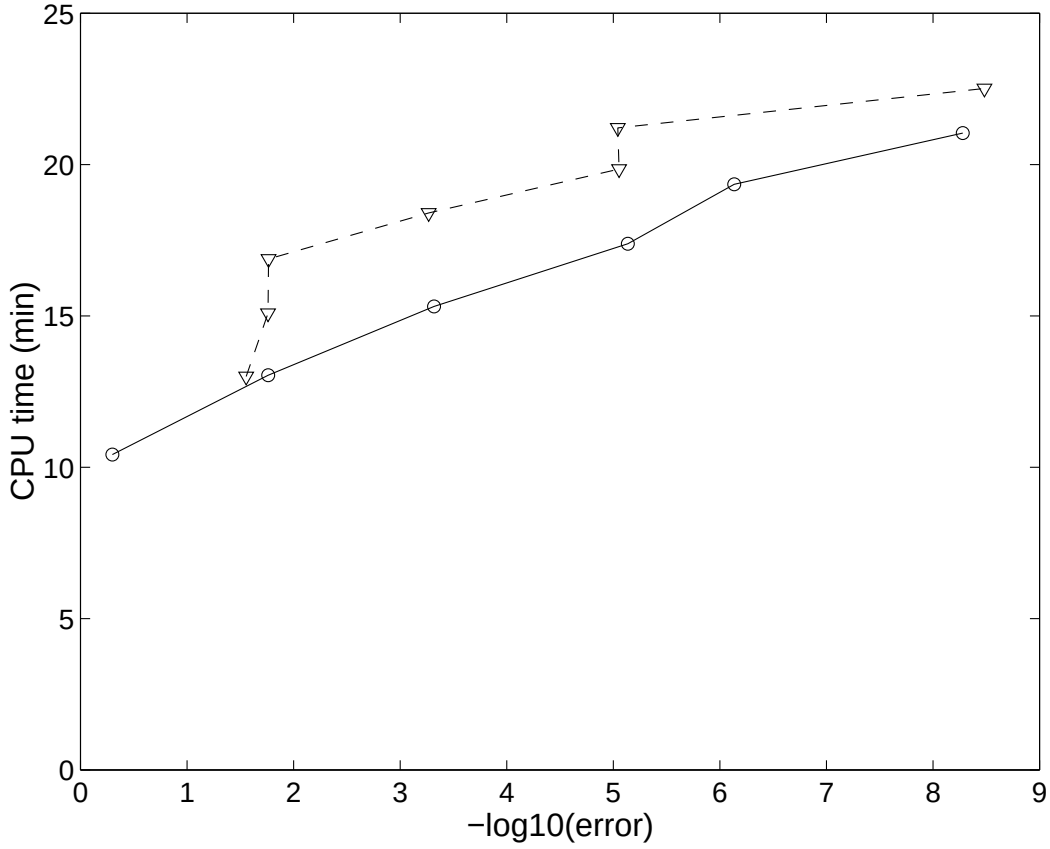

Figure 6 Near field CPU time versus error in the Coulomb energy for $C_{35}H_{36}$ diamond chunk. $\bigtriangledown$ and dashed line represent the original ranges used in the previous work [30, 57, 58]; $\bigcirc$ and solid line represent new ranges.

Chapter 5

Various implementation issues for the PBC code

DFT numerical integration

In calculations with PBC, the numerical integration of the DFT exchange-correlation terms has to be carried out over the volume of the unit cell including weighted contributions —in principle— from all atoms in the infinite system

$$E_{xc} = \int_{\mathbf{r}\in U} \epsilon_{xc}(\mathbf{r})\ d\mathbf{r} = \sum_{I\mathbf{g}} \int_{\mathbf{r}\in U} \epsilon_{xc}(\mathbf{r}) \frac{\pi_{I\mathbf{g}}(\mathbf{r})}{\sum_{J\mathbf{h}} \pi_{J\mathbf{h}}(\mathbf{r})}\ d\mathbf{r} = \sum_{I\mathbf{g}} \int_{\mathbf{r}\in U} \epsilon_{xc}(\mathbf{r}) \Pi_{I\mathbf{g}}(\mathbf{r})\ d\mathbf{r}, \quad (5.1)$$

where U is the unit cell and $\pi_{I\mathbf{g}}(\mathbf{r})$ is the normalized weight of an atomic center I located in cell $\mathbf{g}$. The sum of these weights over all atoms in the system is 1 at any point $\mathbf{r}$. Each of these weights contain pairwise contributions from all other atoms. The translational symmetry of the system allows one to transform the integral over the unit cell into an integral over all space

$$E_{xc} = \sum_{I\mathbf{g}} \int_{\mathbf{r}\in U} \epsilon_{xc}(\mathbf{r})\ \Pi_{I\mathbf{g}}(\mathbf{r})\ d\mathbf{r} = \sum_{I} \int_{all\ \mathbf{r}} \epsilon_{xc}(\mathbf{r})\ \Pi_{Io}(\mathbf{r})\ d\mathbf{r}, \quad (5.2)$$

where we emphasize that in the last term the integration is performed over all $\mathbf{r}$ but the sum is restricted to atoms in the central cell. Such integration is very similar to the integration over an atom in the center of a big but finite cluster of atoms. The Stratmann-Scuseria (SS) weights [59], although originally proposed for systems without periodicity, were designed to deal exactly with this type of situation. Therefore, we utilized them in our PBC program with minor modifications.

Of course, it is also possible to use other weight schemes in a PBC code, and some alternatives are discussed in Ref. [18].

Real space - reciprocal space transformations

In our DFT PBC implementation, we do as much work as possible in real space. Consequently, all matrices are stored in real space form, e.g., $A_{\mu\nu}^{\mathbf{0g}}$, and transformed into $\mathbf{k}$ space only when needed. In the iterative part of the code, we first construct the entire real space Fock matrix $F_{\mu\nu}^{\mathbf{0g}}$, transform it into several $\mathbf{k}$-space matrices, diagonalize them, obtain orbital coefficients and energies, and then construct the reciprocal space density matrices. These density matrices are integrated by numerical quadrature of equation (2.6) to yield the real space density matrix utilized in the following SCF cycle. At SCF convergence, we compute $\mathbf{k}$-space dependent energy weighted density matrices $W^{\mathbf{k}}$ using equation (2.10), and then transform them into $W_{\mu\nu}^{\mathbf{0g}}$ form. As a result, the gradient part of the code also deals with real space quantities only. Overall, $\mathbf{k}$-space integration adds just a few extra steps to the PBC calculation when compared to the molecular case. All the transformations between real and reciprocal spaces are computationally inexpensive and simple to implement.

An extremely important feature of our direct space Gaussian PBC code is that once the real space matrices are available, the major cost of any additional $\mathbf{k}$ point calculation is just the transformation into an orthonormal basis set plus the diagonalization. This is drastically different from the PW implementations where one computes the Fock matrix for each $\mathbf{k}$ point separately. Even though the cost of

such matrix formation may be considerably lower for PWs than for our real space Fock matrix, the relative low cost of additional **k** point calculations when using Gaussians permits denser **k** point meshes in reciprocal space even for systems of relatively large size. Furthermore, the derivatives of the band energies with respect to the **k** vector are also rather straightforward to compute [60]. Such information may be very useful in the Brillouin-zone integration for systems with complicated band structures [61].

Convergence in reciprocal space

The number of **k** points required to achieve convergence of the real space density matrices, and, consequently, of energy and forces, depends on the size of the unit cell and the band gap of the system. The general relation is that doubling a unit cell size in real space halves the corresponding lattice dimension in reciprocal space and therefore requires half as many **k** points in this dimension. Also, the smaller the band gap of the system, the larger the number of **k** points required to achieve similar accuracy. In the limit of zero band gap, the system becomes metallic and the orbital occupations might become **k**-point dependent. Such discontinuity requires more sophisticated and robust methods for reciprocal space integration than just the simple rectangular quadrature used in this work [62, 63, 64, 65, 66].

An example of reciprocal space integration convergence for a system with a fairly small HOMO-LUMO gap is shown in Table 2. These are LSDA/3-21G calculations for PPV, which at this level of theory has a band gap of 1.35 eV. As one can see, 32 **k** points are needed to converge the energy to 10^{-9} Hartrees. For a

smaller number of **k** points, the energy oscillates around the converged value. The polyacetylenes discussed in Chapter 7 usually have band gaps smaller than 1 eV, so those calculations required about 400 **k** points to converge the energy to 10^{-9} Hartree accuracy.

Table 2 Converged SCF energy (in Hartrees) as a function of the number of **k** points for PPV at the LSDA/3-21G level of theory.

No. **k**	Total energy
1	-305.046267410
2	-305.016437317
3	-305.003488239
4	-305.011151754
8	-305.010605979
16	-305.010590467
32	-305.010590432
64	-305.010590432

SCF convergence

As is the case of typical molecular calculations, SCF convergence problems may arise in calculations with PBC. In order to minimize the number of SCF cycles, we employed the direct inversion of the iterative subspace method (DIIS) developed by Pulay [67, 68]. DIIS requires formation of error matrices $R_n = (F_n P_{n-1} S - S P_{n-1} F_n)$ for each SCF cycle, where F_n, P_{n-1}, and S are the Fock, density and overlap matrices, respectively. The matrix R_n approaches zero as the calculation proceeds toward convergence. During the DIIS procedure, one evaluates inner products of the error matrices from different SCF cycles, $B_{ij} = R_i \cdot R_j$, and uses

these B_{ij} products to determine the DIIS mixing coefficients, C_i [68]. The sum of all C_i's is constrained to be 1, and the coefficients are chosen such that the norm of the vector $\sum C_i R_i$ is the smallest among all possible sets of C_i's. Then, the density obtained from the DIIS extrapolated Fock matrix $\tilde{F}_n = \sum C_i F_i$ is closer to convergence than the density obtained from the unextrapolated Fock matrix F_n.

A simple way to incorporate the DIIS procedure into a PBC code is to employ F, P and S matrices just for one point in reciprocal space, for example, the Γ ($\mathbf{k} = 0$) point. For the Γ point, the F, P, and S matrices are real, and all the DIIS steps are then the same as in the case of molecular calculations. The DIIS mixing coefficients are used to form the extrapolated real space Fock matrix $\tilde{F}^{\mathbf{0g}}_{\mu\nu}$. In summary, our strategy is to evaluate the DIIS mixing coefficients from Γ point matrices and use these coefficients to build Fock matrices in all $\mathbf{k}$ points through the extrapolation of the real space $\tilde{F}^{\mathbf{0g}}_{\mu\nu}$ matrix.

In Table 3, we present the SCF energy convergence patterns for LSDA/3-21G calculations of PPV with varying number of $\mathbf{k}$ points used in the reciprocal space integration. The DIIS procedure was always carried out for Γ matrices only. All calculations were performed at the same geometry (starting from the same converged LSDA/STO-3G density for the Γ point). The SCF procedure was considered converged when the RMS change in density matrix elements between successive cycles became smaller than 10^{-8}. One can see that all calculations presented in Table 3 converge in a similar manner and require roughly the same number of SCF cycles. These results demonstrate that the efficiency of the Γ point DIIS does not depend on the number of $\mathbf{k}$ points employed in the reciprocal space integration.

Table 3 SCF energy convergence for different number of **k** points used in reciprocal space integration for PPV at the LSDA/3-21G level. The Γ point DIIS extrapolation is used in all cases. The density is converged to a RMS deviation of 10^{-8}.

Cycle	Γ point	8 **k** points	32 **k** points
2	-0.224691937902	-0.189839234123	-0.189825525804
3	-0.001679363067	-0.001589740097	-0.001590495497
4	-0.010874794608	-0.010126205214	-0.010123523908
5	-0.000051023879	-0.000056762104	-0.000056803464
6	-0.000006455924	-0.000030254529	-0.000030301006
7	-0.000000136088	-0.000000075913	-0.000000075704
8	-0.000000019909	-0.000000027911	-0.000000027601
9	-0.000000000382	-0.000000000328	-0.000000000333
10	-0.000000000086	-0.000000000172	-0.000000000170
11	-0.000000000003	-0.000000000001	0.000000000000
12		-0.000000000003	-0.000000000003
E_t	-305.046267410044	-305.010605978591	-305.010590431688

We have also explored the use of error matrices for points other than Γ point. In such a case, the matrices are complex and yield complex mixing coefficients, which are used to form linear combinations of complex matrices $F^{\mathbf{k}}$ from previous SCF cycles. It turns out that while the density is far away from a stationary point, the imaginary part of the mixing coefficients remains small and the convergence rate is the same as in the previous case with Γ matrices. However, once the stationary point is close, the imaginary parts of the coefficients usually become relatively large and cause oscillations in energy and density. As a consequence, it was not possible to reliably achieve the required accuracy in a small number of SCF cycles with such an approach. After removing the imaginary parts of the DIIS mixing coefficients, the

DIIS procedure worked as in the case of the Γ point matrices described above. So, in our experience DIIS gives useful results only when the mixing coefficients are real. We have also tried to employ error matrices for several **k** points simultaneously with the total inner products formed as $B_{ij} = B_{ij}^{\mathbf{k}1} + ... + B_{ij}^{\mathbf{k}n}$. In this particular situation, the acceleration of the SCF convergence was slightly worse than with matrices for one **k** point only, and the SCF usually took one extra cycle. In summary, we did not find the additional computational effort of dealing with several **k** matrices useful for DIIS and settled on the Γ point scheme described above.

