INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Université d'Ottawa • University of Ottawa

Dedications

To my brother, Chee Keong Goh

and

all sentient beings

Abstract

Algorithms for treating large molecules are developed and implemented within the DeFT density functional package. In order to realistically model large molecules by quantum molecular methods, new algorithms whose computational costs increase linearly with system size must be developed. Moreover, to efficiently exploit the vast computing power provided by massively parallel computer architectures, we would like our new linear scaling algorithms to be scalable on such architectures. When developing these new algorithms, we strive to lose as little accuracy as possible. Many of our new approaches are inspired by Yang's divide and conquer (DAC) philosophy.

An approximate electronic density is developed exploiting the fact that a molecule's electronic density can be accurately fit in a DAC fashion. It is constructed by using true subsystem densities to calculate contributions from nearby subsystems (in the vicinity of the region of interest) and using fitted subsystem densities to calculate contributions from faraway subsystems. This approximate density can ease the computational burden associated with various calculations. Its application to the calculation of the molecular electrostatic potential is illustrated.

An efficient algorithm for carrying out a DAC method on massively parallel computers is developed. This algorithm combines the principles of both coarse and fine parallelism and allows for an uneven distribution of processors among subsystem calculations. The precise approach employed depends on the number of processors available and the number of subsystems to be treated. The processor to subsystem ratio is crucial and can be adjusted to achieve optimal performance. Since each subsystem can

be efficiently run on a cluster of processors, our algorithm should be scalable up to hundreds, if not thousands, of processors when treating very large molecules.

The exchange and correlation (XC) terms within a Kohn-Sham density functional method can also be fit by a DAC approach. This new DAC XC fitting procedure is outlined. The results and timings of benchmark calculations on extended glycine polypeptides are presented. The benefits of using fitted XC terms versus direct numerical integration of the XC terms are discussed. Other modifications made to achieve linear scaling for every XC procedure are also discussed. New grids containing more radial and angular points have also been created to achieve better precision in the final results. A scheme using fewer points for the fitting procedures and more points for the final numerical integration of the energy and energy gradients is presented.

Benchmark calculations on a series of glycine polypeptides are performed to fully analyze the errors introduced by a DAC approach to constructing the electronic density. Various buffer space cutoffs are tested. The errors introduced by dropping various basis functions beyond these cutoffs from subsystem calculations are investigated. The possibility of partially truncating the basis sets beyond the cutoff is tested. Different types of basis functions are found to contribute differently to the DAC errors. This fact is exploited to define a multiple buffer space cutoff approach that will improve the efficiency of DAC calculations.

Acknowledgments

I would like to express my heartfelt thanks to Professor Alain St-Amant, my research supervisor, for his guidance, support, and patience. He has been a wonderful boss, I am thankful to the Triple Gems for that.

I thank the graduate school of University of Ottawa and the chemistry department for financial support.

The love and motivation of my wonderful husband, Edet F. Archibong, is greatly appreciated. Without his encouragement, I would not have attended graduate school. He has been a constant force behind my success throughout my years in the university.

I would also like to thank my sisters, mother, and nephews in Singapore for their love and support. My first sister, Soo Hiang Goh, has been my supplier for those goodies from Singapore. My second sister, Soo Cheng Goh, and her husband, Michael Lim, are my sources to the KPO news in Singapore. The friendship of my third sister, Sophia Goh, is greatly appreciated.

Table of Contents

List of Tables

List of Figures

Chapter 1

General Introduction

1.1 Preliminary

Predicting the properties and behaviour of chemical systems requires an understanding of the interaction between, and among, electrons and nuclei. It was once thought that the laws of classical mechanics could adequately describe their motions and interactions. However, accumulated experimental evidence showed that classical mechanics failed when applied to microscopic particles. The failure of classical mechanic led to the development of the appropriate concepts and the formulation of quantum mechanics. In quantum mechanics [1], all the properties of a system are expressed in terms of a wavefunction that can be obtained by solving the Schrödinger equation,

$$H\Psi = E\Psi \quad . \tag{1.1}$$

In equation 1.1, H is the Hamiltonian operator, E is the total energy of the system and Ψ is the wavefunction which depends on the spatial coordinates of all the particles.

The first step in most quantum mechanical treatments of a molecular system is the separation of the electronic and nuclear motions by invoking the Born-Oppenheimer approximation [2]. This approximation is based on the assumption that the nuclei move much more slowly than the electrons because the nuclear masses are much greater than the mass of the electrons. Consequently, the electrons adjust themselves instantaneously to the positions of the nuclei. The quantum mechanical treatment of the molecular system is thus reduced to solving the electrons' motion in the field of fixed nuclei, and the electronic wavefunction may be obtained from the resulting electronic Schrödinger equation,

$$H(\boldsymbol{r};\boldsymbol{R})\Psi(\boldsymbol{r};\boldsymbol{R}) = E(\boldsymbol{R})\Psi(\boldsymbol{r};\boldsymbol{R}) \quad . \tag{1.2}$$

Here, $\Psi(\boldsymbol{r},\boldsymbol{R})$ is the electronic wave function. It depends explicitly on the $3n$ coordinates of all n electrons, which are collectively denoted as $\boldsymbol{r}$, and implicitly on the nuclear coordinates, collectively denoted as $\boldsymbol{R}$. $\boldsymbol{E}(\boldsymbol{R})$ is the electronic energy. It depends parametrically on the nuclear positions. The electronic Hamiltonian operator, subject to frozen nuclei, is given in atomic units as

$$H(r;R) = -\frac{1}{2}\sum_{i}^{n}\nabla_i^2 - \sum_{A}^{N}\sum_{i}^{n}\frac{Z_A}{|r_i - R_A|} + \sum_{i}^{n}\sum_{j>i}^{n}\frac{1}{|r_i - r_j|} + \sum_{A}^{N}\sum_{B>A}^{N}\frac{Z_A Z_B}{|R_B - R_B|} \quad . \quad (1.3)$$

The first term is the kinetic energy operator for the electrons, the second is the electron-nuclear attraction, the third is the electron-electron repulsion, and the fourth is the nuclear-nuclear repulsion. Z_A and Z_B are the atomic numbers for nuclei A and B, respectively. The electronic Schrödinger equation cannot be solved exactly for many-electron systems because of the inseparability of the electron-electron repulsion term. Hence, an approximate solution is always sought.

1.2 Hartree-Fock Theory

A very common approach to solve the many-electron Schrödinger equation is Hartree-Fock (HF) theory [3, 4], the simplest *ab initio* method in quantum mechanics based upon the variational principle. In this approach, the n-electron Ψ is constructed from a set of spin orbitals. Each spin-orbital contains the Cartesian and spin coordinates of a single electron. To properly ensure an antisymmetric wavefunction, the trial HF wavefunction is approximated by a single determinant [5, 6] of spin orbitals,

$$\Psi_{HF} = |\phi_1(1)\phi_2(2)\cdots\phi_n(n)| \quad . \quad (1.4)$$

The best set of spin orbitals is then obtained by variationally minimizing the energy, subject to the constraint that the spin orbitals remain orthonormal. This leads to a set of integro-differential equations. In practice, the HF integro-differential equations are not easy to solve. An alternative approach, proposed independently by Roothaan [7] and Hall [8], expands the HF orbitals (spatial part of the spin orbitals) as a linear combination within a finite set of atomic basis functions (or LCAO - linear combination of atomic orbitals)

$$\phi_i = \sum_{\mu}^{K} C_{\mu i} \chi_\mu \tag{1.5}$$

where $C_{\mu i}$ are the expansion coefficients of the i^{th} molecular orbital (MO). This approach transforms the integro-differential equations into a set of matrix eigenvalue-type equations,

$$\sum_{\mu=1}^{K} C_{\mu i}\left(F_{\mu\nu} - \varepsilon_i S_{\mu\nu}\right) = 0, \qquad \nu = 1,2,\cdots K \tag{1.6}$$

subject to the orthonormality constraints

$$\sum_{\mu=1}^{K}\sum_{\nu=1}^{K} C_{\mu i}^{*} S_{\mu\nu} C_{\nu j} = \delta_{ij} \quad . \tag{1.7}$$

Here, ε_i is the i^{th} molecular orbital energy, $S_{\mu\nu}$ is an overlap matrix element,

$$S_{\mu\nu} = \int \chi_\mu^{*}(1)\chi_\nu(1) d\boldsymbol{r}_1 , \tag{1.8}$$

and $F_{\mu\nu}$ is a Fock matrix element,

$$F_{\mu\nu} = H_{\mu\nu}^{core} + \sum_{\sigma=1}^{K}\sum_{\lambda=1}^{K} P_{\sigma\lambda}\left[(\mu\nu\,|\,\sigma\lambda) - \frac{1}{2}(\mu\sigma\,|\,\nu\lambda)\right] \quad . \tag{1.9}$$

$H_{\mu\nu}^{core}$ is a matrix element of the one-electron operator

$$H_{\mu\nu}^{core} = \int \chi_\mu^*(1)\left[-\frac{1}{2}\nabla_1^2 - \sum_{A=1}^{M}\frac{Z_A}{|r_1 - R_A|}\right]\chi_\nu(1)dr_1 . \qquad (1.10)$$

The two-electron Coulomb, $(\mu\nu \mid \sigma\lambda)$, and exchange, $(\mu\sigma \mid \nu\lambda)$, integrals are given by

$$(\mu\nu \mid \sigma\lambda) = \iint \chi_\mu^*(1)\chi_\nu(1)r_{12}^{-1}\chi_\sigma^*(2)\chi_\lambda(2)dr_1dr_2 \qquad (1.11)$$

$$(\mu\sigma \mid \nu\lambda) = \iint \chi_\mu^*(1)\chi_\sigma^*(1)r_{12}^{-1}\chi_\nu(2)\chi_\lambda(2)dr_1dr_2 \qquad (1.12)$$

For a closed-shell system, the one-electron density matrix, $P_{\sigma\lambda}$,

$$P_{\sigma\lambda} = 2\sum_{i=1}^{occ} C_{\sigma i}^* C_{\lambda i} \qquad (1.13)$$

is a sum over occupied molecular orbitals only, and the factor 2 indicates that two electrons, of opposite spin, occupy each molecular orbital. The total energy of the system is then equal to the sum of the electronic energy, E_{el},

$$E_{el} = \tfrac{1}{2}\sum_{\mu=1}^{K}\sum_{\nu=1}^{K} P_{\mu\nu}(F_{\mu\nu} + H_{\mu\nu}^{core}) \qquad (1.14)$$

and the internuclear repulsion energy, E_{nuc},

$$E_{nuc} = \sum_{A}^{N}\sum_{B>A}^{N}\frac{Z_A Z_B}{R_{AB}} . \qquad (1.15)$$

The Roothaan-Hall [7, 8] equations are non-linear because the Fock matrix elements depend on the molecular orbital coefficients. Therefore, the equations must be solved in an iterative fashion until the differences in ϕ_i and ε_i in two consecutive iterations are within a predetermined threshold, i.e., a self-consistent field (SCF) has been established.

1.3 Basis Functions

In theory, any set of functions that form a complete set can be employed in atomic and molecular orbital calculations. In practice, however, the choice of the functional form is influenced by two major factors: the rate of convergence of the orbitals when expanded in terms of a particular basis set, and the ease with which the integrals over the basis functions can be evaluated. The basis functions commonly employed in LCAO electronic calculations fall into two classes:

1.3.1 Slater-type Atomic Orbitals

Slater-type atomic orbitals (STOs) [9] have the general form

$$\chi_{n,l,m} = N_{n,l,m,\zeta} r^{n-1} e^{-\zeta r} Y_{l,m}(\theta,\phi) \tag{1.16}$$

and are characterized by quantum numbers n, l, m and exponent ζ. $N_{n,l,m,\zeta}$ is a normalization constant and $Y_{l,m}$ is a spherical harmonic.

STOs provide a rapidly convergent expansion for molecular orbitals. The functions give accurate description of the orbitals near the nucleus and in the long-range, tail, region. Unfortunately, the multi-centered integrals that arise in polyatomic molecular calculations are very difficult to compute, and are, therefore, computationally inefficient.

1.3.2 Gaussian-type Atomic Orbitals

Gaussian-type orbital functions are the most popular choice of basis functions in polyatomic molecular calculations. Cartesian Gaussian functions have the form

$$g_{k,l,m} = N X^k Y^l Z^m e^{-\alpha r^2} \tag{1.17}$$

N is the normalization constant, the nonnegative integers, k, l, m describe the angular shape and direction of the orbitals, and α is the orbital exponent which determines the radial extent.

The main advantage of using Gaussian-type functions [10] lies in the fact that multi-centered integrals can be evaluated efficiently and accurately. However, GTOs as representations of molecular orbitals are less accurate than STOs, because they do not have an appropriate form near the nuclei and in the long-range "tail" region. They do not have a cusp near the nucleus, and they die off too quickly away from the nucleus. Consequently, a linear combination of Gaussian functions, termed a contracted Gaussian-type orbital (CGTO) is used to mimic a single Slater function. Each Gaussian function in a CGTO is called a primitive function. Primitive functions with large exponents are used to simulate the cusp near the nucleus and primitive functions with small exponents are used to make the basis functions die off more slowly.

1.3.3 Common Gaussian Basis Sets

A minimal basis set is the smallest set of nuclear-centered functions used in *ab initio* MO calculations, where each core and valence orbitals in an atom is represented by one function. For example, a minimal basis set for the second row atoms is comprised of a $1s$, $2s$, $2p_x$, $2p_y$, and $2p_z$ function. The minimal basis sets denoted as STO-NG by Pople and coworkers consists of expansions of one Slater-type orbital in terms of N Gaussian functions. There are two features of the Gaussian expansion that distinguish a single STO-NG basis set. First, all the Slater orbitals, ϕ_{nl}, are expressed in terms of the simplest Gaussians of the same symmetry type (i.e., the Slater 2s and higher s orbitals are

expanded in terms of the 1*s* Gaussians while the Slater 3*p* or higher *p* orbitals are expressed in terms of 2*p* Gaussians). Second, the expansions of atomic functions of a given principal quantum number, *n*, often share a common set of Gaussian exponents, α_n. This restriction slightly reduces the overall flexibility of the basis representation. However, evaluation of the various integrals is more efficient with such an exponent restriction.

A way to improve the variational flexibility of a minimal basis set is to use two functions for each function in the minimum basis set. This type of basis set is called a double-zeta basis, denoted as DZ. Since only valence orbitals are important for the description of molecular bonding, a much simpler extension to the minimum basis set, proposed by Pople *et al* [11, 12], is to double only the number of functions representing the valence region. These bases, termed split-valence (SV) basis sets, have the general form K-K'K"G. For example, in a 6-31G basis set, each core orbital is represented by a single contracted function consisting of 6 primitives. Meanwhile, each of the valence orbitals is described by two functions, where one of the two functions representing the valence orbital is a contraction of three primitives while the other consists of only one primitive.

Experience has shown that, for polar molecules and systems that contain small strained rings, it is important to include functions of higher angular quantum number in the basis set in order to allow for the possibility of a non-uniform displacement of charge away from the atomic center. These functions are called polarization functions. Typically, a set of *p*-functions is used to polarize an *s* valence orbital, a set of *d*-functions polarizes *p* valence orbitals, and so on. Basis sets that incorporate polarization functions

are called polarized basis sets. Example of split-valence plus polarization functions are the 6-31G* and 6-31G** basis sets. A 6-31G* basis set adds, to the 6-31G basis set, an extra set of *d*-type primitive functions on each heavy atom. The 6-31G** basis set adds, to a 6-31G* basis set, a set of *p*-type functions on each hydrogen atom.

When the calculation involves anions, it is necessary to include one or more sets of highly diffuse functions in the basis set. An example of a basis set including a set of diffuse function is the 6-31+G*. It is constructed by adding a single set of diffuse *s* and *p* type Gaussian functions to the 6-31G* basis set.

The quality of a basis set can be systematically improved by going to triple-zeta, quadruple-zeta, and so on, and adding basis functions of higher angular momenta.

1.4 Deficiency of the HF Method

The inherent error in HF theory is its approximation of the wavefunction as a single determinant. The latter neglects, in large part, the correlation in the electrons' motion. In the HF approach, an electron is moving in the average field of all other electrons. Nonetheless, in reality, electrons tend to avoid each other. The motion of one will affect that of the others. Electron exchange somewhat correlates the motion of two electrons of the same spin as they cannot occupy the same point in space. However, a single determinant representation of the wavefunction entirely neglects correlation between electrons of opposite spin. As a result, a calculation performed at the HF level leads to a total energy above the exact value. Nevertheless, HF based methods have been successfully applied to certain chemical problems, such as the prediction of equilibrium

geometries. For more challenging cases, such as bond dissociation energies, the single determinant HF methods have proven to be inadequate as a consequence of the neglect of electron correlation. Therefore, more sophisticated models are needed to account for correlation effects.

1.5 Electron Correlation

By convention, the correlation energy is defined as the difference between the complete basis HF and the exact (non-relativistic) energies

$$E(exact) = E(Hartree\text{-}Fock) + E(correlation) \quad . \qquad (1.18)$$

The development of accurate and economical theoretical models that incorporate electron correlation have been, and continue to be, a great challenge in computational chemistry. Ideally, a theoretical model for electron correlation should possess four important characteristics[13]. First, the model must be well defined and applicable to any nuclear arrangement and any number of electrons in a continuous manner. This suggests that no special symmetry requirements can be imposed on the choice of electron configurations. The second feature is size-consistency. This implies that application of the model to a system consisting of several molecules at infinite distance will yield properties that equal the sum of these same properties for the individual molecules at infinite distance. The third characteristic of the model is that the calculated electronic energy be variational. Finally, it should be possible to implement the requisite formulae efficiently on a computer.

Most of the HF methods satisfy these four requirements. However, practical models incorporating correlation usually do not. Typically, correlation methods are developed so as to satisfy the model requirements as closely as possible.

1.5.1 Møller-Plesset Perturbation Theory

In many-body perturbation theory, the Hamiltonian is divided into two parts

$$\boldsymbol{H} = \boldsymbol{H}_0 + \lambda \boldsymbol{V} \tag{1.19}$$

where $\boldsymbol{H}_0$ is the unperturbed Hamiltonian, $\boldsymbol{V}$ is a small perturbation and λ is an ordering parameter. Within this formulation, the exact total energy and wavefunction of a system may be expressed as a power series of perturbation according to Rayleigh-Schrödinger perturbation theory.

In the Møller-Plesset perturbation theory (MPPT) approach, H_0 is defined as the sum of Fock operators,

$$\boldsymbol{H}_0 = \sum_i \boldsymbol{F}(\boldsymbol{i}) = \sum_i \left[\boldsymbol{H}^{core}(\boldsymbol{i}) + \sum_j \mathrm{J}_j(\boldsymbol{i}) - \sum_j \mathrm{K}_j(\boldsymbol{i}) \right], \tag{1.20}$$

where J and K are, respectively, the Coulomb and exchange operators. The perturbation, $\lambda \boldsymbol{V}$, is then equal to

$$\lambda \mathrm{V} = \sum_{i=1}^{n} \sum_{j>i}^{n} \frac{1}{\boldsymbol{r}_{ij}} - \sum_i \sum_j \left(\mathrm{J}_j(\boldsymbol{i}) - \mathrm{K}_j(\boldsymbol{i}) \right). \tag{1.21}$$

By setting λ to one, the practical MPPT method can then be formulated by truncation of the power series. By convention, the total energies correct to a given order are denoted as MP*n*, where *n* is the highest-order energy term allowed. Thus, truncation after second order is MP2, truncation after third order is MP3, and so on. It can be easily

shown that the HF energy is correct to first order, hence, correlation contributions to the energy begin in the second order. Since the HF wavefunction has zero Hamiltonian matrix elements with single excitations by virtue of Brillouin's theorem, and the Hamiltonian contains only one and two-electron terms, the most important electron correlation contributions are due to doubly excited determinants. The second order Møller-Plesset (MP2) approximation is perhaps the most widely used electron correlation method. It has been found that, on average, MP2 recovers roughly 80% of the correlation energy per electron pair. The MP approach has the advantage of being size consistent, but it is not variational.

1.5.2 Configuration Interaction Methods

Another prominent class of electron correlation methods is based on configuration interaction (CI). The CI model improves the HF solutions with a linear combination of configurations obtained by replacing occupied orbitals in the HF determinant with unoccupied, or virtual, orbitals obtained from the same HF calculation. If all the possible excited configurations are included (full CI), the method yields the exact solution within the space spanned by the finite basis set employed. However, the number of configurations in a full CI expansion grows exponentially with system size, hence practical approaches often truncate the CI expansions to a given level of substitutions. The most popular model is configuration interaction restricted to single and double excitations (CISD). CI methods are variational. Unfortunately, truncated CI methods lack size-consistency. Various modified truncated CI methods which are size-consistent

have been developed by various groups. An example is quadratic configuration interaction (QCI) introduced by Pople and coworkers [14]. The QCISD method is an electron correlation model closely connected to the Coupled Cluster Approximation. The QCI with single and double excitations (QCISD) is an approximation to CCSD [15]. The QCISD(T) includes a non-iterative triple excitation is an approximation to CCSD(T). The details of these approximations and their inter-relationships have been discussed extensively [16-19]. The above methods form a well-defined hierarchy of size-consistent correlation treatments. They systematically approach the exact Schrödinger solution. This hierarchy extends in two perpendicular directions, conveniently represented on a two-dimensional chart as shown in Table 1.1. Each intersection of a row and a column represents a well-defined theoretical model chemistry. The horizontal axis contains one particle basis sets of increasing quality and following the vertical axis downward increases the level of correlation treatment, so that towards the lower right corner, one approaches the exact solution of the Schrödinger equation.

Table 1.1 A hierarchy of conventional *ab initio* methods.

Correlation	basis set →					
↓	STO-3G	3-21G	6-31G	6-31G*	.....	complete
HF	HF/STO-3G					HF limit
MP2						
QCISD						
.....						
Full CI						Exact

1.6 Computational cost

A practical concern when applying a quantum mechanical method to a chemical problem is its computational cost. This is usually related to the size of the system under investigation. The computational requirements for the simplest *ab initio* method (the HF approximation) increases formally as the fourth power with the number of basis functions. This scaling reflects the total number of two-electron integrals ($\mu\nu \mid \lambda\sigma$) (equations 1.11) over the basis functions that must, in principle, be evaluated. For a two-electron integral to have non-zero value, each of the atomic orbital pairs, on centers 1 and 2, $\mu\nu$, and on centers 3 and 4, $\lambda\sigma$, must possess an appreciable overlap. An atomic orbital only has an appreciable overlap with atomic orbitals in its vicinity. When the molecule gets sufficiently large, increasing the number of atoms will not affect the number of significant overlaps for that particular atomic orbital. The newly added atoms are simply too far away. Each atomic orbital has a limited number of significant overlaps with others atomic orbitals. As more and more atoms are added to that molecule, the number of non-vanishing atomic orbital pairs, $\mu\nu$, increases only with the number of atomic orbitals, N. Note that the non-vanishing orbital pairs, $\mu\nu$ and $\lambda\sigma$, need not have any appreciable overlap between them. Their interactions die off very slowly (as r^{-1}). Therefore, applying an efficient cutoff scheme, for a sufficiently large molecule, the number of non-negligible two-electron integrals grows as N^2. HF calculations are thus feasible for relatively large molecules.

Upon going to correlated methods, the scaling is much less favorable. For example, using a fixed basis set on molecules of size M, the computational requirements for MP2 and QCISD, scale as M^5 and M^6, respectively. Inclusion of correlation typically

increases computational expenses significantly. This limits highly correlated calculations to rather small molecules. Even with the localized version of the MP2 approach (LMP2) [20, 21], the CPU time scales as M^3 for mid- to large-size systems. An alternative to such correlated *ab initio* approaches is density functional theory.

1.7 Density Functional Theory

The above electron correlation treatments are based on approximating the many-electron wavefunction. Density functional theory (DFT) allows one to forego the need to approximate wavefunctions. Instead, a much simpler quantity, the electron density, is sought. Electron density is a function of only 3 variables, whereas the n-electron wavefunction is a function of $3n$ variables. Consequently, DFT has the potential to greatly simplify electronic structure calculations.

The basic tenet of modern DFT methods is the 1964 theorem of Hohenberg and Kohn (HK) [22]. It states that the exact ground-state molecular energy is a functional only of the electron density, ρ, and the fixed position of the nuclei. They also proved that this approach is both exact and variational, in that any other trial density, ρ', will lead to a higher energy,

$$E[\rho'] \geq E[\rho] \quad . \tag{1.22}$$

Unfortunately, the theorem only applies to the molecular ground state. More importantly, the exact form of the functional is not known.

A major development in the application of DFT is attributed to the work of Kohn and Sham (KS) in 1965 [23]. Their ingenious reformulation of the HK model laid the

foundation for modern DFT based methods. In the KS approach, the total electronic energy functional is partitioned into three parts,

$$E[\rho(r)] = T[\rho(r)] + V[\rho(r)] + E_{xc}[\rho(r)] \tag{1.23}$$

where the three terms to the right are, respectively, the kinetic energy of a non-interacting system of electrons, the classical Coulomb energy, and the remaining effects of exchange, correlation, and the error in the kinetic energy provided by the first term. The density of the system of non-interacting electrons is defined as being equal to the density of the real system of interacting electrons. Therefore, ρ is given by,

$$\rho(r) = \sum_{i}^{occ} |\psi_i(r)|^2 . \tag{1.24}$$

By expressing the density as the sum of the square of the occupied KS orbitals, the total energy of the system can then be obtained by variational minimization of the energy functional, subject to the constraint that the density is normalized to the total number of electrons, n,

$$\int \rho(r) = n \qquad . \tag{1.25}$$

The above procedure leads to a set of effective one-electron Schrödinger equations, commonly referred to as the Kohn-Sham (KS) equations,

$$H_{KS}\psi_i(r) = \varepsilon_i \psi_i(r) \tag{1.26}$$

The KS operator, H_{KS}, has the form (in atomic units)

$$H_{KS} = -\frac{\nabla^2}{2} - \sum_{A}^{nuclei} \frac{Z_A}{|r - R_A|} + \int \frac{\rho(r')}{|r - r'|} dr' + v_{xc}(r) . \tag{1.27}$$

This is analogous to the Fock operator in equation 1.9. The only difference is that the exchange term is replaced by the exchange-correlation (XC) term, $v_{xc}(r)$.

$v_{xc}(\boldsymbol{r})$ is the XC potential. It is the functional derivative of $E_{xc}[\rho(\boldsymbol{r})]$

$$v_{xc}(\boldsymbol{r}) = \frac{\delta E_{xc}[\rho(\boldsymbol{r})]}{\delta \rho(\boldsymbol{r})} \quad . \tag{1.28}$$

It is important to note that the KS orbital, ψ_i, has no physical significance other than in allowing the exact electron density, ρ, to be expressed as equation 1.24. Also, the corresponding KS orbital energy, ε_i, should not be confused with the HF molecular-orbital energies. No Koopmans' theorem [24]relates their values to approximate ionization energies. However, one can deduce (according to Parr and Yang [25]) from the exact long-range behaviour of the exact electron density, $\rho \sim \exp[-2(2I_{min})^{1/2} r]$, where I_{min} is that exact first ionization potential, that the ionization potential energy is equal to the negative value of the highest occupied KS orbital energy. Further, Janak [26] proved that the derivative of the total energy with respect to the i^{th} orbital occupation number is equal to the corresponding ε_i.

In principle, the exact ground state energy of the interacting system can be obtained from the KS formalism, if the true XC energy functional is known. Unfortunately, the exact XC functional is unknown. Hence, the solution of the KS equations requires the use of an approximate form of the true, unknown, XC functional.

1.7.1 Exchange and Correlation Functionals

The simplest and most popular approximation is the "local spin density approximation" (LSDA) [23], which has the form

$$E_{XC}^{LSDA} = \int \rho(\boldsymbol{r}) \varepsilon_{XC}[\rho^{\alpha}(\boldsymbol{r}), \rho^{\beta}(\boldsymbol{r})] d\boldsymbol{r} \tag{1.29}$$

where ε_{XC} is the exchange-correlation energy density at a point $\boldsymbol{r}$ in space, with spin-densities ρ^{α} and ρ^{β} ($\rho = \rho^{\alpha} + \rho^{\beta}$). In the LSDA, the value of ε_{XC} at any point is equal to that of a homogenous electron gas with the same spin densities ρ^{α} and ρ^{β}. Note that, the LSDA does not approximate the heterogeneous distribution of electrons in a real system by a homogeneous distribution. Rather, each point in the real heterogeneous system has the same ε_{XC} as its corresponding homogenous electron gas.

The exchange-correlation energy densities, ε_{XC}, of homogeneous electron gases have been determined accurately by quantum Monte Carlo (MC) simulations [27]. The values obtained from these quantum MC simulations have subsequently been parameterized by Vosko, Wilk, and Nusair (VWN) [28], as well as by Perdew and Zunger (PZ) [29].

It has been found that the LSDA generally gives good molecular geometries and vibrational frequencies [30-33] in strongly bound systems. It does not perform well, however, in thermochemical applications [30, 32, 34, 35]. In particular, LSDA binding energies are systematically overestimated. For weakly bound systems, this tendency to overestimate binding can lead to qualitatively wrong molecular structures.

Improvement upon the LSDA involves the introduction of the gradient of the spin densities, $\nabla\rho^{\alpha}$ and $\nabla\rho^{\beta}$. Unfortunately, the simplicity of the LSDA is lost once

gradient-corrected functional is used, and there seems to be no clear consensus as to what is the best gradient-corrected exchange-correlation functional. Most of the gradient-corrected functionals tend to deal with either the exchange or correlation component only. Popular gradient-corrected functionals for exchange are those of Perdew and Wang (PW) [36] and Becke (B) [37]. For correlation, the functionals of Lee, Yang, and Parr (LYP) [38] and Perdew (P) [39] are popular. These exchange and correlation functionals are mixed and matched to produce a variety of gradient-corrected XC functionals, collectively known as generalized gradient approximations (GGAs). The pair-distribution function can also be used to correct the weaknesses in the LSDA. This approach is adopted independently by Perdew and Wang [40] and Proynov and Salahub [41].

GGAs have been quite successful in thermochemical applications [30, 34, 35, 42]. They also perform well in studies of the energetics and structures of hydrogen-bonded systems [43]. However, for properties that are well described within the LSDA, such as molecular geometries and vibrational frequencies in strongly bound systems, GGAs offer no significant improvement [30].

Both the LSDA and the GGA XC models are based solely on local features of the density. Only the local density and local density gradients are required at each reference point. These local approximations accurately describe XC holes at short-range. Therefore, for special situations, where the exchange-correlation hole is highly delocalized, XC holes will be misrepresented by localized models.

Recently, a different class of functionals which incorporate HF and GGA theories with precision surpassing that of pure GGAs have been proposed. These "hybrid"

functionals were first formulated by Becke [44]. They are based on the "adiabatic connection" formula [45-48] where the exchange and correlation functional is conveniently expressed as:

$$E_{XC} = \int_0^1 h_{xc}^{\lambda} d\lambda \ . \tag{1.30}$$

where λ is a coupling-strength parameter that smoothly turns on inter- electronic Coulomb repulsion between electrons, and h_{xc}^{λ} is the exchange and correlation potential energy at intermediate coupling strength λ. Equation 1.30 literally "connects" the non-interacting KS reference system (λ=0) with the fully interacting real system (λ=1) through a continuum of partially interacting systems $(0 \le \lambda \le 1)$. At $\lambda = 0$, h_{xc}^{0} corresponds to the exact exchange energy (E_x) of the KS orbitals, with no correlation whatsoever. Equation 1.30 shows that the inclusion of exact exchange energy in the approximation of the exchange-correlation energy is important. Based on this, Becke devise a "half-and-half" method where the exchange-correlation energy is approximated as the sum of exact energy and the LSDA approximation of the exchange-correlation potential energy,

$$E_{xc} \cong \tfrac{1}{2} E_x + \tfrac{1}{2} h_{xc}^{LSDA} \ . \tag{1.31}$$

Another possible strategy to improve GGA is thus to graft exact exchange character into the GGA functionals. The hybrid functionals proposed by Becke [44] have the following form

$$E_{xc} = E_{xc}^{LSDA} + a_0 (E_X^{exact} - E_X^{LSDA}) + a_X \Delta E_X^{B88} + a_C \Delta E_C^{PW91} \ . \tag{1.32}$$

where a_0, a_X, a_C are semi-empirical coefficients determined by fitting to experimental data. E_X^{exact} is the "exact" exchange energy, i.e., the exchange energy one would get from

a HF calculation with the Slater determinant constructed from with the KS orbitals. ΔE_X^{B88} is Becke's 1988 [37] gradient corrected exchange, and ΔE_C^{PW91} is the 1991 gradient corrected correlation functional of Perdew and Wang [49]. Notice that coefficient a_0 reflects the importance of nonlocality in the real exchange hole.

These hybrid functionals provide better results than GGAs. In particular, the geometries, heats of reaction, and barrier heights are better than in MP2. However, the computational cost of these functional is higher than the GGA due to the addition of the HF exchange term.

1.8 The Linear Combination of Gaussian-type Orbitals Approach

The most popular approach to solving the KS equations is the linear combination of Gaussian-type orbitals (LCGTO) approach. In this approach, the KS orbitals are expanded in terms of K atom-centered contracted Gaussian functions, $\chi_\nu(\boldsymbol{r})$,

$$\psi_i = \sum_{\nu}^{K} C_{\nu i}\chi_\nu(\boldsymbol{r}) \quad . \tag{1.33}$$

In the above, $C_{\nu i}$ is the set of KS orbital expansion coefficients for the i^{th} KS orbital.

The expansion coefficients are obtained by substituting equation 1.33 into equation 1.26 and applying the variational principle, as in HF theory. This leads to the KS equations,

$$(H_{\mu\nu} - \varepsilon_i S_{\mu\nu})C_{\nu i} = 0 \quad . \tag{1.34}$$

The KS matrix elements, $H_{\mu\nu}$, are given by

$$H_{\mu\nu} = \int \chi_\mu(\boldsymbol{r}) \left\{ \frac{-\nabla^2}{2} - \sum_A^{nuclei} \frac{Z_A}{|\boldsymbol{r} - \boldsymbol{R}_A|} + \int \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' + v_{xc}(\boldsymbol{r}) \right\} \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \qquad (1.35)$$

and the overlap matrix elements, $S_{\mu\nu}$, are given by

$$S_{\mu\nu} = \int \chi_\mu(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \quad . \qquad (1.36)$$

The density ρ is expressed as

$$\rho(\boldsymbol{r}) = \sum_\mu^K \sum_\nu^K \boldsymbol{P}_{\mu\nu} \chi_\mu(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) \,. \qquad (1.37)$$

In the above, $\boldsymbol{P}$ is the density matrix. Its elements are given by

$$\boldsymbol{P}_{\mu\nu} = 2 \sum_i^{\frac{N}{2}} C_{\mu i} C_{\nu i} \,. \qquad (1.38)$$

1.8.1 Fitting the Electronic Density

The evaluation of the Coulomb term in the KS matrix element, $\boldsymbol{H}_{\mu\nu}$, involves the computation of four-centered two-electron integrals, as in HF theory,

$$\int \chi_\mu(\boldsymbol{r}) \left\{ \int \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' \right\} \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} = \sum_\sigma^K \sum_\lambda^K \boldsymbol{P}_{\sigma\lambda} \iint \frac{\chi_\mu(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) \chi_\sigma(\boldsymbol{r}') \chi_\lambda(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \qquad (1.39)$$

The computational cost of this term formally scales as K^4. One approach to easing the computational burden is the use of a fitted density. This was suggested by Sambe and Felton [50]. In this approach, the electronic density is expanded within an auxiliary basis of L atom-centered uncontracted Gaussians, $\{\chi'_k\}$

$$\rho(\boldsymbol{r}) \approx \tilde{\rho}(\boldsymbol{r}) = \sum_k^L c_k \chi'_k(\boldsymbol{r}) \qquad (1.40)$$

The exponents of the auxiliary basis are roughly double those of the orbital basis since the density is defined as the sum of the square of the occupied orbitals. The number of functions used to fit the density is roughly twice the number of contracted basis functions in the orbital basis.