Numerical instability problems

In periodic calculations, large basis sets with diffuse functions may cause instabilities in the SCF procedure due to the limited accuracy of the Fock matrix construction and much more rarely due to the limited accuracy of the diagonalization routines [69]. Such problems can also be encountered in molecular cases if the contributions to the Fock matrix elements are approximated without proper precautions [70]. The usual prescription for restoring the stability of the SCF procedure for both types of problems is to project out the orbitals with small overlap eigenvalues from the basis set, which can be done during the orthonormalization. In order to transform GTOs to an orthonormal basis, one may employ symmetric orthogonalization and use the $S^{-1/2}$ matrix [71]. The latter is computed by diagonalizing S to obtain a matrix V such that $V^{\dagger}SV = s$, where s is a diagonal matrix containing the eigenvalues of S. The $S^{-1/2}$ matrix is then obtained as $S^{-1/2} = Vs^{-1/2}V^{\dagger}$.

Another way to orthonormalize the basis set is called canonical orthogonalization and uses the matrix $U = Vs^{-1/2}$. Columns of U contain the ith eigenvector of the overlap matrix divided by the square root of its eigenvalue s_i. In the case of instabilities, one can throw away columns of U corresponding to very small eigenvalues s_i [72]. As a consequence, during the transformation $U^\dagger FU$ of the Fock matrix F into an orthonormal basis, the offending orbitals are projected out, and the stability is restored. In general, it is desirable to check the $U^\dagger SU$ matrix and make sure that it is sufficiently close to unity because in certain cases where the S matrix has rather small eigenvalues, the diagonalization of S may produce inaccurate results due to numerical problems [70]. If such problems appear, one may resort to the more robust Gram-Schmidt orthogonalization technique [70].

We would like to point out that the SCF instabilities due to any of the two above mentioned causes are usually referred to in the literature as "linear dependency" problems. While the problems related to errors in the accuracy of the Hamiltonian matrix may happen for smallest overlap matrix eigenvalues as large as 10^{-2}-10^{-3}, true linear dependencies do not occur until the smallest overlap matrix eigenvalue becomes on the order of 10^{-6} to 10^{-7} (see discussion in Ref. [69]). Needless to say, practically all references in the literature to "linear dependencies" in calculations with PBC represent the first kind of instabilities, ones that arise from numerical inaccuracies on the Hamiltonian matrix formation rather than true linear dependencies in the basis set.

In our code, kinetic and electrostatic contributions to the Fock matrix are evaluated exactly (the latter via the FMM), while the DFT exchange-correlation quadra-

ture is carried out with high accuracy. Therefore, we might expect that the SCF instabilities would occur only for very small overlap matrix eigenvalues, somewhere in the 10^{-6}-10^{-7} range. Indeed, this is the behavior observed in all of our calculations. For example, in the case of *trans*-polyacetylene, we have successfully carried out calculations using a 6-311G(d,p) basis set (smallest overlap eigenvalues of $\sim 10^{-4}$) with no problems. Using the 6-311+G(d,p) basis set (smallest overlap eigenvalue of $\sim 10^{-7}$), we had to eliminate one orbital at some $\mathbf{k}$ points to make the SCF calculation stable. This shows that PBC calculations are not inherently more prone to linear dependencies than molecular calculations. The key issue seems to be the Hamiltonian matrix evaluation, which needs to be highly accurately done, especially in the infinite Coulomb sums. We achieve this goal by means of the FMM without resorting to any truncation.

Atomic gradients and stress tensor

As mentioned above, our analytic energy gradient code uses only the real space density matrix and the real space energy weighted density matrix. This makes the evaluation of forces in the PBC case somewhat similar to analogous computations for molecular systems, with few additions. We want to remind the reader that the stress tensor [Equation (2.19)] requires derivatives of the unit cell energy with respect to atoms in the neighboring cells, $dE/dx_{I\mathbf{g}}$. Therefore, we simply increase the size of the force array in the computer program and accumulate contributions for each $x_{I\mathbf{g}}$ separately. During the differentiation of the pair $\mu_{\mathbf{0}}(I)\nu_{\mathbf{g}}(J)$, we add the computed values to the elements $dE/dx_{I\mathbf{o}}$ and $dE/dx_{J\mathbf{g}}$, respectively. Then,

the total force dE/dx_I is obtained by adding up all $dE/dx_{I\mathbf{g}}$ terms. The short ranged part of the solid cell stress is computed from these $dE/dx_{I\mathbf{g}}$ parts, and the long ranged (electrostatic) part is treated by the FMM, as described in Ref. [4]. We note that the extra work required to compute the long ranged part of the solid cell force is relatively small because these computations can be efficiently incorporated into the atomic force code. Having a complete set of energy derivatives with respect to geometrical parameters, it is possible to carry out efficiently full geometry optimizations as described in Chapter 6.

Computational scaling and diagonalization alternatives

In energy calculations, each SCF cycle requires two major steps, the Fock matrix construction and the density matrix update. Recent research has shown that one can exploit the locality of the interactions in a physical system and build the Fock matrix in $\mathcal{O}(N)$ CPU time operations [13, 73]. We fully incorporated these recent developments in our PBC code. There are three major contributions to the Fock matrix formation: the kinetic energy term, the electrostatic term, and the exchange-correlation contribution. In large systems, the kinetic energy matrix is sparse, computed only once, and therefore easy to deal with. The electrostatic part of our code uses the periodic FMM, so its scaling is very close to linear [2]. Our periodic exchange-correlation quadrature is a straightforward extension of the one used in molecular calculations whose linear scaling has also been demonstrated [59, 13]. Furthermore, the analytic gradient code resembles the Fock matrix formation, therefore the force calculation also has $\mathcal{O}(N)$ computational cost.

In the applications carried out in this work, the density matrix update is done in the conventional way by diagonalizing the Kohn-Sham Hamiltonian matrix and constructing the density matrix from its eigenvectors. Although this procedure scales as $\mathcal{O}(N^3)$, it has such a small scaling prefactor that the diagonalization cost for systems with up to a few thousand Gaussian basis functions is rather small compared to other steps in the PBC code. The DIIS procedure contains matrix multiplications with regular $\mathcal{O}(N^2)$ matrices, and this step also scales as $\mathcal{O}(N^3)$. Millam and Scuseria [74] were first to demonstrate that in DFT calculations with Gaussians, one can replace the diagonalization step by an $\mathcal{O}(N)$ alternative such as conjugate gradient density matrix search (CGDMS). This method works very well for systems with large band gaps as demonstrated previously [13, 73, 74]. For very large systems, the DIIS procedure uses sparse matrices, and its cost also becomes close to linear. For Γ point calculations, we can use in our PBC code all the sparse matrix multiplication routines developed previously [74, 75]. We have neither implemented nor tested, however, these methods for the complex matrices required for other $\mathbf{k}$ points.

In order to demonstrate the actual scaling properties of our DFT PBC program, we have carried out a series of calculations for PPV with differing numbers of monomers in the unit cell. The results are presented in Table 4. The calculations were carried out at the LSDA/3-21G level of theory and the number of $\mathbf{k}$ points was chosen according to the guidelines outlined above. The SCF took 10 cycles to converge the density to a RMS deviation of 10^{-8}. The total energies for these calculations are given in Table 5. The case with 16 monomers in the unit cell

needs only two **k** points for the converged energy. We also tested one **k** point (Γ). As expected, the energy for the calculation with two **k** points is closer to the converged value than for the Γ point calculation (two last entries in Table 5). Overall, we would like to draw the reader's attention to the remarkable agreement of energies per unit cell from different calculations: they differ only in the 10th decimal!

Table 4 IBM Power3 CPU times (sec) for various steps of the DFT calculation for PPV LSDA/3-21G.

$(C_8H_6)_x$, x	1	2	4	8	16
No. Atoms	14	28	56	112	224
No. Basis	84	168	336	672	1344
No. **k** points	32	16	8	4	2
No. FMM levels	3	4	5	6	7
Timings:					
Form S^{-1}*	0.6	2.0	7	28	115
FMM, FF	1.3	2.5	5	10	20
FMM, NF	6.3	12.1	25	48	98
XC quad	13.0	26.4	54	109	223
DIIS	0.03	0.13	0.9	7.9	73
Γ diag	0.04	0.21	1.3	9.9	78
Diag[a]	0.91	3.2	12.3	52	198
Total SCF	197	416	935	2342	7285
Forces					
FMM, FF	2.7	5.4	11	21	43
FMM, NF	28.4	57.4	115	232	446
XC quad	24.1	49.4	106	243	621
Total Force	55	113	233	500	1123

[a] Timings reported include all **k** points.

Table 5 Total energies (Hartree) for the PPV LSDA/3-21G calculations shown in Table 4.

Unit cell	Total Energy	Energy per PPV unit
$(C_8H_6)_1$	-305.01059043175	-305.01059043175
$(C_8H_6)_2$	-610.02118086376	-305.01059043188
$(C_8H_6)_4$	-1220.04236172718	-305.01059043180
$(C_8H_6)_8$	-2440.08472345487	-305.01059043186
$(C_8H_6)_{16}$ {No. **k**=2}	-4880.16944691366	-305.01059043210
$(C_8H_6)_{16}$ {No. **k**=1}	-4880.16944635247	-305.01059039703

Let us discuss in more detail the timings shown in Table 4. In the SCF part of the calculations, the CPU time required for the evaluation of electrostatic and exchange-correlation terms scales linearly for all practical purposes. On the other hand, the complex diagonalizations and the DIIS procedure scale as $\mathcal{O}(N^3)$. The relative cost of these $\mathcal{O}(N^3)$ steps is such that for the largest system in Table 4 (224 atoms, 1344 basis functions), their total CPU time is roughly similar to the CPU time required for the Fock matrix formation. Comparing systems with different dimensionality and band gaps, one can argue that the Fock matrix formation step in 1D systems is fastest, making the CPU time consumed by the $\mathcal{O}(N^3)$ steps look relatively large. In metallic systems, that require a large number of **k** points regardless of dimensionality [18], the CPU time consumption by diagonalization will be substantial.

In order to achieve a small absolute cost for all matrix operations, we have employed optimized linear algebra routines, such as DGEMM, ZGEMM, DSPEV, and ZHPEV, from the BLAS and LAPACK libraries. On the IBM RS 6000 family

of computers, these routines are included in their machine optimized ESSL library. Similar preoptimized packages are available for other computer architectures. Such libraries ensure that the CPU intensive operations are carried out with the greatest possible efficiency.

Chapter 6

A redundant internal coordinate algorithm for optimization of periodic systems

Introduction

Geometry optimization has become an almost mandatory step in computational studies of individual molecules [35]. Much attention has been devoted to making optimizations reliable and efficient. Newton-Raphson methods are perhaps the most rapidly convergent, but they require the computation of second derivatives or Hessians at each step in the optimization [76]. However, for most levels of theory, second derivatives are significantly more difficult to calculate than gradients (first derivatives), and their computation scales poorly with the size of the system. Consequently, quasi-Newton techniques have emerged as the method of choice since they use only gradients [76]. These algorithms start with an estimate of the Hessian and improve it during the course of the optimization by using a variety of possible updating methods.

It has also become clear that internal coordinates are more efficient than Cartesians for optimizing molecular systems [35]. Redundant internal coordinates are especially useful for polycyclic molecules and other highly connected systems [77, 78, 79, 80, 81, 82]. Such coordinate systems are constructed from the stretches, bends

and torsions involving all of the bonded atoms in a molecule. Thus, the coordinate system automatically reflects the chemical connectivity and takes into account the inherent curvilinear nature of internal motions of molecules. A well chosen internal coordinate system has much less coupling between coordinates than a Cartesian coordinate system, allowing the optimized structure to be found in significantly fewer steps. Furthermore, a diagonal estimate of the Hessian is often sufficient for rapid and reliable convergence of the optimization [83, 84, 85]. Traditional methods for transforming between Cartesian and internal coordinates scale as $\mathcal{O}(N^3)$ with system size. However, with recent developments, this transformation can be carried out with $\mathcal{O}(N^2)$ effort and even $\mathcal{O}(N)$ when sparse matrix techniques are used [86, 87, 88, 89, 90, 91, 92]. When combined with $\mathcal{O}(N)$ methods [13] for electronic structure calculations, this permits systems as large as plasminogen (1226 atoms) to be optimized readily on workstations [93, 94].

To the best of our knowlegde, optimizations of periodic systems carried out to date utilized either fractional or Cartesian coordinate systems. Structural studies with electronic structure methods have been mostly performed for highly symmetric systems with few independent degrees of freedom (e.g. 1-10), so the efficiency of the optimizer has not been a big issue. Furthermore, in most of these cases analytic energy gradients were not available, thus limiting the size of the system being studied. On the other hand, in periodic molecular mechanics calculations energy, forces and even exact Hessians are relatively inexpensive computationally, so high optimization efficiency can be achieved by combining the readily available exact Hessian with a simple fractional or Cartesian coordinate based method [95]. When

optimizing the unit cell parameters, orientation related instabilities can occur if one does not take account of the three rotational degrees of freedom present in the nine Cartesian components of the lattice vectors. To overcome such problems the variable-cell-shape (VCS) algorithm [96] employs six dot products of the lattice vectors instead of their individual Cartesian components. In another approach [95], minimization of the unit cell energy at constant pressure is carried out via the strain matrix containing six unique components, also removing the rotational degrees of freedom from the optimization.

In the present work, we describe an alternative and potentially superior approach that uses redundant internal coordinates to implicitly optimize the lattice vectors of a periodic system. Because periodic structures have connectivities that are similar to cyclic and cage-like molecules, one can anticipate that redundant internal coordinates will be the best choice in this case as well. Our tests demonstrate that when redundant internal coordinates are used, the number of steps required to optimize a periodic system is similar to that required for a comparable molecular system. This number is also small in absolute terms, thus confirming the high efficiency of our redundant internal coordinate algorithm that uses only a simple diagonal guess for the Hessian matrix.

Method

One of our main goals in using redundant internal coordinates for periodic systems is to employ chemically meaningful coordinates such as bond lengths, valence angles and dihedral angles to represent not only the positions of atoms within a unit cell,

but also the relative orientation and spacing between atoms in adjacent unit cells. Thus, changes in the internal coordinates can also adjust the lattice vectors. For example, consider a planar, one dimensional chain, - [P_{-1} - Q_{-1}] -[P_0 - Q_0] -[P_1 - Q_1] -. If one optimizes the bond length within a unit cell P_0 - Q_0, the bond length spanning adjacent cells Q_0 - P_1, and the angles P_0 - Q_0 - P_1 and Q_0 - P_1 - Q_1, then the translation vector T will be optimized implicitly as a combination of these bonds and angles. Force constants or Hessian matrix elements for these internal coordinates should be comparable to those for isolated molecules, and similar initial estimates can be used [83, 84, 85]. These force constants implicitly provide estimates for the Hessian matrix elements for the lattice vectors and their coupling with the other coordinates, which might be rather difficult to estimate otherwise.

An optimization algorithm employing internal coordinates requires a transformation matrix between Cartesian displacements and internal coordinate displacements. In molecular calculations, this is the well-known Wilson B matrix [97], originally used in vibrational analysis. The B matrix for periodic systems is here defined similarly

$$\delta\mathbf{q} = \mathbf{B}\ \delta\mathbf{r} \tag{6.1}$$

where $\mathbf{q} = (q_1, q_2, ...q_m)^T$, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_n, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3)^T$ and $B^a_{i,j} = \partial q_i/\partial r^a_j$. The $\mathbf{r}_i = (r^x_i, r^y_i, r^z_i)$ are the absolute positions of the atoms within the central cell, and $\mathbf{t}_i = (t^x_i, t^y_i, t^z_i)$ are the lattice vectors. Accordingly, the derivatives of the energy with respect to the lattice vectors (stress tensor) should be calculated with the

absolute atomic positions within the cell fixed; we have referred to this quantity as the solid cell stress in Chapter 2.

In the periodic case, there are two types of internal coordinates - intracell and intercell. Intracell coordinates allow one to adjust the relative positions of atoms within the unit cell. Therefore, the periodic B matrix elements for intracell coordinates are computed exactly in the same way as in a comparable molecule. Intercell coordinates span two or more cells and depend on atomic coordinates in cells other than central cell (cell 0). Yet, periodic systems are completely defined by the coordinates in cell 0 and the translational vectors. Coordinates of atoms in other cells can be readily obtained by an appropriate translation. Therefore, intercell coordinates effectively depend on atomic positions within the central cell and translational vectors. Optimization of such coordinates results in the adjustment of atomic positions and translational vectors, without considering the latter explicitly. A rather straightforward way to obtain the periodic B matrix for intercell coordinates is then the following. First, by treating all atoms within a given internal coordinate as independent, one computes the molecular B matrix components for each atom using the usual formulae [97]. Next, by applying the chain rule, these molecular B matrix elements are transformed into periodic ones. Several examples presented below should clarify this procedure.

In the above example, the system is defined via the vector $\mathbf{r} = (\mathbf{P_0}, \mathbf{Q_0}, \mathbf{t})$, where $\mathbf{P_0} = (P_0^x, P_0^y, P_0^z)$, $\mathbf{Q_0} = (Q_0^x, Q_0^y, Q_0^z)$, and $\mathbf{t} = (t^x, t^y, t^z)$. In the following, we represent the molecular-like B matrix components as $\hat{B}$, which are the partial derivatives of internal coordinates with respect to the explicit atomic positions.

The periodic B matrix elements required for our optimization method are total derivatives of a given internal coordinate with respect to an atomic position in cell 0, and the lattice vectors. To simplify notation, we denote the total derivative of a given internal coordinate q_i with respect to the Cartesian component a of atomic position $\mathbf{P_0}$ as

$$B^a_{i,P} \equiv B^a_{i,P_0} = \frac{dq_i}{dP^a_0} \tag{6.2}$$

For intracell coordinates, such as P_0 - Q_0, the periodic B matrix elements are identical to the molecular ones ($\hat{B}$) because all the atoms are located within cell 0 and therefore the corresponding total and partial derivatives are the same. Denoting the P_0 - Q_0 bond length as $q_i = q(\mathbf{P_0}, \mathbf{Q_0})$, one obtains

$$B^a_{i,P} = \frac{dq_i}{dP^a_0} = \frac{\partial q_i}{\partial P^a_0} = \hat{B}^a_{i,P_0}; \quad B^a_{i,Q} = \frac{dq_i}{dQ^a_0} = \frac{\partial q_i}{\partial Q^a_0} = \hat{B}^a_{i,Q_0}. \tag{6.3}$$

For the intercell case, e.g. the Q_0 - P_1 bond, some additional manipulations are necessary to obtain the total derivatives with respect to atoms within the central cell and the lattice vectors. Let $q_j = q(\mathbf{Q_0}, \mathbf{P_1})$ be the intercell bond Q_0 - P_1. First, the molecular B matrix elements for Q^a_0 and P^a_1 are calculated as

$$\hat{B}^a_{j,Q_0} = \frac{\partial q_j}{\partial Q^a_0}; \quad \hat{B}^a_{j,P_1} = \frac{\partial q_j}{\partial P^a_1}. \tag{6.4}$$

Using $P^a_1 = P^a_0 + t^a$, the required total derivatives with respect to P^a_0 and t^a are then obtained as

$$B^a_{j,P} = \frac{dq_j}{dP^a_0} = \frac{\partial q_j}{\partial P^a_1} \cdot \frac{\partial P^a_1}{\partial P^a_0} = \frac{\partial q_j}{\partial P^a_1} = \hat{B}^a_{j,P_1} \tag{6.5}$$

$$B^a_{j,t} = \frac{dq_j}{dt^a} = \frac{\partial q_j}{\partial P^a_1} \cdot \frac{\partial P^a_1}{\partial t^a} = \frac{\partial q_j}{\partial P^a_1} = \hat{B}^a_{j,P_1} \tag{6.6}$$

where we have used that

$$\frac{\partial P_1^a}{\partial P_0^a} = 1; \quad \frac{\partial P_1^a}{\partial t^a} = 1. \tag{6.7}$$

For the angle Q_0 - P_1 - Q_1, denoted as $q_k = q(\mathbf{Q_0}, \mathbf{P_1}, \mathbf{Q_1})$, the periodic B matrix elements are

$$B^a_{k,Q} = \frac{dq_k}{dQ_0^a} = \frac{\partial q_k}{\partial Q_0^a} + \frac{\partial q_k}{\partial Q_1^a} \cdot \frac{\partial Q_1^a}{\partial Q_0^a} = \hat{B}^a_{k,Q_0} + \hat{B}^a_{k,Q_1} \tag{6.8}$$

$$B^a_{k,t} = \frac{dq_k}{dt^a} = \frac{\partial q_k}{\partial P_1^a} \cdot \frac{\partial P_1^a}{\partial t^a} + \frac{\partial q_k}{\partial Q_1^a} \cdot \frac{\partial Q_1^a}{\partial t^a} = \hat{B}^a_{k,P_1} + \hat{B}^a_{k,Q_1} \tag{6.9}$$

The last example illustrates that for intercell internal coordinates depending on more than one atom in an adjacent cell or the same atom appearing in more than one cell, the corresponding periodic B matrix elements have contributions from more than one molecular $\hat{B}$ matrix element. We also note that because of simple expressions describing atomic coordinates in cells other than cell 0 through coordinates in cell 0 and translational vectors, any periodic B matrix element is at most a sum of molecular contributions $\hat{B}$, such as in q_k above.

It is also instructive to consider the bond P_0 - P_1. Let $q_l = q(\mathbf{P_0}, \mathbf{P_1})$ and recall that the molecular $\hat{B}$ matrix elements are translationally invariant (i.e., $\hat{B}^a_{l,P_0} + \hat{B}^a_{l,P_1} = 0$ in the present case), then

$$B^a_{l,P} = \frac{dq_l}{dP_0^a} = \frac{\partial q_l}{\partial P_0^a} + \frac{\partial q_l}{\partial P_1^a} \cdot \frac{\partial P_1^a}{\partial P_0^a} = \hat{B}^a_{l,P_0} + \hat{B}^a_{l,P_1} = 0 \tag{6.10}$$

$$B^a_{l,t} = \frac{dq_l}{dt^a} = \frac{\partial q_l}{\partial P_1^a} \cdot \frac{\partial P_1^a}{\partial t^a} = \hat{B}^a_{l,P_1} \tag{6.11}$$

The only surviving B component is for the lattice vector, as it should be. On the other hand, the bond $q_m = q(\mathbf{P_1}, \mathbf{Q_1})$ will contribute to the following periodic B matrix elements

$$B^a_{m,P} = \frac{dq_m}{dP^a_0} = \frac{\partial q_m}{\partial P^a_1} \cdot \frac{\partial P^a_1}{\partial P^a_0} = \hat{B}^a_{j,P_1} \tag{6.12}$$

$$B^a_{m,Q} = \frac{dq_m}{dQ^a_0} = \frac{\partial q_m}{\partial Q^a_1} \cdot \frac{\partial Q^a_1}{\partial Q^a_0} = \hat{B}^a_{j,Q_1} \tag{6.13}$$

$$B^a_{m,t} = \frac{dq_l}{dt^a} = \frac{\partial q_m}{\partial P^a_1} \cdot \frac{\partial P^a_1}{\partial t^a} + \frac{\partial q_m}{\partial Q^a_1} \cdot \frac{\partial Q^a_1}{\partial t^a} = \hat{B}^a_{m,P_1} + \hat{B}^a_{m,Q_1} = 0 \tag{6.14}$$

where the last result is again due to the translational invariance of the molecular B matrix elements. As expected, the bond $q(\mathbf{P_1}, \mathbf{Q_1})$ does not depend on the translational vector and yields the same B matrix contributions as the bond $q(\mathbf{P_0}, \mathbf{Q_0})$. Consequently, an arbitrary translation of all atoms in a given internal coordinate will lead to a coordinate identical to the original one.

In a general three dimensional periodic system, intracell bonds will be of the type P(0,0,0)-Q(0,0,0), while the intercell bonds are of the type P(c_1,c_2,c_3) - Q(d_1,d_2,d_3), where all of the c_a and d_b indices are either 0 or 1. As a consequence, a given coordinate will contribute to lattice vectors $\mathbf{t}_i$ when c_i or d_i are 1. For example, the bond P(0,0,0) - Q(1,1,1) will have non-zero periodic B matrix elements for atoms P and Q, and all three lattice vectors. Similar considerations apply to angles, and dihedrals. To generate a full set of intra- and intercell internal coordinates, one can replicate the central cell and obtain a 2x2x2 cluster of cells with cell indices 0 and 1. Then, all of the possible bonds, angles, and dihedrals are generated within this

cluster and duplicate ones are thrown away by applying translational invariance. For example, bond P(0,1,1) - Q(1,0,1) is the same as P(0,1,0) - Q(1,0,0) and is eliminated. A similar approach is used for valence angles and dihedrals.