Following the lead of Dunlap [51], the density fitting coefficients, c_k, are obtained by minimizing the Coulomb self-energy of the error in the density.

$$\iint \frac{[\rho(\boldsymbol{r}) - \tilde{\rho}(\boldsymbol{r})][\rho(\boldsymbol{r}') - \tilde{\rho}(\boldsymbol{r}')]}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \quad . \tag{1.41}$$

subject to the constraint that $\tilde{\rho}(\boldsymbol{r})$ is normalized to the total number of electrons in the system, n,

$$\int \tilde{\rho}(\boldsymbol{r}) = n \quad . \tag{1.42}$$

This yields the following equation for fit coefficient c_k,

$$c_k = \sum_j^L \left(S'^{-1}\right)_{kj} [t_j + \Lambda m_j], \tag{1.43}$$

where S'_{kj} is given by

$$S'_{kj} = \iint \frac{\chi'_k(\boldsymbol{r}) \chi'_j(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \quad . \tag{1.44}$$

t_j is given by

$$t_j = \iint \frac{\chi'_k(\boldsymbol{r}) \rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' = \sum_\mu \sum_\nu P_{\mu\nu} \iint \frac{\chi'_k(\boldsymbol{r}) \chi_\mu(\boldsymbol{r}') \chi_\nu(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}', \tag{1.45}$$

and n_j, is given by

$$m_j = \int \chi'_j(\boldsymbol{r}) d\boldsymbol{r} \, . \tag{1.46}$$

The Lagrange multiplier, Λ, guarantees charge conservation [51], and is given by

$$\Lambda = n - \frac{\boldsymbol{mS}'^{-1}\boldsymbol{t}}{\boldsymbol{mS}'^{-1}\boldsymbol{m}}. \tag{1.47}$$

The most time consuming part of the fitting procedure involves the evaluation of the three-centered two-electron integrals required of $\boldsymbol{t}$. The CPU cost for this step scale as K^2L. Since the number of fitting functions, L, roughly equals 2K, the scaling goes as K^3. With a fitted $\tilde{\rho}(\boldsymbol{r})$, the assembly of the Coulomb repulsion component of the KS matrix

$$\begin{aligned} \int \chi_\mu(\boldsymbol{r})\left[\int \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}'\right]\chi_\nu(\boldsymbol{r})d\boldsymbol{r} &\approx \int \chi_\mu(\boldsymbol{r})\left[\int \frac{\tilde{\rho}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}'\right]\chi_\nu(\boldsymbol{r})d\boldsymbol{r} \\ &\approx \sum_k^L c_k \iint \frac{\chi_\mu(\boldsymbol{r})\chi_\nu(\boldsymbol{r})\chi_k(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}d\boldsymbol{r}' \end{aligned} \tag{1.48}$$

is reduce from a K^4 procedure to a K^3 procedure. When handling either a small or a mid-size system, a fitted density approach offers significant savings in CPU time over a non-fitting scheme. However, for a large enough system, an efficient cutoff scheme reduced the construction of the KS matrix to a K^2 procedure, whether a fitted density is used or not. As stated earlier (see section 1.6), a significant contribution by a two-electron integral occurs only when the atomic orbital (AO) pairs, $\chi_\mu(\boldsymbol{r})$ and $\chi_\nu(\boldsymbol{r})$, as well as $\chi_\sigma(\boldsymbol{r})$ and $\chi_\lambda(\boldsymbol{r})$, have non-negligible overlaps. When the system size get large enough, each AO makes a fixed number of non-vanishing overlaps with others AOs. Consequently, the number of non-negligible AO pairs grows linearly with system size, or as K. The number of non-vanishing two-electron integrals thus grows as K^2. With a fitted density, the number of non-vanishing integrals grows as KL. Since L is proportional to K, we again have K^2 scaling. However, a fitting procedure requires a matrix inversion. This step scales as K^3. When the system size gets large enough, this

step will dominate the entire CPU time. Use of a fitting procedure will thus dramatically increase the required CPU time, and it would be best to adopt an approach that foregoes fitting the density altogether.

1.8.2 Fitting the Exchange-Correlation Potential

The exchange and correlation integrals of the KS matrix are given by

$$\int \chi_\mu(\boldsymbol{r}) v_{xc}(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \tag{1.49}$$

where the XC potential, $v_{xc}(\boldsymbol{r})$, is defined as

$$v_{xc}(\boldsymbol{r}) = \frac{\delta E_{xc}[\rho(\boldsymbol{r})]}{\delta \rho(\boldsymbol{r})} \tag{1.50}$$

In general, these integrals can not be evaluated analytically because of the complicated forms used for the approximate XC functionals. Hence numerical quadrature must be used.

Typically, the integration of the XC term involves several steps. First, a grid is generated to cover the entire molecular system. $v_{xc}(\boldsymbol{r})$ must then be evaluated at each grid point. Evaluation of $v_{xc}(\boldsymbol{r})$ requires taking the functional derivative of $E_{xc}[\rho(\boldsymbol{r})]$ (equation 1.50). A GGA functional includes the first derivative of $\rho(\boldsymbol{r})$ in the exchange-correlation expression. Therefore, computing gradient-correlated exchange-correlation potentials would involve second derivatives of $\rho(\boldsymbol{r})$. In a general unrestricted KS calculation, a GGA requires both $v_{xc}^{\alpha}(\boldsymbol{r})$ and $v_{xc}^{\beta}(\boldsymbol{r})$. Therefore, ρ^{α} and ρ^{β} must be

synthesized at each point, as well as each of the three first derivatives and six unique second derivatives of ρ^{α} and ρ^{β} with respect to x, y, and z.

Once the XC potential has been evaluated at each grid point, a direct approach is to numerically integrate the XC integrals,

$$\int \chi_{\mu}(\boldsymbol{r}) v_{xc}(\boldsymbol{r}) \chi_{\nu}(\boldsymbol{r}) d\boldsymbol{r} \approx \sum_{l}^{grid} \chi_{\mu}(\boldsymbol{R}_l) v_{xc}(\boldsymbol{R}_l) \chi_{\nu}(\boldsymbol{R}_l) W_l \,, \tag{1.51}$$

where $\boldsymbol{R}_l$ and W_l are the coordinates and quadrature weights of the grid points. This is done in programs such as DMol [52], ADF [53], and Gaussian 94 [54].

An indirect approach to handling the XC integrals can be taken. First, the XC potential is fit by a second auxiliary basis of M atom-centered uncontracted Gaussians, $\chi_l''(\boldsymbol{r})$.

$$v_{xc}(\boldsymbol{r}) \approx \tilde{v}_{xc}(\boldsymbol{r}) = \sum_{l}^{M} b_l \chi_l''(\boldsymbol{r}) \,. \tag{1.52}$$

Typically, M is equal to the number of basis functions in the $\rho(\boldsymbol{r})$ fitting basis, and the values of the exponents in the XC fitting basis are equal to one third of those found in the $\rho(r)$ fitting basis. The reason for setting the XC exponents to one third of the $\rho(\boldsymbol{r})$ fitting basis is as follows. The exchange-correlation functional can be partitioned into $\boldsymbol{E}_x$ and $\boldsymbol{E}_c$ parts. The simplest exchange functional, $\boldsymbol{E}_x$, is the Slater X_{α} functional [55]. Its functional derivative (i.e., exchange-potential, $\boldsymbol{v}_x$) is proportional to one third of the electronic density. Since exchange is the larger component of the exchange-correlation energy, one could then conclude that $\boldsymbol{v}_{xc}$ will be more or less proportional to the electron density to the one-third power.

The fit coefficients, b_l, are determined by a numerical fitting procedure that minimizes

$$\sum_{l}^{grid}\left[v_{xc}(\boldsymbol{R}_l)-\tilde{v}_{xc}(\boldsymbol{R}_l)\right]^2 W_l \tag{1.53}$$

over the entire set of grid points.

Unlike the fit of $\rho(\boldsymbol{r})$, no constraints are imposed on the XC fit. This results in a much simpler set of equations for the fit coefficients,

$$\boldsymbol{b}_l = \sum_{m}^{M}\left(S^{-1}\right)_{lm} t_m \,, \tag{1.54}$$

where the elements of S'' are given by

$$S''_{lm} = \sum_{l}^{points} \chi_l(\boldsymbol{R}_l)\chi_m(\boldsymbol{R}_l)W_l \,. \tag{1.55}$$

and the elements of t are given by

$$t_l = \sum_{l}^{points} \chi_l(\boldsymbol{R}_l)v_{xc}(\boldsymbol{R}_l)W_l \,. \tag{1.56}$$

With $\tilde{v}_{xc}(\boldsymbol{r})$ in hand, the XC integrals of the KS matrix can be approximated as

$$\int \chi_\mu(\boldsymbol{r})v_{xc}(\boldsymbol{r})\chi_\nu(\boldsymbol{r})d\boldsymbol{r} \approx \int \chi_\mu(\boldsymbol{r})\tilde{v}_{xc}(\boldsymbol{r})\chi_\nu(\boldsymbol{r})d\boldsymbol{r} = \sum_{l}^{M} b_l \int \chi_\mu(\boldsymbol{r})\chi_\nu(\boldsymbol{r})\chi_l(\boldsymbol{r})d\boldsymbol{r} \tag{1.57}$$

which can be easily integrated analytically. Such an approach is used in programs such as deMon [56], early versions of DGauss [32], and DeFT [57].

1.9 Evaluation of Total Energy

The total energy of the LCGTO-DF method can be expressed (for a closed-shell system) as

$$E = 2\sum_{i}^{n_e}\varepsilon_i - \frac{1}{2}\int\int\frac{\rho(r)\rho(r')}{|r-r'|}drdr' + \int\rho(r)[\varepsilon_{xc}(r) - v_{xc}(r)]dr + \sum_{A}^{atoms}\sum_{B>A}^{atoms}\frac{Z_A Z_B}{R_{AB}} \quad (1.58)$$

The first term is the sum of orbital energies. Note that, the energy ε_i contains Coulomb and exchange interactions between an electron in spin-orbital *i* and electrons in all other occupied spin-orbitals (particularly, ε_j). However, ε_j includes Coulomb and exchange interactions between an electron in spin-orbital *j* and electrons in all other occupied spin-orbitals (particularly, ε_i). Therefore, when summing ε_i and ε_j together, the electron-electron interactions between spin-orbitals, *i* and *j*, are counted twice. The second term in equation 1.58 corrects for this double counting. Recall that orbital energies are obtained by solving the KS equations (equation 1.34). The KS equations contain the exchange-correlation potential, $v_{xc}(r)$. But the XC energy functional contains the exchange-correlation energy density, $\varepsilon_{xc}(r)$. Therefore, $(\varepsilon_{xc}(r) - v_{xc}(r))$ in the third term corrects for this fact. The last term is the inter-nuclear repulsion energy.

In most programs, the second and third terms are handled in the same fashion by which the Coulomb and XC terms are handled throughout the course of the SCF procedure. For DeFT, DGauss and deMon, the second term is not expressed as

$$\begin{aligned} -\frac{1}{2}\iint\frac{\rho(r)\rho(r')}{|r-r'|}drdr' &\approx -\frac{1}{2}\iint\frac{\rho(r)\tilde{\rho}(r')}{|r-r'|}drdr' \\ &= -\frac{1}{2}\sum_{\mu}^{K}\sum_{\nu}^{K}P_{\mu\nu}\sum_{k}^{L}c_k\iint\frac{\chi_\mu(r)\chi_\nu(r)\chi_k(r')}{|r-r'|}drdr'. \end{aligned} \quad (1.59)$$

The above expression would exactly eliminate the double-counting resulting from the summation of orbital energies. However, in these programs, it is instead approximated as

$$-\frac{1}{2}\iint\frac{\rho(\boldsymbol{r})\rho(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}d\boldsymbol{r}' \approx -\frac{1}{2}\iint\frac{\tilde{\rho}(\boldsymbol{r})\tilde{\rho}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}d\boldsymbol{r}'$$
$$= -\frac{1}{2}\sum_{k}^{L}\sum_{l}^{L}a_k a_l\frac{\chi_k(\boldsymbol{r})\chi_l(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}d\boldsymbol{r}d\boldsymbol{r}'. \qquad (1.60)$$

This does not correspond exactly to one half of the Coulomb energy. This substitution allows the density to be fit in a variational fashion. Variational fitting is otherwise impossible if equation 1.59 is used in the total energy expression.

The DeFT, DGauss, and deMon programs fit $\varepsilon_{xc}(\boldsymbol{r})$ with the same fitting basis as $v_{xc}(\boldsymbol{r})$,

$$\varepsilon_{xc}(\boldsymbol{r}) \approx \tilde{\varepsilon}_{xc}(\boldsymbol{r}) = \sum_{l}^{M} e_l\chi_l'(\boldsymbol{r}), \qquad (1.61)$$

where e_l is a fit coefficient for the XC energy density. The expression for the third term becomes,

$$\int\rho(\boldsymbol{r})[\varepsilon_{xc}(\boldsymbol{r})-v_{xc}(\boldsymbol{r})]d\boldsymbol{r} \approx \int\rho(\boldsymbol{r})[\tilde{\varepsilon}_{xc}(\boldsymbol{r})-\tilde{v}_{xc}(\boldsymbol{r})]d\boldsymbol{r}$$
$$= \sum_{\mu}^{K}\sum_{\nu}^{K}\boldsymbol{P}_{\mu\nu}\sum_{l}^{M}(e_l - b_l)\int\chi_\mu(\boldsymbol{r})\chi_\nu(\boldsymbol{r})\chi_l'(\boldsymbol{r})d\boldsymbol{r}. \qquad (1.62)$$

In DeFT, once the SCF is converged, equation 1.62 is modified to

$$\sum_{\mu}^{K}\sum_{\nu}^{K}\boldsymbol{P}_{\mu\nu}\sum_{l}^{grid}\chi_\mu(\boldsymbol{R}_l)\chi_\nu(\boldsymbol{R}_l)\varepsilon_{xc}(\boldsymbol{R}_l)\boldsymbol{W}_l - \sum_{\mu}^{K}\sum_{\nu}^{K}\boldsymbol{P}_{\mu\nu}\sum_{l}^{M}b_l\int\chi_\mu(\boldsymbol{r})\chi_\nu(\boldsymbol{r})\chi_l'(\boldsymbol{r})d\boldsymbol{r}\,. \qquad (1.63)$$

Rather than performing an analytical integration of the fitted XC energy density integrals, a direct numerical integration on an augmented grid is carried out instead. This reduces

the grid noise in the total energy and it allows for a reliable energy gradient expression (more on this is in the Chapter 4 dealing with the divide and conquer fit of the XC terms). Inevitably, this procedure increases the amount of CPU time required. However, it is performed only once in the whole SCF cycle, once an SCF has been achieved.

1.10 Linear Scaling Methods

In the time since Hohenberg, Kohn, and Sham's seminal work, DFT has been applied to a wide range of chemical problems. However in the past few years, there has been a surge of reports on the systematic comparison of DFT theories with experiment and conventional *ab initio* treatments. Its success in predicting physiochemical properties such as equilibrium geometries, harmonic frequencies, polarizabilities, dissociation energies and hydrogen bonding has increased DFT's status among the chemistry and biochemistry communities.

DFT methods incorporate electron correlation effects in a way that does not lead to the scaling problem of other HF-based correlation methods. Hence, it opens exciting opportunities for the study of larger and more complex molecular systems which are not feasible by other correlated *ab initio* methods.

However, realistic modeling of truly large molecular systems is still not feasible, mostly due to the unfavorable N^2 scaling with system size arising from the Coulomb problem. As mentioned earlier, the evaluation of the KS Coulomb density-density interaction requires the computation of four centered two-electron integrals. This formally leads to a N^4 scaling, where N is the number of basis functions. Using a fitted

density in the KS matrix construction reduces this to an N^3 scaling procedure. For large enough molecules, however, the number of non-negligible two-electron integrals does not grow quartically with system size. Rather, it grows quadratically. Other procedures of the DFT calculation may show a worse scaling, such as the N^3 generation of charge density via matrix diagonalization. However, this procedure has a much smaller prefactor, and it should only begin to dominate for very large systems. Nevertheless, an alternative to matrix diagonalization must be sought before tackling truly large systems.

Very recently, considerable effort has been devoted to the development of efficient algorithms in which the computational expense scales linearly with system size. The principal basis for achieving linear scaling algorithms is to limit the physical extent of the electronic degrees of freedom to a local region of space.

One of the earliest linear scaling methods is Yang's divide and conquer approach [58], where a system is partitioned into many smaller subsystems that can be treated independently. Each subsystem's density is calculated by the conventional LCAO DFT method. However, the subsystem's Hamiltonian, which includes the potential due to other subsystems, is diagonalized independently. The total density of the system is then obtained by summing all the subsystems' contributions. The effort for each subsystem calculation is roughly that of the calculation on a small molecule of the same size as the subsystem. There is never a need to diagonalize the full Hamiltonian of the large system. As the size of the system increases, each subsystem's computational effort remains essentially constant. Therefore the total computational cost rises linearly with the number of subsystems, and since the number of subsystems rises linearly with total system size, linear scaling is achieved. Divide and conquer approaches have been

successfully applied to study the electronic structure of large molecular systems by semi-empirical methods [59, 60] and crude (Harris functional [61]) DFT calculations [62].

An alternative to diagonalization is based on direct energy minimization. These methods are grouped into two categories. The first category involves searching directly for the optimal density matrix. Baroni and Giannozzi [63] developed a linear scaling method based on the finite-difference real-space discretization of the Hamiltonian and on the recursion method of Haydock, Heine and Kelley [64] to calculate the energy and electronic densities from the one-electron Green's function. Linear scaling is achieved because particular diagonal elements of the Green's function, from which the electron density is computed, can be truncated once a limited region about each point has been explored. For sufficiently large systems, this neighborhood is independent of system size.

Hernández, Gillan and Goringe [65] proposed a linear scaling scheme for density functional pseudopotential calculations. The approach determined the ground state energy by direct minimization of the density matrix using the conjugate-gradient method, subject to the constraint that the eigenvalues of the density matrix lie in the range (0,1). The limited eigenvalue range is imposed by the tight-binding method of Li, Nunes, and Vanderbilt (LNV) [66]. A density matrix, ρ, satisfies this eigenvalue range if it is idempotent (i.e. $\rho^2 = \rho$). The idempotency of ρ can be achieved by expressing it in terms of an auxiliary matrix, σ, subjected to McWeeny "purification transformation" [67]

$$\rho = 3\sigma^2 - 2\sigma^3 .$$

This purification transformation has the property that if σ is a nearly idempotent matrix, then the purified ρ is an even more nearly idempotent matrix. Linear scaling is achieved

by setting density matrix elements to zero when the separation between two basis functions exceeds a chosen cutoff.

Very recently, Millam and Scuseria [68] have proposed another direct minimization method, the conjugate gradient density matrix search (CG-DMS), applicable to *ab initio* calculations. In their approach, the density matrix search minimizes a slightly modified functional of LNV [66], subject to the constraints of the idempotency of ρ, and the correct number of electrons. At convergence, the density matrix and Fock matrix must commute. Modification was required since the original LNV method was developed with the tight-binding formulation, which usually uses an orthogonal basis. To account for the non-orthogonal basis sets commonly used in *ab initio* methods, both the density and Fock matrices are transformed into an orthonormal basis. A Cholesky decomposition of the overlap matrix and its inverse is used to transform to, and back from, an orthonormal basis. It is important to note that the CG-DMS approach alone does not yield linear scaling. The method still involves matrix operations that scale between N^2 and N^3. For large molecules, however, one can reduce the cost of the matrix operations by taking advantage of the sparsity of the matrix, which would not be possible with a standard diagonalization. CG-DMS reduces its cost by storing and computing only the significant elements. For sparse systems, the number of significant elements scales linearly with system size. Therefore, the cost of all matrix operations will also scale linearly with system size.

The second category of alternative methods involves the search for a set of localized molecular orbitals (LMO's). Stewart [69] proposed the use of LMO's in solving semi-empirical SCF equations. In this method, the initial set of LMO's

corresponds to a Lewis dot structure of the molecule. A necessary, and sufficient, condition to achieve a SCF is that the Fock matrix elements coupling the occupied and virtual orbitals must be zero, i.e.,

$$\sum_{i=1}^{n_{occ}}\sum_{j=1}^{n_{vir}}\left|\langle\phi_i \mid \boldsymbol{F} \mid \phi_j\rangle\right| = 0 .$$

This means that there is no mixing of occupied and virtual orbitals through the Fock operator once an SCF is achieved. Annihilation of Fock matrix elements connecting the occupied LMO's and virtual LMO's is done by an Eulerian rotation. This consists of mixing the occupied and virtual orbitals to generate two new LMO'S having a zero interaction between them

$$\begin{aligned}\phi_i' &= \alpha\phi_i + \beta\phi_j \\ \phi_j' &= -\beta\phi_i + \alpha\phi_j\end{aligned},$$

where α and β are defined by

$$\alpha = \sqrt{\tfrac{1}{2}(1+\boldsymbol{D})\Big/\sqrt{4\boldsymbol{F}_{ij}^2+\boldsymbol{D}^2}}$$

$$\beta = \varphi\sqrt{1-\alpha^2}$$

where $\boldsymbol{D} = \boldsymbol{F}_{ii} - \boldsymbol{F}_{jj}$ and $\varphi = 1$ if $\boldsymbol{F}_{ij}$ is negative, and $\varphi = -1$ if $\boldsymbol{F}_{ij}$ is non-negative.

As a consequence of this mixing, the number of atoms in the LMO increases rapidly throughout the course of the SCF procedure. However, many of these new atoms' contributions are very small, and their contribution can be safely deleted from the LMO. Consequently, the size of the LMOs becomes almost constant for large systems, independent of system size. Unfortunately, as it is implemented now, it can only be applied to closed-shell systems and systems that can be represented by a Lewis diagram.

As regards to the construction of the Coulomb matrix, much progress has been made recently for large systems. Near linear scaling effort in the evaluation of the Coulomb elements has been demonstrated by both the fast multipole methods [70] and the Barnes-Hut (BH) [71] tree codes. The common feature of these methods is the subdivision of the space into a three-dimensional tree-like hierarchy of cells. The charge distributions of the molecular system are then sorted into different cells. The complexity of the computation may be reduced by approximating charge distributions within a cell as a multipole expansion. Interactions between cells can be separated according to the extent of the charge distributions, i.e., according to how diffuse the charge distributions are. They are also separated into near and far field interactions. Interaction between two charge distributions is considered as far field if they do not overlap. The integrals associated with these far field charge distributions can be approximated classically through an L-order multipole expansion. No approximation can be made for charge distributions that do overlap (near field), and their integrals are evaluated explicitly. Since the multipole interactions are trivial to evaluate compared to the explicit evaluation of the density-density interactions, the computational effort depends almost entirely on the number of integrals that need to be evaluated explicitly for large systems. As system size increases, the number of near-field integrals increases linearly, and therefore, linear scaling is achieved.

1.11 Parallel Processing

Even if linear scaling is achieved for all the procedures in the QM calculations, vast amounts of wall-clock (real) time are required for calculations on truly large systems. One can only hope that calculations on such systems can be completed within a reasonable amount of wall-clock time. As the speed of single processor computers is fast approaching its physical limit, many researchers, in the pursuit of ever-increasing computing power, have turned to parallel computer architectures [72]. By harnessing the combined power of many smaller computers through parallel processing, parallel machines can easily outperform the fastest serial computers.

To understand what parallel processing is, let's consider an analogy of stacking a set of library books. One person can be hired to stack all the books into their proper places. Regardless of how efficient the worker is, he/she can not complete the task faster than a certain rate. However, this process can be sped up considerably by hiring more than one worker. One simple way to partition the task is to divide the books evenly among the workers. Each worker stacks the books one at a time. This approach may not be the most efficient, as each worker is required to walk all over the library to stack books. An alternative partitioning scheme is to assign the workers to work in different locations with equal numbers of books. Now, all he/she needs to do is to place books on shelves. He/she need not move around needlessly throughout the library. If a stray book is encountered, he/she passes it on to the worker responsible for the location in which it belongs. The second scheme requires less effort from each worker.

The book-stacking metaphor illustrates the basic principles of parallel processing. Each worker corresponds to a processing node. A worker assigned to work in a certain

location with a subset of books is an instance of task partitioning. Passing of stray books between workers is analogous to the communication of information among the processors.

Different problems have different degrees of parallelism. For some problems, assigning subtasks to other processors might be more time consuming than just simply performing the task locally. Other problems are entirely serial, such as the task of digging a pillar hole. One person requires a certain time to dig the hole. Using more people would not reduce this time. Since partitioning of this task is impossible, it is ill suited for parallel processing. Not all problems are equally amenable to parallel processing.

Multiple processors working on a problem does not reduce the total CPU time. Rather, an N processor parallel computer ideally cuts the wall-clock time by a factor of N, dramatically reducing the time one needs to wait to get back an answer. In practice, the efficiency of a parallel machine depends on the size of the problem, the architecture of the parallel computer, the number of processors, and the codes it works on.

1.12 Parallel Processing Architectures

There are two basics classes of parallel architectures [73-75], namely: single instruction multiple data (SIMD) machines and multiple instruction multiple data (MIMD) machines.

1.12.1. SIMD

A single SIMD computer typically consists of a very large number (thousands to tens of thousands) of simple processors, each with its own small local memory, all connected to a common control host processor. In SIMD computing, the same set of instructions is executed synchronously by all processors. Each processor, however, operates on a different partition of data. The advantages of the SIMD architecture are that the processors are always synchronized to the host processor system clock and that the data transfer to each processor is simultaneous. The primary limitation of a SIMD computer is that different processors can not execute different instructions on the same clock cycle. Hence it is only suited for a limited class of problems, characterized by performing the same operations over a large set of data. An example would be solving a differential equation, such as Poisson's equation, by a finite difference approach on a large three dimensional lattice.

1.12.2. MIMD

MIMD computers are characterized by each processor being capable of executing its own instructions on its own data stream, independent of all other processors. Each processor in a MIMD system has its own control unit and operating system. There are two categories of MIMD systems: shared memory and distributed memory.

A Shared-memory MIMD

All the processors in a shared memory MIMD computer have access to a common pool of memory through a shared high-speed bus. This makes modification of an existing

program to operate on such a parallel computer relatively simple and makes communication between processors easy in principle. In practice, shared memory systems have a limited number of processors, a consequence of bus contention and cache inconsistency problems. Bus contention arises when more than one processor attempts to access the same memory at the same time, forcing the other processors to wait. This can severely affect system performance. One solution to alleviate the bus bottleneck is to add a cache to each processor. Cache is a high speed memory devoted solely to that processor. The data that each processor is currently working is stored in the local cache, thus eliminating the need for main memory access. However, storing data in cache can lead to the cache coherency problem. This problem occurs when a processor updates a share variable in its cache. That processor has access to the new value stored in cache. However, the other processors may require it as well. They do not have access to that cache. They will get its value from the shared memory. Unfortunately, they are accessing the old, incorrect, data rather than the correct value stored in the processor's dedicated cache. One solution to the cache coherency problem is allowing only caches to update their values in the shared memory line. The problem of bus contention and cache coherency typically limits the shared memory MIMD computer to a peak size of sixteen to thirty-two processors.

B. Distributed-memory MIMD

Each processor in a distributed memory MIMD computer is essentially a complete computer, with its own local memory. These processors are connected through a communications network that supports passing messages between individual

processors. Since distributed memory systems have no memory bus contention or cache coherency problems, they are typically comprised of a very large number of processors. Though distributed memory systems have no bus contention and cache coherency problems (nothing is shared), they often can experience a network log jam due to excessive inter-processor message passing. It is also more difficult to program for distributed memory systems than shared-memory systems. The programmer must be responsible for the managing and partitioning of data across the distributed memory and the communication between the processors. If any single piece of data is required by a processor, and it is stored in another processor's memory, a message carrying this piece of data must be passed from the latter to the former. The programmer must foresee all such situations and include the requisite message passing into his/her code.

1.13 Parallel Performance

The fundamental measurement of the performance of a parallel application is the speedup it achieves relative to a calculation on a single processor. Speedup is defined as the ratio of the execution time on one node, T_1, to the ratio of the time on multiple (P) nodes, T_p.

$$\text{speedup(P)} = \frac{T_1}{T_P}$$

Ideally, the more processors applied to a perfectly parallelized problem, the more execution time is reduced. In other words, a speedup of P is achieved with P processors. However, we know intuitively that adding more and more processors, so that less and less time is required to execute a problem, can not continue indefinitely. At some point, time

wasted due to inefficiencies in the hardware will become greater than the time spent doing useful work. In parallel processing, more is not necessarily better, as evidenced by Amdahl's law [76].

1.13.1 Amdahl's Law

For a given problem, Amdahl's law says that there is an upper limit to the speedup that can be achieved, regardless of the number of processors. The argument is as follows. The total execution time for a problem on a single processor, T_1, can be divided into two parts. One part of the problem is parallelizable. Adding more processors will indeed speed up the calculation within this section of the code. T_p is the time spent running this part of the program when on a single processor. There is the other part of the problem where more processors will not reduce the wall clock time. This is the nonparallelizable, or inherently sequential, part of the problem. The time spent performing this part of the problem is denoted as T_s. On a single processor, the total wall clock time is $T_1 = T_s + T_p$. On a parallel machine with P processors, the wall clock time is $T_{total} = T_s + \frac{T_p}{P}$.

Figure 1.1: Speedup versus the number of processors. Each curve in this family of curves represents a different percentage of the code that runs in parallel.

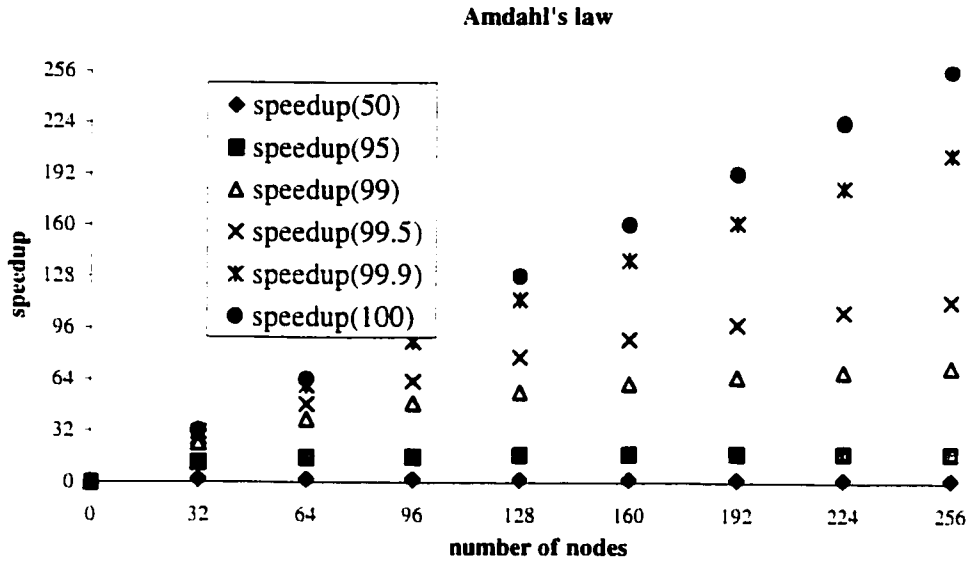


Figure 1.1 shows how the overall speedup of a program can be affected by the fraction of execution time that is parallelizable. Note that even a small amount of serial execution time can severely reduce efficiency when running on a large number of processors. For example, on a 256 processor machine, a mere 1% inefficiency in parallelizability will reduce the speedup from (its theoretical limit of) 256 all the way down to roughly 64; we would only be using this machine at 25% of its capacity. 75% of the machine is being wasted.

The assumption made implicitly in Amdahl's law is that the computational load is static and may be balanced perfectly. This is actually a "best case scenario". The load balance of a parallel computation is a measure of how closely all concurrent tasks will finish in time. A parallel application is perfectly load balanced if all processors terminate

at exactly the same time. If a parallel computation is load imbalanced, it further reduces the observed speedup. Although it is not so costly if one, or a few, processors has less work than most, it can be disastrous if one, or a few, requires a significantly longer time to complete its calculations. This is a serious flaw since the vast majority of the processors must sit idle, awaiting this small number of processors to "catch up" to them in the calculation. Even with perfect load balancing, parallel performance would be poor if too much time is spent communicating data between processors rather than performing useful computations on these processors. Communication costs limit the number of processors that can be effectively used in a parallel computation. As more processors are used to solve a fixed size problem, less work is available for each processor. However, the communications at best remain fixed. In some cases, it increases. And since a single communications operation is thousands of times more expensive (time consuming) than a single processor operation, it is important to increase computational efficiency by maximizing processor use while minimizing inter-processor communication.

A concept closely related to load balance is the granularity of a parallel application [77, 78]. The granularity of a problem is a relative term. It is generally refers to the relationship between the amount of time spent in communication (or synchronization) versus the time spent in computation. Granularity varies widely between the parallel computer architecture and the applications. Generally, an application is classified as fine grain parallelism if the problem has been divided into the smallest possible tasks, with each task performed by a processor. If each processor works on the largest possible component of a problem, this approach is classified as coarse grain parallelism.

The difference between coarse and fine grain parallelism can best be illustrated with an analogy. Suppose ten carts have to be delivered to a warehouse, and there are ten dogs in service. Coarse grain parallelism is assigning each dog a cart, and all ten carts are pulled simultaneously. Fine grain parallelism is assigning all ten dogs to pull the same cart, and all ten carts are pulled in succession. In an ideal situation, if all the carts are equivalent, and all the dogs are identical, the total execution time for both the coarse and fine grain approaches would be the same. However, if all carts have different weights or some dogs are stronger, fine grain parallelism would be the better approach. If there are a large number of dogs (say a thousand) available, fine-grain parallelism will not be efficient, because of the great difficulty in getting them to work in synchronicity. But, fine and coarse grain parallelism are not necessary mutually exclusive. One could partition the load using a so-called combined coarse and fine grain approach. Suppose there are twenty dogs and two carts. A trivial combined coarse/fine grain approach is assigning ten dogs to a cart and both carts are pull concurrently. Lets suppose that a cart is three times heavier than the other. An optimal combined coarse/fine grain approach is assigning three times more dogs to the heavier cart, i.e., the heavier cart is pulled by 15 dogs while the other is pulled by 5 dogs.

In designing a parallel application, it is important to make sure that the granularity of the algorithm is consistent with the granularity of the hardware. For example, if the hardware communication rate is much slower than the computation, fine grain parallelism will not work well. In general, there is a peak size of processors for any fixed size problem. The coarser the granularity of a problem is, the larger this optimal number is

likely to be. However, many problems cannot be partitioned in a coarse grain fashion. They must be done at a fine grain level.

1.14 Message Passing Paradigm

A dominating concern in the development of a parallel program is the choice of communication paradigms and specific programming protocols. The main criterion for choosing a parallel paradigm is portability. Portability refers to writing a program in such a way that it may be easily recompiled and run on other machines. In the last few years, a number of portable parallel communication systems based on the popular message-passing paradigms have been developed. An application in a message-passing model can be viewed as a collection of processes with private local data, where each process, identified by a unique name, interacts with others by sending and receiving messages to and from other named processes.

The first widely used portable message-passing system was the Parallel Virtual Machine (PVM) [79]. The concept behind PVM is to link separate hosts, possibly of different types, to create a single so-called "virtual machine". In other words, PVM allows one to take virtually any network of heterogeneous UNIX-based computers and treat them as a single parallel computer. The major deficiency of PVM is its rather limited message-passing subset, providing some basic send/receive operations and some simple collective communications. PVM does not possess the rich set of features that more formal message passing systems, such as the Message Passing Interface (MPI) [80], provide.

MPI is fast becoming the standard for the message passing paradigm in the parallel computing world. MPI differs from other message passing paradigms in that it is a message passing interface and not a complete parallel computing programming environment. This is not a problem for people who work on one machine. However, it can cause problems for people who attempt to work on a variety of platforms, because the user must learn and remember each vendor's particular implementation to configure and start a parallel program. This can be alleviated partly by layering MPI over PVM, which provides such an environment. There is a number of such implementations of MPI, such as CHIMP [81], MPICH [82] and LAM [83]. Each implementation of MPI requires a different specification of the environment, and each has different procedures for submitting and running jobs. In most practical MPI implementations, a fixed set of processes, one for each processor, is created at program initialization. Processes are not allowed to either be created or be destroyed during program execution.

MPI provides a unique feature that supports modular programming through a mechanism known as a communicator. It enables the MPI programmer to define modules that encapsulate internal communication structures. A communicator is composed of process groups and a communication context. Process groups allow a subset of processes to communicate among themselves using local process identifiers. They can also be used to specify which processes are involved in a collective communication or to enable separate groups of processes to work on separate tasks. The communication context provides an additional criterion for a message selection. A receive operation can only receive a message that was sent in the same context. Since

each internal communication is associated with a unique context, there is no danger of mixing up the internal communications by different program modules.

The MPI standard in its entirety has about 125 functions, but only six functions are absolutely needed in every MPI program. With only six functions, as shown in Table 1.2, many useful and efficient programs can be written. The other functions add flexibility (data-type), robustness (non-blocking send/receive), efficiency ("ready" mode), modularity (groups, communicators) and convenience (collective operations, topologies). These can all be forgone in an MPI program.

Table 1.2 The six-function version of MPI.

MPI_INIT	Initialize an MPI computation
MPI_COMM_SIZE	Determine number of processes
MPI_COMM_RANK	Determine my process identifier
MPI_SEND	Send a message
MPI_RECV	Receive a message
MPI_FINALIZE	Terminate a computation

References

(1) E. Schrödinger, *Ann. Physik* **79**, 361 (1926).

(2) M. Born and J. P. Oppenheimer, *Ann. Phys.* **79**, 361 (1927).

(3) D. R. Hartree, *Proc. Cambridge Philos. Soc.* **24**, 328 (1928).

(4) V. A. Fock, *Z. Phys.* **15**, 126 (1930).

(5) J. C. Slater, *Phys. Rev.* **35**, 509 (1930).

(6) J. C. Slater, *Phys. Rev.* **34**, 1293 (1929).

(7) C. C. J. Roothaan, *Rev. Mod. Phys.* **23**, 69 (1951).

(8) G. G. Hall, *Proc. Roy. Soc. (Lodon)* **A205**, 541 (1951).

(9) J. C. Slater, *Phys. Rev.* **36**, 57 (1930).

(10) S. F. Boys, *Proc. Roy. Soc. (London)* **A200**, 542 (1950).

(11) J. S. Binkley, J. A. Pople and W. J. Hehre, *J. Am. Chem. Soc.* **102**, 939 (1980).

(12) M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro and W. J. Hehre, *J. Am. Chem. Soc.* **104**, 2797 (1982).

(13) W. J. Hehre, L. Radom, P. v. R. Schleyer and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley-Interscience: New York; 1986).

(14) J. A. Pople, M. Head-Gordon and K. Raghavachari, *J. Chem. Phys.* **87**, 5968 (1987).

(15) R. J. Barlett, *J. Phys. Chem* **93**, 1697 (1989).

(16) J. Paldus, J. Cizek and B. Jeziorski, *J. Chem. Phys.* **90**, 4356 (1989).

(17) J. A. Pople, M. Head-Gordon and K. Raghavachari, *J. Chem. Phys.* **90**, 4635 (1989).

(18) J. Paldus, J. Cizek and B. Jeziorski, *J. Chem. Phys.* **93**, 1485 (1990).

(19) K. Raghavachari, M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* **93**, 1486 (1990).

(20) P. Pulay and S. Saebo, *J. Chem. Phys.* **86**, 1901 (1984).

(21) R. B. Murphy, M. D. Beachy, R. A. Friesner and M. N. Ringnalda, *J. Chem. Phys.* **103**, 1481 (1995).

(22) P. Hohenberg and W. Kohn, *Phys. Rev. B* **136**, 864 (1964).