With this method, optimization of periodic systems requires only minor modifications to an algorithm for redundant internal coordinate optimization of molecules. Details of our molecular optimization algorithm have been described previously [80, 86, 87]. Since all of the machinery commonly used for molecular optimizations is available, it can be used in novel ways for periodic cases. For example, if the lattice vector $\mathbf{t}_1$ is to be held constant, one simply freezes the distance P(0,0,0) - P(1,0,0). Similarly, to freeze the angle between lattice vectors $\mathbf{t}_2$ and $\mathbf{t}_3$, one can constrain the angle P(0,1,0) - P(0,0,0) - P(0,0,1); to keep the P - Q bond at a given angle to the $\mathbf{t}_1$ lattice vector, one would constrain P(0,0,0) - Q(0,0,0) - Q(1,0,0). Therefore, in our method one can use the existing capabilities for applying constraints if it is desired to freeze some or all of the lattice parameters.

Examples

All calculations were carried out with the *Gaussian* [26] suite of programs extended for calculations on periodic systems as described in this work. The computational method used here is the PBE DFT functional [98] together with a relatively small 3-21G basis set. The optimization thresholds were *Max_Force* = 0.00045, *RMS_Force* = 0.0003, *Max_Displacement* = 0.0018, *RMS_Displacement* = 0.0012 in the usual atomic units. The optimization was stopped when all these conditions were satis-

fied. This particular set of values corresponds to the default convergence criteria in the *Gaussian* package [26].

Table 6 Comparison of optimizations of periodic and molecular systems. E denotes energy (in a.u.), $\sqrt{\sum F^2}$ - RMS_Force (in a.u.), $\sqrt{\sum \Delta x^2}$ - $RMS\_Displacement$ (in a.u.).

Step	PPV polymer - $[C_8H_6]_\infty$			Styrene molecule - C_8H_8		
	$E(\Delta E)$	$\sqrt{\sum F^2}$	$\sqrt{\sum \Delta x^2}$	$E(\Delta E)$	$\sqrt{\sum F^2}$	$\sqrt{\sum \Delta x^2}$
1	-306.3538719	0.01074	0.16948	-307.5260006	0.00808	0.19220
2	(-0.0121769)	0.00446	0.05579	(-0.0066980)	0.00298	0.07283
3	(-0.0025239)	0.00188	0.00421	(-0.0018739)	0.00132	0.00512
4	(-0.0000945)	0.00062	0.00086	(-0.0000860)	0.00034	0.00091
5	(-0.0000095)	0.00014	0.00015	(-0.0000065)	0.00007	0.00028
	-306.3686767			-307.5346650		

First we analyze the optimization of a one dimensional polymer, poly(*p*-phenylenevinylene) $[C_8H_6]_\infty$ (PPV) (Figure 7a). The initial geometry was chosen to be: C-H bond = 1.09 Å, C-C aromatic bond = 1.39 Å, C-C double bond = 1.36 Å, conjugated C-C single bond = 1.44 Å; all valence angles = 120^o. All atoms lay within a plane, so the dihedrals are either 0^o or 180^o. The **k** space integration employed 16 points. For comparison we also optimized a molecule structurally similar to the unit cell of PPV - styrene C_8H_8 (Figure 7b). Its initial geometrical parameters (bonds and angles) were set to values identical to those in the PPV case. A simple valence force field was employed for the initial guess of the force constants, which is the default for optimizations in *Gaussian*. The optimization convergence pattern for both structures is shown in Table 6. It is evident that for these structures

the two procedures perform similarly, confirming that optimization of the periodic system in internal coordinates is just as efficient as in the non-periodic case. The molecular framework in both systems is relatively rigid, and a minimum is found in only 5 cycles. The PBC coordinate selection algorithm defined 15 bonds, 24 valence angles and 36 dihedrals, totalling 75 coordinates, compared to 39 non-redundant degrees of freedom.

Table 7 Convergence of the PPV polymer optimization. E denotes energy (in a.u.), T - translational vector (in Å), dE/dT - force (in a.u.).

Step	$[(C_8H_6)_1]_\infty$			$[(C_8H_6)_2]_\infty$		
	$E(\Delta E)$	$T(\Delta T)$	dE/dT	$\frac{1}{2}\cdot E(\Delta E)$	$\frac{1}{2}\cdot T(\Delta T)$	dE/dT
1	-306.3538719	6.4485	0.03497	-306.3538719	6.4485	0.03497
2	(-0.0121769)	(0.3871)	-0.01402	(-0.0132568)	(0.3497)	-0.01123
3	(-0.0025239)	(-0.1191)	-0.00210	(-0.0014688)	(-0.0811)	-0.00201
4	(-0.0000945)	(-0.0014)	-0.00079	(-0.0000720)	(-0.0029)	-0.00062
5	(-0.0000095)	(-0.0007)	-0.00014	(-0.0000073)	(0.0001)	-0.00013
	-306.3686767	6.7142		-306.3686769	6.7143	

It is instructive to compare the optimization of PPV using one and two (C_8H_6) polymer units per cell. Table 7 lists energies at each step, translational vectors, and the derivative of the energy with respect to this vector. For convenience, the energies and translational vector for the larger unit cell are scaled by 1/2. While for the doubled unit cell the translational vector is twice that for a single unit cell, the derivative dE/dT is the same in both cases. This is due to the fact that for our definition of dE/dT only one bond is affected by dT [5], and it is the same bond in

both cases (the bond crossed by the cell box in Fig 7a). Overall, both optimizations proceed similarly; the small differences are probably caused by slightly different force constant matrices with the redundant degrees of freedom projected out.

We also optimized urea in its $P\bar{4}2_1m$ crystal phase at zero pressure. This is a highly symmetric system, with each urea molecule NH_2CONH_2 occupying a site of C_{2v} symmetry. Currently our program is capable of utilizing only limited symmetry - the point group of the unit cell. Therefore in our calculations we use a unit cell with 4 urea molecules (32 atoms), and the point group of the unit cell was constrained to be S_4 (Fig 8). During the optimization, atomic displacements are symmetrized via the operations of the unit cell point group. We have also frozen all dihedral angles to further reduce the actual number of degrees of freedom being optimized. The purpose of this procedure is to combat numerical noise present in the forces and to ensure symmetry of the final structure. We used a $2 \times 2 \times 4$ mesh of $\mathbf{k}$ points for the reciprocal space integration. The atoms in the unit cell have 99 degrees of freedom; however, the number of independent degrees of freedom is actually much smaller. The optimization employed 204 redundant internal coordinates, including 36 bonds, 24 hydrogen bonds, 48 angles, 16 linear angles and 80 dihedrals. The optimization convergence pattern starting from the experimental geometry found in Ref. [99] (also used in Refs. [100, 101]) is shown in Table 8. In the early stages of the optimization the lattice vectors and energy change substantially indicating that the "strong" intramolecular coordinates are being brought to their optimum values. In later steps the energy changes are significantly smaller, with a slower decrease in the RMS force. This is due to the fact that molecules in solid urea are

bound by weak hydrogen bonds with relatively shallow minima, and finding a local minimum is more demanding than optimizing covalent distances and angles. Such behavior is not uncommon in weakly bonded systems.

Table 8 Convergence of the urea optimization. The three translational vectors are [T_1, T_2, T_2]. E denotes energy (in a.u.), $\sqrt{\sum F^2}$ - RMS_Force (in a.u.), $\sqrt{\sum \Delta x^2}$ - $RMS\_Displacement$ (in a.u.).

Step	$E(\Delta E)$	$\sqrt{\sum F^2}$	$\sqrt{\sum \Delta x^2}$	T_1	T_2
1	-895.3003190	0.01504	0.02673	4.684	7.870
2	(-0.0162205)	0.00583	0.01400	4.710	7.902
3	(-0.0039320)	0.00336	0.04535	4.673	7.863
4	(-0.0050651)	0.00194	0.02123	4.587	7.736
5	(-0.0005542)	0.00178	0.00459	4.558	7.679
6	(-0.0002201)	0.00089	0.00796	4.563	7.670
7	(0.0000039)	0.00048	0.00481	4.565	7.646
8	(-0.0000332)	0.00051	0.00247	4.564	7.660
9	(-0.0000651)	0.00019	0.00098	4.572	7.661
10	(-0.0000056)	0.00014	0.00354	4.572	7.659
11	(-0.0000058)	0.00008	0.00207	4.575	7.653
12	(-0.0000077)	0.00006	0.00050	4.574	7.659
	-895.3264243				

The optimization algorithm described in this work was also recently used to optimize fluorinated carbon nanotubes [8], and polyglycines in various conformations, some of which were helices bound by weak hydrogen bonds [9, 10]. Overall, a general "black box" redundant internal coordinate optimization algorithm such as the one developed here is a very useful tool in structural studies of periodic systems.

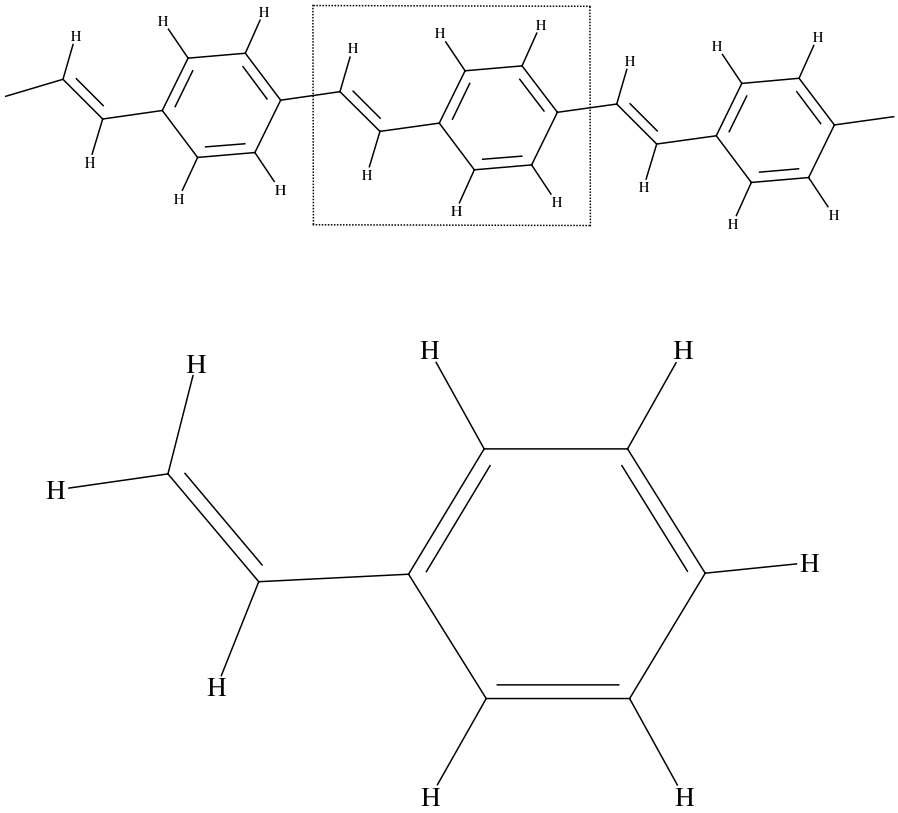

Figure 7 The structures of (a) poly(*p*-phenylenevinylene); (b) styrene.

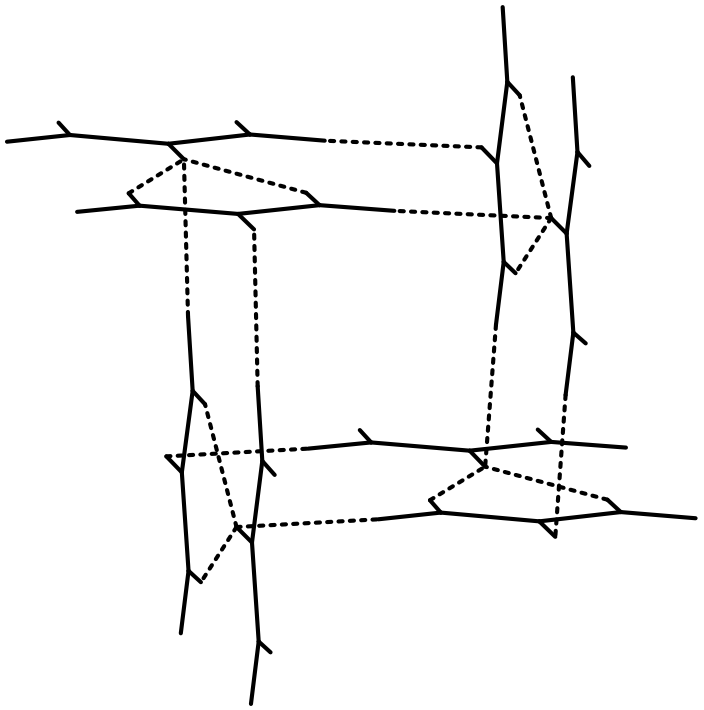

Figure 8 A fragment of the urea crystal. The four molecules shown in front define the simulation cell used in this optimization.

Chapter 7

Benchmark calculations

In order to demonstrate the capabilities of our DFT PBC code with Gaussian basis sets, we present in this chapter benchmark studies of several 1D periodic systems. Here, we focus more on the chemically relevant data as opposed to previous chapters where our primary concerns were methodological and computational details. Our first benchmark case is polyacetylene (PA), a system where pure DFT methods remarkably fail to quantitatively reproduce the experimental C-C bond length alternation [28]. The second benchmark system is PPV. Lastly, we have carried out optimizations for a series of carbon as well as boron-nitride nanotubes.

From the wide variety of currently available DFT functionals, we have chosen for benchmark purposes the local spin density approximation (LSDA) [102], the generalized gradient approximation functional, PBE [98], and VSXC [103], the kinetic-energy dependent functional recently developed in our research group. We have combined these functionals with what is considered in calculations of extended systems a medium quality 3-21G basis set and also a relatively high quality 6-31G(d) basis set. The geometries of all periodic systems have been optimized in redundant internal coordinates, as described in Chapter 6. Optimizations were stopped when the RMS force became smaller than 10^{-5} a.u. This criterion corresponds to the *Gaussian* keyword *opt* = *tight*. The PPV vibrational frequencies were computed

by numerical differentiation of analytic forces. The IR intensities are non-zero only for vibrations that change the dipole moment of the cell in the directions *perpendicular* to the periodicity axis. The dipole derivatives with respect to the atomic positions, required for IR intensities, were also computed numerically.

Polyacetylene - PA

Trans-PA (Figure 9a) is a prototypical system, probably one of the most studied polymers because of its small size and extreme sensitivity of the predicted bond length alternation (BLA) between single and double bonds with respect to the theoretical method used. The latter has motivated many researchers to apply various available methods to this quite difficult problem [23, 28, 38, 41, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115].

Experimentally, Yannoni and Clarke have determined that the two CC bonds in *trans*-PA are 1.36 and 1.44 Å [116], while Kahlert *et al.* [117] reported values of 1.36 Å and 1.45 Å. Despite the good agreement between these studies, these results cannot be relied on with total certainty because of the low quality of the PA samples, as pointed out in Ref. [118]. The literature on PA is vast, and we do not attempt to review it here; we nevertheless briefly mention a few recent representative studies. Using a double zeta (DZ) basis set, Suhai [110] reported BLAs of 0.107, 0.083, and 0.084 Å at the HF, MP2, and MP4 levels of theory, respectively. His DFT BLAs are 0.016 and 0.012 Å with the LSDA and BLYP functionals, respectively. Fogarasi *et al.* [109] obtained C-C bond-lengths of 1.325 and 1.462 Å, at the HF/6-31G(d) level of theory. Hirata *et al.* [111] computed

1.373 and 1.423 Å at the MP2/6-31G(d) level. At the B3LYP/6-31G(d) level of theory, Hirata *et al.* [113] reported 1.369 and 1.426 Å.

From these results, it is evident that DFT methods, such as LSDA and GGAs, fail to predict the BLA of *trans*-PA, yielding results in the 0.01-0.03 Å range [28]. These results are too small compared to those calculated with the MP2 and B3LYP methods [111, 23] and compared to experiment. Our results, presented in Table 9, follow the same trend. The VSXC functional, which typically mimics B3LYP quality results [119, 120, 121], increases the BLA compared to PBE, but not significantly.

Table 9 Structural parameters for polyacetylene (Å and degrees), computed with different DFT functionals.

	LSDA 3-21G	LSDA 6-31G(d)	PBE 3-21G	PBE 6-31G(d)	VSXC 3-21G	VSXC 6-31G(d)
trans-$(C_2H_2)_x$						
Gap (eV)	0.11	0.07	0.15	0.11	0.20	0.16
T_l	2.455	2.455	2.481	2.481	2.464	2.471
$R_{C=C}$	1.383	1.384	1.395	1.395	1.387	1.391
R_{C-C}	1.394	1.392	1.411	1.408	1.408	1.407
R_{C-H}	1.101	1.102	1.099	1.100	1.093	1.097
A_{CCC}	124.3	124.4	124.2	124.5	123.7	124.0
cis-$(C_2H_2)_x$						
Gap (eV)	0.76	0.72	0.75	0.71	0.90	0.85
T_l	2.199	2.205	2.233	2.242	2.205	2.219
$R_{C=C}$	1.373	1.375	1.386	1.387	1.379	1.382
R_{C-C}	1.413	1.412	1.430	1.427	1.429	1.429
R_{C-H}	1.098	1.098	1.096	1.097	1.087	1.091
A_{CCC}	125.8	126.0	126.3	126.8	125.3	125.8

The band gap of *trans*-PA is in the 0.07-0.20 eV range, which is small. Other authors have pointed out the connection between BLA and the band gap [107]. DFT

methods underestimate the band gap at the experimental geometry. Similarly, small BLAs are associated with very small band gaps. In order to obtain well converged energies and geometries for PA, we have employed 400 **k** points in the reciprocal space integration. If the number of **k** points is reduced substantially, the predicted alternation is significantly larger. For example, in a calculation with 21 **k** points, the LSDA/6-31G(d) BLA is 0.020 Å; employing just 11 **k** points leads to a BLA value of 0.033 Å. These BLAs are much larger than the converged value of 0.008 Å at the same level of theory. These results seem to rationalize the very large LDA BLA of 0.061 Å reported by Springborg [115], that was obtained in a calculation with 11 **k** points. Clearly, the latter result [115] is far from converged. According to our estimates, in order to obtain a converged BLA value, one ought to use about 200-400 **k** points, which is in accord with the conclusions of Ref. [113].

We have also optimized the geometry of two isomers of *trans*-PA, denoted *cis*-PA and *meta*-PA (Figure 9b,c). The predicted BLA for *cis*-PA is around 0.04-0.05 Å, significantly larger than that of *trans*-PA. Hirata *et al* [113] B3LYP/3-21G C-C bond lengths for *cis*-PA are 1.366 and 1.438 Å, and B3LYP/6-31G(d) values are 1.369 and 1.435 Å (0.066 Å BLA). In the case of *cis*-PA, the BLA also seems to be underestimated by the pure DFT methods. We would also like to point out the agreement between our LSDA geometries, both for *trans*-PA and *cis*-PA, with those reported in Ref. [113]. The bond lengths predicted with 3-21G and 6-31G(d) basis sets agree with each other within 0.001 Å and the angles within 0.1^o, even though the authors of Ref. [113] employ an auxiliary basis to expand the electron density while we do not.

The other isomer, *meta*-PA, has non-equivalent single and double bonds (not listed in Table 9), with BLA values within the 0.035-0.04 Å range, depending on the particular basis set and functional. These BLAs are much closer to the value in *cis*-PA than in *trans*-PA.

The relative energies between the *cis*- and *trans*-PA isomers are presented in Table 10. Again, our LSDA values are practically the same as those in Ref. [113] and agree quite well with their B3LYP/3-21G and B3LYP/6-31G(d) energy differences of 2.0 and 2.3 kcal/mol, respectively [113]. At the HF/4-31G level, Teramae [106] reported an energy difference of 2.1 kcal/mol. Despite the differences in calculated BLAs at the equilibrium geometries, all methods predict the *cis*-PA/*trans*-PA energy difference in a close range.

Table 10 Relative energies per monomer (kcal/mol) of polyacetylene isomers at different levels of theory.

	LSDA 3-21G	LSDA 6-31G(d)	PBE 3-21G	PBE 6-31G(d)	VSXC 3-21G	VSXC 6-31G(d)
trans-$(C_2H_2)_x$	0.00	0.00	0.00	0.00	0.00	0.00
cis-$(C_2H_2)_x$	1.87	2.24	2.21	2.50	0.72	1.34
meta-$(C_2H_2)_x$	0.88	1.06	1.04	1.18	0.21	0.51

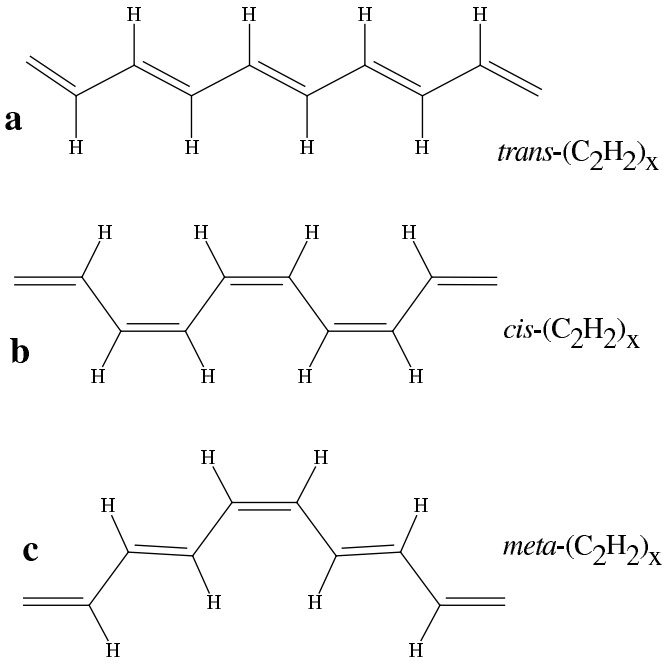

Figure 9 Isomers of polyacetylene (PA). (a) *trans*-PA; (b) *cis*-PA; (c) *meta*-PA

Poly(p-phenylenevinylene) - PPV

PPV (Figure 7a) has been studied quite extensively in many theoretical papers because of its importance for applications on light emitting diodes (LED) [122]. To the best of our knowledge, geometry optimizations of the PPV polymer have previously been carried out only at the semiempirical AM1 level of theory [123]; other authors have also computed the LDA band structure without geometry optimization [124]. In this work, we have used our DFT PBC code with analytic energy gradients to find the equilibrium geometry of PPV using three DFT functionals and two basis sets. The results can be found in Table 11, where for comparison purposes, we also present AM1 values from Ref. [123]. The structural data agree reasonably well with each other. Compared to AM1, the DFT results indicate much larger conjugation of the vinyl unit with the benzene ring. For example, AM1 predicts a 0.013-0.016 Å bond alternation within the ring, while DFT yields 0.027-0.031 Å. In order to produce results comparable to experimentally available data for PPV, we have computed $\mathbf{k} = 0$ frequencies and their intensities by numerical differentiation of forces and dipole moments (Table 12). The theoretical results compare very well with the experimental results [125]. Such agreement indicates that the chosen functionals seem to perform quite adequately for PPV.

Table 11 Geometrical parameters of PPV (Å and degrees).

	LSDA 3-21G	LSDA 6-31G(d)	PBE 3-21G	PBE 6-31G(d)	VSXC 3-21G	VSXC 6-31G(d)	AM1
R_1	1.351	1.353	1.364	1.365	1.358	1.361	1.344
R_2	1.441	1.438	1.460	1.456	1.456	1.455	1.451
R_3	1.409	1.406	1.423	1.420	1.417	1.417	1.403
R_4	1.379	1.378	1.392	1.391	1.388	1.389	1.390
R_5	1.407	1.405	1.421	1.418	1.416	1.416	1.406
R_{H1}	1.100	1.101	1.097	1.098	1.092	1.095	
R_{H2}	1.096	1.098	1.095	1.096	1.087	1.091	
R_{H3}	1.095	1.097	1.093	1.095	1.085	1.089	
A_1	126.3	126.6	126.6	127.1	125.9	126.6	
A_2	119.1	119.1	118.9	119.0	118.3	118.1	
A_3	121.5	121.8	121.7	122.0	121.5	121.9	
A_4	120.9	120.9	121.0	121.1	120.4	120.4	
A_5	117.6	117.3	117.3	116.9	118.1	117.7	
Gap (eV)	1.35	1.26	1.38	1.30	1.48	1.38	

Table 12 PPV harmonic frequencies (cm^{-1} and their IR intensities (km/mol) calculated with DFT methods and a 6-31G(d) basis set.