(23) W. Kohn and L. J. Sham, *Phys. Rev. A* **140**, 1133 (1965).

(24) T. A. Koopmans, *Physica* **1**, 104 (1933).

(25) R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules* (Oxford University Press; New York; 1989).

(26) J. F. Janak, *Phys. Rev. B* **18**, 7165 (1978).

(27) D. M. Ceperley and B. J. Alder, *Phys. Rev. Lett.* **45**, 566 (1980).

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

(29) J. P. Perdew and A. Zunger, *Phys. Rev. B* **23**, 5048 (1981).

(30) B. G. Johnson, P. M. W. Gill and J. A. Pople, *J. Chem. Phys.* **98**, 5612 (1993).

(31) R. M. Dickson and A. D. Becke, *J. Chem. Phys.* **99**, 3898 (1993).

(32) J. Andzelm and E. Wimmer, *J. Chem. Phys.* **96**, 1280 (1992).

(33) I. Papai, A. St-Amant, J. Ushio and D. Salahub, *Int. J. Quantum Chem., Quantum Chem. Symp.* **24**, 29 (1990).

(34) A. D. Becke, *J. Chem. Phys.* **96**, 2155 (1992).

(35) A. D. Becke, *J. Chem. Phys.* **97**, 9173 (1992).

(36) J. P. Perdew and Y. Wang, *Phys. Rev. B* **33**, 8800 (1986).

(37) A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

(38) C. Lee, W. Yang and R. G. Parr, *Phys. Rev. B* **37**, 785 (1988).

(39) J. P. Perdew, *Phys. Rev. B* **33**, 8822 (1986).

(40) J. P. Perdew and Y. Wang, *Phys. Rev. B* **46**, 12947 (1992).

(41) E. I. Proynov and D. R. Salahub, *Phys. Rev. B* **49**, 7874 (1994).

(42) L. Fan and T. Ziegler, *J. Am. Chem. Soc.* **114**, 10890 (1992).

(43) F. Sim, A. St-Amant, I. Papai and D. R. Salahub, *J. Am. Chem. Soc* **114**, 4391 (1992).

(44) A. D. Becke, *J. Chem. Phys.* **98**, 5648 (1993).

(45) J. Harris and R. O. Jones, *J. Phys. F* **4**, 1170 (1974).

(46) O. Gunnarsson and B. I. Lundqvist, *Phys. Rev. B* **13**, 4274 (1976).

(47) D. C. Langreth and J. P. Perdew, *Phys. Rev. B* **15**, 2884 (1977).

(48) J. Harris, *Phys. Rev. A* **29**, 1648 (1984).

(49) J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, *Phys. Rev. A* **46**, 6671 (1992).

(50) H. Sambe and R. H. Felton, *J. Chem. Phys.* **62**, 1122 (1975).

(51) B. I. Dunlap, J. W. D. Connolly and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

(52) B. Delley, *J. Chem. Phys.* **92**, 508 (1990).

(53) E. J. Baerends, D. E. Ellis and P. Ros, *Chem. Phys.* **2**, 41 (1973).

(54) M. J. Frish, G. W. Trucks, H. B. Schlegel, P. M. W. Gill, B. G. Johnson, M. A. Robb, J. R. Cheeseman, T. A. Keith, G. A. Peterson, J. A. Montgomery, K. Raghavachari, M. A. Al-Laham, V. G. Zakrzewski, J. V. Ortiz, J. B. Foresman, J. Cioslowski, B. B. Stefanov, A. Nanayakkara, M. Challacombe, C. Y. Peng, P. Y. Ayala, W. Chen, M. W. Wong, J. L. Andres, E. S. Replogle, R. Gomperts, R. L.

Martin, D. J. Fox, J. S. Binkley, D. J. DeFrees, J. Baker, J. P. Stewart, M. Head-Gordon, C. Gonzalez and J. A. Pople GAUSSIAN 94, GAUSSIAN, Inc., Pittsburgh, PA, 1995

(55) J. C. Slater, *Phys. Rev.* **81**, 385 (1951).

(56) A. St-Amant, PhD thesis, University of Montreal, 1992.

(57) A. St-Amant the latest version of DeFT can be downloaded from (http://www.chem.uottawa.ca/DeFT.html).

(58) W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

(59) S. L. Dixon and K. M. Merz, *J. Chem. Phys.* **104**, 6643 (1996).

(60) T. S. Lee, D. M. York and W. Yang, *J. Chem. Phys.* **105**, 2744 (1996).

(61) J. Harris, *Phys. Rev. B* **31**, 1770 (1982).

(62) W. Yang, *Phys. Rev. A* **44**, 7823 (1991).

(63) S. Baroni and P. Giannozzi, *Europhys. Lett.* **17**, 547 (1992).

(64) R. Haydock, V. Heine and M. J. Kelly, *J. Phys. C* **5**, 2845 (1972).

(65) E. Hernández, M. J. Gillan and C. M. Goringe, *Phys. Rev. B* **53**, 7147 (1996).

(66) X. P. Li, R. W. Nunes and D. Vanderbilt, *Phys, Rev. B* **47**, 10891 (1993).

(67) R. McWeeny, *Rev. Mod. Phys.* **32**, 335 (1960).

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

(69) J. J. P. Stewart, *Int. J. Quantum. Chem.* **58**, 113 (1996).

(70) L. Greengard and V. Rokhlin, *J. Comput. Phys.* **73**, 325 (1987).

(71) J. Barnes and P. Hut, *Nature* **324**, 446 (1986).

(72) A. Trew and G. Wilson, *Past, Present, Parallel: A Survey of Available Parallel Computer Systems* (Springer-Verlag; London; 1991).

(73) D. Schutzer, in *Parallel Processing and the Future Data Center: Computing in the Land of the Lilliputians*, (Van Nostrand Reinhold; New York; 1994).

(74) H. S. Morse, in *Practical Parallel Computing*, (AP Professional; Boston; 1994).

(75) V. Kumar, A. Grama, A. Gupta and G. Karypis, in *Introduction to Parallel Computing: Design and Analysis of Algorithms*, (The Benjamin/Cumming Publishing Company, Inc.; Redwood City; 1994).

(76) G. Amdahl, *AFIPS Comput. Conf.* **30**,483 (1967).

(77) T. G. Mattson, In Parallel Computing in Computational Chemistry, ACS Symposium series, M. J. Comstock, ed. (Am. Chem. Soc.; Washington, DC; 1995

(78) L. Baker and B. J. Smith, in *Parallel Programming*, (McGraw-Hill; New York; 1996).

(79) A. Beguelin, J. Dongarra, A. Geist, R. Manchek and V. Sunderam, *A User's Guide to PVM Parallel Virtual Machine/*,Oak Ridge National Laboratory,Technical Report, TM-11826 (1991).

(80) W. Gropp, E. Lusk and A. Skjellum, in *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, (MIT Press; Cambridge; 1995).

(81) CHIMP version of MPI is available by annoymous **ftp** from **ftp.epcc.ed.ac.uk** in the directory **pub/chimp/release**

(82) MPICH is available by anonymous **ftp** from **info.mcs.anl.gov** in the directory **pub/mpi/mpich**, file **mpich.tar.Z.**

(83) LAM is available by anonymous **ftp** from **tbag.osc.edu** in the directory **pub/lam**

Chapter 2

Using a Fitted Electronic Density to Improve the Efficiency of a Linear Combination of Gaussian-Type Orbitals Calculation

2.1 Introduction

In the LCGTO-DFT [1-3] approach, the computational burden of constructing the Coulombic contribution to the KS matrix can be eased by using a fitted density [4]. A fitted density transforms the four-centered two-electron integrals into three-centered two-electron integrals. This allows the construction of Coulomb matrix elements to be reduced from a quartically scaling procedure to a cubically scaling procedure. For large molecules, however, efficient cutoff schemes [5] reduce this step to a quadratic procedure, whether or not a fitted density is used. However, the density fitting procedure requires the inversion of an N by N overlap matrix. This procedure scales cubically with system size. For small to mid-sized molecules, the prefactor associated with matrix inversion is still relatively small and has a very small effect on the overall CPU time. As the system gets larger, however, this step will eventually dominate the entire cost of the fitting procedure. Therefore, for calculations on large systems, using a fitted density becomes a very expensive procedure. Instead of saving CPU time, fitting leads to much larger overall CPU times. Moreover, in view of the recent advances made in linear-scaling algorithms [6-8] for the construction of the Coulomb matrix, the benefit of using a fitted density is highly questionable unless something is done to alleviate its poor scaling behaviour.

These new linear-scaling algorithms are based on the continuous [6] (or Gaussian[8]) fast multipole method. The continuous fast multipole method divides the Coulombic interactions of Gaussian charge distributions into near-field and far-field contributions. The interaction between two Gaussian charge distributions is classified as near-field when there is significant overlap between them (they are not "well separated").

When two Gaussian charge distributions do not overlap (they are "well separated"), the Coulombic interaction between them is classified as far-field. Recall that the total electronic density in an LCGTO approach can be express as sums of Gaussian charge distributions

$$\rho(\boldsymbol{r}) = \sum_{\mu\nu} \boldsymbol{P}_{\mu\nu} \chi_{\mu}(\boldsymbol{r}) \chi_{\nu}(\boldsymbol{r}) \quad . \tag{2.1}$$

In equation 2.1, $\boldsymbol{P}_{\mu\nu}$ is a density matrix element given by equation 1.13. $\chi_{\mu}(\boldsymbol{r})$ and $\chi_{\nu}(\boldsymbol{r})$ are the Gaussian basis functions. Under the Gaussian product rule, the product of two Gaussians is a third Gaussian centered in between. We will call this third Gaussian a charge distribution. Two Gaussian functions must overlap somewhat to produce a charge distribution having a non-vanishing contribution to the total electronic density. However, a Coulombic interaction between two charge distributions is weighted by a $\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}$ factor that dies off very slowly. Therefore, no overlap is necessary for two charge distributions to have a significant Coulombic interaction. For large systems, the vast majority of the Coulombic interactions are far-field contributions that can be approximated by multipole expansions. The remaining near-field contributions require explicit evaluation of the two-electron integrals. The multipole expansions of those far-field contributions are very trivial to evaluate in comparison (in the sense that much less CPU time is required). Therefore, the CPU time is dominated by the explicit evaluation of the two-electron integrals of the near-field contributions.

When a fitted density is used, the number of two-electron integrals that need to be explicitly evaluated is much smaller than that required with a true density. The number of density fitting functions is much smaller than the number of charge distributions that

arise from the orbital bases from pairs of primitive functions having appreciable overlap. To prove this point, we carried out a series of calculations on the extended glycine polypeptides using triple-ζ plus polarization orbital bases with 34 density fitting functions on heavy atoms [9] and double-ζ plus polarization orbital bases with 13 density fitting functions on hydrogen atoms [9]. Note that the polypeptides used are not as they are in nature. Each end of the polypeptide is terminated by a peptide linkage. Each peptide thus corresponds to a section of a larger peptidic structure with the conventional carboylate and anime terminii.. In our test cases, both the true and fitted $\rho(r)$ that need to be explicitly evaluated grow linear with system size, as shown in Figure 2.1. However, note that the scale for the fitted density is 50 times larger. Though the number of integrals that require explicit evaluations increases linearly in each case, a factor of 100 in CPU time is saved with a fitted density since 100 times fewer integrals need to be evaluated. For the largest peptide, the glycine decapeptide, with 69 atoms, 891 orbital basis functions, and 1716 auxiliary basis functions, the number of two-electron integrals that require explicit evaluation is cut from 7 billion to 69 million when a fitted density is employed.

The above test cases show that an *ab initio* calculation on a large system can be more efficient with a fitted density. But to effectively make use of a fitted density in any calculation, the cubic scaling of the fitting procedure has to be eliminated. Luckily, a linear scaling method had recently been developed [10] where the electronic density is accurately fit in a divide and conquer (DAC) manner. This linear scaling fitting procedure allows us to realistically apply a fitted density to *ab initio* calcualtions on large

systems. This will thus allow us to improve the efficiency of various aspects of these calculations.

However, a fitted density may not be sufficiently accurate to calculate all molecular properties of interest. One such property is the electrostatic potential (ESP) of a molecule. The ESP is the potential experienced by a unit positive charge in the presence of a molecule's nuclear and electronic distributions. It is commonly evaluated over a set of grid points on the molecule's van der Waals surface (the details of how this is done will be presented later). Obviously, when evaluating the ESP at a grid point in space, using a fitted density to describe $\rho(\boldsymbol{r})$ in the nearby surrounding is most likely not accurate enough. However, it might be acceptable to use the less accurate fitted density in regions of space located further away. Recall that a fitted density reduces the number of floating point operations in a calculation. Therefore, it would be beneficial if a fitted density can be used to replace the true density whenever possible. With this in mind, we will show how the DAC expression for $\rho(\boldsymbol{r})$ can be modified to derive an approximate $\rho(\boldsymbol{r})$ whose quality varies depending on its location. We will use this approximate $\rho(\boldsymbol{r})$ to evaluate a molecule's ESP. The quality of this ESP will be judged by the errors generated in the set of ESP-fitted atomic charges. To quantify this error further, the ESP-fitted charges will be used to calculate the dipole moment and the free energy of solvation. The difference between the dipole moments and free energies of solvation generated by the true and approximate densities will then be measured to ascertain the quality of the approximate density. An analysis of the CPU time savings afforded by the approximate $\rho(\boldsymbol{r})$ will also be presented.

2.2 DAC Fit of Electronic Density

In Yang's divide and conquer (DAC) schemes [11, 12], the total electronic density (expressed in equation 2.1) is partitioned into a sum of subsystem contributions,

$$\rho(\boldsymbol{r}) = \sum_{\alpha}^{subsystems} \rho^{\alpha}(\boldsymbol{r}) \quad . \tag{2.2}$$

To fit the electronic density by a DAC approach, Yang's density matrix DAC formulation [12] is used. In this formulation, a subsystem density is defined as

$$\rho^{\alpha}(\boldsymbol{r}) = \sum_{\mu}^{K} \sum_{\nu}^{K} p_{\mu\nu}^{\alpha} P_{\mu\nu} \chi_{\mu}(\boldsymbol{r}) \chi_{\nu}(\boldsymbol{r}) \tag{2.3}$$

with the partition function $p_{\mu\nu}^{\alpha}$ defined as

$$p_{\mu\nu}^{\alpha} = \begin{cases} 1 & \text{if both } \chi_{\mu}(\boldsymbol{r}) \text{ and } \chi_{\nu}(\boldsymbol{r}) \text{ belong to subsystem } \alpha \\ \frac{1}{2} & \text{if either } \chi_{\mu}(\boldsymbol{r}) \text{ or } \chi_{\nu}(\boldsymbol{r}) \text{ belongs to subsystem } \alpha \\ 0 & \text{if neither } \chi_{\mu}(\boldsymbol{r}) \text{ and } \chi_{\nu}(\boldsymbol{r}) \text{ belong to subsystem } \alpha \end{cases} \quad . \tag{2.4}$$

This partition scheme is consistent with that used within a Mulliken population analysis [13]. In a Mulliken population analysis, electronic density is totally attributed to an atom if $\chi_{\mu}(\boldsymbol{r})$ and $\chi_{\nu}(\boldsymbol{r})$ are both centered on that atom, and it is otherwise equally split between two atoms if $\chi_{\mu}(\boldsymbol{r})$ and $\chi_{\nu}(\boldsymbol{r})$ are centered on two different atoms. It can be shown that the partition functions obey the following constraint for each pair of basis functions,

$$\sum_{\alpha}^{subsystems} p_{\mu\nu}^{\alpha} = 1 \quad . \tag{2.5}$$

As a consequence, note that no approximations have been made at this point as the sum of these subsystem densities still recovers the exact LCGTO density. However, $p_{\mu\nu}^{\alpha}$ localizes $\rho^{\alpha}(\boldsymbol{r})$'s contribution to $\rho(\boldsymbol{r})$ to a region in the vicinity of subsystem α's

atoms. Therefore, only the auxiliary fitting functions centered on subsystem α's atoms and nearby buffer atoms (i.e., atoms within a predefined cutoff distance of any atom of subsystem α) should be required to accurately fit $\rho^{\alpha}(\boldsymbol{r})$.

The fit of $\rho^{\alpha}(\boldsymbol{r})$ follows rather closely the conventional $\rho(\boldsymbol{r})$ fitting procedure. Rather than minimizing the expression in equation 1.41,

$$\iint \frac{\left[\rho^{\alpha}(\boldsymbol{r}) - \tilde{\rho}^{\alpha}(\boldsymbol{r})\right]\left[\rho^{\alpha}(\boldsymbol{r}') - \tilde{\rho}^{\alpha}(\boldsymbol{r}')\right]}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \tag{2.6}$$

is minimized instead. A new normalization constraint is imposed on each subsystem fit [10]:

$$\int \tilde{\rho}^{\alpha}(\boldsymbol{r}) d\boldsymbol{r} = \sum_{A \in \alpha} Q_A \tag{2.7}$$

where Q_A is the Mulliken population [13] of atom A. This is the proper normalization constraint since the DAC and Mulliken approaches are consistent, as stated earlier. No particular physical significance is being attributed to Mulliken populations. The subsystem density fit coefficient vector, $\boldsymbol{c}^{\alpha}$, is obtained from

$$\boldsymbol{c}^{\alpha} = (\boldsymbol{S}'^{\alpha})^{-1}(\boldsymbol{t}^{\alpha} + \lambda^{\alpha}\boldsymbol{m}^{\alpha}) , \tag{2.8}$$

where $(\boldsymbol{S}'^{\alpha})^{-1}, \boldsymbol{t}^{\alpha}$, and $\boldsymbol{m}^{\alpha}$ are defined by equations similar to equations 1.44 to 1. 46. The superscript α indicates that these vectors and matrices (in equation 2.8) only span those auxiliary basis functions centered on an atom of subsystem α or on one of its buffer atoms. The Lagrange multiplier, λ^{α}, in equation 2.8 is now given by

$$\lambda^{\alpha} = \sum_{A \in \alpha} Q_A - \frac{\boldsymbol{m}^{\alpha}(\boldsymbol{S}'^{\alpha})^{-1}\boldsymbol{t}^{\alpha}}{\boldsymbol{m}^{\alpha}(\boldsymbol{S}'^{\alpha})^{-1}\boldsymbol{m}^{\alpha}} \quad . \tag{2.9}$$

The $\boldsymbol{t}$ vector in equation 1.46 is modified to

$$t_i^{\alpha} = \sum_{\mu}^{N} \sum_{\nu}^{N} p_{\mu\nu}^{\alpha} P_{\mu\nu} \iint \frac{\chi_i'(\boldsymbol{r}) \chi_{\mu}(\boldsymbol{r}') \chi_{\nu}(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \qquad (2.10)$$

Since $p_{\mu\nu}^{\alpha}$ vanishes unless at least one orbital function is centered on an atom of subsystem α, the above equation can be reduced to

$$t_i^{\alpha} = \sum_{\mu \in \alpha} \sum_{\nu}^{N} p_{\mu\nu}^{\alpha} P_{\mu\nu} \iint \frac{\chi_i'(\boldsymbol{r}) \chi_{\mu}(\boldsymbol{r}') \chi_{\nu}(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r} d\boldsymbol{r}' \qquad (2.11)$$

where $p_{\mu\nu}^{\alpha}$ has a value of 1 if $\chi_{\nu}(\boldsymbol{r})$ is centered on an atom of subsystem α, and $\frac{1}{2}$ otherwise. The first summation in equation 2.11 runs over only those basis functions centered on atoms of subsystem α. Their number is independent of overall system size. The elements of t^{α} span only the auxiliary functions on the atoms and buffer atoms of subsystem α. For a given buffer space cutoff, their number is also fixed regardless of overall system size. The second summation runs over the entire set of basis functions in the system. One might therefore think that the CPU time spent constructing t^{α} will increase linearly with system size. However, this is not the case in practice. Non-negligible contributions to t^{α} only occur when $\chi_{u}(\boldsymbol{r})$ and $\chi_{\nu}(\boldsymbol{r})$ have non-negligible overlap. Since a basis function has a finite extent, it only has a significant overlap with basis functions in its vicinity. As system size increases, the number of basis functions having significant overlap with an orbital basis function centered on an atom in subsystem α will increase. However, at some point, increasing system size by adding atoms far away from subsystem α will have no effect on the number of orbital functions with significant overlap. The atoms being added are simply too far away. At this point, the CPU time spent constructing t^{α} will reach a constant value and become independent of total system size. Therefore, for sufficiently large systems, the overall fitting

procedure will scale linearly only with the number of subsystems. The procedure will thus scale linearly with overall system size.

The key to achieving linear scaling with a divide and conquer method is the ability to use truncated bases for subsystem calculations within localized regions. A previous test [10] on a single molecule suggested that a better than 0.01 kcal mol^{-1} accuracy can be achieved with a 5.0 Å buffer space cutoff. This means that any atom within 5.0 Å of any subsystem atom is added to the buffer space. To confirm that such a cutoff is indeed adequate, we carry out a series of calculations on extended glycine polypeptides using triple-ζ plus polarization orbital bases with 34 $\rho(\boldsymbol{r})$ fitting functions on heavy atoms and double-ζ plus polarization orbital bases with 13 $\rho(\boldsymbol{r})$ fitting functions on hydrogen atoms. The results are tabulated in Table 2.1. The test calculations are performed with the exchange-correlation (XC) potential also being fit by a DAC procedure [14]. For these XC DAC fits, a 7.0 Å cutoff is used. Details of this DAC XC fitting procedure will be given in Chapter 4. In these calculations, each polypeptide (as an example, the pentapeptide, given in Figure 2.4) is divided into a series of -$NHCH_2CO$- subsystems, a terminal –COH subsystem and a terminal –NH_2 subsystem. From Table 2.1, we confirm that a 5.0 Å cutoff is indeed adequate for our series of extended glycine polypeptides. We note that the differences between total energies obtained with the conventional fitting procedures and the total energies obtained with the DAC fitting procedures are consistently less than 0.002 kcal mol^{-1}. We also note that there is no appreciable increase in error with increasing system size. The total energies obtained with the DAC fitting procedures tend to be lower. This is consistent with the fact that a perfect fit of $\rho(\boldsymbol{r})$ always yields a total energy that serves as an upper

bound [4]. There is no such variational aspect to the DAC XC fits, thus explaining why, on occasion, the energy with the DAC fits can be higher.

Figure 2.2 shows the CPU times associated with the conventional and DAC fits of $\rho(\boldsymbol{r})$ for a series of extended glycine polypeptides. Clearly, a DAC fit of $\rho(\boldsymbol{r})$ does indeed scale linearly with system size. The conventional fit scales quadratically. The DAC fit is marginally more expensive for small peptides and clearly becomes the method of choice for system with more than 40 atoms. To get a deeper understanding of why this is so, we refer to Table 2.3, which lists the number of atoms in each polypepetide's subsystems. Using a DAC approach for the dipeptide is clearly not efficient since the largest subsystem covers the entire system and the others are just marginally smaller. In a sense, we are performing a conventional fit three times. The tetrapeptide has 5 subsystems, which, in descending order, contain, 23, 18, 18, 11, and 11, atoms. We see that the largest subsystem is just marginally smaller than the total number of atoms, 27. Though the smaller subsystems clearly require much less CPU time than the global conventional fit, not much CPU time can be saved as the largest subsystem is clearly as large as the original molecule. The total CPU time is slightly larger than that for the conventional non-DAC approach. As system size is increased to the tetrapeptide and beyond, we see that the largest subsystem in each peptide has reached a maximum number of atoms, 23. This means that the CPU time for any subsystem calculation has reached a maximum value. As the number of subsystem increases, the total CPU time increases proportionally. Hence, the total CPU time increases linearly with system size. Note that the extended conformation of the polypeptides is used in these benchmark

calculations. For a general globular structure, much larger systems will have to be studied before linear scaling becomes apparent.

Calculations have also been performed on our series of extended glycine polypeptides using smaller subsystems in DAC fits. Each seven atom $-NHCH_2CO-$ subsystem is further divided into a $-NH-$, a $-CH_2-$, and a $-CO-$ subsystem. We found that this has very little effect on the overall CPU time of the DAC fit. Each subsystem fit requires less CPU time, but there are more subsystems requiring fitting. The two effects cancel themselves almost perfectly. In general, smaller subsystems are somewhat less efficient for smaller molecules. However, they are slightly more efficient for larger molecules.

2.3 The Electrostatic Potential and the Electrostatic Potential-Fitted Charges

The ESP at any point $\boldsymbol{r}$ in space is defined as

$$V(\boldsymbol{r}) = \sum_A \frac{Z_A}{|\boldsymbol{r} - \boldsymbol{R}_A|} - \sum_{\mu\nu} \boldsymbol{P}_{\mu\nu} \int \frac{\chi_\mu \chi_\nu}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' \quad . \tag{2.12}$$

In the above, Z_A is the nuclear charge, χ_μ is an atomic orbital, and $\boldsymbol{P}_{\mu\nu}$ is s density matrix element. The first term represents the electrostatic potential at a point $\boldsymbol{r}$ generated by the nuclei. The second term is the electrostatic potential generated by the electronic distribution.

In molecular mechanical approaches, charges are often assigned to atoms. Often, these charges are generated by an approach that attempts to find the best set of atomic charges at reproducing the ESP evaluated by a quantum mechanical approach.

To evaluate the ESP of a molecule and subsequently obtain these fitted atomic charges, it is important to generate a good set of points around the molecule to sample its ESP. Since we are interested in intermolecular interactions, the ESP is sampled beyond the molecule's van der Waals surface. A procedure which is well defined for all molecules and capable of generating a uniform surface of points is the surface generation algorithm of Connolly [15]. This algorithm computes a spherical surface of points around each atom at a specific multiple of the atom's van der Waals' radius. The molecular surface is then produced by merging all the atom surfaces and eliminating those points that are within this specific multiple of the van der Waals' radius of any other atom. Finally, any remaining points that can not be touched by a 1.4 Å (the approximate van der Waals radius of a water molecule) sphere without touching other points is discarded. This last step eliminates any points in "nooks or cranny" that are not solvent accessible. It has been established by Singh and Kollman [16] that using four surfaces with shells of 1.4, 1.6, 1.8 and 2.0 times the van der Waals' radii and a density of between one to five points per angstrom^2 yields an adequate sampling of points.

The potential-derived (PD) atom-centered charges (ESP-fitted charges) can be obtained by a least-squares fitting procedure [17-19]. The least-squares fit criterion with a single constraint is given by minimizing the function

$$z(q_1, q_2, \ldots\ldots, q_n) = \sum_{i=1}^{m} (V_i - \sum_{j=1}^{n} \frac{q_j}{r_{ij}})^2 + \lambda \left(\sum_{j=1}^{n} q_j - q_{tot} \right) \quad . \qquad (2.13)$$

In the above, V_i is the quantum mechanically calculated ESP at point i, q_j is the net charge on atom j, r_{ij} is the distance between atom j and the grid point i, m is the number of grid points, n is the number of atoms, and λ is a Lagrange multiplier. The last term

imposes the constraint that the net atomic charges must sum to the net molecular charge, q_{tot}. The minimum of z (equation 2.13) and the corresponding q's may be found by solving the following system of equations:

$$\frac{\partial z}{\partial \lambda} = 0 = \sum_{j=1}^{n} q_j - q_{tot} \tag{2.14}$$

$$\frac{\partial z}{\partial q_k} = 0 = -\sum_{i=1}^{m} \frac{2}{r_{ik}} \left(V_i - \sum_{j=1}^{n} \frac{q_j}{r_{ij}} \right) + \lambda \tag{2.15}$$

The above equations are simplified to

$$\sum_{j=1}^{n} q_j = q_{tot} \tag{2.16}$$

$$\sum_{j=1}^{n} \sum_{i=1}^{m} \frac{q_j}{r_{ij} r_{ik}} = \sum_{i=1}^{m} \frac{V_i}{r_{ik}} + \lambda . \tag{2.17}$$

Note that λ is an arbitrary constant. It is left as $+\lambda$ (rather than $-\frac{\lambda}{2}$) in equation 2.17.

This can be written in matrix form as

$$Aq = B \tag{2.18}$$

where

$$A_{jk} = \sum_{i=1}^{m} \frac{1}{r_{ij} r_{ik}} \tag{2.19}$$

and

$$B_k = \sum_{i=1}^{m} \frac{V_i}{r_{ik}} . \tag{2.20}$$

2.4 The Finite-Difference Poisson-Boltzman Solvantion Model

A Solvation model proposed by Tannor *et al* [20] combines *ab initio* quantum chemical calculations with a continuum description of the solvent to obtain accurate solvation free energies. In this model, the solute is described by a charge distribution in a cavity immersed in a solvent represented as a polarizable dielectric continuum. The molecular charge distribution of the solute induces a surface charge density in the solvent on the solute/solvent boundary. This in turn influences the molecular charge distribution of the solute. The response of the solvent to the molecular charge distribution is given by Poisson's equation

$$\nabla\varepsilon(\boldsymbol{r})\nabla V(\boldsymbol{r}) + 4\pi\rho(\boldsymbol{r}) = 0 \qquad (2.21)$$

where $V(\boldsymbol{r})$ is the electrostatic potential distribution in the solvent. The dielectric function, $\varepsilon(\boldsymbol{r})$, has a value of 1 within the van der Waals surface of the solute and a value of 80 (the dielectric constant of water) in the solvent region.

The above equation may be solved by a finite difference method implemented within the DelPhi program [21]. In this method, the region of space containing the solute and the surrounding solvent is divided into a cubic lattice with equal spacing between the grid points. A typical lattice is made up of 65 points in each of the three directions with a spacing of 0.5-1.0 Å. In the simplified model, $\rho(\boldsymbol{r})$ is zero everywhere, except at the positions of a solute atom where $\rho(\boldsymbol{r})$ is equal to the ESP-fitted charge of that atom. Since Poisson's equation is solved on a cubic lattice, the ESP-fitted atomic charges must reside on these lattice points. Each atom finds itself in a cube defined by eight lattice points. The charge is partitioned among the eight lattice points so as to conserve the center of charge (and thus, the dipole of the solute). Once the solute's atomic charges are

set up on the cubic lattice, the value of $V(r)$ at each point of the lattice can be obtained from

$$V(i,j,k) = \frac{\sum_{l=1}^{6} \varepsilon(l) V(l) + 4\pi q_o^f (i,j,k)/h}{\sum_{l=1}^{6} \varepsilon(l)} \tag{2.22}$$

where $l = i \pm 1, j \pm 1, k \pm 1$ refers to the six adjoining points on the lattice. h is the lattice spacing. q_o^f is the solute charge assigned to the grid point. The induced charge at lattice point o is obtained from

$$q_o^i = \frac{h\left(6V_o - \sum_{l=1}^{6} V(l)\right)}{4\pi} - q_o^f \tag{2.23}$$

The mathematics is such that an induced charge may only be generated at lattice points adjacent to the solute/solvent dielectric boundary.

A quantum mechanical calculation is then performed on the solute in the presence of the surface induced charges. These induced charges will polarize the electronic structure of the solute, which leads to larger ESP-fitted charges on the atoms. Hence, a series of quantum mechanical and FDPB calculations are carried out until the induced surface charges and the ESP-fitted charges become self consistent. The electrostatic solvation free energy can then be obtained from

$$\Delta G_{el}^o = \frac{1}{2} \sum_k \sum_o \left(\frac{q_k^f q_o^i}{r_{ko}} \right) \tag{2.24}$$

where r_{ko} is the distance between atom k and lattice point o on the dielectric boundary. q_k^f is the charge assigned to atom k. The non-electrostatic component of solvation is given by a surface area dependent term obtained from experimental data on alkanes for

which it is assumed that all contributions to the free energy of solvation is non-electrostatic in nature.

We now reconsider fitting the ESP using the Besler, Merz, and Kollman approach. The ESP for a series of extended glycine polypeptides, ranging from the dipeptide to the decapeptide is computed using the true density and the fitted density. The errors introduced to the ESP by using the fitted density, rather than the true density, are reported in Table 2.2 as the root mean squares (RMS) value per grid point. From Table 2.2, we see that about an 8% RMS error is introduced in the ESP at any point in space. When this set of ESP values at the grid points is used to fit the atomic charges, the RMS error in the atom-centered charges is well over 0.04 units of charge. The RMS error in the dipole moments calculated form these atomic charges is roughly 0.07 Debye. Finally, using this set of charges in a FDPB calculation, a roughly 2.0 kcal mol^{-1} RMS error is introduced into the electrostatic contribution to the free energy of solvation (equation 2.4). Clearly, this is not sufficiently accurate for most applications. It would seem that using a true density for these calculations is inevitable. However, the true and fitted density are not necessarily mutually exclusive in a calculation. In the next section, we will show an elegant way of combining the true and the fitted densities in a calculation. We will call this combined density a hybrid DAC density. This hybrid DAC density is derived by modifying the DAC expression for a fitted density and adding components of the true density. The quality of the hybrid DAC density varies depending on its location. In regions requiring great precision, it can be made extremely accurate. In regions where less precision is acceptable, it can be made significantly less accurate (without sacrificing much precision in the final value of interest).

2.5 Hybrid DAC Density

A hybrid DAC density exploits the fact that a DAC fit of $\rho(\boldsymbol{r})$ is divided into a sum of clearly defined subsystem contributions, $\sum_{\alpha}^{subsystems} \rho^{\alpha}(\boldsymbol{r})$. Equation 2.2 can be rewritten as

$$\rho(\boldsymbol{r}) = \rho^{\sigma}(\boldsymbol{r}) + \sum_{\alpha \neq \sigma}^{subsystems} \rho^{\alpha}(\boldsymbol{r}) \quad . \tag{2.25}$$

The above equation is derived by explicitly removing subsystem σ 's contribution to the density from the sum. Now imagine that we are interested in calculating the ESP at a point in space located nearest to an atom in subsystem σ. Obviously, an accurate description of $\rho(\boldsymbol{r})$ in the vicinity of subsystem σ is required since the electrostatic interaction dies off as $1/\boldsymbol{r}$. Further away from the subsystem σ, a less accurate description of $\rho(\boldsymbol{r})$ may be quite sufficient. We can then introduce the following approximation for $\rho(\boldsymbol{r})$:

$$\rho(\boldsymbol{r}) \approx \rho^{\sigma}(\boldsymbol{r}) + \sum_{\alpha \neq \sigma}^{subsystems} \tilde{\rho}^{\alpha}(\boldsymbol{r}) \tag{2.26}$$

where we replace the remaining true subsystem densities, $\rho^{\alpha}(\boldsymbol{r})$, with their fitted counterparts, $\tilde{\rho}^{\alpha}(\boldsymbol{r})$. We call this a hybrid DAC density. With it, $\rho(\boldsymbol{r})$ is very accurately described in the vicinity of the point of interest. Further away from the point of interest, it is less accurate. This approach is possible only because the DAC approach partitions the densities into clearly defined, and properly normalized, subsystem contributions. No double counting of electronic density occurs. If this approximation is too drastic, a concept similar to buffer space can be invoked. True $\rho^{\alpha}(\boldsymbol{r})$ contributions

are used within a selected cutoff and fitted $\rho^{\alpha}(\boldsymbol{r})$ contributions are used beyond it, thus yielding the following expression,

$$\rho(\boldsymbol{r}) \approx \sum_{\sigma}^{\text{nearby subsystems}} \rho^{\sigma}(\boldsymbol{r}) + \sum_{\alpha}^{\text{faraway subsystems}} \tilde{\rho}^{\alpha}(\boldsymbol{r}) \quad . \tag{2.27}$$

This cutoff separating the subsystems into nearby and faraway sums may be either raised or lowered to any value deemed appropriate for the problem at hand. The exact nature of the hybrid DAC density changes from point to point. The decision as to which summation, nearby or faraway, a subsystem is placed is made based upon its position relative to the immediate point of interest. When evaluating the ESP on a grid, equation 2.27 will be used and the cutoff value remains fixed from point to point. However, the subsystems included within each summation will change from point to point.

2.6 Benchmark Calculations

A hybrid DAC density approximates a system's true density by combining true subsystem contributions from "nearby subsystems" with a fitted subsystem contributions from "faraway subsystems". The quality of a hybrid DAC density can be improved systematically by extending the cutoff value encapsulating the subsystems characterized as being "nearby". We wish to establish the errors introduced by hybrid DAC densities of varying quality. Therefore, benchmark calculations are performed on a series of extended glycine polypeptides ranging from the dipeptide to the decapeptide, using triple-ζ plus polarization orbital bases with 34 $\rho(\boldsymbol{r})$ fitting functions on heavy atoms and double-ζ plus polarization orbital bases with 13 $\rho(\boldsymbol{r})$ fitting functions on hydrogen

atoms [9]. A 5.0 Å cutoff is used for the DAC fit of $\rho(\boldsymbol{r})$. A 7.0 Å cutoff is used for the DAC XC fits. In these calculations, a hybrid DAC density is used to evaluate each peptide's ESP. Four levels of hybrid DAC densities are tested (see Figure 2.5). In our test calculations, a level 0 hybrid density uses the true density only within the immediate subsystem of interest. A level 1 hybrid density extends the use of the true density to the adjoining subsystems. For level 2 and 3 hybrid densities, use of the true density is extended to two and three $-NHCH_2CO-$ subsystems, respectively, beyond either side of the immediate subsystem of interest. Since we are working with extended conformations, this approach is acceptable. In general, for globular systems, a distance criterion will have to be employed. In terms of distances, level 1, 2, and 3 hybrid DAC densities roughly correspond to using a subsystem's true density if any one atom of that subsystem is within 3.0, 6.0, or 9.0 Å, respectively, of any one atom in the immediate subsystem of interest. When evaluating the electrostatic potential on a grid, a subsystem is considered as being the immediate subsystem of interest if it contains the atom nearest to the grid point in question.

2.7 Results and Discussion

The CPU times involved evaluating the ESP on a collection of grid points using a series of hybrid DAC $\rho(\boldsymbol{r})'s$ and the true $\rho(\boldsymbol{r})$ for a series of glycine polypeptides are plotted in Figure 2.3. Note that as the molecules get larger, the number of grid points rises linearly. It is apparent that evaluating the ESP on a collection of grid points with a true $\rho(\boldsymbol{r})$ scales quadratically with system size. Part of this quadratic scaling is

attributed to the linear growth in the number of grid points. From equation 2.12, we see that the computational time at each grid point depends almost entirely on the number of pairs of primitive Gaussians in the orbital basis having appreciable overlap. The time involved evaluating the nuclear contribution is negligible. When the system is sufficiently large, each primitive has a fixed number of appreciable overlaps with other primitives. However, as system size increases, the number of primitives (each with a fixed number of non-vanishing overlaps) increases linearly. The linear growth in the number of points, combined with the linear increase in CPU time per grid point, leads to the overall quadratic scaling when the ESP is evaluated with the true density.

In Figure 2.3, we see that whenever a hybrid DAC $\rho(\boldsymbol{r})$ is used to evaluate the ESP, linear scaling is eventually achieved. For each level of hybrid DAC density, the CPU time spent on the region using a true density (the "nearby subsystems") dominates the overall CPU time required for the evaluation of the ESP at a grid point. The size of this region is independent of overall system size. Consequently, the CPU time required by any point for "nearby subsystems" becomes independent of system size. Therefore, for large systems, linear scaling is observed. The number of fitting functions employed for "faraway subsystems" rises linearly with system size. Therefore, combined with the fact that the number of grid points rises linearly, evaluation of ESP using a hybrid DAC $\rho(\boldsymbol{r})$ should be expected to scale quadratically. However, linear scaling is seen in practice since the time spent evaluating the ESP with the fitted $\rho(\boldsymbol{r})$ is trivial for even the largest systems. For example, for the 69 atom decapeptide, less than 12 seconds of CPU time are required to evaluate the full ESP with the fitted $\rho(\boldsymbol{r})$ over the entire grid of some 11000 points. Even with the crudest level 0 hybrid DAC $\rho(\boldsymbol{r})$, the same calculation takes

over 750 seconds. For small systems, the CPU times with the hybrid DAC $\rho(\boldsymbol{r})$'s are identical to those using the true $\rho(\boldsymbol{r})$ since the definition of "nearby" encompasses the entire system. However, linear scaling is quickly achieved once subsystems begin to be classified as "faraway subsystems".

Table 2.2 shows the errors introduced in the ESP of the extended glycine polypeptides when hybrid DAC densities are employed. It is clearly shown that errors can be systematically reduced by increasing the hybrid DAC $\rho(\boldsymbol{r})$ quality (i.e., with a higher level hybrid DAC density). The test for our hybrid DAC approximations to the true $\rho(\boldsymbol{r})$ is their ability to yield accurate free energies of solvation within the FDPB [20] solvation model outlined above. The errors associated with the solvation free energy, as calculated with hybrid DAC $\rho(\boldsymbol{r})$'s , are given in Table 2.2. The inherent errors within this solvation model are on the order of 0.6 kcal mol^{-1} [20]. Therefore, using either the fitted $\rho(\boldsymbol{r})$ or the level 0 hybrid DAC $\rho(\boldsymbol{r})$ is not acceptable. Their RMS errors are either well over, or of the same magnitude as, the intrinsic 0.6 kcal mol^{-1} error of the model. With either a level 2 or a level 3 hybrid DAC $\rho(\boldsymbol{r})$, the RMS errors in the solvation free energy are reduced to 0.015 and 0.009 kcal mol^{-1}, respectively. These errors are, for all intents and purposes, negligible. These small errors are even more impressive considering that the free energies of solvation for the larger peptides are as large as –50.0 kcal mol^{-1}. Please note that the errors quoted are relative to the values obtained using the "true" densities, i.e., the density obtained within a conventional, non-DAC, DFT calculation with the given basis set. These are not relative to any experimental values. Our goal with the hybrid DAC density is to not further contribute to the intrinsic errors of the solvation model. Hence we have these stringent error

requirements. Figure 2.3 also shows that considerable CPU time can be saved for large systems using our new approximation for $\rho(\boldsymbol{r})$. It is clear that hybrid DAC $\rho(\boldsymbol{r})$'s may be an excellent way of speeding up calculations without sacrificing too much accuracy.

Test calculations have also been performed where these hybrid DAC densities are used to evaluate the XC potential at grid points in a DFT calculation. The CPU time is reduced when using level 0 or level 1 hybrid DAC $\rho(\boldsymbol{r})$'s. However, the errors introduced are not acceptable. Using higher level hybrid DAC $\rho(\boldsymbol{r})$'s do reduce the errors to acceptable levels. Unfortunately, at this point, the CPU time saved is negligible. Contributions to the value of $\rho(\boldsymbol{r})$ at a grid point are almost exclusively arising from "nearby subsystems". This would suggest that the usefulness of hybrid DAC $\rho(\boldsymbol{r})$'s are limited to treating terms that die off slowly, such as electrostatic interactions.

2.8 Conclusion

We have shown that a DAC approach to fitting $\rho(\boldsymbol{r})$ does exhibit linear scaling with system size. For our test series of glycine polypeptides, the errors in the total energy introduced by the DAC $\rho(\boldsymbol{r})$ fits (and the DAC XC fits) are consistently well under 0.002 kcal mol^{-1} with an average error of about 0.0006 kcal mol^{-1}. It is possible to achieve linear scaling in constructing the Coulomb matrix without using a fitted $\rho(\boldsymbol{r})$. However, using a fitted $\rho(\boldsymbol{r})$ can dramatically reduce the number of floating point operations in a calculation. If a fitted $\rho(\boldsymbol{r})$ is not sufficiently accurate in a calculation, we have shown that a hybrid DAC $\rho(\boldsymbol{r})$ combing both the true and the fitted densities can be constructed. The quality of a hybrid density can be improved systematically by

extending the cutoff within which the true $\rho^{\alpha}(\boldsymbol{r})$ contributions are used. Linear scaling is eventually achieved for large systems using any hybrid DAC $\rho(\boldsymbol{r})$ when evaluating the ESP on a grid. A hybrid DAC $\rho(\boldsymbol{r})$ could also be used to construct the Coulomb contribution to either the Fock or Kohn-Sham matrix elements in *ab initio* or density functional LCGTO calculations. This would be a compromise between those methods that currently make use of a fitted $\rho(\boldsymbol{r})$ and those methods that evaluate the exact Coulomb repulsion contribution with the true LCGTO $\rho(\boldsymbol{r})$. Since we do not yet have the capability of performing four-centered two-electron integrals within our DeFT software package, we have not been able to test out this possibility. This work is under way.

Table 2.1

Errors in the total energy introduced by DAC fits in calculations on extended glycine polypeptides.

molecule	conventional fits (hartrees)	DAC fits (hartrees)	error (kcal mol^{-1})
dipeptide	-375.0348774	-375.0348776	+0.0001
tripeptide	-581.4549341	-581.4549363	+0.0014
tetrapeptide	-787.8757065	-787.8757069	+0.0003
pentapeptide	-994.2961476	-994.2961473	-0.0002
hexapeptide	-1200.7168353	-1200.7168364	+0.0007
heptapeptide	-1407.1373028	-1407.1373034	+0.0004
octapeptide	-1613.5580470	-1613.5580485	+0.0009

Table 2.2

Errors in the ESP of extended glycine polypeptides calculated with hybrid DAC $\rho(r)$'s.

	errors			
	$\phi(r)$ (%)[a]	$q_A(e)$[b]	μ(D)[c]	ΔG^{el}_{solv} (kcal mol^{-1})[d]
fitted $\rho(r)$	8.02	0.0413	0.073	1.788
level 0 hybrid DAC $\rho(r)$	4.08	0.0233	0.048	0.276
level 1 hybrid DAC $\rho(r)$	0.90	0.0052	0.023	0.064
level 2 hybrid DAC $\rho(r)$	0.33	0.0011	0.014	0.015
level 3 hybrid DAC $\rho(r)$	0.17	0.0006	0.012	0.009

[a] Mean error in the ESP at a grid point.

[b] RMS error in the ESP-fitted atomic charges; e = elementary charge = 1.6022×10^{-19} C.

[c] RMS error in the dipole moment calculated from the ESP-fitted charges, in Debyes.

[d] RMS error in the electrostatic contribution to the free energy of solvation, as calculated in a FDPB calculation.

Data is for nine extended glycine polypeptides, from the dipeptide to the decapeptide. A 5.0 Å buffer space cutoff is used for the DAC fit of $\rho(r)$; a 7.0 Å buffer space is used for the DAC fit of the XC terms.

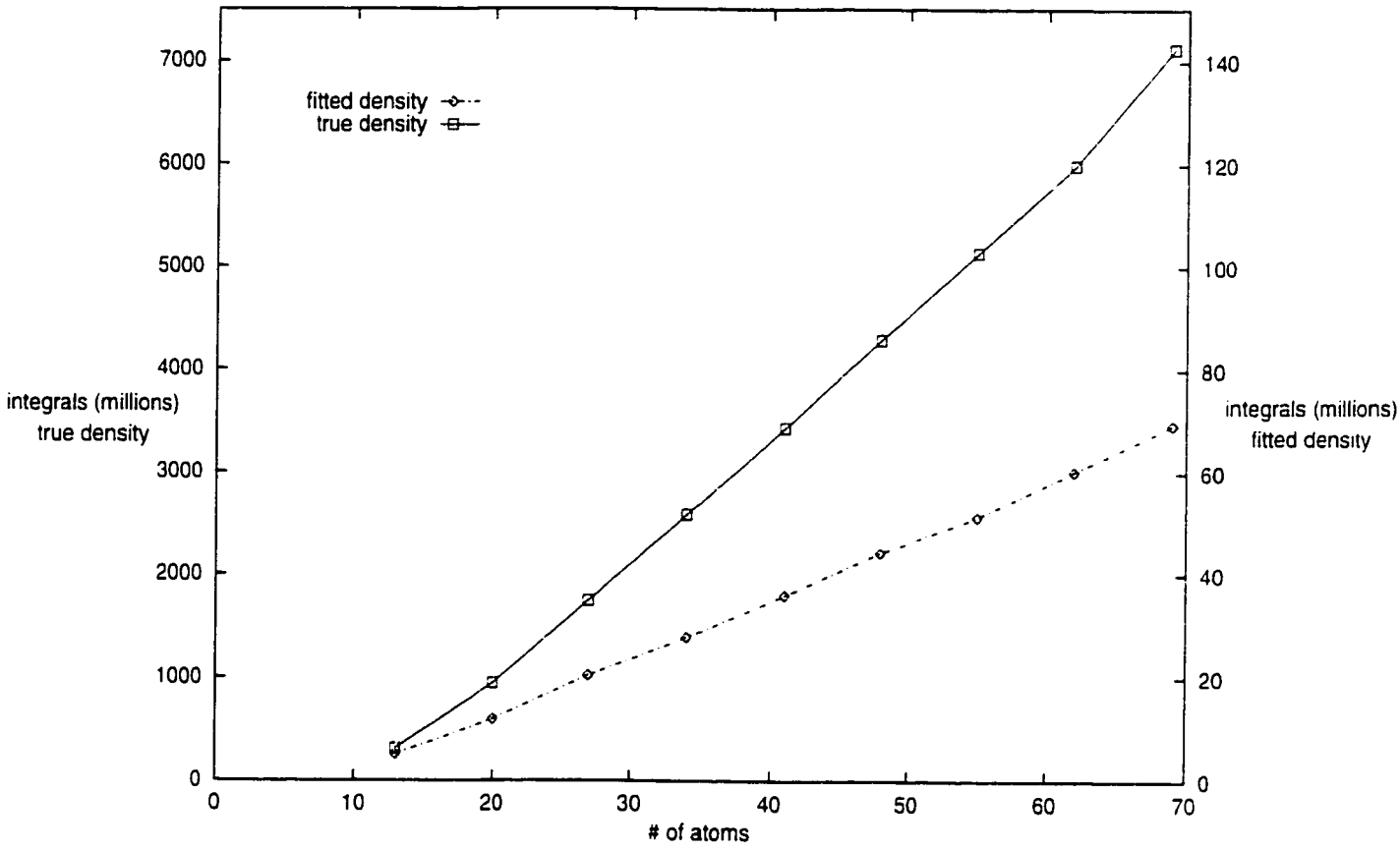
Table 2.3

Total number of subsystem and buffer atoms included in subsystem fits for the DAC fit of $\rho(r)$ for a series of extended glycine polypeptide fragments.

peptides	number of atoms per subsystem	total number of atoms
dipeptide	11,13, 11	13
tripeptide	11,18, 18, 11	20
tetrapeptide	11,18, 23, 18,11	27
pentapeptide	11,18, 2*23, 18,11	34
hexapeptide	11,18, 3*23, 18,11	41
heptapeptide	11,18, 4*23, 18,11	48
octapeptide	11,18, 5*23, 18,11	55
nonapepetide	11,18, 6*23, 18,11	62
decapeptide	11,18, 7*23, 18,11	69
undecapeptide	11,18, 8*23, 18,11	76
dodecapeptide	11,18, 9*23, 18,11	83
trisdecapeptide	11,18, 10*23, 18,11	90
tetradecapeptide	11,18, 11*23, 18,11	97

Figure 2.1

Number of two-electron integrals which need explicit evaluation using either the true $\rho(r)$ or a fitted $\rho(r)$ to handle short-range interactions while using fast multipole methods to handle long-range interactions.



Note that the number of integrals with the true $\rho(r)$ is given on the left, and that for the fitted $\rho(r)$ is given on the right on a scale that is 50 times larger. A threshold of 10^{-10} was used to determine if two Gaussian distributions are sufficiently "well separated" [6].

Figure 2.2

CPU times on an IBM 3BT workstation associated with the conventional and DAC fits of $\rho(r)$ for a series of extended glycine polypeptides.

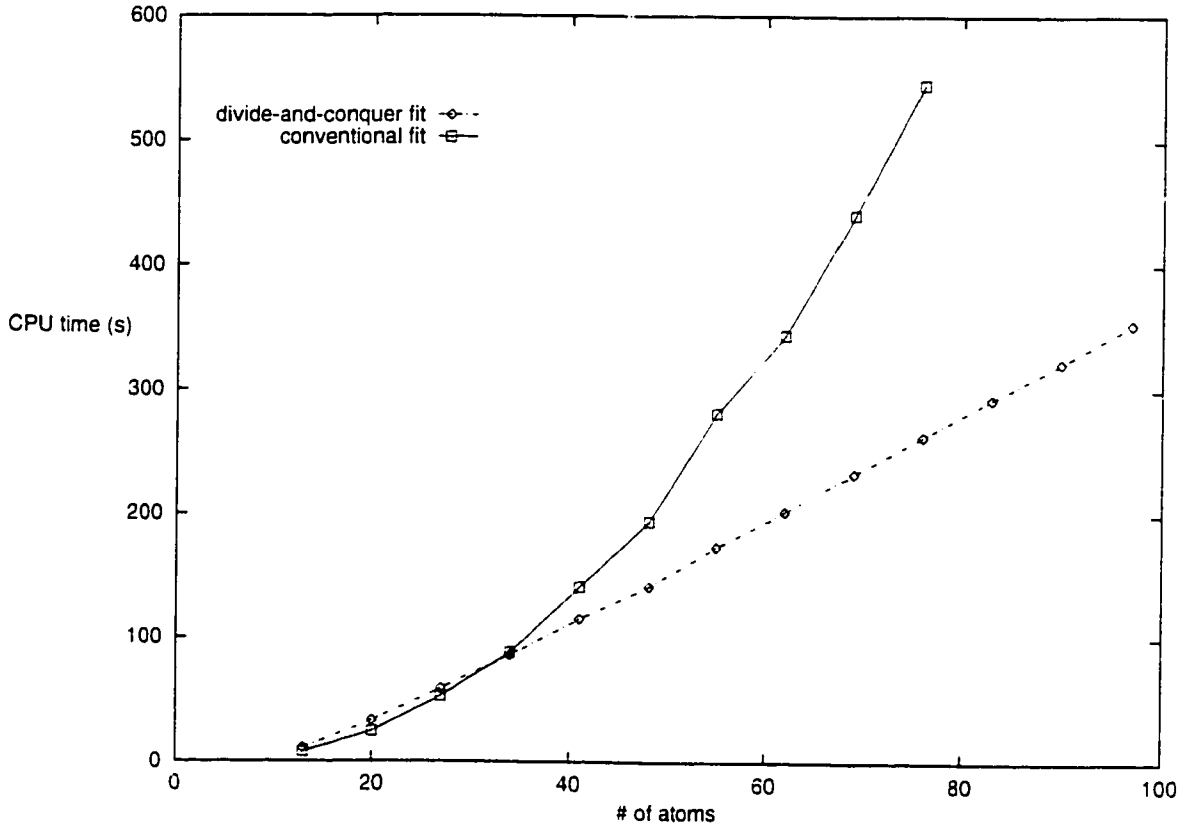


Figure 2.3

CPU times on an IBM 3BT workstation associated with the evaluation of the ESP using various levels of hybrid $\rho(r)$'s and the true $\rho(r)$ for a series of extended glycine polypeptides.

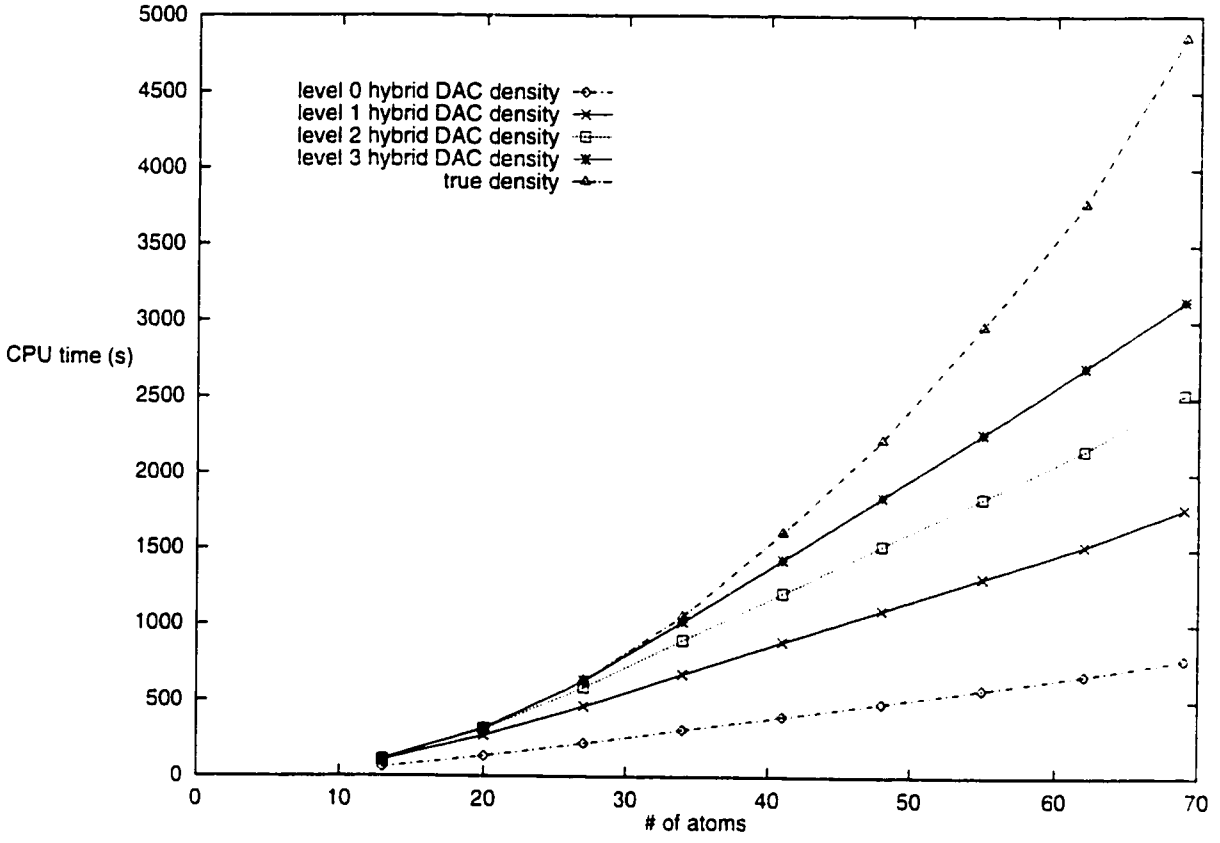


Figure 2.4

The partitioning of the glycine pentapeptide fragment into smaller subsystems.

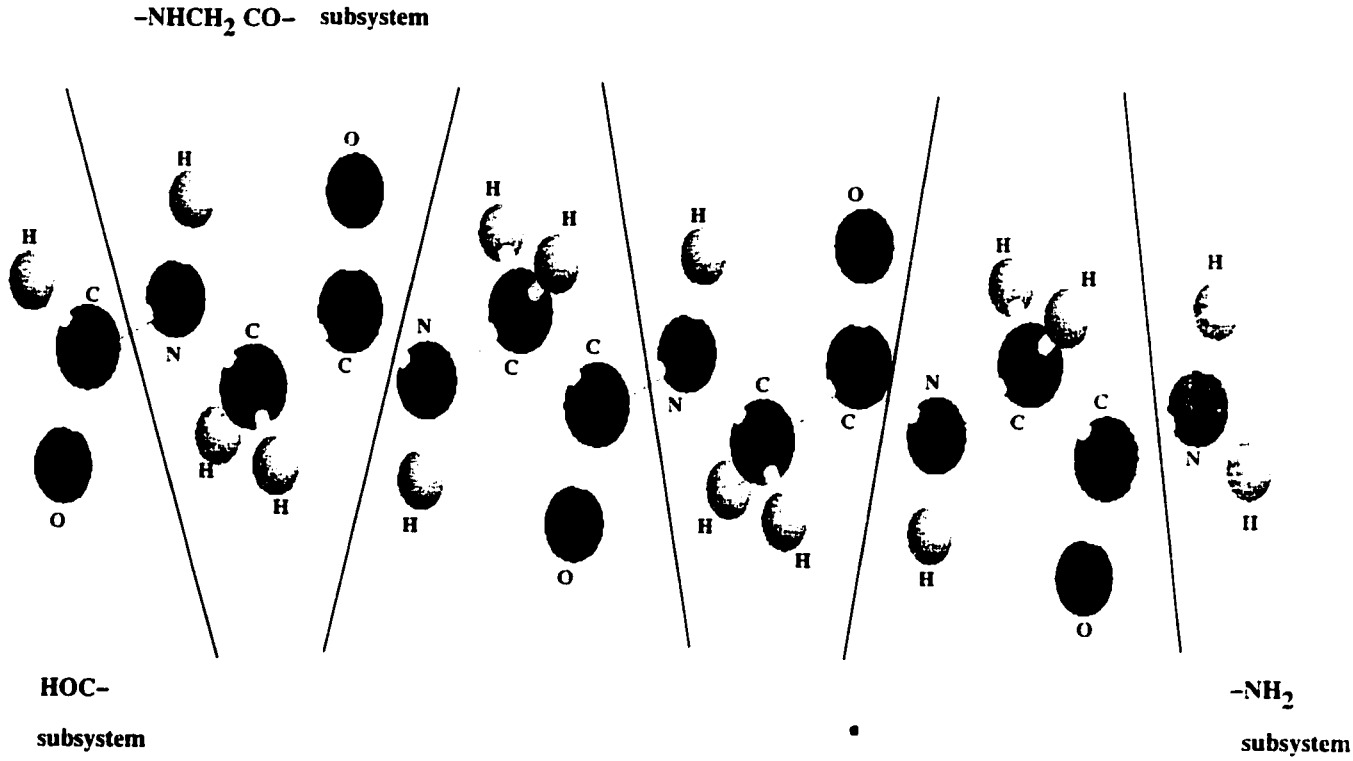
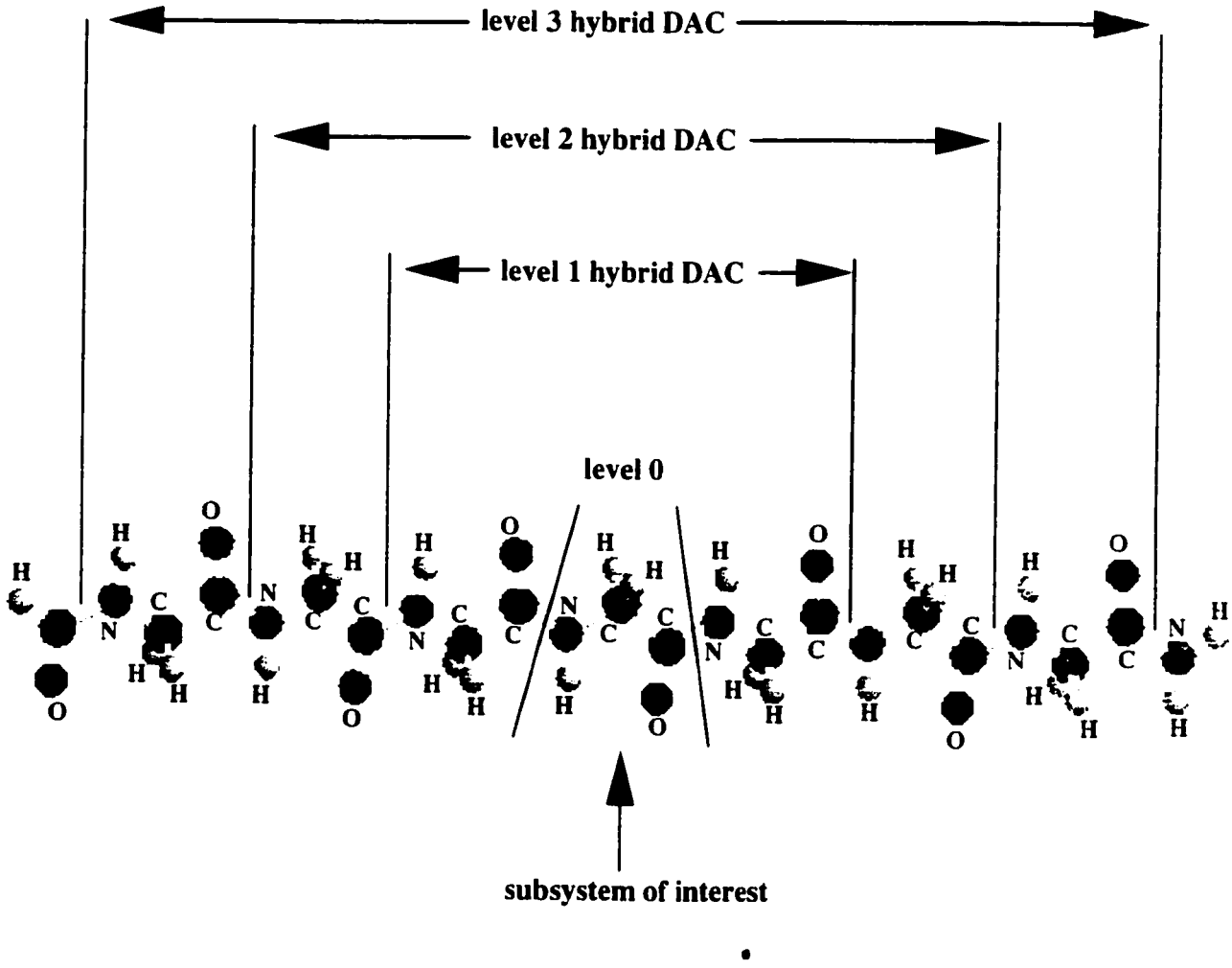


Figure 2.5

Four different hybrid DAC partitioning schemes for the glycine octapeptide.



References

(1) J. Andzelm and E. Wimmer, *J. Chem. Phys.* **96**, 1280 (1992).

(2) A. St-Amant, PhD thesis, University of Montreal, 1992.

(3) A. St-Amant the latest version of DeFT can be downloaded from (http://www.chem.uottawa.ca/DeFT.html).

(4) B. I. Dunlap, J. W. D. Connolly and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

(5) M. Haser and R. Ahlrichs, *J. Comput. Chem.* **10**, 104 (1989).

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

(7) M. Challacombe, E. Schwegler and J. Almlof, *J. Chem. Phys.* **104**, 4685 (1996).

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

(9) N. Godbout, D. R. Salahub and J. Andzelm, *Can. J. Chem.* **70**, 560 (1992).

(10) R. T. Gallant and A. St-Amant, *Chem. Phys. Lett.* **256**, 569 (1996).

(11) W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

(12) W. Yang, *J. Mol. Struct. (THEOCHEM)* **225**, 461 (1992).

(13) R. S. Mulliken, *J. Chem. Phys.* **23**, 1833 (1955).

(14) A. St-Amant, S. K. Goh and R. T. Gallant, in *Recent Developments and Applications of Modern Density Functional Theory*, J. Seminario, Ed.(Elsevier: Amsterdam; 1996).

(15) M. L. Connolly, *J. Appl. Cryst.* **16**, 548 (1983).

(16) U. C. Singh and P. A. Kollman, *J. Comp. Chem.* **5**, 129 (1984).

(17) B. H. Besler, K. M. Merz and P. A. Kollman, *J. Comput. Chem.* **11**, 431 (1990).

(18) S. R. Cox and D. E. Williams, *J. Comput. Chem.* **2**, 304 (1981).

(19) D. E. Williams and J. M. Yan, *Adv. Atomic Mol. Phys.* **23**, 87 (1988).

(20) D. J. Tannor, B. Marten, R. Murphy, R. A. Friesner, D. Sitkoff, A. Nicholls, M. Ringnalda, W. A. Goddard and B. Honig, *J. Amer. Chem. Soc.* **116**, 11875 (1994).

(21) A. Nicholls and B. Honig, *J. Comput. Chem.* **12**, 435 (1991).

Chapter 3

A Scalable Divide and Conquer Algorithm Combining Coarse and Fine Grain Parallelism

3.1 Introduction

Recently, tremendous progress has been made in the development of *ab initio* and DFT methods for electronic structure calculations. Particularly, the very recent advances in linear scaling algorithms [1-5] offer great promise for the realistic modeling of truly large systems. However, such quantum mechanical modeling of large systems still requires an enormous amount of computer time. The prospect of completing these calculations within a reasonable amount of wall-clock (or real) time relies in large part on their ability to exploit massively parallel processor (MPP) architectures. Consequently, the potential of these linear scaling algorithms is largely determined by their parallel scalability, i.e., their ability to reduce wall-clock times by an amount proportional to the number of processors, even when that number is several hundreds, if not thousands. Experience suggests that it is extremely difficult to construct software that will maintain its efficiency as the number of processors is increased. For a given application program, Amdahl's law [6] states that the total CPU time required to execute on a single processor is $T_1 = T_{serial} + T_{parallel}$. When executing over p processors, the total CPU time becomes

$$T_p = T_{serial} + \frac{T_{parallel}}{p} \qquad (3.1)$$

where T_{serial} and $T_{parallel}$ are the CPU times for the serial and the parallel portions of the program. In the above, p is the number of processors. The speed-up of a program is defined as

$$\text{speedup(P)} = \frac{T_1}{T_p} \qquad (3.2)$$

Equations 3.1 and 3.2 suggest that the parallel speedup of a program is limited by the fraction of the code that is run in serial. A plot of the speedup as a function of the

number of processors is given in Figure 1.1. For any fixed size problem, the speedup increases as more processors are added until an optimal number of processors is reached. Adding more processors beyond this optimal value will provide very little to the speedup. In effect, as these processors are added, they are wasted, and the total CPU time collectively consumed increases dramatically. It is extremely difficult to achieve high efficiency over a large number of processors. As an example, consider a program that has 99.9% parallel code (i.e., 0.1% code is serial). When this program is run over 16 processors, a speedup of 15.8 is obtained (i.e., 98.5% parallel efficiency). In essence, an almost perfect efficiency is achieved. However, once the number of processors is increased to 64, we begin to see some degradation in performance. The efficiency achieved decreases to 94%, i.e., a speedup of 60.2. When run on a large size MPP computer, say over 512 processors, a speedup of only 338.8 is obtained for a mere 66.2% efficiency. This means that about 34% of the processors are wasted. Though some programs have reportedly achieved excellent efficiencies on MPP systems, in some cases up to hundreds of processors, the plateau in speedups will inevitably be reached [7-11].

One of the most promising linear scaling methods is the so-called "divide and conquer" (DAC) approach [1, 2]. In this method, a large molecular system is divided into a collection of relatively small subsystems. A set of eigen-equations analogous to those in the conventional quantum method on the global system is independently solved for each subsystem. The results obtained from each subsystem are then pieced together and used to approximate the results for the entire molecular system. Each of the subsystem calculations exhibits the same scaling behaviour as its global counterpart. However, the cost of a subsystem calculation quickly becomes independent of the overall system size

since only those basis functions that are within a cutoff criterion are included in the subsystem calculation. Once the CPU times associated with these localized subsystem calculations become a constant, the total computational effort will only rise linearly with the number of subsystems, and therefore, rises linearly with overall system size. These DAC schemes have successfully been applied within density functional [1, 2] and semi-empirical [12, 13] calculations. DAC approaches for the fits of the electronic density [4, 14] and the exchange-correlation potential [15] have also been used within a density functional program. Though several of the DAC schemes proposed thus far are equally suited for HF and post-HF applications, they have yet to be implemented with these *ab initio* methods. The linear scaling behaviour of DAC schemes has been established, and provided that the system under study is large enough, the DAC approach outperforms the conventional approach when run on a single processor.

The fact that subsystem calculations within the DAC formulation are carried out independently shows that the DAC philosophy is inherently parallel. There are two possible types of parallelism: coarse grain parallelism and fine grain parallelism [16, 17]. Coarse grain parallelism is achieved by assigning a processor to each subsystem and carrying out all subsystem calculations in unison. Fine grain parallelism can, as in the conventional non-DAC calculation, be achieved by dividing the workload of a subsystem calculation among all the available processors, with each subsystem calculation then being carried out in succession. In a later section, it will become apparent that neither of these two approaches is very efficient on an MPP machine. The fine grain approach suffers the inevitable Amdahl's law effect [6] with a large number of processors. Improper load balancing in the coarse grain approach greatly hinders parallel

performance, as subsystem calculations vary considerably in their CPU demands. Shorter subsystem calculations will be completed long before those of the larger subsystems. Their processors will be needlessly idle for significant periods of time while waiting for the larger subsystem calculations to be terminated.

As coarse and fine grain parallelisms are not necessarily mutually exclusive in an algorithm, we have developed an elegant scheme to combine these two different approaches within a DAC philosophy. It can be efficiently run over both a small and a large number of processors. The main feature of our algorithm is its flexibility. The precise approach of our algorithm varies in accordance with the number of processors available and the number (and nature) of subsystems involved. The key to the success of our algorithm is reasonably efficient fine grain processing within individual subsystem calculations. If a single subsystem can be efficiently run over 10-20 processors, our combined coarse/fine grain algorithm should be scalable to several hundreds, if not thousands, of processors when working on truly large systems. Note that our fine grain parallelism need not be so efficient so as to achieve scalability on such a large number of processors by itself. The ability to efficiently exploit MPP machines makes DAC approaches even more appealing on these architectures than they already are on conventional single processor workstations.

Though our new algorithm is easily extended to any DAC approach, as an illustration, we will only focus on employing this algorithm for the DAC fit of a molecule's electronic density [4, 14]. The minor issues that must be addressed before this algorithm is extended to other DAC calculations will be discussed. In the following section, we will show how the efficiency of a particular parallelization scheme for a

given number of processors and subsystems can be easily predicted. Rough guidelines can be established to develop software that will automatically determine the specific approach that is best suited for the given problem and architecture at hand. It will also be demonstrated that a molecule's subsystems can be redefined, if desired, so as to obtain a greater efficiency for a given number of processors.

3.2 Serial Algorithm

The DAC fit of the electronic density has been implemented in our DeFT density functional program. The details of this fitting procedure have been described in Chapter 2. Here, we will briefly summarize the approach. The DAC approach partitioned the true LCGTO electronic density into a sum of subsystem contributions,

$$\rho(\boldsymbol{r}) = \sum_{\alpha}^{subsystems} \rho^{\alpha}(\boldsymbol{r}) = \sum_{\alpha}^{subsystems} \sum_{\mu}^{N} \sum_{\nu}^{N} \boldsymbol{p}_{\mu\nu}^{\alpha} \boldsymbol{P}_{\mu\nu} \chi_{\mu}(\boldsymbol{r}) \chi_{\nu}(\boldsymbol{r}) \quad . \tag{3.3}$$

In the above equation, $\boldsymbol{P}_{\mu\nu}$ is a density matrix element, $\chi_{\mu}(\boldsymbol{r})$ and $\chi_{\mu}(\boldsymbol{r})$ are orbital functions. The density matrix partition function, $\boldsymbol{p}_{\mu\nu}^{\alpha}$, has a value of 1 if both orbital functions are centered on atoms in subsystem α, a value of $\frac{1}{2}$ if only one of the orbital functions is, and 0 if neither of the orbital functions is. The sum of these subsystem densities still recovers the exact LCGTO density. $\boldsymbol{p}_{\mu\nu}^{\alpha}$ localizes $\rho^{\alpha}(\boldsymbol{r})$'s contribution to $\rho(\boldsymbol{r})$ within the vicinity of subsystem α's atoms. To fit this subsystem density accurately, only auxiliary functions centered on subsystem atoms and nearby buffer atoms need be included. Each subsystem fit minimizes

$$\iint \frac{\left[\rho^{\alpha}(\boldsymbol{r}) - \tilde{\rho}^{\alpha}(\boldsymbol{r})\right]\left[\rho^{\alpha}(\boldsymbol{r}') - \tilde{\rho}^{\alpha}(\boldsymbol{r}')\right]}{|\boldsymbol{r} - \boldsymbol{r}'|} \tag{3.4}$$

subject to the constraint that $\tilde{\rho}^{\alpha}(\boldsymbol{r})$ is normalized to the Mulliken population of subsystem α.