	LSDA		PBE		VSXC		Expt.
Sym	Freq.	Int.	Freq.	Int.	Freq.	Int.	Freq.
B_g	124	0	115	0	104	0	
A_u	225	0.2	222	0.1	225	0.2	
B_g	327	0	317	0	318	0	
A_g	321	0	319	0	323	0	
A_u	397	0	393	0	388	0.05	
B_u	421	0.01	419	0.04	412	0.01	429
A_u	555	12	550	11	549	9	558(s)[a]
A_g	629	0	627	0	617	0	
A_g	657	0	655	0	651	0	
B_g	696	0	689	0	681	0	
B_u	799	0.1	786	0.06	799	0	785
B_g	802	0	798	0	808	0	
A_u	824	30	816	31	820	33	837(s)
B_g	867	0	851	0	852	0	
A_g	905	0	887	0	898	0	
A_u	910	0.4	910	0.2	920	2	
B_g	935	0	927	0	936	0	
A_u	957	35	961	31	971	32	966(s)
B_u	1005	0	1000	0	1016	0.1	1013
B_u	1108	7	1114	4	1104	5	1108
A_g	1168	0	1169	0	1171	0	
A_g	1224	0	1209	0	1215	0	
B_u	1215	3	1226	2	1233	1	1211
B_u	1313	2	1295	2	1309	1	1271
A_g	1291	0	1297	0	1308	0	
A_g	1313	0	1327	0	1338	0	
B_u	1413	2	1386	0.1	1398	0.2	1339
B_u	1474	3	1442	4	1458	4	1424(s)
B_u	1548	0.01	1524	0.01	1548	0.1	1518(s)
A_g	1572	0	1533	0	1558	0	
A_g	1609	0	1573	0	1597	0	
A_g	1681	0	1644	0	1666	0	
A_g	3068	0	3080	0	3118	0	
B_u	3076	14	3088	15	3127	15	
B_u	3106	2	3107	7	3142	14	
A_g	3108	0	3108	0	3145	0	
B_u	3125	16	3131	28	3182	22	
A_g	3127	0	3133	0	3184	0	

[a] (s) labels strong bands.

Single wall carbon and boron-nitride nanotubes

We have optimized a series of achiral zigzag (n,0) and armchair (n,n) carbon and boron-nitride (BN) single-wall nanotubes (SWNT) at the PBE/3-21G level of theory. The values of n were chosen from the range 3-14. We have also considered the structures with effective $n = \infty$, i.e. graphite and planar BN. In the SWNT calculations, the minimum unit cell contained $4n$ atoms regardless of the type, while the 2D graphite and BN unit cells contained 4 atoms. During geometry relaxation coordinates for all atoms of the same type were symmetrized in order to obtain a final structure with perfect cylindrical symmetry. In the **k** space integration, 128 **k** points were employed for metallic structures and 32 **k** points were used for SWNTs with non-zero gaps.

The electronic structure of the carbon SWNT was found to be in agreement with the previous calculations at lower levels of theory [126]. All carbon SWNTs of the (n,n) chirality were found to be metallic, while the (n,0) SWNTs were metallic up to n=6, and became semiconducting for larger n. On the other hand, all BN SWNTs had significantly larger band gaps. In armchair (n,n) BN SWNTs, the smallest band gap 4.1 eV was computed for (3,3) BN SWNT, and this value rapidly increased to the planar BN value of 4.5 eV, reaching this value in (5,5) SWNT. At the same time in zigzag (n,0) structures, the smallest band gap of 1.2 eV found in (3,0) SWNT increased more slowly with n and reached the planar BN value only at about n=14. While the trends for band gaps found in our calculations agree qualitatively with the systematic tight-binding studies [127], we find that latter

calculations underestimate the band gap for smaller radius SWNTs compared to our results, while for larger SWNTs the values are in a better agreement. On the other hand, better quality LDA calculations [128] predict the band gap for (4,4) BN SWNT to be close to the gap of planar BN sheet, similar to what we see in our calculations.

One of the important characteristics in the SWNTs is the strain energy, E_s, as a function of their curvature $1/R$. The plots of E_s versus the tube radius for carbon and BN SWNTs are shown in Figures 10 and 11, while Figure 12 contains an overlay of these two. The elasticity theory predicts that E_s should be proportional to the square of the nanotube curvature $1/R^2$, so when plotting the former as a function of the latter almost straight lines appear for all tubes but the smallest ones (Figure 13). In agreement with the previous tight-binding predictions [129], our data also demonstrates (Figure 12) that BN SWNTs have smaller strain energies than carbon SWNTs for the same radius. It is worth noting, however, that in our calculations the relative difference between strain energies for carbon and BN SWNTs decreases with increasing tube radius, mostly due to the subquadratic scaling of the BN SWNTs strain energy with the tube radius, discussed below.

To examine in more detail what values of parameter a are observed for the equation $E_s = C/R^a$, we fitted the logarithms of the corresponding variables for the largest diameter SWNTs. For carbon tubes we found that $a = 2.00$ with high precision. The value of constant C determined from larger diameter SWNTs is 1.95 eV·Å^2/atom, and is independent of whether (n,0) or (n,n) structures are considered. This number is in good agreement with the LDA-based values of

1.95±0.05 eV·Å^2/atom computed both for armchair and zigzag tubes [130], and 2.00 eV·Å^2/atom value obtained from the pseudopotential-density-functional theory calculations for (n,n) tubes [131]. Quite interestingly, in the latter case the authors have also obtained $C = 2.16$ eV·Å^2/atom for the (10,0) structure, while in our calculations (and Ref. [130]) no difference was found between (10,0) and a (n,n) SWNT of the matching radius. The lack of data for other (n,0) SWNTs in Ref. [131] does not allow us to make any conclusions about the nature of such a discrepancy.

In contrast to carbon SWNTs, both for zigzag and armchair BN tubes we have found the subquadratic dependence of the strain energy on the tube curvature with $a = 1.95$. This effect might be due to the buckling of the BN SWNTs, with B atoms displaced toward the tube axis and N atoms pushed outwards. The plot of the degree of buckling versus the tube radius is shown in Figure 14. Similar to the strain energy, the buckling does not depend on the SWNT type [(n,0) or (n,n)] and is a function of the tube radius only. Using log-log fits we established that for large diameter tubes, the degree of buckling is sublinear with respect to the curvature $1/R$ and probably a result of the electronic effects.

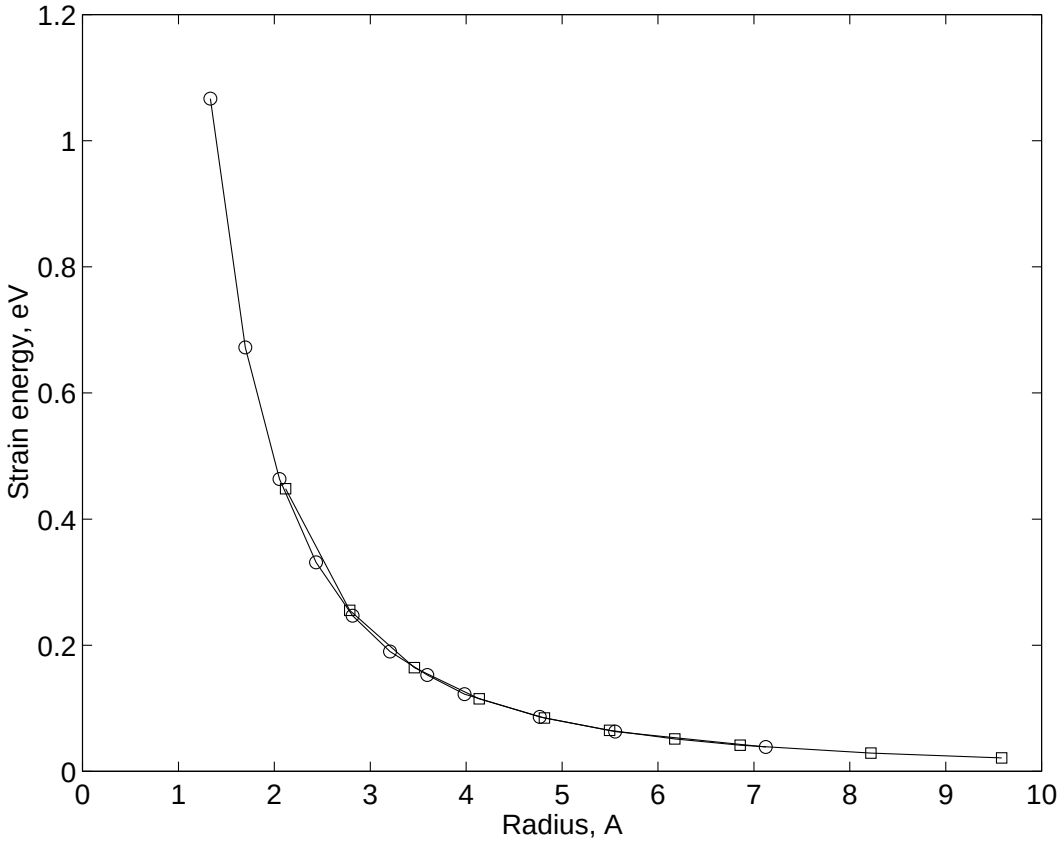

Figure 10 Curvature strain energy as a function of the equilibrium carbon tube radius. ◯ represent data points for (n,0) tubes; □ represent data points for (n,n) tubes.

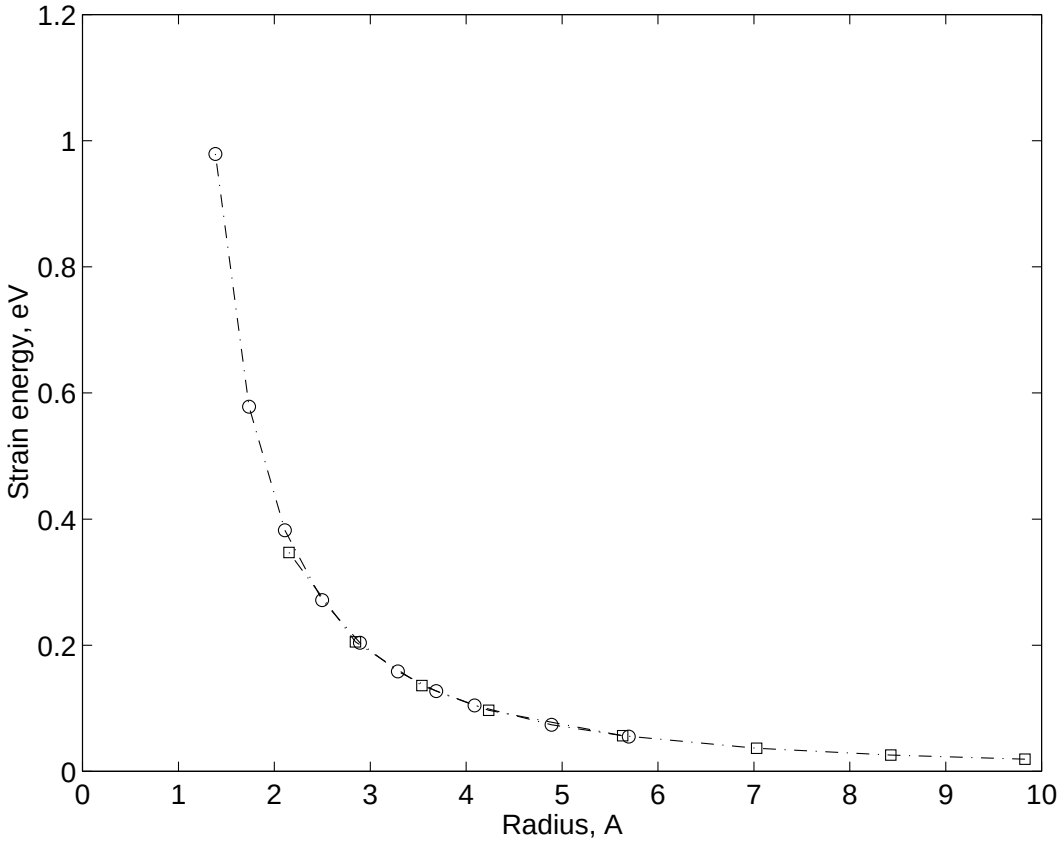

Figure 11 Curvature strain energy as a function of the equilibrium BN nanotube radius. ◯ represent data points for $(n,0)$ tubes; □ represent data points for (n,n) tubes.