Subsystem fits are carried out in exactly the same fashion as the global fit. To obtain fitted densities, two- and three-centered integrals must be evaluated. These molecular integrals can be evaluated using Obara and Saika's (OS) recursive scheme [18]. The recurrence relation for the electron repulsion integrals (ERIs) given by OS was originally for the four-centered integrals encountered in conventional Hartree-Fock calculations,

$$[\boldsymbol{ab} \,\|\, \boldsymbol{cd}] = \iint [\varphi(\boldsymbol{r};\zeta_a,\boldsymbol{a},\boldsymbol{A})\varphi(\boldsymbol{r};\zeta_b,\boldsymbol{b},\boldsymbol{B})]|\boldsymbol{r} - \boldsymbol{r}'|^{-1}[\varphi(\boldsymbol{r}';\zeta_c,\boldsymbol{c},\boldsymbol{C})\varphi(\boldsymbol{r}';\zeta_d,\boldsymbol{d},\boldsymbol{D})]d\boldsymbol{r}d\boldsymbol{r}', \tag{3.5}$$

where each orbital function is described by the position of the electron, an orbital exponent, the angular momentum index, and the function center, respectively. The angular momentum index identifies the nature of the Cartesian Gaussian by listing the angular momentum along the x, y, z axes. For example, a Cartesian d_{xx} Gaussian would have a momentum index of (2,0,0) while a Cartesian d_{yz} would have a index of (0,1,1). Within DeFT, since the use of a fitted density reduces these to three-centered integrals, ζ_d is set to zero. This means that $\varphi(\boldsymbol{r}';\zeta_d,\boldsymbol{d},\boldsymbol{D})$ has a value of one over all space. The function on the third center is also replaced by a fitting function, $\kappa(\boldsymbol{r}';\zeta_c,\boldsymbol{c},\boldsymbol{C})$. This reduces the four-centered integrals of Equation (3.5) to three-centered integrals of the following form

$$[\boldsymbol{ab} \,\|\, \boldsymbol{c}] = \iint [\varphi(\boldsymbol{r};\zeta_a,\boldsymbol{a},\boldsymbol{A})\varphi(\boldsymbol{r};\zeta_b,\boldsymbol{b},\boldsymbol{B})]|\boldsymbol{r} - \boldsymbol{r}'|^{-1}[\kappa(\boldsymbol{r}';\zeta_c,\boldsymbol{c},\boldsymbol{C})]d\boldsymbol{r}d\boldsymbol{r}'. \tag{3.6}$$

These integrals are evaluated via a recursive procedure. This means that angular momentum is "built" on function centers and integrals are obtained from linear combinations of "simpler" integrals (integrals with lower angular momentum). The key recurrence relation for the three-centered integrals is

$$\begin{aligned}[(\boldsymbol{a}+\mathbf{1}_i)\boldsymbol{b}\,\|\,\boldsymbol{c}]^{(m)} &= (\boldsymbol{P}_i-\boldsymbol{A}_i)(\boldsymbol{ab},\boldsymbol{c})^{(m)}+(\boldsymbol{W}_i-\boldsymbol{P}_i)(\boldsymbol{ab},\boldsymbol{c})^{(m+1)} \\ &+\frac{1}{2\zeta}\boldsymbol{N}_i(\boldsymbol{a})\left\{[(\boldsymbol{a}-\mathbf{1}_i)\boldsymbol{b},\boldsymbol{c}]^{(m)}-\frac{\rho}{\zeta}[(\boldsymbol{a}-\mathbf{1}_i)\boldsymbol{b},\boldsymbol{c}]^{(m+1)}\right\} \\ &+\frac{1}{2\zeta}\boldsymbol{N}_i(\boldsymbol{b})\left\{[\boldsymbol{a}(\boldsymbol{b}-\mathbf{1}_i),\boldsymbol{c}]^{(m)}-\frac{\rho}{\zeta}[\boldsymbol{a}(\boldsymbol{b}-\mathbf{1}_i),\boldsymbol{c}]^{(m+1)}\right\} \\ &+\frac{1}{2(\zeta+\zeta_c)}\boldsymbol{N}_i(\boldsymbol{c})[\boldsymbol{ab},(\boldsymbol{c}-\mathbf{1}_i)]^{(m+1)} \qquad . \end{aligned} \tag{3.7}$$

In the above, i is one of the three Cartesian axes, i.e., x, y, or z, and $-\mathbf{1}_i$ subtracts one unit of angular momentum along the appropriate axes from the angular momentum indices $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$. $\boldsymbol{P}_i$, $\boldsymbol{W}_i$, and ζ are defined as follows

$$\boldsymbol{P}_i=\frac{\zeta_a\boldsymbol{A}_i+\zeta_b\boldsymbol{B}_i}{\zeta_a+\zeta_b}, \tag{3.8}$$

$$\boldsymbol{W}_i=\frac{\zeta\boldsymbol{P}_i+\zeta_c\boldsymbol{C}_i}{\zeta+\zeta_c}, \tag{3.9}$$

and

$$\zeta=\zeta_a+\zeta_b. \tag{3.10}$$

Finally, $\boldsymbol{N}_i$ is a generalized Kronecker delta that returns the value of the angular momentum of a given function along a given axis. For example, for a f_{xxz} Cartesian Gaussian, $\boldsymbol{N}_x(f_{xxz})$ would return 2, $\boldsymbol{N}_y(f_{xxz})$ would return 0, and $\boldsymbol{N}_z(f_{xxz})$ would return 1.

To build angular momentum on the fitted function, the recurrence relation is

$$[\boldsymbol{ab}\,\|\,(\boldsymbol{c}+\mathbf{1}_i)]^{(m)}=(\boldsymbol{W}_i-\boldsymbol{C}_i)[\boldsymbol{ab}\,\|\,\boldsymbol{c}]^{(m+1)}$$

$$+\frac{1}{2\zeta_c}N_i(\boldsymbol{c})\left\{[\boldsymbol{ab}\,\|\,(\boldsymbol{c}-\mathbf{1}_i)]^{(m)}-\frac{\rho}{\zeta_c}[\boldsymbol{ab}\,\|\,(\boldsymbol{c}-\mathbf{1}_i)]^{(m+1)}\right\}$$

$$+\frac{1}{2(\zeta+\zeta_c)}\left\{N_i(\boldsymbol{a})[(\boldsymbol{a}-\mathbf{1}_i)\boldsymbol{b}\,\|\,\boldsymbol{c}]^{(m)}+N_i(\boldsymbol{b})[\boldsymbol{a}(\boldsymbol{b}-\mathbf{1}_i)\,\|\,\boldsymbol{c}]^{(m+1)}\right\} \qquad (3.11)$$

The superscript (m) means that integrals with $m = 0$ are true ERIs. Integrals with $m > 0$ are auxiliary integrals (their definition will be given further on). As stated, the above three-centered ERI recurrence relations express a given ERI of higher angular momentum as a linear combination of integrals of lower angular momentum. For example, consider the $[\boldsymbol{p}_i\boldsymbol{p}_j\,\|\,\boldsymbol{s}]$ $(i, j = x, y, z)$ integrals. The recurrence formula expresses this integral as

$$[\boldsymbol{p}_i\boldsymbol{p}_j\,\|\,\boldsymbol{s}]=(\boldsymbol{P}_j-\boldsymbol{B}_j)[\boldsymbol{p}_i\boldsymbol{s}\,\|\,\boldsymbol{s}]^{(0)}-(\boldsymbol{W}_j-\boldsymbol{P}_j)[\boldsymbol{p}_i\boldsymbol{s}\,\|\,\boldsymbol{s}]^{(1)}$$

$$+\frac{\delta_{ij}}{2\zeta}\left\{[\boldsymbol{ss}\,\|\,\boldsymbol{s}]^{(0)}-\frac{\zeta_c}{\zeta+\zeta_c}[\boldsymbol{ss}\,\|\,\boldsymbol{s}]^{(1)}\right\}. \qquad (3.12)$$

Note that in Equation (3.12), the $N_i(a)$ functions have been replaced by the conventional Kronecker delta. In the above equation, intermediary integrals, such as $[\boldsymbol{p}_i\boldsymbol{s}\,\|\,\boldsymbol{s}]^{(m)}$, can be further reduced to intermediary integrals of even lower angular momentum, a set of three $[\boldsymbol{ss}\,\|\,\boldsymbol{s}]^{(m)}$ $(m=0,1,2)$ integrals. It is clear from this example that the OS recursive formula requires explicit formulae for these integrals comprised solely of s type functions. They are

$$[\boldsymbol{ss}\,\|\,\boldsymbol{s}]^{(m)}=(\zeta+\zeta_c)^{-1/2}\boldsymbol{K}_{\boldsymbol{AB}}\boldsymbol{F}_m(\boldsymbol{T}) \qquad (3.13)$$

where

$$\boldsymbol{T}=\frac{\zeta\zeta_c}{\zeta+\zeta_c}(\boldsymbol{P}-\boldsymbol{C})^2, \qquad (3.14)$$

$$\boldsymbol{F}_m(\boldsymbol{T})=\int_0^1 \boldsymbol{t}^{2m}\exp[-\boldsymbol{Tt}]\,\boldsymbol{dt}, \qquad (3.15)$$

and

$$K_{AB} = 2^{1/2} \frac{\pi^{3/4}}{\zeta_a + \zeta_b} \exp\left[-\xi (\boldsymbol{A} - \boldsymbol{B})^2\right], \tag{3.16}$$

$$\xi = \frac{\zeta_a \zeta_b}{\zeta_a + \zeta_b}. \tag{3.17}$$

It has been found that the OS recurrence scheme is most efficient for ERI's involving only *s* and *p* functions. Therefore, the Head-Gordon and Pople (HGP) "horizontal transfer" of angular momentum scheme [19]

$$\left[(\boldsymbol{a} + \mathbf{1}_i)\boldsymbol{b} \| \boldsymbol{c}\right]^{(m)} = \left[\boldsymbol{a}(\boldsymbol{b} + \mathbf{1}_i) \| \boldsymbol{c}\right]^{(m)} - (A_i - B_i)\left[\boldsymbol{ab} \| \boldsymbol{c}\right]^{(m)}, \tag{3.18}$$

is also used to evaluate integrals involving *d* functions or functions of higher angular momentum within DeFT. Note that this approach does not build up angular momentum. It merely transfers it from one center to another. The OS recurrence relations must still be used to build up the required angular momentum on a center. It has been shown that with the HGP method, the number of floating-point operations (FLOPs) is significantly reduce from the original OS method on its own.

3.3 Computational Strategy

DeFT employs contracted Cartesian Gaussians as basis functions. The contracted function, χ_a, takes the form

$$\chi_a(\boldsymbol{r}) = \sum_{k=1}^{L} d_{ak} \varphi_{ak}(\boldsymbol{r}), \tag{3.19}$$

where d_{ak} is the contraction coefficient, L is the length of contraction, and the primitive Cartesian Gaussian centered at $\boldsymbol{A}$ with exponent ζ_k is

$$\varphi_{ak}(\boldsymbol{r}) = (x - A_x)^{a_x}(y - A_y)^{a_y}(z - A_z)^{a_z} \exp\left[-\zeta_k(\boldsymbol{r} - \boldsymbol{A})^2\right], \tag{3.20}$$

where

$$\boldsymbol{a} = (a_x, a_y, a_z) \tag{3.21}$$

is a set of three nonnegative integrals, the sum of which is the angular momentum of the Gaussian. A set of functions associated with the same angular momentum, orbital exponents, contraction coefficients, and the same center $\boldsymbol{A}$ constitute a shell.

The most time-consuming portion of the fitting procedure is the evaluation of the three-centered ERIs required for the construction of the $\boldsymbol{t}$ vector (Equation 2.10). An efficient coding of the ERIs is essential. The overall structure of the ERI program within DeFT is comprised of two main steps. The first step calculates the parameters, $\boldsymbol{P}$, ζ, $\exp[-\xi(\boldsymbol{A} - \boldsymbol{B})^2]$, and contraction coefficient products for each combination of shell pairs. The second step uses recursive formulae (both the OS and HGP methods) to evaluate the three-centered ERIs between the non-vanishing primitive shell pairs (pairs of primitives having an appreciable overlap) and the auxiliary fitting functions. The above steps are done separately for each unique combination of angular momenta in the orbital basis, i.e., for a program that permits only *s*, *p*, and *d* functions, subroutines are required for the *ss*, *ps*, *pp*, *ds*, *dp*, and *dd* combinations of shell pairs. As an illustration, consider the evaluation of the ERI's for the *ps* shell pairs. The subroutines for the first and second steps are shown in Scheme I:

Scheme I (for all integrals between *p* and *s* functions in the orbital basis)

Step I

Do over all the s contracted shells (on center A)

Do over all the p contracted shells (on center B)

Do over all the primitive functions within the s contracted shells

Do over all the primitive functions within the p contracted shells

Calculate the parameters P, ζ, $\exp[-(\boldsymbol{A}-\boldsymbol{B})^2]$, and the products of contraction coefficients for each primitive shell pairs

Discard any primitive shell pairs associated with sufficiently small $\exp[-\xi(\boldsymbol{A}-\boldsymbol{B})^2]$

Continue

Step II

Do over all the auxiliary fitting functions

Do over all the non-vanishing primitive pair shells

Evaluate the three-centered ERIs using the OS and HGP recursive formulae

Continue

As can be seen from Equation 3.7, the same explicit recurrence formulae are required for all ERIs having the same combination of angular momenta (such as [psIIs]), irrespective of function centers. Therefore, identical operations are performed in the second part of the ERI subroutines for each combination of angular momenta. This step is ideally suited for vector processing. DeFT adopts the strategy of explicitly programming each possible combination of angular momenta for ERIs with absolutely no use of logic (i.e., no IF statements). The advantages of such a strategy are twofold. First, explicitly programming the ERIs enables efficient vectorization as IF statements do not allow the compiler to enable vectorization. The second advantage is that the number of FLOPs required for integral evaluations are dramatically reduced. The FLOP counts are dramatically reduced since the majority of the mathematical operations in Equation

3.7 include the generalized Kronecker's delta, $N_i(n)$, many of which have a value of either zero or one. By explicitly programming each unique integral, the $N_i(n)$ function can be explicitly replaced by its numerical value, and when $N_i(n)$ is equal to zero, the accompanying term is simply omitted. Such an approach requires a great deal of code development as many integrals require explicit calculation. As an example, the *dd* subroutines require the explicit programming of 36 [*dd||s*], 108 [*dd||p*], and 216 [*dd||d*] integrals.

Fitting the electronic density with a DAC approach [4] requires only minor changes to the conventional fitting procedure. As presented in Scheme II, the charge fitting routine in the SCF cycle routine is being called within the loop over all the subsystems. Each call performs the fitting of a particular subsystem density with elements spanning only the auxiliary fitting functions of subsystem and buffer atoms.

Scheme II

Do over all the subsystems

    Call the charge density fitting routine, cdftmm (Scheme I)

Continue

3.4 Parallel Algorithm

The construction of a fully parallel distributed-memory version of DeFT is currently in progress. This is being done with the Message-Passing Interface (MPI) [20] implementation developed jointly by Argonne National Laboratories and Mississippi State University.

The ERI program is easily parallelized. For the first step of Scheme I, the outermost loop is partitioned over the available processors. Note that proper load balancing is difficult to achieve since the number of non-vanishing primitive shell pairs is not easy to determine beforehand for a given shell. Luckily, the CPU time required for this step is relatively small compared to the second step of the program. For the second step, the proper load balancing is achieved by partitioning the list of non-vanishing primitive shell pairs over the processors. Again, the subsystem fits are carried out in succession. This strategy is classified as a fine grain approach since only the parallelism within a given subsystem is exploited.

This straightforward fine grain approach performs well if the number of processors is not too large. Otherwise, as dictated by Amdahl's law [6] (see Equation 3.1), the serial portion of the code will dominate the overall CPU time. Therefore, even if the ratio $\frac{T_{parallel}}{T_{serial}}$ is very large, it is extremely difficult to achieve high efficiency on a very large number of processors. Obviously, to effectively exploit machines with hundreds, or even thousands, of processors, the fine grain approach by itself is not the method of choice.

In order to efficiently utilize the enormous computing power provided by MPP machines with hundreds of processors, new algorithms have to be developed. We propose a new algorithm that combines the coarse grain parallelism inherently provided by the DAC philosophy with the fine grain parallelism possible within conventional DFT applications. A combined coarse/fine grain approach performs subsystem calculations in unison where each subsystem calculation is done by a group of processors. Since the subsystems' sizes vary considerably and the scaling within each calculation is cubic, the

computational effort associated with each calculation can be either many times larger, or smaller, than its counterparts. Therefore, it would be extremely inefficient if a similar number of processors is assigned to every subsystem calculation. Instead, a fraction of the total number of processors, resembling as closely as possible the fraction of the total CPU time required by that subsystem calculation, is assigned to that subsystem. A large difference between subsystem times means that efficient load balancing can only be achieved when a reasonably large number of processors become available. For example, if a subsystem calculation is ten times larger than the other, at least eleven processors (ten for the larger subsystem and one for the smaller subsystem) are needed between these two subsystems to achieve optimal load balancing. However, if the number is much smaller than eleven, then processors would necessarily be removed from the larger subsystem calculation as the smaller subsystem must always have at least a single processor at its disposal. Optimal load balancing can no longer achieved.

A way to achieve better load balancing when the ratio of processors to subsystems becomes too small is to perform multiple passes within the combined coarse/fine grain approach. Instead of performing all subsystem calculations in unison, within a single pass, concurrent calculations are performed on only a fraction of the total number of subsystems. The remaining subsystem calculations are handled in subsequent passes. This increases the ratio of processors to subsystems within any one pass. This provides the required flexibility to achieve optimal load balancing. In our simple algorithm, subsystems are reordered, in descending order, according to their expected CPU time on a single processor. The first pass performs n_1 subsystem calculations in unison, followed by n_2 subsystem calculations within the second, and so on. By placing them in order of

descending CPU time, we ensure that subsystem calculations within any one pass are as similar in cost as possible, thus making proper load balancing as easy as possible. We also choose n_1, n_2,....., so that the total CPU times within the passes are as similar as possible. Though this latter point is not an essential part of the algorithm, this decision is made solely to simplify the coding of an automated procedure. In testing the efficiency of the combined coarse/fine grain approach, only the single pass and two pass situations will be considered. Situations where it would become beneficial to use three or more passes will be discussed in a later section. It should be noted that a straightforward fine grain scheme is equivalent to the combined coarse/fine grain scheme where the number of passes is equal to the number of subsystems since $n_1 = n_2 = \ldots = 1$.

In using this combined scheme, an advanced knowledge of the approximate cost of each subsystem calculation is crucial. Otherwise, optimal load balancing cannot be achieved through a coarse grain distribution of the subsystem calculations over appropriately sized clusters of processors. Fortunately, these DAC methods are being employed within an iterative self-consistent field (SCF) procedure. Information collected from a previous iteration can be used to establish the times that will be required for subsystem calculations in the current iteration. The CPU time associated with each subsystem is barely affected from iteration to iteration. A reasonably accurate estimate of the time required by a subsystem calculation should be fairly easy to establish for the first iteration. In our test calculations on the DAC fit of $\rho(r)$, given the number of auxiliary fitting functions being used and the number of non-vanishing primitive shell pairs in the orbital basis, a reasonably accurate estimate of the time required by a subsystem calculation can be made. Since the success of a combined coarse/fine grain scheme

depends largely on the ratios between subsystem times (and not the absolute values), the formulae used to make these estimates need not be changed upon going from one computer platform to another. In our test case, to be presented shortly, subsystem times obtained from a previous run on a single processor will be used. Considering the fact that subsystem calculations are not 100% parallelized, our algorithm can be further refined by using Amdahl's law to take into account drops in efficiency over larger numbers of processors when assigning processors to subsystems. This modification, however, will only begin to significantly alter the actual number of processors being assigned when the ratio of processors to subsystems is exceedingly large. However, the goal of our flexible algorithm is precisely to avoid such a situation, whenever possible. Therefore, assigning processors to subsystems will be based on the assumption that each subsystem calculation is 100% efficient over the cluster of processors assigned to it.

A combined coarse/fine grain approach can be easily incorporated within an existing code with a DAC fitting procedure. It has already been established that DAC schemes can be implemented within existing codes with relative ease. Once a parallel version (a straightforward fine grain approach) of the DAC fitting program is constructed, our combined coarse/fine grain algorithm can also be easily incorporated. We have chosen to parallelize our code within an MPI programming environment since it supports modular programming and it has now become the industry standard. The MPI communicator mechanism can be used to create a local communication space for a subset of processors assigned to a common subsystem. This allows different components of a program to run concurrently without the danger of mixing up data associated with each. A combined coarse/fine grain approach can be carried out either within a single pass or

over multiple passes. Here, we will describe the algorithm for a general multiple pass approach. The algorithm for this section of the code is shown in scheme III. A combined coarse/fine grain approach simply replaces the outer loop of scheme II by a loop over the number of passes. Within each pass, a local communication space for each subsystem calculation is created (of course, the number of processors assigned to each subsystem has to be predetermined).

SCHEME III

```
Do over all passes
Do over all subsets of processors assigned to subsystem calculations
        Do over all processors in subset
            Enroll processor into a group
        Continue
Continue
Do over all subsets of processors assigned to subsystem calculations
        Create a new group using only members enrolled in the above step
        Create a new communicator for this new group
Continue
Do over all subsets of processors assigned to subsystem calculations
            Assign each subsystem within the pass to a new communicator
Continue
        Call charge density fitting routine, cdftmm (scheme I)
        Destroy the new group communicator
Continue
```

This is done as follows. First, processors are enrolled into groups in numbers predetermined by the relative subsystem CPU times. New groups are then created from the default group (the entire set of processors) and assigned an MPI communicator. With

each group of processors and its associated new communicator, the fitting of a subsystem electronic density is carried out using virtually the same MPI code as for the straightforward approach. The only change is to replace the default communicator associated with the entire collection of processors, MPI_COMM_WORLD, by the new communicator associated with the subset of processors assigned to the subsystem calculation, MPI_COMM_GROUP. Once each subgroup of processors completes its subsystem calculation, its associated communicator is destroyed. In subsequent passes, new communicators are created, as appropriate, for the given number of subsystems in that pass and their relative CPU times.

Finally, the issue of memory and disk storage must be addressed. The serial code of a DAC fitting procedure stores much of the information required for a subsystem calculation (most importantly, the inverse of the auxiliary basis' overlap matrix) on disk. In general, the memory capacity of a single processor is about 64 - 256 MB. Therefore, it is impossible to store all this information in memory except for a relatively small molecule. Our combined coarse/fine grain algorithm is designed for a parallel machine, especially for a distributed memory MIMD machine. In a distributed memory MIMD machine, *each* processor has its own private memory of 64 – 256 MB. To exploit the vast amount of total memory available when running on many such processors, our code has been modified. If the number of processors available for subsystem calculations is greater than or equal to the number of subsystems, all required information will be stored in memory rather than disk. This approach is entirely scalable since the memory requirement for each subsystem is independent of overall system size. In other words, no matter how large the system, all information for a single subsystem will fit in the memory

available on a single node. Storing information in memory minimizes disk I/O operations by making full use of the available memory. Superlinear speedups are thus often observed in the following section. A superlinear speedup over n processors is a speedup by a factor m, where m is grater than n. Similar modifications have been made in other scientific codes, and they are responsible for the superlinear speedups often seen when these codes are ported to massively parallel platforms [20].

3.5 Computational Details

Benchmark calculations are performed on the PM3 optimized extended conformations of the glycine pentapeptide (34 atoms), heptapeptide (48 atoms), and nonapeptide (62 atoms). Two partitioning schemes, A and B, are tested. Each polypeptide in partitioning scheme A is divided into $-NHCH_2CO-$ subsystems, a terminal HCO- subsystem, and a terminal $-NH_2$ subsystem. Partitioning scheme B further subdivides each of the $-NHCH_2CO-$ groups into NH, CH_2, and CO fragments. A 5.0 Å cutoff is used for the DAC fit of $\rho(\boldsymbol{r})$ within both partitioning schemes. A (7111/411/1*) orbital basis is used for heavy atoms, and a (41/1*) orbital basis is used for hydrogen atoms [21]. The fitting basis for heavy atoms contains 7 s functions, 3 sets of p functions, and 3 sets of d functions. To make integral evaluation more efficient, the three p and d sets of functions, along with three of the s functions, are constrained to have the same exponents. A similar constraint is also in place for the fitting basis of a hydrogen atom, whose auxiliary basis consists of four s functions, a single set of p functions, and a single set of d functions [21]. Test calculations are performed on a Cray T3E computer

with up to 48 processors. The speedup of the parallel implementation is measured relative to the same parallel implementation on a single processor.

3.6 Results and Discussion

The wall-clock time required by each subsystem fit on a single Cray T3E processor is listed in Table 3.1. Note that only the heptapeptide's results are shown for partitioning scheme B. The wall-clock times in Table 3.1 include both the CPU time and the disk I/O time required to read the inverse of the subsystem's overlap matrix. Using partitioning scheme A for the pentapeptide, heptapeptide, and nonapeptide, the sums of the subsystem I/O times add up to roughly 2.0 *s*, 4.3 *s*, and 5.6 *s*, respectively. The I/O time roughly increases linearly with system size. This reflects the fact that a maximum number of auxiliary basis functions within any one subsystem fit has been attained and the I/O cost is now rising only with the addition of more subsystems. Likewise, the CPU time associated with each subsystem calculation has almost attained a maximum value. To support this observation, from Table 3.1, we see that within partitioning scheme A, the longest subsystem time observed for the heptapeptide (23.5 s) is far more similar to that observed for the nonapeptide (24.1 s) than it is to that observed for the pentapeptide (21.1 s). Many of the central $-NHCH_2CO-$ subsystems have exactly the same number of auxiliary basis functions assigned to their buffer space. However, those at the very center of the polypeptide have slightly longer wall-clock times as they have a greater number of non-vanishing primitive pairs arising within the orbital basis. In a more general, globular, peptide, the maximum time associated with any one subsystem calculation

would be considerably higher. Nevertheless, their subsystem times would also quickly achieve a maximum value and the overall CPU time would increase linearly with system size. From Table 3.1, we also observe much smaller subsystem time within partitioning scheme B. However, the sum of the CPU times for the three smaller fragments (NH, CH_2, and CO) originating from each $-NHCH_2CO-$ group is roughly equal to the CPU time observed for the corresponding $-NHCH_2CO-$ subsystem within partitioning A. Overall, it seems that partitioning scheme B is slightly more efficient (for the heptapeptide, 117.3 s as opposed to 121.0 s). From our experience, the overall efficiency of a DAC fit of $\rho(\boldsymbol{r})$ on a single processor is fairly insensitive to the choice of subsystems (so long as the subsystems' sizes do not cover a significant fraction of the total system, otherwise, the cubic scaling of a conventional non-DAC approach would again arise). This insensitivity to the chosen partitioning scheme has also been observed previously over a range of sizes for these glycine polypeptides [14]. An important observation to be taken from Table 3.1 is that, within any polypeptide, the individual subsystem CPU times can span a full order of magnitude. Consequently, this presents a great load balancing challenge. Finally, within partitioning scheme A, we note that the total wall-clock times associated with subsystem fits exhibit essentially perfect linear scaling. Upon going from the pentapeptide to the heptapeptide and the heptapeptide to the nonapeptide, two extra $-NHCH_2CO-$ subsystems are added in each instance, and the total CPU time increases by 48.3 seconds in each instance. The increase in CPU time is independent of the overall size of the system. This is the essence of linear scaling. Linear scaling is also observed within scheme B.

Table 3.1 also shows the CPU times for the conventional, non-DAC, fits of $\rho(\boldsymbol{r})$. The CPU time requirements are 73.2 s for the peptapeptide, 163.8 s for the heptaptide, and 281.4 s for the nonapeptide (note that the times reported neglect the time spent inverting the overlap matrices of the auxiliary bases before the start of the SCF procedure). These times scale slightly worse than quadratically with system size. Without efficient screening of the primitive pairs in the orbital bases, these times would scale cubically with system size.

The parallel speedup curves for the extended polypeptides listed in Table 3.1 are shown in Figures 3.1 to 3.4. Note that only when the number of processors is equal to or greater than the number of subsystems will a single pass combined coarse/fine grain approach be attempted. A minimum of one processor per subsystem is a necessary condition for a single pass combined coarse/fine grain approach. The more general rule of thumb is that the minimum number of processors required is the number of subsystems to be tackled within a single pass. Therefore, the minimum number of processors required to attempt the two pass combined coarse/fine grain approach is equal to one half the total number of subsystems. In the speedup curves for the straightforward fine grain scheme, a noticeable jump in parallel efficiency is observed whenever the number of processors equals, or becomes greater than, the number of subsystems. These superlinear speedups (the speedup is greater than the number of processors) result from the program's switch from storing all the necessary information for the subsystem fits on disk to storing it all in memory. A great deal of I/O time is eliminated once this switch to storing information in high speed memory is made.

Figure 3.1 displays the speedup curves for the pentapeptide using partitioning scheme A. With the speedup curve for fine grain parallelization, we see that a reasonable level of parallel efficiency is maintained up to 24 processors. A speedup of 23.3, corresponding to an efficiency of 97.8%, is obtained for 24 processors. However, once the number of processors goes beyond 24, the efficiency of the fine grain approach drops dramatically. When running on 48 processors, a speedup of only 33.8, corresponding to an efficiency of a 70.4%, is obtained.

The single pass combined coarse/fine grain approach on the pentapetide starts off, as expected, with a poor performance over eight processors. Since each system requires at least one processor, the best way to distribute eight processors over six subsystems is to assign an extra processor to each of the two most time-consuming subsystems. However, even if the times for these two subsystems are halved, each still requires about 10 s, compared to only 2 s for the shortest subsystem calculation. We see that optimal load balancing is far from being achieved. As more and more processors become available, the parallel efficiency of a single pass approach improves dramatically. From an efficiency of only 68.8% over 8 processors, the efficiency of the combined coarse/fine grain approach increases sharply to roughly 90% for 16 to 24 processors. The actual efficiency depends on how well load balancing can be achieved through coarse grain parallelism with the given number of processors. Using a larger number of processors in our proposed algorithm does not necessarily mean that better load balancing will be achieved. As an illustration, suppose that we have two subsystem fits of equal complexity running on 16 processors, a 100% parallel efficiency can theoretically be achieved. However, this perfect efficiency can not be achieved when run over 17

processors. Since our algorithm can only assign an integer number of processors to a subsystem, one of the two subsystems will get an extra processor, and perfect load balancing can no longer be maintained. In general, however, more processors provide greater flexibility in achieving optimal load balancing through coarse grain partitioning of subsystems over groups of processors. Consequently, a higher efficiency is often observed. For the pentapeptide, with a single pass combined coarse/fine grain approach, once the processor count reaches 32, 40 and 48, a sufficient number of processors is available to provide considerable flexibility in load balancing. Hence, near perfect efficiencies are observed over 32, 40, and 48 processors. This translates to speedups of 31.3, 40.0, and 47.8. Had it been possible to store in memory all information required for the subsystem fits and eliminate the disk I/O time when run on a single processor (we again point out that this is quite impossible considering the limited memory available on a single node), the speedups would drop to 30.4, 38.8, and 46.5. These are still excellent results. Analyzing the results further, we see that when running on 48 processors, no subsystem calculation is performed on more than 14 processors. The number of processors assigned to subsystems, in descending order are, 14, 13, 9, 8, 2, and 2, in accord with their wall-clock times on a single processor. Amdahl's law should therefore pose no great impact on any of the subsystem calculations, especially when taking into account that the smaller subsystems, those most susceptible to poor scalability, have the least number of processors assigned to them.

Within the two pass approach, the first pass handles only the two largest subsystems. The remaining four are handled in the second pass. From the speedup curve of Figure 3.1, we see that performing the combined coarse/fine grain approach over two

passes has a better overall parallel efficiency than over a single pass. Speedups of 2.9, 7.9, 17.2, 25.0, 32.8, 40.2, and 47.8 are obtained over 4, 8, 16, 24, 32, 40, and 48 processors, respectively. Again, superlinear speedups arise from the elimination of disk I/O time (beyond 4 processors in this case), and excellent results are still maintained up to 48 processors as in the single pass run. However, at this point, each of the subsystem calculations within the first pass is performed on 24 processors. This is very near the point where Amdahl's law will begin causing a noticeable reduction in the parallel efficiency of a subsystem calculation. In the second pass, 21, 19, 5, and 3 processors are assigned to the remaining four subsystems. Though not as many processors are assigned to individual subsystem calculations within the second pass, it too is facing a problem. Problems arise because many processors are assigned to smaller subsystem calculations. Hence, the effects of Amdahl's law may play an even greater role considering the smaller size of the subsystems. When going beyond 48 processors, the parallel efficiency is expected to decline at a rate similar to that seen in the straightforward fine grain approach beyond 24 processors. The single and two pass versions of the combined coarse/fine grain approach to the pentapeptide's subsystem fits have virtually identical efficiencies on 48 processors. Beyond 48 processors, however, the single pass scheme will be more efficient. No individual subsystem calculation will be performed on more than 24 processors within the single pass scheme until we reach about 90 processors. Amdahl's law should therefore not come into play until we reach this point. Consequently, if a single pass approach is attempted on 90 processors for the pentapeptide, it is reasonable to expect that its parallel efficiency will be very much similar to that of the two pass approach on 48 processors, i.e., very near 100%.

The parallel performances for the heptapeptide and nonapeptide systems using partitioning scheme A are shown in Figures 3.2 and 3.3, respectively. A general observation is that the overall efficiency of a straightforward fine grain approach is little affected by the size of the global system. Since the computational burden associated with each subsystem is independent of the overall system size and the ratio $\frac{T_{parallel}}{T_{serial}}$ remains virtually unchanged, we should expect the same efficiency regardless of the size of the global system. However, for a single pass combined coarse/fine grain approach, the speedups obtained for both the heptapeptide and the nonapeptide are not as good as the speedups for the pentapeptide. For any given number of processors, the processor to subsystem ratio becomes smaller upon going to larger systems. Hence, it is harder to achieve optimal load balancing through coarse grain parallelism. Comparing the performance on 48 processors, the heptapeptide obtained a speedup of 46.3, while the pentapeptide obtained a speedup of 47.8. The speedup decreases even further, to 45.6, upon going to the nonapeptide. However, as more processors are added, it will be easier for the nonapeptide to achieve better load balancing, and near ideal speedups would be expected up to roughly 170 processors. At that point, still no more than 24 processors are being assigned to the largest subsystem calculations, and Amdahl's law has yet to take its toll on efficiency. In the two pass approach, however, no obvious drops in efficiency are observed upon going to larger systems. Speedups of 48.4 and 47.5 on 48 processors are obtained, respectively, for the heptapeptide and the nonapeptide. By breaking up the run into two passes, we are ensuring that the processor to subsystem ratio within any single pass is sufficiently large to achieve adequate load balancing. In fact, despite a smaller processor to subsystem ratio for the heptapeptide, it has a better speedup than the

pentapeptide. This could be due to the fact that fewer processors are being assigned to each subsystem calculation. Twenty-four processors are assigned to each of the two largest subsystems for the pentapeptide within the first pass. For the heptapetide, the three largest subsystems within the first pass are assigned, in descending order, 17, 16, and 15 processors. In the second pass, the largest number of processors assigned to any subsystem is 21 for the pentapeptide, and 19 for the heptapeptide. Therefore, the heptapeptide is less affected by Amdahl's law within individual subsystem calculations. They should be even less affected within the nonapeptide calculation. Nevertheless, a slightly smaller speedup is observed for the nonapeptide. This probably arises from the smaller processor to subsystem ratio, and load balancing is thus harder to achieve when assigning processors to subsystems.

Figure 3.4 shows the speedup curves for the heptapeptide using partitioning scheme B. It is obvious that partitioning scheme B is not preferred when computing over a large cluster of processors. This is in contrast to the results shown in Table 3.1 that indicated using smaller subsystems in scheme B makes it marginally more efficient than scheme A when computing on a single processor. For the straightforward fine grain approach, a speedup of only 21.4 is obtained on 48 processors. In fact, the speedup barely increases from 40 to 48 processors. It is doubtful that the speedup would ever be greater than 22, regardless of how many more processors are made available. This is considerably worse than the speedups obtained for the heptapeptide using partitioning scheme A. However, this is to be expected. Most of the subsystems within scheme B are much smaller than they are within scheme A. Since each subsystem calculation requires very little CPU time, it is harder to achieve high parallel efficiency within a

straightforward fine grain approach. The overhead associated with executing the serial code at the beginning of each subsystem calculation plays a far more important role. This will always occur with partitioning scheme B, independent of the overall system size.

The results are considerably better using a combined coarse/fine grain approach. However, they are still not nearly as good as those obtained within scheme A. The ratio of processors to subsystems is considerably lower within scheme B than it is within scheme A. As a result, load balancing is harder to achieve and a lower efficiency results. A considerable improvement in efficiency, 79.3% to 89.9%, is observed upon going from 40 to 48 processors within a single pass. This indicates that the overall efficiency is still highly sensitive to the processor to subsystem ratio and improvement can still be brought to efficiency by adding more processors. The ratio of processors to subsystems is much higher when performing a two pass approach. Hence, a slightly better overall efficiency is observed. Particularly, on 40 and 48 processors, efficiencies of 91.5%, and 94.8% are obtained, respectively. This strongly supports the previous observation that many more processors can still be added before calculations using partition scheme B within a single pass approach reach their optimal efficiencies. Unfortunately, we do not have the resources to test out this hypothesis. Assume that, within scheme B, Amdahl's law will only begin to seriously affect a subsystem calculation when more than 12 processors are used for any given one. We assumed 24 for scheme A since larger subsystem calculations have better scalability (see Figures 3.2 and 3.4). By extrapolation, near linear speedups should be observed up to 140 processors with the combined coarse/fine grain approach using a single pass. As for scheme A, assuming that Amdahl's law only becomes a problem on more than 24 processors, near linear speedups should be

maintained up to roughly 125 processors. Also taking into account the fact that scheme B provides a slightly lower overall time on a single processor, scheme B should become the method of choice if such a large number of processors do become available.

3.7 Conclusion

From these preliminary benchmarks calculations (the actual results shown for up to a maximum of 48, and our extrapolations beyond 48 processors), we establish that a DAC approach to quantum mechanical calculations could be exploited to achieve high efficiency over a very large number of processors. However, the precise method of choice depends entirely on the number of processors available. With a small number of processors, a straightforward fine grain approach is more efficient than the combined coarse/fine grain schemes. Once more processors become available, the combined schemes are far more efficient. The poor scalability of a straightforward fine grain approach arises from the fact that each subsystem calculation is severely hampered by the effects of Amdahl's law. If the processor to subsystem ratio is relatively small, multiple passes are preferred within the combined coarse/fine grain scheme. By so doing, the processor to subsystem ratio is increased and better load balancing is achieved within each individual pass. However, when a larger number of processors becomes available, it will be best to reduce the number of passes. By reducing the processor to subsystem ratio, we prevent the Amdahl's law from becoming a greater problem in any one subsystem calculation. Through the benchmarking of the various parallelization schemes, the results provide us with insights as to how many processors are needed before a particular approach becomes more efficient than another. These insights may be

used to design a section of computer code that will automatically determine the preferred approach for the conditions under which the program is running. This will ensure that near linear speedups are maintained on any number of processors, without resorting to user intervention. This is yet to be implemented within DeFT.

It is important to note that our preliminary results were obtained for relatively small systems. These are a more stringent test of a code's scalability over a large number of processors than are large systems. Memory constraints in some of DeFT's modules, unrelated to the DAC procedures, prevent us from carrying out test calculations on larger systems. Linear systems were studied so as to establish linear scaling with as small systems as possible. However, we fully anticipate that linear scaling would also be observed for 3D systems, though much larger systems would have to be studied before linear scaling became evident. With only 48 processors available, adequate load balancing can be achieved for much larger systems through multiple passes. If more processors do become available, near linear speedups should be maintained up to several hundreds, if not thousands, of processors. If the largest number, P, of processors for which linear speedups are still possible within a straightforward fine grain approach is known, a reliable estimate can made of the total number of processors that can be exploited in an efficient manner within a combined fine/coarse grain scheme. This is done by finding out at which point any one of the subsystems (in fact, the most CPU time demanding) in the single pass combined coarse/grain approach will be assigned to more than P processors. Obviously, the scalability of the combined coarse/fine grain approach would greatly be extended whenever P's value increases. Slight improvements in the

fine grain parallelization can consequently introduce huge improvements within our new combined coarse/fine grain algorithms.

Though, we have only applied these combined schemes to the DAC fit of the electronic density, they can be extended just as efficiently to the DAC fit of the exchange-correlation potential and the DAC construction of the electronic density. It is important to note that the allocation of processors to subsystems employed for the fit of the electronic density need not necessarily be the same within the other DAC procedures. Finally, the efficiency of any combined coarse/fine grain approach depends entirely on the efficiencies of the subsystem calculations within the fine grain scheme only. The less efficiently a straightforward fine grain approach works within individual subsystem calculations, the more important it is to exploit the inherent coarse grain parallelism within the DAC method. The fact that a DAC algorithm can provide for linear scaling in total CPU time with system size, combined with the scalability of the combined coarse/fine grain schemes, means that our DAC new algorithm is ideally suited for applications on truly large systems on today's massively parallel supercomputer architectures.

Table 3.1

Wall-clock time (in seconds, single Cray T3E processor) required by each subsystem in a divide and conquer fit of $\rho(r)$ for a series of extended glycine polypeptides.

Subsystem #	Pentapeptide[a]	Heptapeptide[a]	Heptapeptide[b]	Nonapeptide[a]
1	3.0	3.1	3.1	3.2
2	13.4	13.5	2.5	13.6
3	21.1	21.9	4.8	22.1
4	20.8	23.5	6.9	23.7
5	12.4	23.3	4.9	24.1
6	2.0	21.5	7.2	23.7
7		12.2	8.6	23.3
8		2.0	5.2	21.3
9			7.7	12.3
10			9.9	2.0
11			5.4	
12			7.7	
13			8.8	
14			5.1	
15			6.6	
16			7.8	
17			4.4	
18			5.0	
19			3.7	
20			2.0	
Total	72.7	121.0	117.3	169.3
Conventional[c]	73.2	163.8	163.8	281.4

[a] Using partitioning scheme A

[b] Using partitioning scheme B

[c] Time required for a conventional LCGTO-DFT fit of $\rho(r)$

Figure 3.1

Speedup on a Cray T3E for the DAC fit of $\rho(r)$ for the glycine pentapeptide using partitioning scheme A.

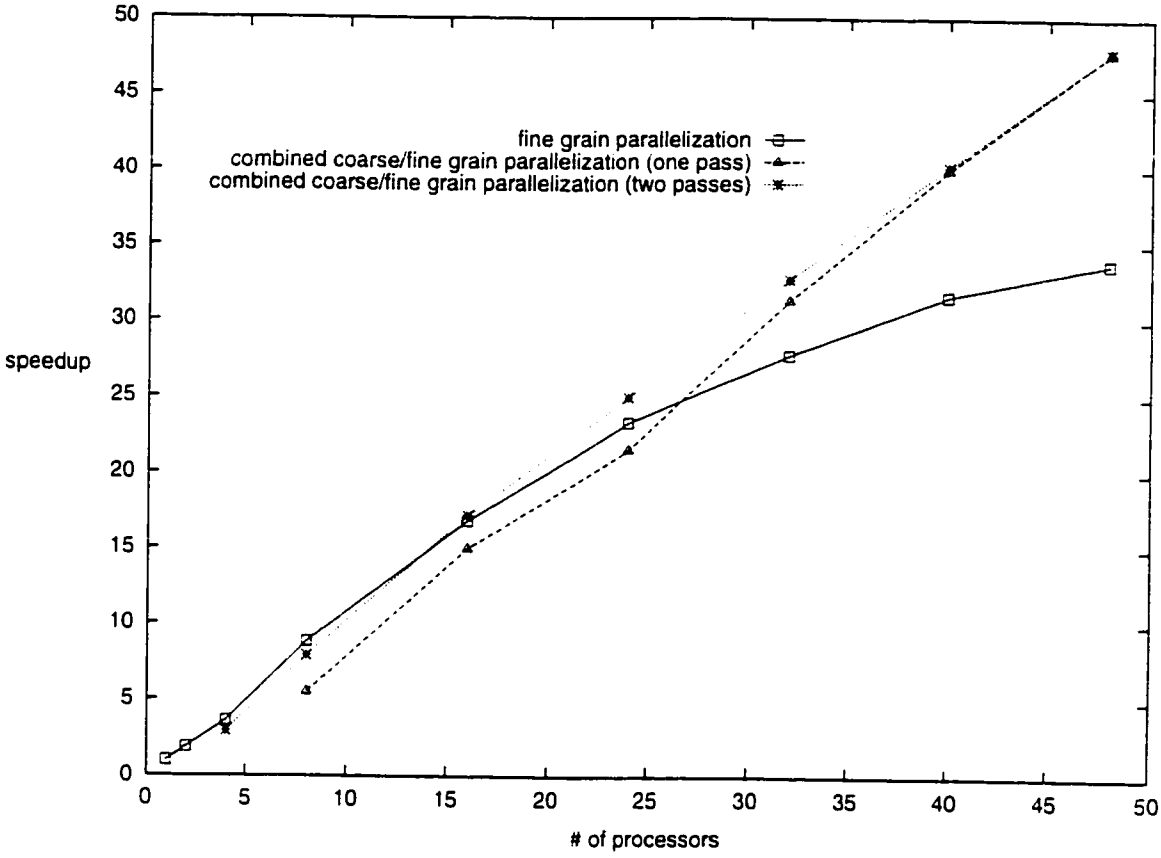


Figure 3.2

Speedup on a Cray T3E for the DAC fit of $\rho(r)$ for the glycine heptapeptide using partitioning scheme A.

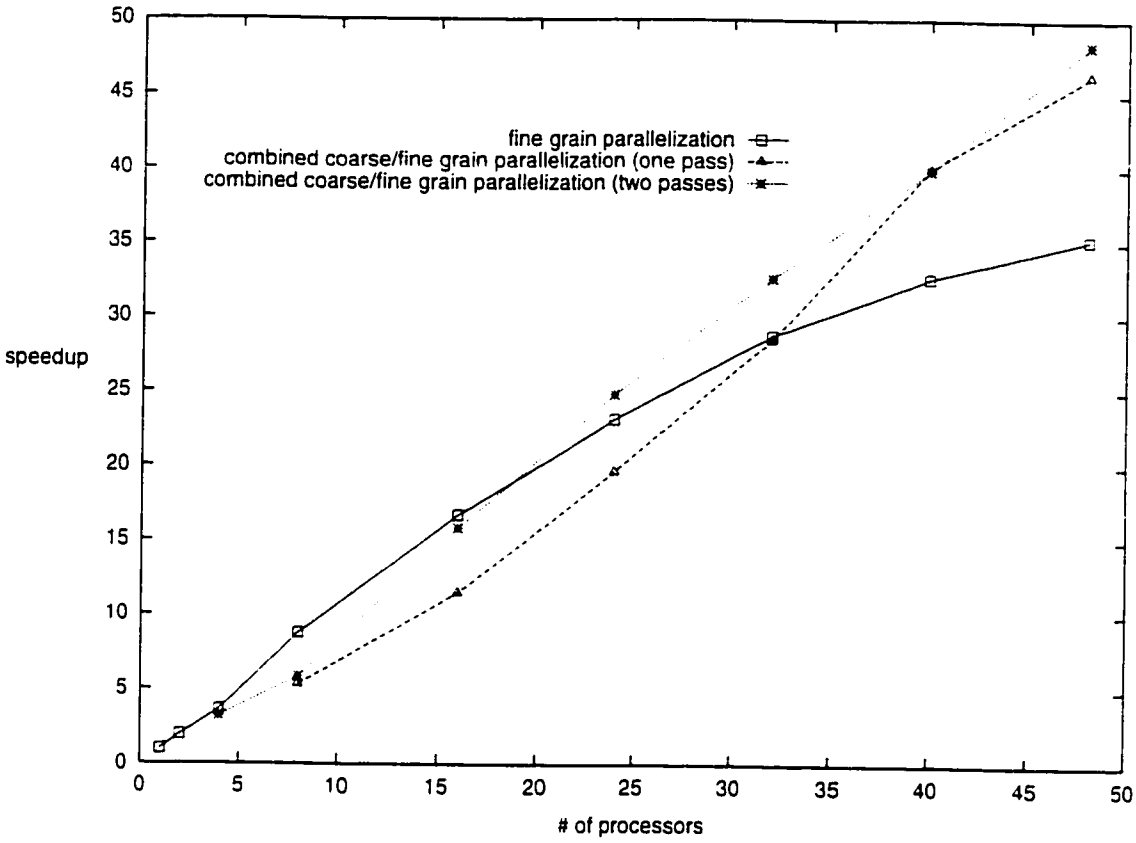


Figure 3.3

Speedup on a Cray T3E for the DAC fit of $\rho(r)$ for the glycine nonapeptide using partitioning scheme A.

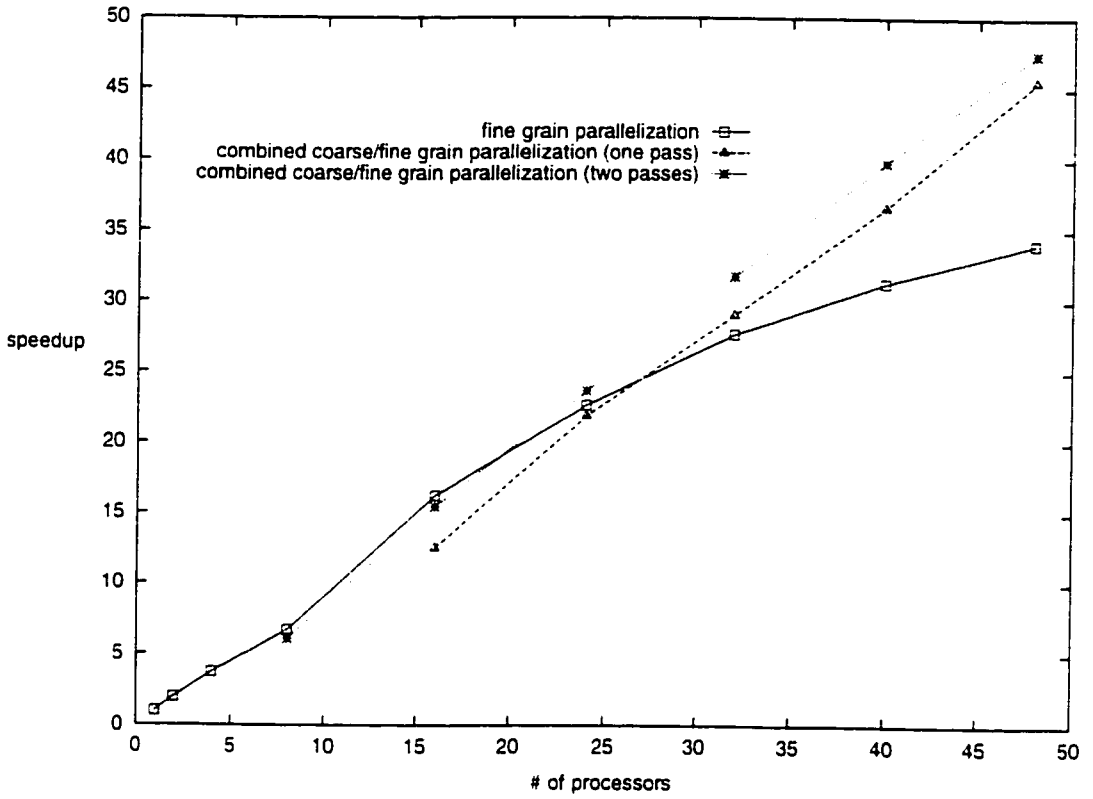
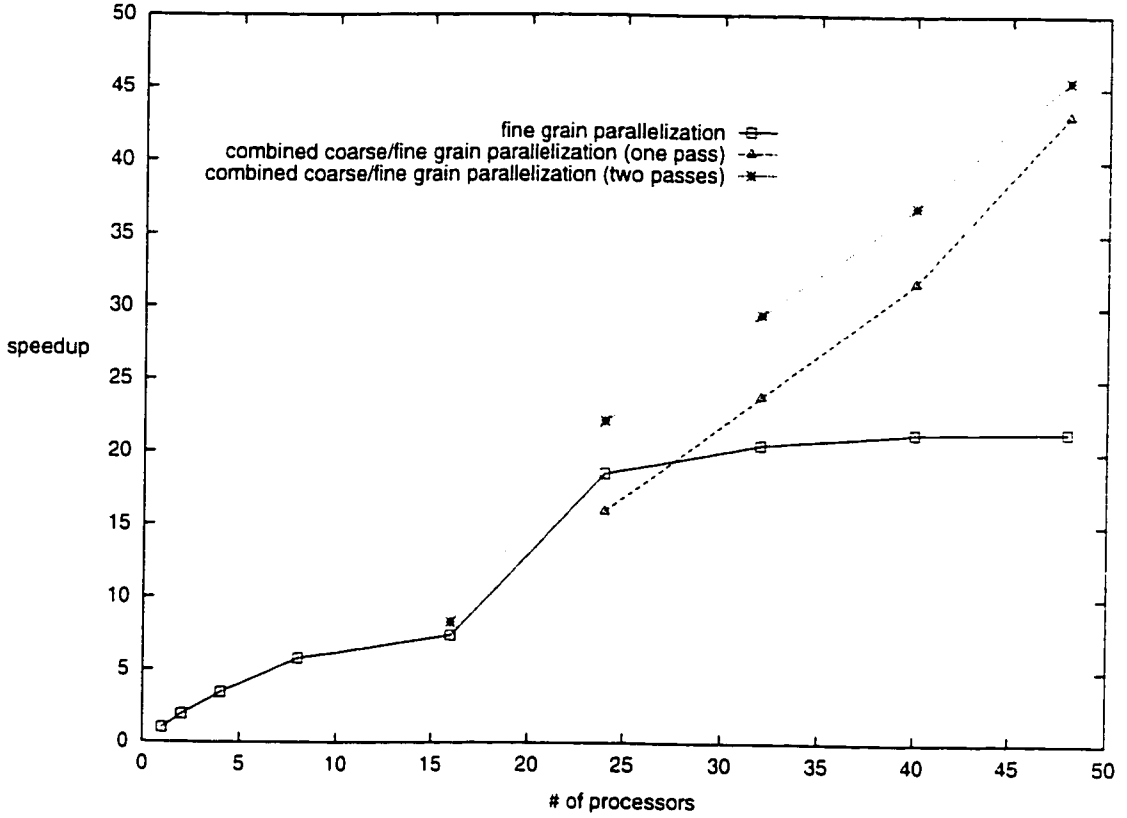


Figure 3.4

Speed up on a Cray T3E for the DAC fit of $\rho(r)$ for the glycine heptapeptide using partitioning scheme B.



References

(1) W. Yang, *J. Mol. Struct. (THEOCHEM)* **225**, 461 (1992).

(2) W. Yang and T. S. Lee, *J. Chem. Phys.* **103**, 5674 (1995).

(3) M. Challacombe, E. Schwegler and J. Almlof, *J. Chem. Phys.* **104**, 4685 (1996).

(4) R. T. Gallant and A. St-Amant, *Chem. Phys. Lett.* **256**, 569 (1996).

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

(6) G. Amdahl, *AFIPS Comput. Conf.* **30**,483 (1967).

(7) R. A. Kendall, R. J. Harrison, R. J. Littlefield and M. F. Guest, in *Reviews in Computational Chemistry*, **Vol.** 6, K. B. Lipkowitz and D. B. Boyd, Eds.(VCH Publisher; New York; 1995).

(8) D. E. Bernholdt, E. Apra, H. A. Fruchtl, M. F. Guest, R. J. Harrison, R. A. Kendall, R. A. Kutteh, X. Long, J. B. Nicholas, J. A. Nichols, H. L. Taylor, A. T. Wong, G. I. Fann, R. J. Littlefield and J. Nieplocha, *Int. J. Quantum Chem. Symp.* **29**, 475 (1995).

(9) M. W. Schmidt, K. K. Baldridge, J. J. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis and J. A. Montgomergy, *J. Comput. Chem.* **14**, 1347 (1993).

(10) M. Fayereisen and R. A. Kendell, *Theor. Chim. Acta* **84**, 289 (1993).

(11) M. E. Colvin, C. L. Janssen, R. A. Whiteside and C. H. Tong, *Theor. Chim. Acta* **84**, 301 (1993).

(12) S. L. Dixon and K. M. Merz, *J. Chem. Phys.* **104**, 6643 (1996).

(13) T. S. Lee, D. M. York and W. Yang, *J. Chem. Phys.* **105**, 2744 (1996).

(14) S. K. Goh and A. St-Amant, *Chem. Phys. Lett.* **264**, 9 (1997).

(15) S. K. Goh, R. T. Gallant and A. St-Amant, *Int. J. Quantum Chem.* **69**, 405 (1998).

(16) L. Baker and B. J. Smith, in *Parallel Programming*, (McGraw-Hill; New York; 1996).

(17) T. G. Mattson, In Parallel Computing in Computational Chemistry, ACS Symposium series, M. J. Comstock, ed. (Am. Chem. Soc.; Washington, DC; 1995

(18) S. Obara and A. Saika, *J. Chem. Phys.* **84**, 3963 (1986).

(19) M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* **89**, 5777 (1988).

(20) W. Gropp, E. Lusk and A. Skjellum, in *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, (MIT Press; Cambridge; 1995).

(21) N. Godbout, D. R. Salahub and J. Andzelm, *Can. J. Chem.* **70**, 560 (1992).

Chapter 4

Toward Linear Scaling with Fitted Exchange-Correlation Terms in the LCGTO-DF Method via a Divide and Conquer Approach

4.1 Introduction

Density functional theory (DFT) based methods have gained widespread acceptance within the chemistry and biochemistry communities [1-4] in recent years. The principal advantage of such methods is their ability to include electron correlation without the unfavorable scaling problems of the HF based methods. The set of working equations in the linear combination of Gaussian-type orbital (LCGTO) implementation of DFT is very similar to the HF equations, except that the HF exchange potential is replaced by an exchange-correlation (XC) potential. Unlike the traditional *ab initio* methods, the exchange-correlation potential and energy density in the LCGTO equations cannot be fully treated by analytical means. These terms are commonly treated using numerical integration on a three dimensional grid of points. Programs such as DGauss [5], deMon [6], and DeFT [7], however, take a less straightforward approach. XC terms are fitted numerically within an auxiliary set of basis functions. The fitted terms are then integrated analytically. The reasoning for a two-step approach is as follows. It has been established that significantly less grid points are needed for the fitting procedure than for the integration procedure [8, 9]. The total energy is less susceptible to grid noise if grid points are only used for fitting purposes. However, regardless of how the XC terms are handled, the computational cost formally scales cubically with the size of the molecule.

As mentioned in earlier chapters, only when linear scaling is achieved will realistic modeling of truly large molecules be possible. The scaling properties in the current LCGTO-DFT implementations are affected by three fundamental procedures. Recently, the first of these procedures, construction of the two-electron Coulomb matrix, has been reduced to near linear scaling using fast multipole methods [10-12]. For the

second procedure, diagonalization of the KS matrix, the cubic scaling of this procedure can be eliminated with Yang's divide and conquer (DAC) scheme [13, 14]. Recently, Scuseria has introduced a linear scaling conjugate gradient optimization of the density matrix[15], eliminating the need to diagonalize the Fock matrix. For those DFT methods that employ density fitting procedures, a fitting technique that scales linearly with system size, based on the divide and conquer concept, has also been developed [16]. As for the remaining procedure, the numerical quadrature of the XC energy functional and XC potential, its scaling has recently been addressed by Stratmann, Scuseria, and Frisch [17]. They propose a new weighting scheme for density functional quadrature that is faster and more efficient than the widely used algorithm of Becke [18]. By using a microbatching algorithm (points in a localized region of space are handled concurrently) together with the new weight function, they have demonstrated that near linear scaling can be achieved for all steps involved in the XC quadrature. The exact procedure will be outlined in this chapter.

Some of these exciting improvements brought to the numerical integration of the XC terms have been implemented in our DeFT program. The new weighting scheme of Stratmann, Scuseria, and Frisch [17] has so far been only tested on the straightforward numerical integration of the XC quadrature. However, DeFT requires these XC terms to first be fit within an auxiliary basis. We will also like this fitting step to scale linearly with system size. Just as the fits of $\rho(\boldsymbol{r})$ can be done in a linear fashion with a divide and conquer (DAC) approach [16], a DAC procedure can similarly be developed [19] for the XC fitting procedure. The accuracy and the scaling properties of these new DAC procedures will be presented. The details of our DAC XC fitting procedure will be

described. The advantages, and disadvantages, to fitting the XC terms, rather than performing straightforward numerical integration, will be discussed. The error introduced by these DAC XC fitting procedures and their CPU time requirements will be examined.

4.2 Fitting of the XC Terms

The XC contribution to the KS matrix elements and the total energy require the following integrals,

$$\int \chi_\mu(\boldsymbol{r}) v_{xc}^{\sigma}(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \tag{4.1}$$

and

$$\int \rho(\boldsymbol{r}) \varepsilon_{xc}(\boldsymbol{r}) d\boldsymbol{r} \ . \tag{4.2}$$

Here $\chi_\mu(\boldsymbol{r})$ and $\chi_\nu(\boldsymbol{r})$ are the basis functions in the LCGTO orbital basis. $v_{xc}^{\sigma}(\boldsymbol{r})$ is the XC potential associated with spin $\sigma(\alpha \text{ or } \beta)$, and $\varepsilon_{xc}(\boldsymbol{r})$ is the XC energy density. Due to the complex nature of the XC functionals, these XC integrals in equations 4.1 and 4.2 are, in practice, evaluated by numerical integration,

$$\int \chi_\mu(\boldsymbol{r}) v_{xc}^{\sigma}(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \approx \sum_{I}^{grid} \chi_\mu(\boldsymbol{R}_I) v_{xc}^{\sigma}(\boldsymbol{R}_I) \chi_\nu(\boldsymbol{R}_I) W_I \tag{4.3}$$

and

$$\int \rho(\boldsymbol{r}) \varepsilon_{xc}(\boldsymbol{r}) d\boldsymbol{r} \approx \sum_{I}^{grid} \rho(\boldsymbol{R}_I) \varepsilon_{xc}(\boldsymbol{R}_I) W_I \ , \tag{4.4}$$

over the ensemble of grid points positioned at $\boldsymbol{R}_I$ with quadrature weights W_I.

However, in DeFT, the XC terms are first fitted numerically within an auxiliary basis of M uncontracted Gaussians, $\chi''(\boldsymbol{r})$,

$$v_{xc}^{\sigma}(\boldsymbol{r}) \approx \tilde{v}_{xc}^{\sigma}(\boldsymbol{r}) = \sum_{l}^{M} \mathrm{b}_{l}^{\sigma} \chi_{l}''(\boldsymbol{r}) \tag{4.5}$$

and

$$\varepsilon_{xc}(\boldsymbol{r}) \approx \tilde{\varepsilon}_{xc}(\boldsymbol{r}) = \sum_{l}^{M} \mathrm{e}_{l} \chi_{l}''(\boldsymbol{r}). \tag{4.6}$$

This XC auxiliary basis is different from that used in the $\rho(\boldsymbol{r})$ fitting procedure. The fit coefficient vectors, $\mathbf{b}^{\sigma}$ and $\mathbf{e}$, are obtained via a least squares fitting procedure that minimizes the errors in the XC terms,

$$\sum_{l}^{grid} \left[\tilde{v}_{xc}^{\sigma}(\boldsymbol{R}_l) - v_{xc}^{\sigma}(\boldsymbol{R}_l)\right]^2 W_l \tag{4.7}$$

and

$$\sum_{l}^{grid} \left[\tilde{\varepsilon}_{xc}(\boldsymbol{R}_l) - \varepsilon_{xc}(\boldsymbol{R}_l)\right]^2 W_l, \tag{4.8}$$

over the entire set of grid points. The solutions for the fit coefficient vectors have the following forms,

$$\mathbf{b}^{\sigma} = \mathbf{S}^{-1}\mathbf{t}^{v_{xc}^{\sigma}}, \tag{4.9}$$

and

$$\mathbf{e} = \mathbf{S}^{-1}\mathbf{t}^{\varepsilon_{xc}}. \tag{4.10}$$

The elements of the XC overlap matrix, $\mathbf{S}$, are given by

$$S_{lm} = \sum_{l}^{grid} \chi_{l}''(\boldsymbol{R}_l)\chi_{m}''(\boldsymbol{R}_l)W_l, \tag{4.11}$$

and the elements of $\mathbf{t}^{v_{xc}^{\sigma}}$ and $\mathbf{t}^{\varepsilon_{xc}}$ are given by

$$t_I^{v_{xc}^\sigma} = \sum_I^{grid} \chi_I(\boldsymbol{R}_I) v_{xc}^\sigma(\boldsymbol{R}_I) \boldsymbol{W}_I \tag{4.12}$$

and

$$t_I^{\varepsilon_{xc}} = \sum_I^{grid} \chi_I(\boldsymbol{R}_I) \varepsilon_{xc}(\boldsymbol{R}_I) \boldsymbol{W}_I \,. \tag{4.13}$$

With these XC fitted terms, the approximate XC integrals can then be integrated analytically

$$\int \chi_\mu(\boldsymbol{r}) \tilde{v}_{xc}^\sigma(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} = \sum_I^M \boldsymbol{b}_I \int \chi_I(\boldsymbol{r}) \chi_\mu(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \tag{4.14}$$

and

$$\int \rho(\boldsymbol{r}) \tilde{\varepsilon}_{xc}(\boldsymbol{r}) d\boldsymbol{r} = \sum_\mu^K \sum_\nu^K \boldsymbol{P}_{\mu\nu} \sum_I^M \boldsymbol{e}_I \int \chi_I(\boldsymbol{r}) \chi_\mu(\boldsymbol{r}) \chi_\nu(\boldsymbol{r}) d\boldsymbol{r} \,. \tag{4.15}$$

Note that equations 4.9 and 4.10 require the inverse of the XC overlap matrix. This proves to be problematic as inversion scales cubically with system size. A DAC approach will allow us to circumvent this problem.

4.3 DAC Fitting of the XC Terms

The total density of a large system, in a DAC scheme, is expressed as a sum of subsystem contributions,

$$\rho(\boldsymbol{r}) = \sum_\alpha^{subsystems} \rho^\alpha(\boldsymbol{r}) \,. \tag{4.16}$$

Using this scheme to fit the XC terms introduces a new problem. Since the dependence on the electronic density, $\rho(\boldsymbol{r})$, in the $v_{xc}^\sigma(\boldsymbol{r})$ terms is not linear, one cannot simply construct the subsystem densities, $\rho^\alpha(\boldsymbol{r})$, and obtain their corresponding subsystem XC

potentials, $v_{xc}^{\sigma^\alpha}(\boldsymbol{r})$. To make this point clearer, consider a simple Slater exchange [20] potential which is proportional to $\rho^{1/3}$. Clearly, $(\rho^\alpha + \rho^\beta)^{1/3} \neq (\rho^\alpha)^{1/3} + (\rho^\beta)^{1/3}$. Instead, the total density, $\rho(\boldsymbol{r})$, must first be synthesized over the entire set of grids points. The electronic density at a grid point, $\boldsymbol{R}_l$, is expressed as

$$\rho(\boldsymbol{R}_l) = \sum_{\mu}^{K} \sum_{\nu}^{K} P_{\mu\nu} \chi_\mu(\boldsymbol{R}_l) \chi_\nu(\boldsymbol{R}_l). \tag{4.17}$$

Once $\rho(\boldsymbol{R}_l)$ is known at each grid point, then $v_{xc}^{\sigma}(\boldsymbol{R}_l)$ may be evaluated. Afterwards, one may choose to decompose the XC potential,

$$v_{xc}^{\sigma}(\boldsymbol{R}_l) = \sum_{\alpha}^{subsystems} v_{xc}^{\sigma^\alpha}(\boldsymbol{R}_l), \tag{4.18}$$

into a sum of subsystem contributions. This is done using the partitioning function, $g^\alpha(\boldsymbol{r})$, of Yang's density formulation [14] of DAC DFT. It is defined as

$$g^\alpha(\boldsymbol{r}) = \frac{\sum_{i \in \alpha}^{atoms} \left[\rho_i^{atomic}\left(\left|\boldsymbol{R}_i - \boldsymbol{r}\right|\right)\right]^2}{\sum_{j}^{atoms} \left[\rho_j^{atomic}\left(\left|\boldsymbol{R}_j - \boldsymbol{r}\right|\right)\right]^2}. \tag{4.19}$$

This partition function clearly obeys the constraint,

$$\sum_{\alpha}^{subsystems} g^\alpha(\boldsymbol{r}) = 1. \tag{4.20}$$

In equation 4.19, the summation in the denominator runs over all the atoms in the molecule, while that in the numerator only runs over those atoms that are within subsystem α. $\rho_j^{atomic}\left(\left|\boldsymbol{R}_j - \boldsymbol{r}\right|\right)$ is the spherical atomic density of atom j (positioned at $\boldsymbol{R}_j$) at point $\boldsymbol{r}$ in space. These spherical atomic densities are tabulated within the DeFT program and were obtained from prior atomic DFT calculations for all the elements.

A subsystem's contribution to the XC terms can now be expressed as

$$v_{xc}^{\sigma^\alpha}(\boldsymbol{r}) = g^\alpha(\boldsymbol{r}) v_{xc}^{\sigma}(\boldsymbol{r}) \tag{4.21}$$

and

$$\varepsilon_{xc}^{\alpha}(\boldsymbol{r}) = g^\alpha(\boldsymbol{r}) \varepsilon_{xc}(\boldsymbol{r}) . \tag{4.22}$$

The partition function, $g^\alpha(\boldsymbol{r})$, localizes subsystem α's contribution about the atoms of subsystem α. Therefore, only those auxiliary XC fitting functions that are centered on subsystem α's atoms and nearby "buffer" atoms [14, 16] are needed for the fits of $v_{xc}^{\sigma^\alpha}(\boldsymbol{r})$ and $\varepsilon_{xc}^{\alpha}(\boldsymbol{r})$.

Each subsystem fit follows very closely the conventional XC fitting procedure. The expressions in equations 4.7 and 4.8 become

$$\sum_{l}^{grid} \left[\tilde{\varepsilon}_{xc}^{\alpha}(\boldsymbol{R}_l) - g^\alpha(\boldsymbol{R}_l) \varepsilon_{xc}(\boldsymbol{R}_l) \right]^2 W_l \tag{4.23}$$

and

$$\sum_{l}^{grid} \left[\tilde{v}_{xc}^{\sigma^\alpha}(\boldsymbol{R}_l) - g^\alpha(\boldsymbol{R}_l) v_{xc}^{\sigma}(\boldsymbol{R}_l) \right]^2 W_l . \tag{4.24}$$

Note that both quantities in equation 4.23 and 4.24 formally sum over the entire set of grid points. However, since $g^\alpha(\boldsymbol{R}_l)$ survives only at positions where the square of the spherical atomic densities of subsystem α's atoms fall within a predetermined threshold value, only those points in the vicinity of subsystem α need to be considered. This number of grid points for each subsystem is independent of the total size of the system.

The above least squares fitting procedures lead to the following solutions for the subsystem fit coefficient vectors,

$$\mathbf{b}^{\sigma^\alpha} = (\mathbf{S}^\alpha)^{-1} (\mathbf{t}^\alpha)^{v_{xc}^{\sigma^\alpha}} \tag{4.25}$$

and

$$\mathbf{e}^{\alpha} = (\mathbf{S}^{\alpha})^{-1}(\mathbf{t}^{\alpha})^{\varepsilon_{xc}^{\alpha}} . \tag{4.26}$$

Here, the elements in the overlap matrix for subsystem α, $\mathbf{S}^{\alpha}$, are equal to their global counterparts, given in equation 4.11. However, the matrix now spans only those fitting functions that are actually involved in the subsystem fit. The elements of the subsystem's $\mathbf{t}^{\alpha}$ vectors differ from their global counterparts by a factor $\mathbf{g}^{\alpha}(\mathbf{r})$,

$$(t^{\alpha})_l^{\varepsilon_{xc}^{\alpha}} = \sum_{I}^{grid} \chi_l(\boldsymbol{R}_I)\varepsilon_{xc}(\boldsymbol{R}_I)\boldsymbol{g}^{\alpha}(\boldsymbol{R}_I)\boldsymbol{W}_I \tag{4.27}$$

and

$$(t^{\alpha})_l^{v_{xc}^{\sigma\alpha}} = \sum_{L}^{grid} \chi_l(\boldsymbol{R}_I)v_{xc}^{\sigma}(\boldsymbol{R}_I)\boldsymbol{g}^{\alpha}(\boldsymbol{R}_I)\boldsymbol{W}_I . \tag{4.28}$$

The DAC approximations to the conventional $\mathbf{b}^{\sigma}$ and $\mathbf{e}$ vectors can then be obtained by summing up the subsystem fit coefficients,

$$\mathbf{e} \approx \mathbf{e}^{\mathrm{DAC}} = \sum_{\alpha}^{subsystems} \mathbf{e}^{\alpha} \tag{4.29}$$

and

$$\mathbf{b}^{\sigma} \approx \mathbf{b}^{\sigma^{\mathrm{DAC}}} = \sum_{\alpha}^{subsystems} \mathbf{b}^{\sigma^{\alpha}} . \tag{4.30}$$

The final expression for the DAC approximation to $\varepsilon_{xc}(\boldsymbol{r})$ becomes

$$\varepsilon_{xc}(\boldsymbol{r}) \approx \sum_{l}^{M} e_l^{DAC}\chi_l(\boldsymbol{r}), \tag{4.31}$$

and that for $v_{xc}^{\sigma}(\boldsymbol{r})$ becomes

$$v_{xc}^{\sigma}(\boldsymbol{r}) \approx \sum_{l}^{M} b_l^{\sigma^{DAC}}\chi_l(\boldsymbol{r}). \tag{4.32}$$

It is important to note that the final DAC approximation to the XC terms has three sources of errors. The first two sources, an imperfect grid and an incomplete fitting basis, are also present in the conventional XC fitting procedure. The third source is the DAC truncation of the XC fitting basis. Distant XC fitting functions that would otherwise contribute to the fit of a localized subsystem contribution, although to a very small degree, are omitted from the subsystem fit. The last source of error may be eliminated entirely by extending the buffer space to include all XC fitting functions. However, by so doing, nothing would be gained from the DAC approach.

To define the buffer space cutoff, a distance, rather than a topological, criterion will be employed. A distance criterion is better suited for globular systems as two atoms may be in close proximity even through a large number of covalent bonds separates them. If the distance between an atom to any one of the subsystem's atoms falls within the buffer space cutoff, its XC fitting functions will be included in that subsystem fitting procedure. The shorter the cutoff, the quicker the calculation will be able to achieve linear scaling as system size increased. The results of benchmark calculations on extended glycine polypeptide with a number of different buffer space cutoffs will be presented further on.

4.4 Grid Design and Numerical Integration

In the DeFT program, the fitting of the XC terms and the numerical integration of the final XC energy both require a grid. A new linear scaling procedure [17] has recently been added to the latest version of DeFT to generate a new set of grid points. Also, larger

grids, with more radial and angular points, have been added to DeFT. These new augmented grids are better as they reduce grid "noise" and therefore achieve greater precision in the final results. The nature of these new grids will now be outlined.