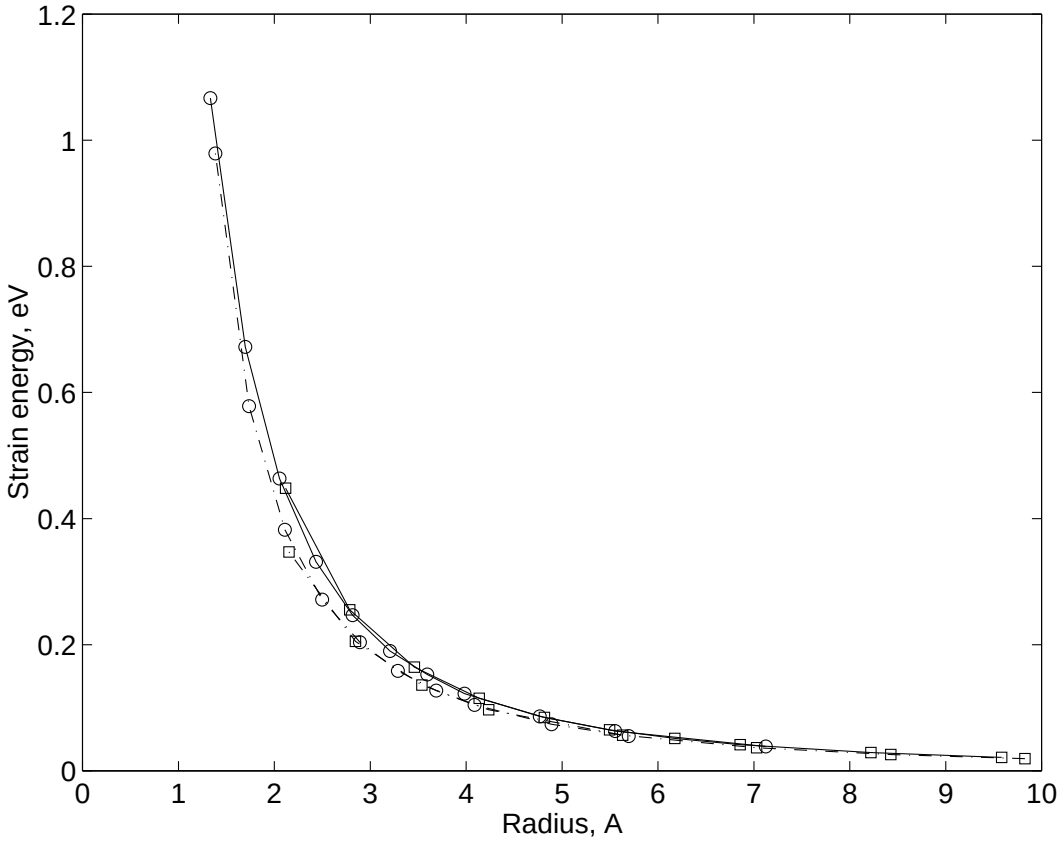

Figure 12 Curvature strain energy for carbon and BN tubes as a function of the equilibrium radius. ◯ and solid lines represent data for (n,0) carbon tubes; □ and solid lines represent data for (n,n) carbon tubes; ◯ and dashdotted lines represent data for (n,0) BN tubes; □ and dashdotted lines represent data for (n,n) BN tubes.

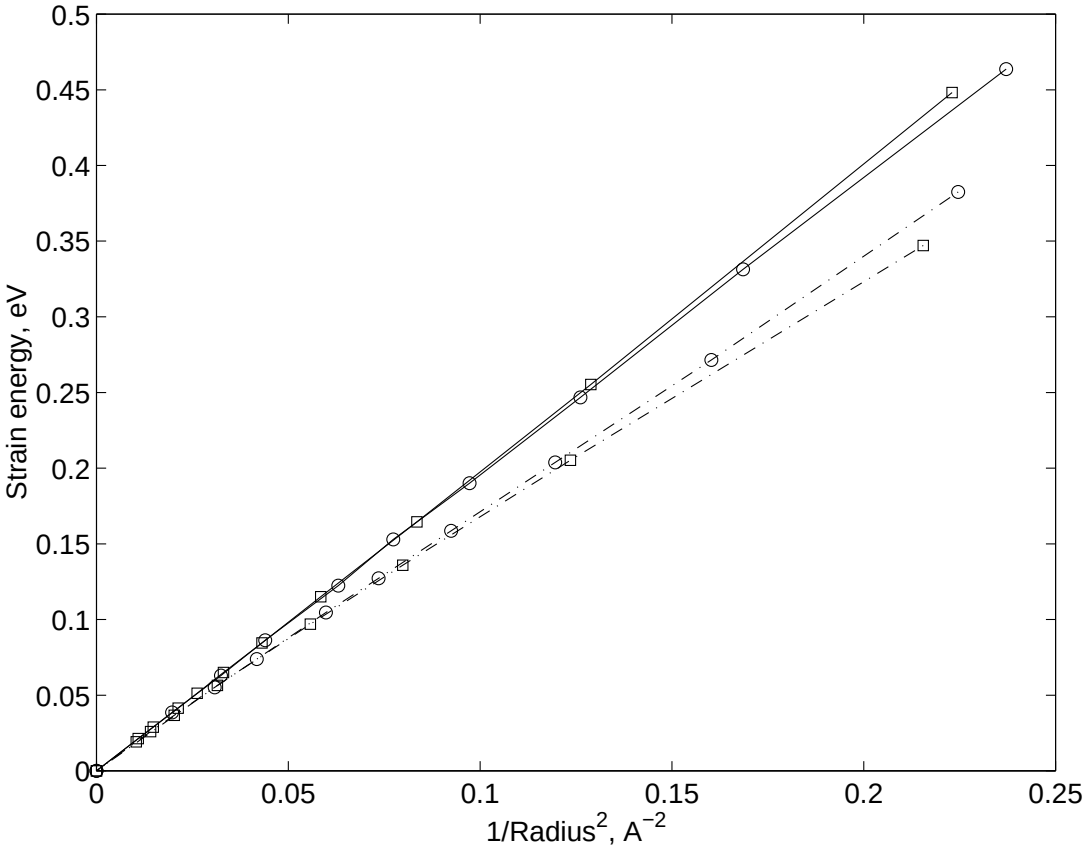

Figure 13 Curvature strain energy for carbon and BN tubes as a function of the square of their inverse radius. ◯ and solid lines represent data for (n,0) carbon tubes; □ and solid lines represent data for (n,n) carbon tubes; ◯ and dashdotted lines represent data for (n,0) BN tubes; □ and dashdotted lines represent data for (n,n) BN tubes.

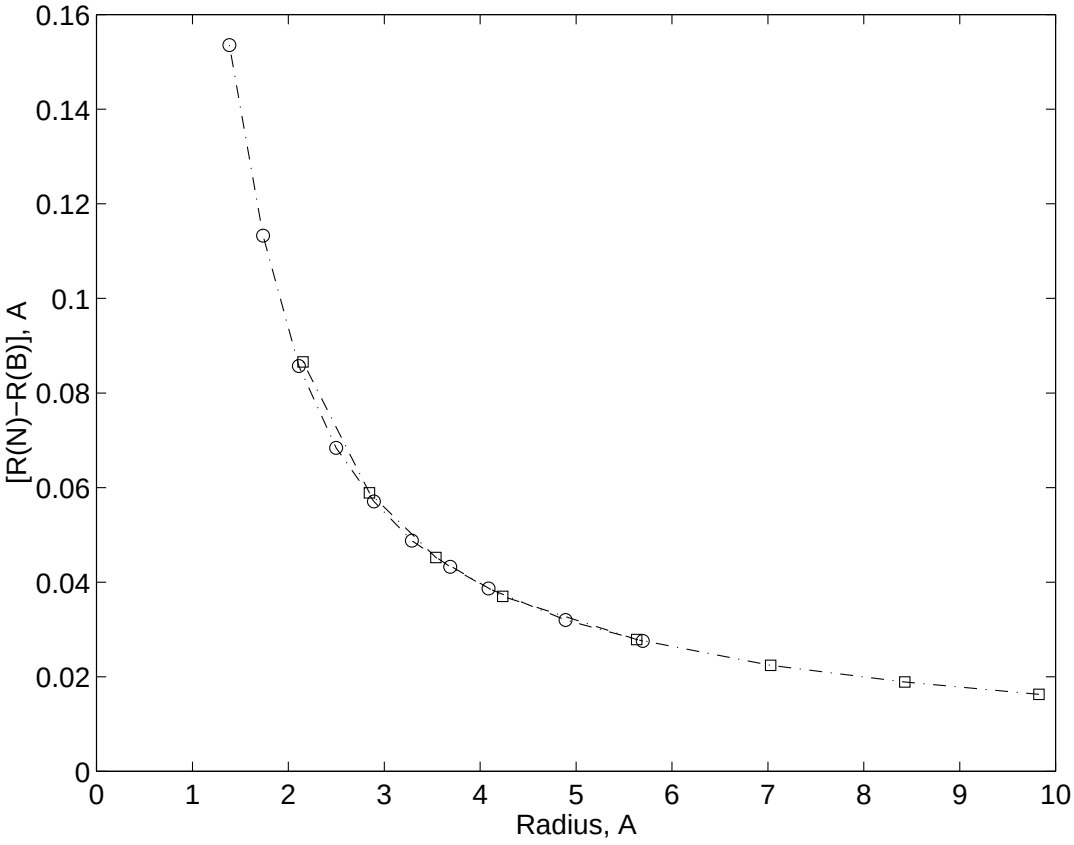

Figure 14 Buckling in the BN tube equilibrium structures versus the tube radius. ◯ represent data points for $(n,0)$ tubes; □ represent data points for (n,n) tubes.

Chapter 8

Conclusions

In this work, we have presented an efficient implementation of a periodic DFT method for electronic structure calculations with Gaussian basis sets. To the best of our knowledge, this is the first periodic code that employs the Fast Multipole Method for all electrostatic interactions. As demonstrated here, the periodic FMM for Gaussian charge distributions permits high accuracy, favorable $\mathcal{O}(N)$ scaling and relatively low absolute computational cost for one of the most demanding stages of the periodic calculation - the Coulomb problem. Furthermore, high accuracy in all parts of the self-consistent energy code helps to avoid linear dependency problems when dealing with diffuse basis sets.

The algorithm developed here also incorporates a full set of analytic first energy derivatives with respect to geometrical parameters, specifically atomic forces and stress tensor. Combining these derivatives with a novel redundant internal coordinate algorithm for optimization of periodic systems, a general efficient "black box" tool for studying a variety of periodic systems is created. The benchmark studies carried out in this work convincingly demostrate that the current implementation offers a unique combination of speed, accuracy and versatility, and is, perhaps, the first code that makes "chemical accuracy" calculations for systems larger than a hundred atoms practical.

Future work on the PBC program could proceed along several different paths. Ever important computational efficiency may be improved via adding an efficient **k** space integration scheme, an ability to utilize $\mathcal{O}(N)$ sparse matrix methods in the density update step, and the capability to reduce the required CPU time by employing the system's symmetry. On the other hand, the work on new features could involve implementing the periodic Hartree-Fock exchange, which is required for a variety of very accurate hybrid density functionals and the Hartree-Fock (HF) method. The periodic HF could serve as a frame for implementing MP2 and higher order correlated methods among which the atomic orbital based formulations are probably the simplest to extend to periodic systems [14, 16, 132].

Another possible area of activity may include analytic second derivatives to make more practical evaluation of realistic phonon dispersion curves in solids. Implementation of the time-dependent density functional theory would permit computations of visible and UV spectra. Technologically important questions such as reactions on surfaces and in porous solids and defects in periodic structures could be addressed by developing methods specifically suited for treating a perturbed region within a perfect periodic structure.

While the current PBC code is already very useful for a variety of applications, it can also serve as a foundation for implementing other computational methods for periodic systems which accurately evaluate a variety of other useful physical and chemical properties. Such a set of tools would complement methods already available for studying molecules and advance further the capabilities of computational chemistry.

References

[1] K. N. Kudin and G. E. Scuseria, Chem. Phys. Lett. **283**, 61 (1998).

[2] K. N. Kudin and G. E. Scuseria, Chem. Phys. Lett. **289**, 611 (1998).

[3] K. N. Kudin and G. E. Scuseria, J. Chem. Phys. **111**, 2351 (1999).

[4] K. N. Kudin and G. E. Scuseria, Phys. Rev. B **61**, 5141 (2000).

[5] K. N. Kudin and G. E. Scuseria, Phys. Rev. B **61**, 16440 (2000).

[6] K. N. Kudin, G. E. Scuseria and H. B. Schlegel, J. Chem. Phys., in press.

[7] K. N. Kudin and G. E. Scuseria, J. Chem. Phys. **113**, 7779 (2000).

[8] K. N. Kudin, H. F. Bettinger and G. E. Scuseria, Phys. Rev. B, in press.

[9] R. Improta, V. Barone, K. N. Kudin and G. E. Scuseria, J. Chem. Phys., in press.

[10] R. Improta, V. Barone, K. N. Kudin and G. E. Scuseria, J. Am. Chem. Soc., submitted.

[11] A. D. Becke, J. Chem. Phys. **96**, 2155 (1992).

[12] G. E. Scuseria, J. Chem. Phys. **97**, 7528 (1992).

[13] G. E. Scuseria, J. Phys. Chem. A. **103**, 4782 (1999).

[14] P. Y. Ayala and G. E. Scuseria, J. Chem. Phys. **110**, 3660 (1999).

[15] M. Schütz, G. Hetzer, and H-J. Werner, J. Chem. Phys. **111**, 5691 (1999).

[16] G. E. Scuseria and P. Y. Ayala, J. Chem. Phys. **111**, 8330 (1999).

[17] C. Pisani, R. Dovesi, and C. Roetti, *Hartree-Fock Ab Initio Treatment of Crystalline Systems*, Lecture Notes in Chemistry, (Springer-Verlag, Heidelberg, 1988), Vol. 48.

[18] C. Pisani (ed.), Lecture Notes in Chemistry, (Springer-Verlag, Heidelberg, 1996), Vol. 67.

[19] J. W. Mintmire, J. R. Sabin, and S. B. Trickey, Phys. Rev. B **26**, 1743 (1982).