To construct a finite grid around each atom, a spherical polar coordinate system (r, ϑ, ϕ) first has to be defined. A series of radial shells are generated within the coordinate system. On each radial shell, an angular grid is defined. Since the electron density distribution varies more rapidly near the nuclei, one should design a grid that takes this into account. Close to the nuclei, the radial shells are densely packed. Since the density near the nucleus is more spherically symmetric, each radial point contains relatively few angular points. Further away from the nuclei, the radial shells are spaced more widely. However, to reflect the greater angular anisotropy, there is a higher number of angular points on each shell. These radial shells are generated by a standard Gauss-Chebyshev [21] quadrature scheme. To apply this scheme, the standard Gauss-Chebyshev interval $-1 \le x \le 1$ must first be mapped onto the radial interval $0 \le r \le \infty$ [18] by the following transformation

$$r_i = r_m \frac{(1 + x_i)}{(1 - x_i)}. \qquad (4.33)$$

This transformation produces a "telescopic" radial grid that has very tight grid spacing near the nucleus, with progressively wider spacing as we move further away. In the above equations, x_i is a Gauss-Chebyshev quadrature point, defined by

$$x_i = \cos\left(\frac{\pi i}{N+1}\right). \qquad (4.34)$$

where N is the number of radial points. The parameter, r_m, which enables the radial grid to adapt to the size of the atom, is taken as [18] half of the covalent radius of the atom

about which the grid is currently centered. The associated weight for the Gauss-Chebyshev points are scaled to reflect the thickness of the radial shell over the integration interval. As for the angular grid, any set of points and weights designed for quadrature along the surface of a unit sphere may be used. In this chapter, angular grids of 12, 26, 50, 110, and 194 points, will be investigated.

Once the grid for each atom within a molecule is constructed, a molecular grid may then be generated by superimposing all these atomic grids. As a result, each atom's grid now overlaps with that of every other atom. To avoid the double counting of space, the associated weights of the grid points must be readjusted. This is done by assigning a weight function, w_k, to each nucleus k such that

$$\sum_{k}^{atoms} w_k(\boldsymbol{r}) = 1. \tag{4.35}$$

In principle, we would like to see $w_k(\boldsymbol{r})$ have a value near one in the vicinity of atom k and a value close to zero near others atoms. For the sake of numerical stability, the weight functions should smoothly drop from values near one to values near zero. There should be no discontinuous jumps. The range of a grid about an atom still goes from 0 to ∞. In essence, the molecular space is divided into "fuzzy", overlapping, analytically continuous cells. These are the so-called "fuzzy" Voronoi polyhedra of Becke [18].

Various efficient schemes to construct Voronoi polyhedra have been proposed. The most popular approach among them is that proposed by Becke [18]. To construct Voronoi polyhedra, Becke introduced a two-center system, known as confocal elliptical coordinates (λ, μ, ϕ). At each grid point, $\boldsymbol{r}_g$, the key variable, μ, between all pairs of atoms is defined as

$$\mu_{ij} = (r_{ig} - r_{jg})/R_{ij} \,. \tag{4.36}$$

Here r_{ig} and r_{jg} are the distances from the grid point (g) to atoms i and j, and R_{ij} is the internuclear separation. μ_{ij} has a value of -1 on center i and along the nuclear axis beyond i. μ_{ij} has a value of 1 on center j and along the nuclear axis beyond j. Between atoms i and j, μ falls smoothly from -1 at atom i to $+1$ at atom j. Along the perpendicular bisector of R_{ij}, μ_{ij} has a value of exactly 0. Using a step function,

$$s(\mu_{ij}) = \begin{cases} 1, & -1 \le \mu_{ij} \le 0 \\ 0, & 0 \le \mu_{ij} \le +1 \end{cases}, \tag{4.37}$$

a Voronoi polyhedron about nucleus i can be defined as

$$P_i(r) = \prod_{j \neq i} s(\mu_{ij}) \,. \tag{4.38}$$

Becke referred to this product as a "cell function". It has a value of one if r lies inside the cell. It vanishes if r lies outside. The step discontinuity at $\mu_{ij} = 0$ can be "smoothed out" into analytically continuous, mutually overlapping, regions with the aid of the function

$$h(\mu_{ij}) = \tfrac{3}{2}\mu_{ij} - \tfrac{1}{2}\mu_{ij}^3 \,. \tag{4.39}$$

Note that this function possess properties

$$\begin{aligned} h(1) &= 1, \quad h(-1) = -1, \\ h'(1) &= 0, \quad h'(-1) = 0. \end{aligned} \tag{4.40}$$

This indicates that h is continuous in the range {-1,1} and its derivatives equal zero at the end points. This polynomial function in equation 4.39 is iterated three times,

$$g_3(\mu_{ij}) = h\{h[h(\mu_{ij})]\} \,. \tag{4.41}$$

The cutoff function $s(\mu_{ij})$ is then obtained from

$$s(\mu_{ij}) = \tfrac{1}{2}[1 - g_3(\mu_{ij})]. \tag{4.42}$$

It was found that the cutoff function is too soft and is not properly extinguished near nuclei with only one or two iterations of $h(\mu_{ij})$. In other words, $s(\mu_{ij})$ is too unlike the step function and the transition from one to zero is too slow. Employing more than three iterations of $h(\mu_{ij})$, the cutoff function will become too much like a step function. This leads to numerical instabilities. With the new cutoff functions, "fuzzy" Voronoi polyhedra are defined as before with equation 4.38.

The nuclear weight function is defined by

$$w_k(\boldsymbol{r}_g) = \boldsymbol{P}_k(\boldsymbol{r}_g) \Big/ \sum_{m}^{atoms} \boldsymbol{P}_m(\boldsymbol{r}_g). \tag{4.43}$$

Note that the denominator includes all atoms in the system (i.e., $m = k$ as well). This weighting scheme formally scales cubically since the number of points increases linearly and the number of atom pairs to be treated at each point increases quadratically with system size. Recently, Stratmann, Scuseria, and Frisch [17] proposed a new weighting scheme capable of generating results as accurate as those with Becke's approach. However, it can be evaluated significantly faster. In their new weighting scheme, the cutoff function is defined as

$$s(\mu_{ij}) = \tfrac{1}{2}\left[1 - g(\mu_{ij})\right], \tag{4.44}$$

where

$$g(\mu_{ij};a) = \begin{cases} -1, & \mu_{ij} \le -a, \\ z(\mu_{ij},a), & -a \le \mu_{ij} \le a, \\ +1, & \mu_{ij} \ge a. \end{cases} \tag{4.45}$$

In the above equation, $z(\mu_{ij},a)$ is given by

$$z(\mu_{ij},a)=\tfrac{1}{16}\left[35(\mu_{ij}/a)-35(\mu_{ij}/a)^3+21(\mu_{ij}/a)^5-5(\mu_{ij}/a)^7\right]. \qquad (4.46)$$

The new z function also satisfies the requirements

$$\begin{aligned} z(\mu_{ij}=a)&=1, \quad z(\mu_{ij}=-a)=-1. \\ z'(\mu_{ij}=a)&=0, \quad z'(\mu_{ij}=-a)=0. \end{aligned} \qquad (4.47)$$

meaning that the new z function is continuous. In addition, the second and third derivatives also vanish at $\pm a$. This ensures a numerical more stable algorithm. It was found that a value of a = 0.64 leads to an optimal result.

The advantage of using the cell function of equations 4.44 to 4.46 is that the normalized weight associated with a grid point can be more easily and efficiently determined. The normalized weight on a grid point automatically has a value of 1 if the following condition holds

$$r_{ig} < \tfrac{1}{2}(1-a)R_{in} \Rightarrow \begin{array}{c} F_i(\boldsymbol{r}_g)=1, \\ \text{and } w_i(\boldsymbol{r}_g)=1. \end{array} \qquad (4.48)$$

In the above, r_{ig} is the distance from the grid point g to its parent atom i, R_{in} is the distance to the nearest atomic neighbor of i, and the constant a is defined as above. To get a clearer picture of equation 4.48, imagine that, for an atom i, a sphere of radius $0.18R_{in}$ can be drawn. Each $s(\mu_{ij})$ within the sphere has a value of 1 for any atom j. For all r_{ig} that satisfies equation 4.48, it turns out that $\mu_{ij} \le -a$ for all j atoms. Therefore, the g function (equation 4.45) equals –1. Substituting the value of g into equation 4.44, a value of one is obtained for the cell function. This means that for a grid point r_{ig} within the sphere, its unnormalized weight function, $P_i(r_{ig})$ as defined in equation 4.38, consists of a product of terms all having a value of 1. In other words, the unnormalized weight for

grid point r_{ig} associated with atom i equals 1. In addition, the symmetry properties of the cell function dictate that if

$$s(\mu_{ij}) = 1 \;\Rightarrow\; s(\mu_{ji}) = 0. \tag{4.49}$$

Equation 4.49 indicates that inside the sphere, the cell function has a value of 1. Outside the sphere, it has a value of zero. Consequently, all terms in the denominator of equation 4.43 are zero except for $P_i(r_{ig})$. The normalized weight $w_i(r_{ig})$ associated with atom i equals 1. The process of predetermining whether a grid point's weight has a value of one is trivial (i.e., if condition 4.48 is met, $P_i(r_g)$ and $w_i(r_g)$ immediately set to one). In comparison, determining the weight of a grid point using equation 4.44 requires lengthy calculations. Therefore, by prescreening those grid points with weights equal to 1, substantial computational time can be saved.

It is important to note that, in Becke's weighting scheme, many cell functions that are within the neighborhood of $\mu_{ij} \geq -1$ have values near one, but not quite one. In the vicinity of $\mu_{ij} \leq 1$, many cell functions have values near zero, but not quite zero (see Figure 4.1). This means that these cell functions always require explicit evaluation. To avoid the explicit calculations of those cell functions that have a value either very close to one or very close to zero, Stratmann *et al* devised a cutoff scheme [17]. For a cell function that falls within $\mu_{ik} < -a$ for any atom k, its value is set equal to one. A cell function that has a value of exactly one can be removed from the product defining the unnormalized weight. For those cell functions where $\mu_{ik} > a$, their values are set equal to zero. If a grid point has a strictly zero value cell function, its weight evaluation can be avoided. Though screening of near zero weight is possible in both weighting schemes,

Becke's scheme has a much smaller near zero value space. Hence, the weighting scheme of Stratmann *et al* is significantly faster than Becke's scheme. The latest version of DeFT employs this weighting scheme of Stratmann *et al.*

Note that the new weighting scheme by itself does not scale linearly with system size. For an atom n, the number of grid points that require explicit evaluation increases with system size. Each of these grid points includes all atoms in its evaluation. Therefore, the computational cost in the weight evaluation increases quadratically with system size.

4.5 Linear Scaling Strategy

In general, linear scaling can only be achieved if the computational effort involved at each grid point is independent of overall system size. In the case of weight evaluation [17], linear scaling is achieved when each grid point weight evaluation only includes a reduced set of atoms. This is possible since each atom's basis functions have a finite extent. One can imagine that each basis function $\chi_\alpha(\boldsymbol{r})$ centered at $\boldsymbol{R}_\alpha$ is included in a sphere. For any point $\boldsymbol{r}$ outside this sphere, its contribution is less than a predetermined threshold value, ε

$$\left|\chi_\alpha(\boldsymbol{r})\right| \le \varepsilon \ . \tag{4.50}$$

Therefore, a basis function is deemed significant to a grid point $\boldsymbol{r}_g$ if

$$\left|\boldsymbol{r}_g - \boldsymbol{R}_\alpha\right| \le \lambda_\alpha \tag{4.51}$$

where λ_α is the radius of the sphere. A list of significant basis functions, S_g, satisfying equation 4.51 can be created for a given grid point. There is a different list of significant basis functions for each grid point. The number of basis functions in each list is size independent.

If all basis functions centered on an atom make a negligible contribution to a given grid point, that atom can be safely excluded from the calculation of that grid point's weight without sacrificing too much in accuracy. In practice, an atom is considered non-significant to a given grid point if its most diffuse function is not included in S_g. In DeFT, a threshold of 10^{-10} atomic units is used to determine the radius of the sphere. For each atom, the radius of the most diffuse function λ_α is saved in a list. At each grid point, an atom is included in the weight evaluation when condition (4.51) holds. Consequently, the number of atoms involved in the weight evaluation is constant at each grid point. Hence the computational effort at a grid point is independent of overall system size, and hence, the overall computational effort increases linearly with system size.

4.6 Evaluating the Density and the XC Terms on the Grid

Once the molecular grid is generated, the total density must be synthesized over the entire set of grid points. Either one of two expressions,

$$\rho(\boldsymbol{R}_I) = \sum_{\mu}^{K} \sum_{\nu}^{K} \boldsymbol{P}_{\mu\nu} \chi_\mu(\boldsymbol{R}_I) \chi_\nu(\boldsymbol{R}_I) \tag{4.52}$$

or

$$\rho(\boldsymbol{R}_I) = \sum_{i}^{occupied} |\psi_i(\boldsymbol{R}_I)|^2 = \sum_{i}^{occupied} \left| \sum_{\mu}^{K} \boldsymbol{C}_{\mu i} \chi_\mu(\boldsymbol{R}_I) \right|^2 . \tag{4.53}$$

may be used. In the above, $\boldsymbol{P}_{\mu\nu}$ is a density matrix element obtained from the expansion coefficients of the occupied KS orbitals, $\psi_i(\boldsymbol{r})$,

$$\boldsymbol{P}_{\mu\nu} = 2 \sum_{i}^{occupied} \boldsymbol{C}_{\mu i} \boldsymbol{C}_{\nu i} \tag{4.54}$$

Using equation 4.53 to construct the electronic density is more efficient for small molecules since the first summation runs over only the occupied molecular orbitals. This approach was used in the earlier version of DeFT. KS orbitals are delocalized and will span the full extent of the molecule, regardless of its size. The computational requirement increases with both the number of grid points and the number of occupied KS orbitals. The computational effort will not grow with the number of basis functions, K, since only a finite number of basis functions make a significant contribution to any one grid point, regardless of system size. For large molecules, using equation 4.53 to evaluate the density will thus scale quadratically. Equation 4.52 better exploits the fact that orbital basis functions make significant contributions only to grid points around the vicinity of the atom on which they are centered. This is done in large part with the aid of equation 4.51. Each grid point includes a fixed number of significant basis functions. Increasing system size does not change this. It only increases the number of grid points. Therefore, using equation 4.52 to evaluate density scales linearly with system size. The latest version of DeFT uses equation 4.52 to evaluate the density on the grid. As it is implemented, the actual construction of the density matrix via equation 4.54 does formally exhibit cubic scaling. However, the actual time involved in this step is trivial

compared to others in the numerical quadrature for the systems being considered in the present study.

Grid points are generated and stored as a series of concentric spheres, growing out from the first atom, followed by the second, the third, and so on. These grid points are then handled as batches for the XC quadrature. It has been found that when working on small molecules, a large batch of grid points, as large as memory allows, maximizes computational efficiency [22]. In treating large molecules, however, the large batch (macrobatch) approach has been shown to be inefficient. Instead, microbatches of all grid points (from every atom) that fit within a small cubic box is recommended [17]. If the extent of a basis function cannot reach an outer edge of the cube, it can immediately be concluded that this basis function will not contribute to any one of the grid points in that batch.

The microbatch strategy is yet to be implemented in our DeFT program. In the present implementation, however, this concept is exploited, as best possible, within the existing DeFT structure. The distance, $\boldsymbol{R}_{ij}$, between any pair of atoms, i and j, is calculated and stored in a matrix. When working on any one batch of grid points, the distance, $\boldsymbol{R}_{\max}$, between the last point of the batch and the atom about which the batch is centered is determined. DeFT's storage of grid points assures us that the last point in any batch is as far away from the atom as any other point in the batch (points are stored in an outwardly fashion). A basis function centered on atom j can possibly contribute to any one of the grid points in a batch centered on atom i only if $| \boldsymbol{R}_{ij} - \boldsymbol{R}_{\max} |$ is less than that basis function's extent. Otherwise, this basis function can be ignored for the current batch of points. By construction, half of the radial points are within one half of the

covalent radius of the atom. Therefore, most of the batches have their grid points confined to a very small region of space. This method is thus only slightly less efficient than the optimal microbatch approach when dealing with the outer most spheres of radial grid points. To implement the microbatch approach within DeFT's current structure, we would have to be able to store each grid point's coordinates and weights in memory. This is not possible for large system. Work is in progress to overhaul DeFT's structure to permit microbatching.

In a DAC XC fitting procedure, the outermost loop runs over all the subsystem calculations. The first subsystem calculation involves determining the extent of each contraction in the orbital basis and each fitting function in the XC auxiliary basis. A threshold value of 10^{-16} atomic units is used to determine the extent of a basis function for equation 4.52, and a value of 10^{-20} atomic units is used to determined the extent of the squared spherical atomic density, $\left|\rho^{atomic}(\boldsymbol{r})\right|^2$.

Since the density (or the spin densities, in an open-shell calculation) must be synthesized over the entire grid, this is done in the first subsystem calculation. Here, we choose to discuss the implementation within the LSDA. Therefore, the derivatives of $\rho(\boldsymbol{r})$ are not considered, though the concept can easily be extended to any gradient-corrected functional. Once the density has been synthesized, the XC terms, $v_{xc}(\boldsymbol{r})$ and $\varepsilon_{xc}(\boldsymbol{r})$, can then be evaluated at each point. Each XC term is divided by the denominator of equation 4.19 and saved on disk (or in memory, if running on a massively parallel computer with sufficient memory available). Note that the orbital basis functions, the density, and the XC terms need only be evaluated once for the first subsystem calculation. In the subsequent subsystem calculations, the XC terms are simply read back

from disk. Within each subsystem calculation, XC terms are weighted by the numerator of equation 4.19 to obtain that subsystem's contribution to the XC terms, $v_{xc}^{\sigma^\alpha}(\boldsymbol{r})$ and $\varepsilon_{xc}^{\alpha}(\boldsymbol{r})$. The values of the M^α XC fitting functions involved in the subsystem's fitting procedure are evaluated within each subsystem loop. There is a slight duplication of effort involved in this step, since a single XC fitting function may be included in different subsystem fits. Each subsystem's work on the grid for the current SCF cycle is completed once t^α is constructed for the XC potential and the XC energy density.

4.7 Analytic Evaluation of the XC Integrals

With the DAC fitted XC potential and the DAC fitted XC energy density, the XC contributions to the KS matrix and $\boldsymbol{E}_{xc}[\rho(\boldsymbol{r})]$ may be calculated via equations 4.14 and 4.15, respectively. The analytic computation of the XC terms involves the evaluation of simple three-center overlap integrals. This step formally scales as K^3 as the number of integrals rises cubically. This step, however, is significantly less demanding than the evaluation of the three-center two-electron integrals of the Coulomb components. Again, significant CPU times can be saved by exploiting the fact that orbital basis functions and XC fitting functions have finite extents.

Each function can only overlap with functions that fall within its extent. When the molecule gets sufficiently large, a function has an appreciable overlap with a fixed number of other functions. Making the molecule bigger and adding new functions to faraway regions will not increase the number of appreciable overlaps for this function. Therefore, as the molecular system gets sufficiently large, the number of non-vanishing

primitive shell pairs rises linearly only with the number of fitting functions, and therefore linearly overall with system size. As it is implemented, the time spent generating this list of appreciable overlaps scales quadratically. But the actual CPU time spent on the list is negligible compared to the time spent evaluating the final integrals. If the scaling of this step ever becomes a problem, a more efficient approach can be adopted.

In DeFT's evaluation of these overlap integrals, a list of non-vanishing primitive shell pairs for each unique combination of orbital angular momenta: *ss*, *ps*, *pp*, *ds*, *dp*, and *dd*, is generated. The three-center overlap base integrals [23] (S_a and S_b are associated with orbital basis functions and S_c is associated with a fitting function),

$$\langle S_a \mid S_b \mid S_c \rangle = \frac{\zeta^{3/2}}{(\zeta+\zeta_c)^{3/2}} \langle S_a \mid S_b \rangle \exp\left\{ \frac{\zeta\zeta_c}{\zeta+\zeta_c} (\boldsymbol{P}-\boldsymbol{C})^2 \right\}, \tag{4.56}$$

where

$$\zeta = \zeta_a + \zeta_b, \tag{4.57}$$

and

$$\boldsymbol{P} = \frac{\zeta_a \boldsymbol{A} + \zeta_b \boldsymbol{B}}{\zeta_a + \zeta_b}, \tag{4.58}$$

are evaluated and a list of non-vanishing base integrals is generated. In theory, this step also has a quadratic scaling. The actual time spent in this step is still small, but no longer totally negligible. Once the list is generated, the full set of three-centered overlap can then be evaluated. The size of the list scales linearly with system size. An explicitly coded subroutine is dedicated to each of the six unique combinations of angular momenta within the orbital bases. The integrals in these subroutines are evaluated using the recurrence relations of Obara and Saika [23] to build up angular momentum, and the

horizontal recurrence relation of Head-Gordon and Pople [24] is used to transfer angular momentum between centers.

4.8 Assessment of Grid Quality

In any grid-based calculation, the precision of the result depends on the quality of the grid employed. If the grid is sufficiently dense, the computed value will converge to the true value of the integrand. For a finite grid clearly, establishing the number of grid points required to achieve an acceptable level of precision is essential.

Test calculations on methanol are performed to assess the quality of various default grids within DeFT. A total of 26 different sets of nuclear coordinates are created by rotating the initial methanol geometry about the x, y, and z axes by angles produced from a random number generator. If the grids were perfect, all 26 methanol molecules would have the same total energy, since the internal coordinates are not altered. But atom-centered grids do not rotate along with the molecule. Rotational invariance does not exist. A way to create a rotational invariant grid has been proposed [25, 26]. It is achieved by establishing a standard orientation for each molecule in 3D space. A rotationally invariant grid has the advantage of always providing the same answer, irrespective of a molecule's orientation. It also seems to remove many of the problems encountered while evaluating the harmonic frequencies of low-lying normal modes [26]. However, one must realize that the rotationally invariant grid relies on the arbitrary definition of a standard orientation. Hence the total energy obtained is just one of the

many possible values that would otherwise have been obtained with a grid that was not artificially constrained to be rotationally invariant.

The root mean square (RMS) deviations in the total energy can be used to assess the quality of a grid when calculations are performed without invoking a standard orientation. With benchmark calculations, we can establish how many radial and angular points are required to lower the RMS deviations in the total energy under a desired value. Once this is done, a standard orientation can then be adopted, so that in future studies, the total energy is rotationally invariant. However, if this is done, when studying the energetics of a chemical process, the established RMS error should always be taken into consideration when deciding if an energy difference is the result of a true effect or simply the result of "grid noise".

Test calculations are performed within the LSDA with triple-ζ + polarization basis sets on all heavy atoms [27] and double-ζ + polarization basis sets on all hydrogen atoms [27]. For heavy atoms, each auxiliary basis set ($\rho(\boldsymbol{r})$ fitting and XC fitting) contains four *s* functions and another four sets of *s*, *p*, and *d* functions constrained to have the same exponents. The notation for such an auxiliary basis set is (4,4;4,4) [27]. Hydrogen atoms' auxiliary basis sets are composed of three *s* functions and a single set of *s*, *p*, and *d* functions. These auxiliary bases are labeled as (3,1;3,1) [27]. The results of test calculations are listed in Table 4.1. Two sets of results are reported for each grid. The first value represents the total energy obtained when the fitted XC energy density is used to calculate $\boldsymbol{E}_{xc}[\rho(\boldsymbol{r})]$ via equation 4.15. The second value is obtained when equation 4.4 is used to integrate $\boldsymbol{E}_{xc}[\rho(\boldsymbol{r})]$ numerically. From the data shown, several conclusions can be drawn.

First, the total energy is far less prone to grid noise if $E_{xc}[\rho(r)]$ is calculated via equation 4.15. It is a full order of magnitude more stable than the $E_{xc}[\rho(r)]$ values evaluated by numerical integration. This is consistent with the general observation that, when the grid is used solely for fitting purposes, fewer points are needed to achieve a given level of numerical precision. However, the numerically integrated final total energy is still required to construct potential energy surfaces. The energy gradient expression derived for the total energy when $E_{xc}[\rho(r)]$ is evaluated via equation 4.15 is not valid unless the XC fitting basis is very large. In fact, test have shown that the XC fitting functions need to be so large that they make their use unfeasible, in practice, for an accurate energy gradient [28]. The strategy adopted within DeFT is therefore as follows. A modest auxiliary fitting basis is used strictly for fitting XC terms only. Once an SCF is achieved, equation 4.4 is employed for the final total energy evaluation. Note that the numerical integration discussed thus far occurs only after the SCF has converged. During the SCF, XC contributions to the KS matrix elements are obtained from analytic integration with the fitted XC potentials (equations 4.14 and 4.15). It is highly probable that if numerical integration was used for the XC terms throughout the SCF, the level of grid noise would be even higher than those already listed in Table 4.1.

From Table 4.1, we see that for a given number of angular points, very little is gained in terms of accuracy when more than 40 or 48 radial points are used to perform the fits. The same can be said when more than 64 radial grid points are used to perform the numerical integration. For a grid that contains 96 radial points and either 50, 110, or 194 angular points, the average total energy calculated with the fitted XC energy density are all within 1 μhartree of each other. For numerical integration on the 96 radial point

grid with 110 and 194 angular points, the average total energies differ by 35 μhartrees. Therefore, for an accuracy of better than 35 μhartrees, a total of 60 fewer angular points are required when the total energy is calculated with the fitted XC energy density (obtained on the 96 radial point grid) than when the total energy is calculated by direct numerical integration.

The final point to be made from Table 4.1 is that, using either an analytic integration with a fitted quantity or a direct numerical integration on any grid, the average total energies between the two differ by about 1 *m*hartree. This indicates that XC fitting bases employed in the calculations are not complete. However, this is not a serious problem, as it is the relative energies that are always of interest. With an incomplete orbital basis, we can always rely on a systematic cancellation of errors. The same can be expected with an incomplete fitting basis. A BSSE-like error may arise within the XC fitting basis. However, it is almost certainly not as important as that within the orbital basis. This is because only a fraction of the total energy arises from $E_{xc}[\rho(r)]$. Also, it is significantly easier for the XC auxiliary basis to fit the nodeless XC terms than it is for the orbital basis to properly describe a full set of occupied molecular orbitals which may each possess a far more complex nodal structure.

In these test calculations, we also tried randomly rotating the angular grid points of each radial shell. This is because angular points are generally aligned along common rays. Therefore, to get a better sampling of the space and hopefully reduce grid noise, we randomly rotate each radial shell. In general, for a small number of angular grid points, a randomly rotated grid did reduce the grid noise when a fitted quantity is used in the total energy evaluation. This is precisely the situation for which the randomly rotate grids

were originally proposed and for which they have enjoyed much success [29]. But randomly rotating a grid of 194 angular points always tends to increase the grid noise slightly. For the numerically integrated energies, random rotations generally lead to a bigger RMS deviation in the total energy. Since random rotations eliminate the possibility of establishing a standard orientation for a rotationally invariant total energy and they are only useful for small angular grids with questionable value, it has been decided that randomly oriented grids will not be a supported option within the future versions of DeFT.

For the test calculations on methanol, the same grid was used for both the XC fits and the final numerical integration of $E_{xc}[\rho(r)]$. This, however, is not absolutely necessary. Therefore, the possibility of using a much smaller grid for the fits of the XC terms, and a larger grid for the final numerical integration, is explored. We realize that the absolute value of the fluctuations in the total energy will most likely rise with system size. Therefore, considering the very small size of methanol, we set to achieve an accuracy within 0.01 kcal mol^{-1} for its total energy. With this target precision, a grid containing 40 radial and 110 angular points is used for the fits of XC terms, and the one-time numerical integration of $E_{xc}[\rho(r)]$ at the end of the SCF is carried out on a grid of 64 radial and 194 angular points. Note that each of these surpass 0.01 kcal mol^{-1} accuracy in Table 4.1. Test calculations are again performed on the same ensemble of 26 geometries. The RMS deviation in the numerically integrated energies is still only 0.0072 kcal mol^{-1}, the same value obtained when 64 radial points and 194 angular points were used for the fitting procedures. This indicates that if numerical integration of $E_{xc}[\rho(r)]$ is used to construct the potential energy surfaces, fitting of the XC terms can

be done on a smaller grid with negligible loss in accuracy, thus affording considerable CPU savings.

With a default grid of 40 radial points and 110 angular for fitting along with 96 radial points and 194 angular points for numerical integration, the possibility of grid pruning [25, 30](i.e., using less angular points near the nuclei where $\rho(\boldsymbol{r})$ is more spherically symmetric) has also been investigated. We have chosen to be very cautious and therefore sacrifice very little in terms of precision when establishing pruned grids. In the fitting procedures, the final pruned grid uses 12 angular points when $r_i \leq 0.20\,\boldsymbol{R}^*$, 26 when $r_i \leq 0.60\,\boldsymbol{R}^*$, 50 when $r_i \leq 1.00\,\boldsymbol{R}^*$, and 110 when $r_i \geq 1.00\,\boldsymbol{R}^*$. $\boldsymbol{R}^*$ is the covalent radius of an atom with respect to grid point r_i. The same cutoffs are used in the numerical integration, but the angular grid points are increased from 12, 26, 50, and 110, to 26, 50, 110, and 194, respectively. In general, pruning reduces the total number of grid points by just over 50%. However, it sacrifices very little in terms of numerical precision. Using a pruned grid for the fits, the RMS deviation in the energy given by equation 4.15 increases merely from 0.0039 to 0.0041 kcal mol^{-1}. However, the RMS deviation remains unchanged at 0.0072 kcal mol^{-1} for the numerically integrated energies (equation 4.4). Any attempts to extend the cutoff values tend to raise the RMS deviations sharply. With the pruned grids, and after discarding points with negligible weights, there are roughly 1800 points per atom in the fitting procedure and about 5300 points per atom in the final numerical integration. Note that roughly 12416 (64 × 194) points would have had to be used, for both the fits and the numerical integration, had we insisted on using the same quality grid for both procedures and had not made use of pruned grids. And this is done with very little loss of accuracy.

To establish the errors associated with larger systems, pruned grids are tested in a set of 26 calculations on both the C_5 (Figure 4.4) and C_7 (Figure 4.5) glycine dipeptide conformers. The RMS deviations for the C_5 conformer are 0.021 kcal mol^{-1} for the total energy calculated with the fitted energy density and 0.017 kcal mol^{-1} for the total energy calculated by numerical integration of $E_{xc}[\rho(r)]$. For the more compact C_7 conformer, the RMS errors fall slightly to 0.018 and 0.015 kcal mol^{-1}, respectively. The fluctuations in the total energy do indeed increase with system size. This point should always be taken into consideration when deciding whether an energy difference is indeed significant or just an artifact of grid noise. In the test calculations on the dipeptide, energies calculated with the fitted XC energy density have a slightly larger variation than the numerically integrated values. However, recall that significantly fewer points, roughly one third as many points as in the final numerical integration, are being used in the fitting procedure. If we equate our RMS deviations to errors relative to calculation using a "perfect" grid [25], our deviations in the final numerical integrated results are somewhat smaller than the average error obtained with the SG-1 grid [25] for systems of similar size. Note that, a perfect grid as defined by Gill *et al.* [25], is a grid having 96 radial points, 32 theta points and 64 phi points, and the SG-1 grid is defined as containing 50 radial points and 194 angular points (note that grid pruning is also performed on this grid). In the study carried out by Gill *et al.*, numerical integration is used both during the SCF cycle and the final total energy evaluation, and pruned grids are used for both. The total energies presented in their report are relative to the 'perfect' grid. The grid used in our study for the numerical integration contains roughly twice as many points as the SG-1

grid. However, considerably fewer grid points are used throughout the course of an SCF cycle, where they are used solely for the fitting purposes.

4.9 Establishing the Optimal Buffer Space Cutoff for the DAC XC Fitting Procedures

The DAC approach divides a large molecule into many smaller subsystems. XC fitting is then carried out within each and every subsystem. It has been established that to obtain a reliable DAC result, each subsystem calculation must also include basis functions that are centered on atoms within its vicinity, i.e., on the so-called buffer atoms. For the DAC fit of $\rho(\boldsymbol{r})$, a 5.0 Å cutoff has been found to be sufficient to achieve an accuracy of 0.01 kcal mol^{-1} in the total energies [31] of molecules of several dozen atoms. It is important to have similar confidence in the results generated via a DAC XC fit. To ascertain the cutoff value required to attain a given level of precision in a DAC XC fit, a series of calculations with various cutoff values are performed on the extended conformer of the glycine heptapeptide. These calculations are performed within the LSDA using 6-31G** orbital basis sets, (4,3;4,3) auxiliary basis sets for heavy atoms, and (3,1;3,1) auxiliary basis sets for hydrogen atoms. A 5.0 Å cutoff is used for the DAC $\rho(\boldsymbol{r})$ fit in each calculation. The results are shown in Table 4.2. The last entry in the table is obtained using an infinite cutoff in the DAC XC fits. This ensures that the conventional (non-DAC XC fit) result is indeed exactly reproduced. The last column represents the error obtained in the total energy relative to the energy in the last entry, i.e., the error arising from the DAC $\rho(\boldsymbol{r})$ is not included in these values. Note that the

last entry is 0.006 kcal mol^{-1} lower than the energy obtained when $\rho(r)$ is fit by the conventional, non-DAC, scheme. The error associated with a DAC $\rho(r)$ fit tends to lower the total energy. An exact fit to $\rho(r)$ provides an upper bound to the total energy within a given orbital basis [32]. Unfortunately, no such variational principle exists for the XC fits.

An error of 9.6 kcal mol^{-1} is produced when no buffer space is assigned to the subsystem XC fits. By simply including only those atoms directly bound to the subsystem's atoms (this corresponds to a 2.0 Å cutoff in these tests), the error is reduced to 1.9 kcal mol^{-1}. This is still too high to be acceptable. The results only become somewhat acceptable when the buffer space is extended to 4.0 Å. With this cutoff, an error of 0.06 kcal mol^{-1} is obtained, which is of the same order as the grid noise. This may be acceptable in many instances. However, if we insist that the errors generated by the DAC XC fits be of the same order as those of the DAC fits of $\rho(r)$, a 7.0Å cutoff must be used. For this specific case, the error almost vanishes entirely with such a 7.0 Å cutoff. However, this is fortuitous. An 8.0 Å cutoff generates a 0.003 kcal mol^{-1} error. This increase and the subsequent change in the sign of the error as larger cutoffs are used clearly indicate that the XC fitting procedure is not variational. However, the general trend towards smaller absolute errors as the XC buffer space is extended is maintained. With this data and prior experience from preliminary work on the DAC XC fitting procedures, we have settled upon a 7.0 Å cutoff for the DAC XC fits. This is considerably larger than that required for the DAC $\rho(r)$ fit. This is most likely due to the more diffuse nature of the XC fitting functions (remember that its exponents are one third those in the $\rho(r)$ fitting basis, see Chapter 1). A larger cutoff value is needed

before the effect of dropping distant diffuse XC fitting functions is not reflected in the total energy.

Now that an adequate buffer space cutoff of 7.0 Å has been established for the DAC XC fit, it is important to study the effect of system size on the magnitude of the DAC errors. Together with the 5.0 Å buffer space cutoff for the DAC $\rho(\boldsymbol{r})$ fit, test calculations are performed on a series of extended glycine polypetides, ranging from the dipeptide to the nonapeptide. The results are presented in Table 4.3. It is clear that the error is slowly, but systematically, increasing with the length of the peptide chain. This contradicts the earlier finding [31] obtained with larger orbital and auxiliary bases which indicated no noticeable increase in errors with system size. However, those earlier studies used a far less reliable grid. Our new results are most likely more representative of what will be encountered in future DAC applications. The error becomes progressively more negative, i.e., the energy calculated with the DAC XC fits is too low. Remember that the DAC $\rho(\boldsymbol{r})$ fit is variational, with the perfect fit serving as the upper bound. This means that the errors introduced are always associated with a negative sign. However, no such variational exist for the DAC XC fits. The errors introduced by the imperfect DAC XC fits can either be positive or negative. Therefore, the systematic growth in the absolute value of the error, as shown in Table 4.3, is most likely attributable to the errors in the DAC $\rho(\boldsymbol{r})$ fit. Recall that a 7.0 Å buffer space cutoff for the DAC XC fits in our test calculations were decided upon based on test calculations on the glycine heptapeptide. And, at this particular buffer space cutoff, the errors introduced from the DAC XC fits are fortuitously low. It is possible that this effect is common to all the extended glycine polypeptide conformations. More tests will

unfortunately have to be performed before the suitability of a 7.0 Å DAC XC buffer space cutoff is firmly established.

Though the error in the DAC fits does increases systematically with system size, it is important to note that the error is only on the order of 0.0001 kcal mol^{-1} per atom in these test calculations. This is considerably smaller than the increase in the grid noise in the total energy, which roughly doubles from 0.007 kcal mol^{-1} for methanol to 0.015 kcal mol^{-1} for the dipeptide. This corresponds to an increase of roughly 0.001 kcal mol^{-1} per atom, a fully order of magnitude larger. Therefore, the DAC errors are not necessary our greatest concern if much larger systems than the nonapeptide are to be studied. Rather, it is grid noise.

4.10 Assessing the Performance of the XC Algorithm

Whenever any changes are made to an existing algorithm, it is essential to document the performance. Benchmark calculations are performed on a series of extended glycine polypeptides, ranging from the 13 atom dipeptide to the 83 atom dodecapeptide. The results are obtained at the LSDA level using a 6-31G** orbital basis together with (4,3;4,3) and (3,1;3,1) auxiliary fitting bases on heavy atoms and hydrogen atoms, respectively. Graphical representations of the timings associated with various steps in DeFT's XC modules are presented in Figure 4.2 and Figure 4.3. These graphs are used to determine the scaling behaviour of the various components of the XC modules. To obtain an effective scaling exponent, Ω, the last three points, corresponding

to the three largest systems (with 69,76, and 83 atoms), are fitted using an N^{Ω} functional form.

A) Grid Construction

As evidenced in Figure 4.2, the generation of the grid and the final quadrature weight evaluation (using Stratmann et al.'s weighting scheme that includes only the set of significant atoms for each grid point) does indeed exhibit near-linear scaling. The asymptotic exponent obtained from our benchmark calculations is Ω=1.27.