[20] J. C. Boettger and S. B. Trickey, Phys. Rev. B **29**, 6425, 6434 (1985).

[21] P. J. Feibelman, Phys. Rev. B **35**, 2626 (1987).

[22] J. E. Jaffe and A. C. Hess, J. Chem. Phys. **105**, 10983 (1996).

[23] S. Hirata and S. Iwata, J. Chem. Phys. **107**, 10075 (1997).

[24] D. K. Remler and P. A. Madden, Mol. Phys. **70**, 921 (1990).

[25] G. Lippert, J. Hutter, and M. Parrinello, Mol. Phys. **92**, 477 (1997).

[26] Gaussian 99, Development Version (Revision B.08+) M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, Jr., R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, C. Adamo, J. Jarmillo, R. Cammi, C. Pomelli, J. Ochterski, G. A. Petersson, P. Y. Ayala, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B.

Johnson, W. Chen, M. W. Wong, J. L. Andres, C. Gonzalez, M. Head-Gordon, E. S. Replogle, and J. A. Pople, Gaussian, Inc., Pittsburgh PA, 1998.

[27] J. B. Nicholas, Topics in Catalysis **4**, 157 (1997).

[28] C. H. Choi and M. Kertesz, J. Chem. Phys. **107**, 6712 (1997).

[29] L. Greengard and V. Rokhlin, J. Comp. Phys. **73** (1987).

[30] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).

[31] K. E. Schmidt and M. A. Lee, J. Stat. Phys. **63**, 1223 (1991); erratum: **87**, 955 (1996).

[32] C. G. Lambert, T. A. Darden, and J. A. Board, J. Comp. Phys. **126**, 274 (1996).

[33] F. Figueirido, R. M. Levy, R. Zhou, and B. Berne, J. Chem. Phys. **106**, 9835 (1997); erratum: **107**, 7002 (1997).

[34] M. Challacombe, C. A. White, and M. Head-Gordon, J. Chem. Phys. **107**, 10131 (1997).

[35] For reviews see:

H. B. Schlegel "Geometry optimization on potential energy surfaces" in *Modern Electronic Structure Theory*, D. R. Yarkony, ed., (World Scientific Publishing, Singapore, 459-500, 1995); H. B. Schlegel "Geometry optimization" in *Encyclopedia of Computational Chemistry*, Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer and P. R. Schreiner ed. (Wiley, New York, 1998)

[36] S. F. Boys, Proc. R. Soc. London Ser. A **200**, 542 (1950).

[37] J. Q. Sun and R. J. Bartlett, J. Chem. Phys. **109**, 4209 (1998).

[38] H. Teramae, T. Yamabe, C. Satoko, and A. Imamura, Chem. Phys. Lett. **101**, 149 (1983).

[39] M. Kertesz, Chem. Phys. Lett. **106**, 443 (1984).

[40] P. J. Feibelman, Phys. Rev. B. **44**, 3916 (1991).

[41] S. Hirata and S. Iwata, J. Mol. Struct. THEOCHEM **451**, 121 (1998).

[42] S. Hirata and S. Iwata, J. Chem. Phys. **109**, 4147 (1998).

[43] P. Pulay, Mol. Phys. **17**, 197 (1969).

[44] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, Int. J. Quantum Chem. Symp. **S13**, 225 (1979).

[45] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3780 (1985).

[46] O. H. Nielsen and R. M. Martin, Phys. Rev. B **32**, 3792 (1985).

[47] D. L. Strout and G. E. Scuseria, J. Chem. Phys. **102**, 8448 (1995).

[48] L. Greengard, *The rapid evaluation of potential fields in particle systems*, MIT Press, Cambridge, MA, 1988.

[49] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, Chem. Phys. Lett. **253**, 268 (1996).

[50] P. Ewald, Ann. Phys. **64**, 253 (1921).

[51] C. A. White, B. G. Johnson, P. W. M. Gill, and M. Head-Gordon, Chem. Phys. Lett. **230**, 8 (1994).

[52] R. Euwema and G. Surratt, J. Phys. Chem. Solids **36**, 67 (1975).

[53] M. Häser and R. Ahlrichs, J. Comput. Chem. **10**, 104 (1989).

[54] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **109**, 8764 (1998).

[55] M. Challacombe, E. Schwegler, and J. Almlöf, J. Chem. Phys. **104**, 4685 (1996).

[56] J. M. Perez-Jorda and W. Yang, J. Chem. Phys. **107**, 1218 (1997).

[57] J. C. Burant, M. C. Strain, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **248**, 43 (1996).

[58] J. C. Burant, M. C. Strain, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **258**, 45 (1996).

[59] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, Chem. Phys. Lett. **257**, 213 (1996).

[60] J.-M. Andre, J. Delhalle, G. Kapsomenos, and G. Leroy, Chem. Phys. Lett. **14**, 485 (1972).

[61] C. J. Pickard and M. C. Payne, Phys. Rev. B **59**, 4685 (1999).

[62] O. Jepsen and O. K. Andersen, Solid St. Commun. **9**, 1763 (1971).

[63] G. Lehmann and M. Taut, Phys. Status Solidi B **54**, 469 (1972).

[64] A. Baldereschi, Phys. Rev. B **7**, 5212 (1973).

[65] D. J. Chadi and M. L. Cohen, Phys. Rev. B **8**, 5747 (1973).

[66] H. J. Monkhorst and J. D. Pack, Phys. Rev. B **13**, 5189 (1976).

[67] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).

[68] P. Pulay, J. Comp. Chem. **3**, 556 (1982).

[69] S. Suhai, P. S. Bagus, and J. Ladik, Chem. Phys. **68**, 467 (1982).

[70] M. J. Frisch, personnal communication.

[71] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry. Introduction to Advanced Electronic Structure Theory*, Dover Publications, Inc., 1996.

[72] P.-O. Löwdin, Rev. Mod. Phys. **39**, 259 (1967).

[73] M. Challacombe, J. Chem. Phys. **110**, 3232 (1999).

[74] J. M. Millam and G. E. Scuseria, J. Chem. Phys. **106**, 5569 (1997).

[75] A. D. Daniels and G. E. Scuseria, J. Chem. Phys. **110**, 1321 (1999).

[76] R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1981) and similar texts on optimization algorithms.

[77] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, J. Am. Chem. Soc. **101**, 2550 (1979).

[78] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay, J. Am. Chem. Soc. **114**, 8191 (1992).

[79] P. Pulay and G. Fogarasi, J. Chem. Phys. **96**, 2856 (1992).

[80] C. Peng, P. Y. Ayala, H. B. Schlegel, and M. J. Frisch, J. Comput. Chem. **17**, 49 (1996).

[81] J. Baker, A. Kessi, and B. Delley, J. Chem. Phys. **105**, 192 (1996).

[82] M. v. Arnim and R. Ahlrichs, J. Chem. Phys. **111**, 9183 (1999).

[83] H. B. Schlegel, Theor. Chim. Acta **66**, 333 (1984).

[84] T. H. Fischer and J. Almlof, J. Phys. Chem. **96**, 9768 (1992).

[85] J. M. Wittbrodt and H. B. Schlegel, J. Mol. Struct. THEOCHEM **398**, 55 (1997).

[86] Ö. Farkas and H. B. Schlegel, J. Chem. Phys. **109**, 7100 (1998).

[87] Ö. Farkas and H. B. Schlegel, J. Chem. Phys. **111**, 10806 (1999).

[88] B. Paizs, G. Fogarasi, and P. Pulay, J. Chem. Phys. **109**, 6571 (1998).

[89] J. Baker, D. Kinghorn, and P. Pulay, J. Chem. Phys. **110**, 4986 (1999).

[90] S. R. Billeter, A. J. Turner, and W. Thiel, Phys. Chem. Chem. Phys. **2**, 2177 (2000).

[91] B. Paizs, J. Baker, S. Suhai, and P. Pulay, J. Chem. Phys., in press.

[92] K. Németh, O. Coulaud, G. Monard and J. G. Ángyán, J. Chem. Phys., in press.

[93] A. D. Daniels, G. E. Scuseria, Ö. Farkas, and H. B. Schlegel, Int. J. Quantum Chem. **77**, 82 (2000).

[94] A. v. d. Vaart, D. Suarez, and K. M. Merz, Jr., J. Chem. Phys., in press.

[95] J.D. Gale, JCS Faraday Trans. **93**, 629 (1997).

[96] I. Souza and J. L. Martins, Phys. Rev. B **55**, 8733 (1997).

[97] E.B. Wilson, J.C. Decius and P.C. Cross, *Molecular Vibrations*, (McGraw-Hill, New York, 1955).

[98] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996); *ibid. (E)* **78**, 1396 (1997).

[99] S. Swaminathan, B.M. Craven, R.K. McMullan, Acta Crystallogr. B **40**, 300 (1984).

[100] B. Rousseau, R. Keuleers, H.O. Desseyn, H.J. Geise, C. Van Alsenoy, Chem. Phys. Lett. **302**, 55 (1999).

[101] M. S. Miao, V.E. Van Doren, R. Keuleers, H. O. Desseyn, C. Van Alsenoy, J. L. Martins, Chem. Phys. Lett. **316**, 297 (2000).

[102] S. H. Vosko, L. Wilk, and M. Nusair, Can. J. Phys. **58**, 1200 (1980).

[103] T. Van Voorhis and G. E. Scuseria, J. Chem. Phys. **109**, 400 (1998).

[104] A. Karpfen and R. Holler, Solid St. Commun. **37**, 179 (1981).

[105] R. Dovesi, Int. J. Quantum Chem. **26**, 197 (1984).

[106] H. Teramae, J. Chem. Phys. **85**, 990 (1986).

[107] J. W. Mintmire and C. T. White, Phys. Rev. B **35**, 4180 (1987).

[108] M. Springborg, J.-L. Calais, O. Goscinski, and L. A. Eriksson, Phys. Rev. B **44**, 12713 (1991).

[109] G. Fogarasi, R. Liu, and P. Pulay, J. Phys. Chem. **97**, 4036 (1993).

[110] S. Suhai, Phys. Rev. B **51**, 16553 (1995).

[111] S. Hirata, H. Torii, and M. Tasumi, J. Chem. Phys. **103**, 8964 (1995).

[112] J. Q. Sun and R. J. Bartlett, J. Chem. Phys. **104**, 8553 (1996).

[113] S. Hirata, H. Torii, and M. Tasumi, Phys. Rev. B **57**, 11994 (1998).

[114] B. Champagne, E. A. Perpete, S. J. A. v. Gisbergen, E.-J. Baerends, J. G. Snijders, C. Soubra-Ghaoui, K. A. Robins, and B. Kirtman, J. Chem. Phys. **109**, 10489 (1998).

[115] M. Springborg, J. Am. Chem. Soc. **121**, 11211 (1999).

[116] C. S. Yannoni and T. C. Clarke, Phys. Rev. Lett. **51**, 1191 (1983).

[117] H. Kahlert, O. Leitner, and G. Leising, Syn. Met. **17**, 467 (1987).

[118] Q. Zhu and J. E. Fisher, Solid St. Commun. **83**, 179 (1992).

[119] S. Sadhukhan, D. Munoz, C. Adamo, and G. E. Scuseria, Chem. Phys. Lett. **306**, 83 (1999).

[120] A. D. Rabuck and G. E. Scuseria, Chem. Phys. Lett. **309**, 450 (1999).

[121] J. Jaramillo and G. E. Scuseria, Chem. Phys. Lett. **312**, 269 (1999).

[122] J. H. Burroughes *et al.*, Nature **347**, 539 (1990).

[123] J. Cornil, D. A. dos Santos, D. Beljonne, and J. L. Brédas, J. Phys. Chem. **99**, 5604 (1995).

[124] M. E. Vaschetto and M. Springborg, J. Mol. Struct. THEOCHEM **460**, 141 (1999).

[125] B. Tian, G. Zerbi, R. Schenk, and K. Müllen, J. Chem. Phys. **95**, 3191 (1991).

[126] X. Blase, L. X. Benedict, E. L. Shirley, and S. G. Louie, Phys. Rev. Lett. **72**, 1878 (1994).

[127] A. Rubio, J. L. Corkill, and M. L. Cohen, Phys. Rev. B **49**, 5081 (1994).

[128] X. Blase, J.-Ch. Charlier, A. De Vita, and R. Car, Appl. Phys. A **68**, 293 (1999).

[129] E. Hernandez, C. Goze, P. Bernier, and A. Rubio, Phys. Rev. Lett. **80**, 4502 (1998).

[130] J. Kurti, G. Kresse, and H. Kuzmany, Phys. Rev. B **58**, 8869 (1998).

[131] D. Sanchez-Portal, E. Artacho, J. M. Soler, A. Rubio, and P. Ordejon, Phys. Rev. B **59**, 12678 (1999).

[132] P. Y. Ayala and G. E. Scuseria, Chem. Phys. Lett. **322**, 213 (2000).