B) DAC XC Fitting Procedure

Within each SCF cycle, all three common steps in the XC fitting procedure (collectively, we call them an SCF XC fitting procedure) exhibit near-linear scaling behaviour. The most time-consuming step, the synthesis of $\rho(\boldsymbol{r})$ at each grid point (equation 4.17) achieves a scaling of $N^{1.22}$. The evaluation of the orbital basis functions and the XC fitting functions over the grid together scale as $N^{1.32}$. The least time-consuming step, the evaluation of the localized contributions to the XC terms (equations 4.21 and 4.22) and the evaluation of the t^{α} vectors (equations 4.27 and 4.28) has a combined asymptotic exponent of 1.32. The total CPU time of these three steps roughly reflects the total time required for the XC fitting procedures in an SCF cycle. Overall, the SCF XC fitting procedure has a combined scaling of $N^{1.26}$.

C) Construction and Inverting Subsystem XC Overlap Matrices

An $M^{\alpha} \times M^{\alpha}$ XC overlap matrix for each subsystem must be constructed and inverted within the first iteration of any SCF calculation using a DAC XC fitting procedure. The CPU time requirements for these steps are shown in Figure 4.3. Both the construction and inversion steps exhibit near-linear scaling behaviour. The size for each subsystem is independent of overall system. The CPU times required to construct and invert each subsystem is fixed. Increasing the system size only increases the number of subsystems (i.e., we will only have to construct and invert more subsystem matrices). Therefore, the total CPU time required for these two steps only increase with the number of subsystem, and thus increases linearly with system size. This is the essence of the DAC approach to achieving near-linear scaling. The conventional non-DAC fitting procedure would display a cubic scaling due to the inversion of the global XC overlap matrix. Inversion is an N^3 procedure. The asymptotic scaling exponents for the construction of the matrices and for the inversion of the matrices are 1.15 and 1.22, respectively. Both require similar amounts of CPU time.

On a single Cray T3E processor, constructing and subsequently inverting all the subsystem the XC overlap matrices for the largest peptide studied roughly requires a total CPU time of 3000 s. This is roughly equivalent to twelve times the total CPU time spent on the XC fitting procedure within one SCF cycle. Therefore, the overhead is equal to the fitting cost incurred over twelve iterations, roughly the number of iterations required to achieve an SCF solution. At first glance, this may seem excessive. However, these two time-consuming tasks make possible the use of a fitting procedure. This, in turns, allows the use of roughly three times fewer grid points. Taking this into consideration,

the overhead is reduced to only four SCF cycles, had fitting not been possible. If the fitting procedure is not used, the time required to numerically integrate the XC contributions to the $K(K+1)/2$ KS matrix elements also has to be taken into account (see equation 4.3 and 4.4). From the studies performed by Stratmann, Scuseria, and Frisch [17], the numerical integration of the XC contributions to the KS matrix requires roughly the same amount of CPU time as that taken to synthesize $\rho(\boldsymbol{r})$ on the grid. However, the analytic evaluation of the fitted XC terms, equations 4.14 and 4.15, are not nearly as costly. Finally, the times shown in Figure 4.3 will not be affected if these calculations were instead performed using gradient-corrected functionals. But, it will dramatically increase the times in Figure 4.2 associated with the evaluation of the orbital basis functions and the synthesis of $\rho(\boldsymbol{r})$ since the first and second derivatives of $\rho(\boldsymbol{r})$ have to be evaluated when gradient-corrected functionals are used. Their linear scaling behaviour would remain intact however. Therefore, the apparently large overhead associated with the construction and inversion of the subsystem XC overlap matrices should not dissuade one from using DAC XC fitting procedures.

D) Analytic Evaluation of the XC Integrals

The CPU times associated with the analytic integration of the approximate XC terms are shown in Figure 4.2. The times reported for this step include the actual time required to evaluate the final three-centered integrals, the time spent creating the list of non-vanishing overlap integrals, and the time spent summing up the integrals in the total energy expression and the KS matrix elements. The curve clearly shows that near-linear scaling is achieved. The origin of the spurious point corresponding to the 56 atom

octapeptide is not known. Nevertheless, this portion of the code scales as $N^{1.38}$ over the last three points. This is the worse scaling behaviour observed in any XC module within DeFT. The poor scaling could partly be attributed to the fact that an XC integral could only be ignored if there is no appreciable overlap between the XC fitting function and primitive pair of orbital basis functions at *any* point in space. In contrast, when working on the grid, only the current grid point of interest need be considered. Therefore, only when the system size gets larger than those studied here will the number of non-vanishing integrals truly rises linearly with system size. This worst scaling could also partly be attributed to the fact that the pre-screening of the three-centered base overlap integrals still scales as N^2.

Though this portion of the code has the worst scaling, it is important to note that the time spent evaluating these integrals is considerably less than that spent synthesizing $\rho(\boldsymbol{r})$ on the grid. If a gradient-corrected functional is employed, the discrepancy between the two would only become greater. The use of a gradient-corrected functional will greatly increase the time spent evaluating the orbital basis functions, the density, and their derivatives on the grid. However, the time spent evaluating the three-centered overlap integrals does not change since their nature is not affected by the nature of the XC functional. Consequently, synthesizing $\rho(\boldsymbol{r})$ on the grid will roughly require an order of magnitude more CPU time than the analytic evaluation of the integrals, as opposed to the factor of two currently experienced for these calculations within the LSDA.

4.11 Conclusion

From the results obtained in our test calculations on the extended glycine polypeptides, near-linear scaling has clearly been achieved for every XC step within DeFT. No single module has a scaling worse than $N^{1.4}$ once the systems contain at least 83 atoms. When treating more globular systems, however, this favorable scaling will only be achieved for systems containing considerably more atoms than 83. Test on larger systems will have to wait until memory considerations in DeFT (unrelated to the XC procedures discussed here) are addressed. The DAC philosophy adopted in the XC fitting procedure can also be exploited to run an LCGTO-DF application on massively parallel computer architectures. Using coarse grain parallelization to perform subsystem calculations over groups of processors of various sizes, together with fine grain parallelization within each, leads to an efficient use of a large number of processors. This subject is more fully discussed in Chapter 3. Having achieved near-linear scaling in the XC fits, and considering the advantages of using XC fitting procedures, discussed earlier, the use of fitted XC terms is a viable option for calculations on larger systems.

Table 4.1

RMS errors in the total energy of CH_3OH employing various grids.

grid points		total energy (fitted)		total energy (integrated)	
radial	angular	RMS error (kcal mol^{-1})	average (hartrees)	RMS error (kcal mol^{-1})	average (hartrees)
24	26	0.1395	-114.829962	1.8127	-114.830471
32	26	0.1061	-114.830045	1.4863	-114.830988
40	26	0.1130	-114.830070	1.4707	-114.831039
48	26	0.1133	-114.830063	1.4641	-114.831047
64	26	0.1132	-114.830061	1.4695	-114.831077
80	26	0.1132	-114.830061	1.4692	-114.831068
96	26	0.1132	-114.830061	1.4694	-114.831068
24	50	0.0545	-114.830006	0.6200	-114.830950
32	50	0.0400	-114.830070	0.4558	-114.831512
40	50	0.0404	-114.830101	0.4414	-114.831567
48	50	0.0407	-114.830094	0.4480	-114.831571
64	50	0.0407	-114.830092	0.4464	-114.831604
80	50	0.0406	-114.830092	0.4452	-114.831596
96	50	0.0406	-114.830092	0.4455	-114.831595
24	110	0.0031	-114.830006	0.0887	-114.830899
32	110	0.0050	-114.830070	0.0567	-114.831434
40	110	0.0039	-114.830099	0.0499	-114.831469
48	110	0.0038	-114.830093	0.0511	-114.831492
64	110	0.0038	-114.830091	0.0499	-114.831527
80	110	0.0038	-114.830091	0.0499	-114.831518
96	110	0.0038	-114.830091	0.0504	-114.831518
24	194	0.0007	-114.830005	0.0236	-114.830849
32	194	0.0006	-114.830069	0.0175	-114.831404
40	194	0.0004	-114.830098	0.0115	-114.831439
48	194	0.0003	-114.830092	0.0054	-114.831457
64	194	0.0003	-114.830090	0.0072	-114.831491
80	194	0.0003	-114.830090	0.0053	-114.831483
96	194	0.0003	-114.830090	0.0058	-114.831483

Calculations are carried out within the LSDA using triple-ζ + polarization orbital basis sets and (4,4;4,4) auxiliary basis sets on heavy atoms, and double-ζ + polarization orbital basis sets and (3,1;3,1) auxiliary basis sets on hydrogen atoms.

Table 4.2

Errors in the total energy of the extended conformation of the glycine heptapeptide introduced by the DAC XC fitting procedures.

XC buffer space (Å)	total energy (hartrees)	error (kcal mol^{-1})
no buffer space	-1406.6256282	+9.5819
2.0	-1406.6378133	+1.9350
3.0	-1406.6407124	+0.1157
4.0	-1406.6408018	+0.0596
5.0	-1406.6408541	+0.0267
6.0	-1406.6408799	+0.0105
7.0	-1406.6408966	+0.0001
8.0	-1406.6408918	+0.0031
9.0	-1406.6408968	- 0.0001
10.0	-1406.6409005	- 0.0031
11.0	-1406.6408981	- 0.0009
12.0	-1406.6408979	- 0.0008
13.0	-1406.6408983	- 0.0010
14.0	-1406.6408977	- 0.0006
15.0	-1406.6408969	- 0.0001
∞	-1406.6408967	+0.0000

Calculations are carried out within the LSDA using 6-31G** orbital basis sets, (4,3;4,3) auxiliary basis sets on heavy atoms, and (3,1;3,1) auxiliary basis sets on hydrogen atoms. A 5.0 Å buffer space cutoff is used for the DAC $\rho(\boldsymbol{r})$ fitting procedure in each instance.

Table 4.3

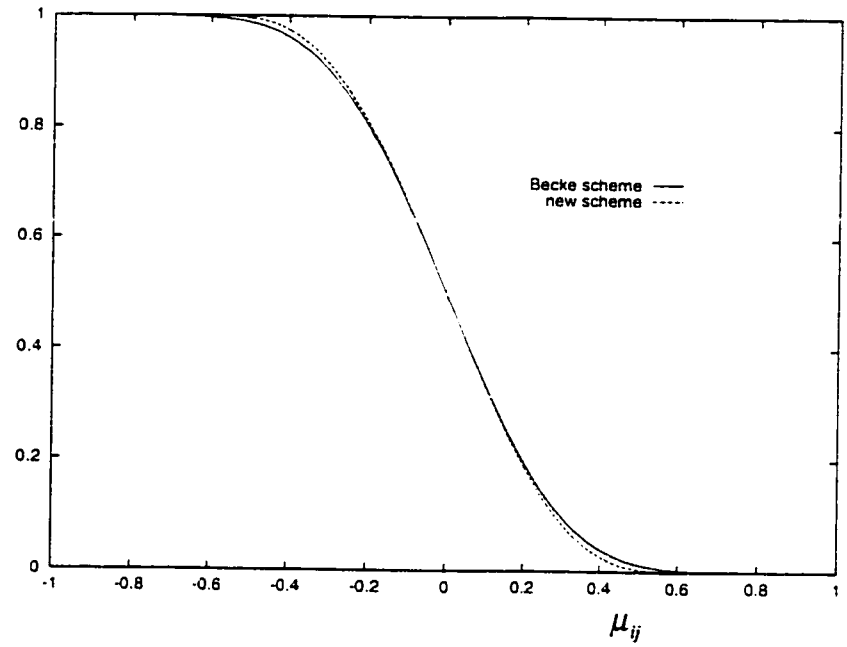
Errors in the total energy for a series of extended glycine polypeptides when using a 7.0 Å buffer cutoff for the DAC XC fits and a 5.0 Å buffer cutoff for the DAC $\rho(r)$ fit.

Molecule	Total energy		Error (kcal mol^{-1})
	Conventional fits (hartrees)	DAC fits (hartrees)	
Dipeptide	-374.8962674	-374.8962675	-0.0000
Tripeptide	-581.2448831	-581.2448861	-0.0019
Tetrapeptide	-787.5939193	-787.5939243	-0.0031
Pentapeptide	-993.9429009	-993.9429072	-0.0040
Hexapeptide	-1200.2918664	-1200.2918739	-0.0047
Heptapeptide	-1406.6408878	-1406.6408966	-0.0055
Octapeptide	-1612.9899057	-1612.9899155	-0.0062
Nonapeptide	-1819.3389079	-1819.3389188	-0.0068

Calculations are carried out within the LSDA using 6-31G** orbital basis sets, (4,3;4,3) auxiliary basis sets on heavy atoms, and (3,1;3,1) auxiliary basis sets on hydrogen atoms.

Figure 4.1

Comparison of cell functions obtained for two different weighting schemes: Becke and the new weighting scheme of Stratmann *et al.*



An expended version of the above graph

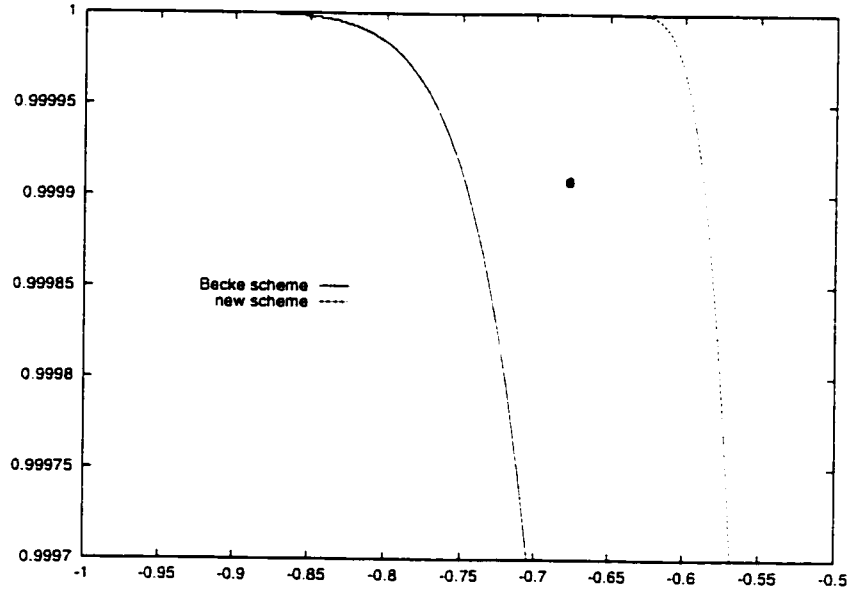
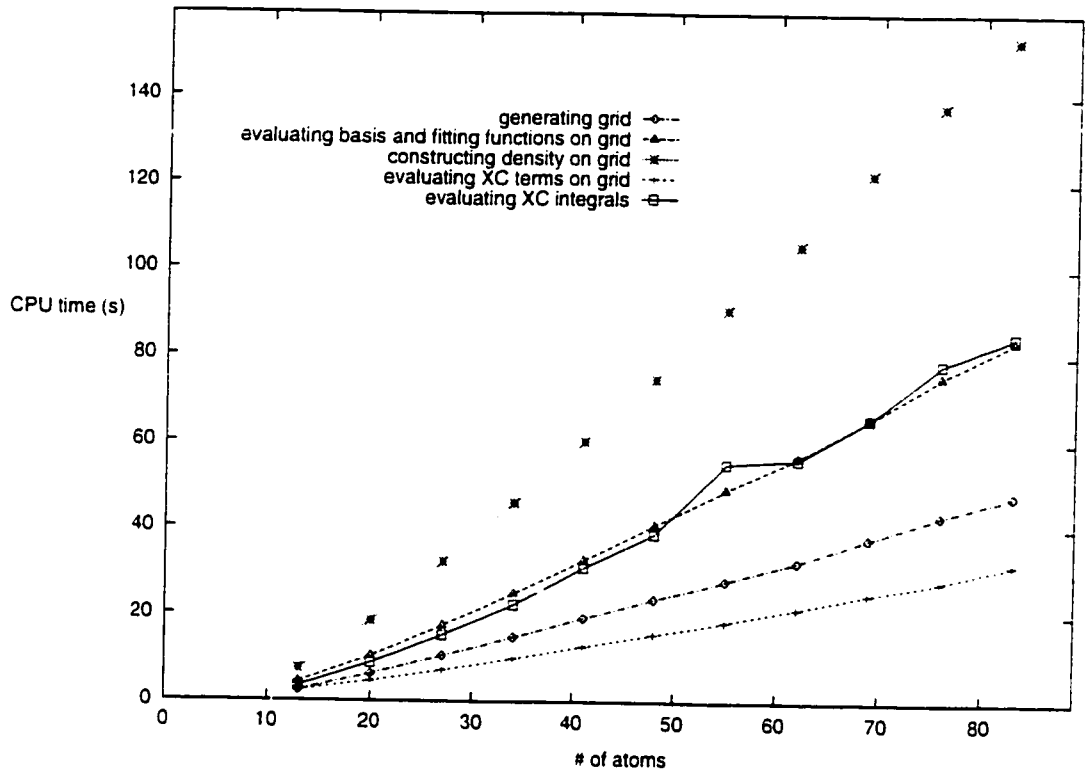


Figure 4.2

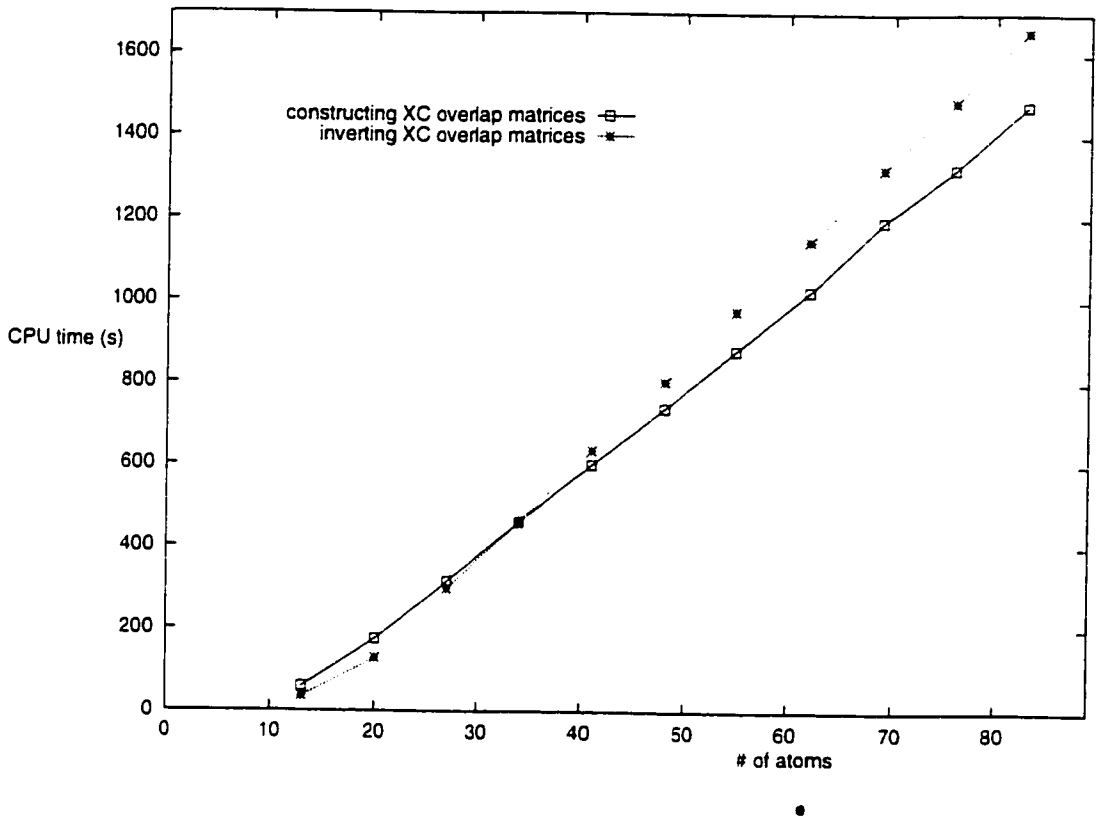
CPU times, on a single Cray T3E processor, associated with various portions of DeFT's XC routines.



These times are for 6-31G** LSDA calculations on extended glycine polypeptides using (4,3:4,3) and (3,1:3,1) auxiliary fitting bases on heavy atoms and hydrogen atoms, respectively.

Figure 4.3

CPU times, on a single Cray T3E processor, associated with the portions of DeFT's XC routines executed a single time at the outset of the SCF calculation.



These times are for 6-31G** LSDA calculations on extended glycine polypeptides using (4,3;4,3) and (3,1;3,1) auxiliary fitting bases on heavy atoms and hydrogen atoms, respectively.

Figure 4.4

The C_5 conformation of the glycine dipeptide.

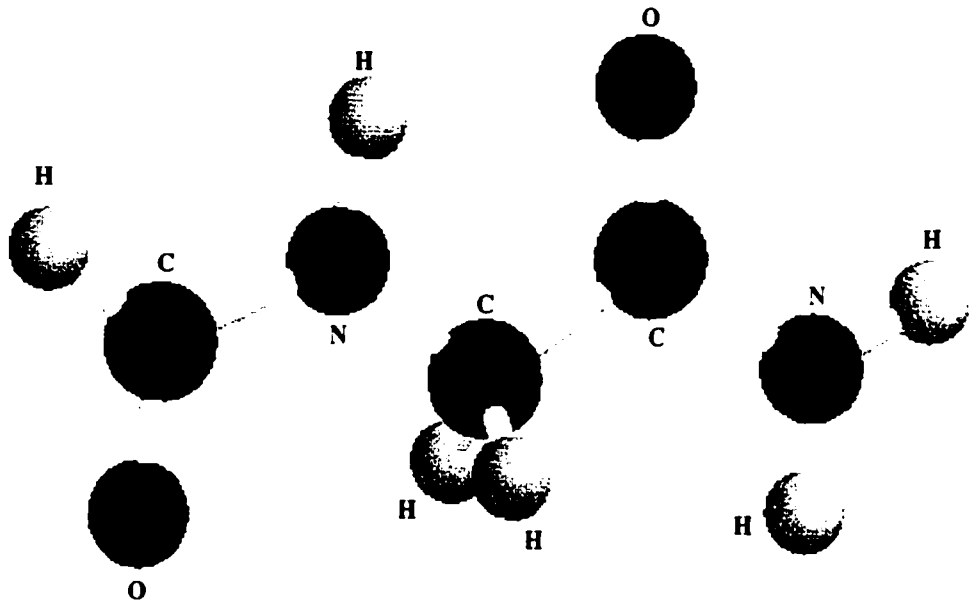
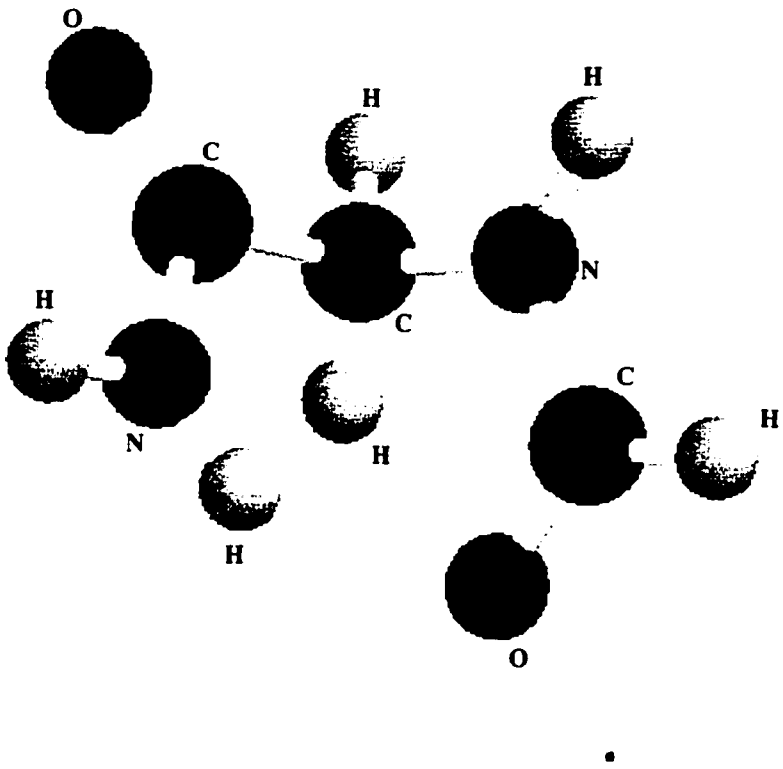


Figure 4.5

The C_7 conformation of the glycine dipeptide.



References

(1) J. K. Labanowski and J. W. Andzelm, *Density Functional Methods in Chemistry* (Springer-Verlag; New York; 1991).

(2) T. Ziegler, *Chem. Rev.* **91**, 651 (1991).

(3) A. St-Amant, in *Reviews in Computational Chemistry*, **Vol.** 7, K. B. Lipkowitz and D. B. Boyd, Eds.(VCH Publishers, New York; New York; 1995).

(4) J. M. Seminario and P. Politzer, *Modern Density Functional Theory: A Tool for Chemistry* (Elsevier Science B. V.; Amsterdam; 1995).

(5) J. Andzelm and E. Wimmer, *J. Chem. Phys.* **96**, 1280 (1992).

(6) A. St-Amant, PhD thesis, University of Montreal, 1992.

(7) A. St-Amant, the latest version of DeFT can be downloaded from (http://www.chem.uottawa.ca/DeFT.html).

(8) A. St-Amant, in *Quantum Mechanical Simulation Methods For Studying Biological Systems*, D. Bicout and M. Field, Eds.(Les Editions de Physique; France; 1996).

(9) B. I. Dunlap, *Int. J. Quantum Chem.* **58**, 123 (1996).

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

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

(12) M. Challacombe, E. Schwegler and J. Almlof, *J. Chem. Phys.* **104**, 4685 (1996).

(13) W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

(14) W. Yang, *J. Mol. Struct. (THEOCHEM)* **225**, 461 (1992).

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

(16) R. T. Gallant and A. St-Amant, *Chem. Phys. Lett.* **256**, 569 (1996).

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

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

(19) A. St-Amant, S. K. Goh and R. T. Gallant, in *Recent Developments and Applications of Modern Density Functional Theory*, J. Seminario, Ed.(Elsevier: Amsterdam; 1996).

(20) J. C. Slater, *Phys. Rev.* **81**, 385 (1951).

(21) M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions* (Dover: New York; 1965).

(22) B. G. Johnson, Ph. D, Carnegie Mellon University, 1993.

(23) S. Obara and A. Saika, *J. Chem. Phys.* **84**, 3963 (1986).

(24) M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* **89**, 5777 (1988).

(25) P. M. W. Gill, B. G. Johnson and J. A. Pople, *Chem. Phys. Lett.* **209**, 506 (1993).

(26) B. G. Johnson, P. M. W. Gill and J. A. Pople, *Chem. Phys. Lett.* **220**, 377 (1994).

(27) N. Godbout, D. R. Salahub and J. Andzelm, *Can. J. Chem.* **70**, 560 (1992).

(28) R. Fournier, J. Andzelm and D. R. Salahub, *J. Chem. Phys.* **90**, 6371 (1989).

(29) R. S. Jones, J. W. Mintmire and B. I. Dunlap, *Int. J. Quantum Chem.* **S22**, 77 (1988).

(30) C. W. Murray, N. C. Handy and G. J. Laming, *Mol. Phys.* **78**, 997 (1993).

(31) S. K. Goh and A. St-Amant, *Chem. Phys. Lett.* **264**, 9 (1997).

(32) B. I. Dunlap, J. W. D. Connolly and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

Chapter 5

Improving the Efficiency and Reliability of the Divide and Conquer Approach to Constructing the Electronic Density

5.1 Introduction

The design and development of linear scaling algorithms for quantum mechanical calculations [1-7] has been and continues to be an active area of research. These methods eliminate the unfavourable scaling of the conventional approaches and hence offer the possibility of studying much larger systems. The feasibility of accurate first-principles electronic structure calculations exhibiting linear scaling was first demonstrated by Yang in 1991 [1] with a divide and conquer (DAC) approach. The first divide and conquer scheme is Yang's electron density formulation. It was developed solely within the framework of density functional theory. Recently, a more generalized divide and conquer approach [3] has been developed by Yang. It is known as the density matrix formulation. This new approach is applicable to any quantum mechanical method that makes use of molecular orbitals (MO). This thus opens the possibility of applying the divide and conquer philosophy to traditional Hartree-Fock *ab initio* and semi-empirical methods.

The density matrix formulation of the divide and conquer scheme will be given in detail. However, the basic philosophy is as follows. In the divide and conquer scheme, a large system is divided into a collection of subsystems. Each subsystem density is determined by solving a set of local eigen-equations. Subsystem densities are then pieced together with the aid of a common Fermi energy that ensures proper normalization of the total electronic density. It has been found that if subsystem matrices span only the basis functions centered on atoms within the actual subsystem, the DAC approach is too severe an approximation. Therefore, the concept of buffer atoms [2, 3] is introduced. The buffer atoms of a subsystem α are atoms that are not part of subsystem α but are in the

vicinity of subsystem α. Their basis functions are, however, included in the local basis set of subsystem α. Buffer atoms are introduced to yield a better representation of the density contribution from subsystem α. The accuracy of the divide and conquer approach can be systematically improved by increasing the number of buffer atoms in a subsystem calculation.

In practice, a finite number of buffer atoms is used in any divide and conquer method. An atom belonging to a specific subsystem may find its basis functions being included in several nearby subsystem calculations as buffer atoms. Once an atom is beyond a certain cutoff distance from a subsystem, it is considered to be too far for it to have a significant impact on that subsystem calculation. Its basis functions are thus not included in such subsystem calculations. As system size gets larger, the CPU time associated with any subsystem calculation will eventually reach a maximum value and become independent of overall system size. At that point, the total CPU time will simply increase linearly with the number of subsystems, and thus linearly with the overall system size. In theory, linear scaling in any MO approach is thus achieved.

Such divide and conquer schemes have been implemented within DFT [2, 3] and semi-empirical methods [8, 9] with various degree of success. However, to achieve a given level of precision, there is no clear consensus regarding the nature of the basis functions required in the buffer spaces. Further, the origin of the errors involved with these DAC calculations have not yet been clearly established. In this chapter, a detailed investigation into the errors of a DAC calculation will be performed. Benchmark calculations are carried out on a series of glycine polypeptides using the linear combination of Gaussian-type orbitals (LCGTO) density functional program, DeFT [10].

5.2 An Approximate Divide and Conquer Density

In the LCGTO approach [11, 12], the electronic density can be defined as

$$\rho(\boldsymbol{r}) = \sum_{\mu}^{K}\sum_{\nu}^{K} \boldsymbol{P}_{\mu\nu}\chi_{\mu}(\boldsymbol{r})\chi_{\nu}(\boldsymbol{r}). \tag{5.1}$$

Here, $\chi_{\mu}(\boldsymbol{r})$ and $\chi_{\nu}(\boldsymbol{r})$ are atomic basis functions, and K is the number of basis functions. The density matrix, $\boldsymbol{P}$, for a closed-shell system is given by

$$\boldsymbol{P}_{\mu\nu} = 2\sum_{i}^{M} \boldsymbol{C}_{\mu i}\boldsymbol{C}_{\nu i} \tag{5.2}$$

where M is the number of doubly occupied MOs. $\boldsymbol{C}_{\mu i}$ and $\boldsymbol{C}_{\nu i}$ are the expansion coefficients for the i^{th} MO.

The set of MO expansion coefficients is determined by the algebraic eigenvalue equation [13]

$$\mathbf{HC} = \mathbf{SC}\varepsilon, \tag{5.3}$$

where **H** is the Kohn-Sham or Fock matrix, **S** is the overlap matrix, and ε is the vector containing the eigenvalues associated with the eigenvectors contained in **C**.

The density matrix formalism of the DAC approach [3] divides the density matrix into a sum of subsystem contributions

$$\boldsymbol{P}_{\mu\nu} = \sum_{\alpha}^{subsystems} \boldsymbol{P}_{\mu\nu}^{\alpha} = \sum_{\alpha}^{subsystems} \boldsymbol{p}_{\mu\nu}^{\alpha}\boldsymbol{P}_{\mu\nu}. \tag{5.4}$$

This is done with the aid of a symmetric partition matrix, $\boldsymbol{p}^{\alpha}$, which obeys the following constraint

$$\sum_{\alpha}^{subsystems} \boldsymbol{p}_{\mu\nu}^{\alpha} = 1. \tag{5.5}$$

Here, α is the subsystem index. Clearly, if the constraint of equation 5.5 is obeyed, the left and right hand sides of equation 5.4 are always equal, regardless of the precise nature of the partition matrix. The elements in the partition matrix can be defined as

$$p_{\mu\nu}^{\alpha} = \begin{cases} 1 & \text{if } \mu \in \alpha \text{ and } \nu \in \alpha \\ \frac{1}{2} & \text{if } \mu \in \alpha \text{ and } \nu \notin \alpha \\ 0 & \text{if } \mu \notin \alpha \text{ and } \nu \notin \alpha \end{cases} \quad . \tag{5.6}$$

Substituting equation 5.2 into equation 5.4, the density matrix expression becomes

$$P_{\mu\nu} = 2 \sum_{\alpha}^{subsystems} p_{\mu\nu}^{\alpha} \sum_{i}^{M} C_{\mu i} C_{\nu i} \quad . \tag{5.7}$$

The second summation in equation 5.7 still runs over the M doubly-occupied MO's. It can be replaced by a sum over the K occupied and virtual MO's,

$$P_{\mu\nu} = 2 \sum_{\alpha}^{subsystems} p_{\mu\nu}^{\alpha} \sum_{i}^{K} \eta(\varepsilon_F - \varepsilon_i) C_{\mu i} C_{\nu i} \, . \tag{5.8}$$

The Fermi energy, ε_F, is the energy of the highest occupied molecular orbital. In the above, $\eta(\varepsilon_F - \varepsilon_i)$ is a Heavyside function. It returns a value of 1 if the i^{th} MO's eigenvalue, ε_i, is less than or equal to ε_F. It has a value 0 if $\varepsilon_i > \varepsilon_F$. At this stage, no approximations have been made. The density matrix elements given by equations 5.2 and 5.8 are identical. The expression for the total density becomes

$$\rho(\boldsymbol{r}) = 2 \sum_{\alpha}^{subsystems} \sum_{\mu}^{K} \sum_{\nu}^{K} p_{\mu\nu}^{\alpha} \sum_{i}^{K} \eta(\varepsilon_F - \varepsilon_i^{\alpha}) C_{\mu i} C_{\nu i} \chi_{\mu}(\boldsymbol{r}) \chi_{\nu}(\boldsymbol{r}) \quad . \tag{5.9}$$

To improve computational efficiency, approximations must now be made. The partition matrix for subsystem α, as defined in equation 5.6 helps localize subsystem α's contribution to $\rho(\boldsymbol{r})$ to the vicinity of those atoms of subsystem α. Therefore, an approximate DAC $\rho(\boldsymbol{r})$ can be constructed by replacing the true set of MO coefficients,

$C_{\mu i}$ in equation 5.9, with localized subsystem MO sets of coefficients, $C^{\alpha}_{\mu i}$. Subsystem sets of MO coefficients are obtained by solving

$$\mathbf{H}^{\alpha}\mathbf{C}^{\alpha} = \mathbf{S}^{\alpha}\mathbf{C}^{\alpha}\varepsilon^{\alpha}. \tag{5.10}$$

The matrix elements of the H^{α} and S^{α} matrices appearing in equation 5.10 are identical to their global counterparts in equation 5.3, i.e., their numerical values are equivalent. However, the subsystem matrices and vectors span only those elements associated with the K^{α} basis functions centered on actual subsystem atoms or nearby buffer atoms. The elements in $\mathbf{C}^{\alpha}$ and ε^{α} are affected by the basis set truncation within a subsystem calculation. They are no longer equivalent to their global counterparts. Substituting the subsystem MO coefficients and eigenvalues into equation 5.8, the approximate DAC $\rho(r)$ becomes

$$\rho(r) \approx 2 \sum_{\alpha}^{subsystems} \sum_{\mu}^{K^{\alpha}} \sum_{\nu}^{K^{\alpha}} p^{\alpha}_{\mu\nu} \sum_{i}^{K^{\alpha}} \eta(\varepsilon_F - \varepsilon^{\alpha}_i) C^{\alpha}_{\mu i} C^{\alpha}_{\nu i} \chi_{\mu}(r) \chi_{\nu}(r). \tag{5.11}$$

In practice, the above Heavyside function leads to convergence problems [2] in an SCF procedure. Therefore, the discontinuous Heavyside function is replaced by a steep, but continuous, Fermi function to produce the final expression for the approximate DAC $\rho(r)$

$$\rho(r) \approx 2 \sum_{\alpha}^{subsystems} \sum_{\mu}^{K^{\alpha}} \sum_{\nu}^{K^{\alpha}} p^{\alpha}_{\mu\nu} \sum_{i}^{K^{\alpha}} \frac{1}{1 + e^{-\beta(\varepsilon_F - \varepsilon^{\alpha}_i)}} C^{\alpha}_{\mu i} C^{\alpha}_{\nu i} \chi_{\mu}(r) \chi_{\nu}(r). \tag{5.12}$$

Here, β is an adjustable parameter. Its value has no significant influence on the calculated total energy as long as it is large enough to ensure a sufficiently steep Fermi function. The value of ε_F is adjusted such that the DAC approximate $\rho(r)$ is appropriately normalized [2, 3],

$$\sum_{\alpha}^{subsystems} 2\sum_{\mu}^{K^\alpha}\sum_{\nu}^{K^\alpha} p_{\mu\nu}^{\alpha}\sum_{i}^{K^\alpha}\frac{1}{1+e^{-\beta(\varepsilon_F-\varepsilon_i^\alpha)}}C_{\mu i}^{\alpha}C_{\nu i}^{\alpha}\int\chi_\mu(\boldsymbol{r})\chi_\nu(\boldsymbol{r})d\boldsymbol{r} = N\,, \qquad (5.13)$$

where N is the total number of electrons.

If buffer space were to always encompass the entire molecule, both equations 5.12 and 5.9 would essentially produce the same $\rho(\boldsymbol{r})$. Each set of the subsystem MOs would be the same as the true, global, set. The only remaining error originates from the replacement of the Heavyside function by the Fermi function. However, this has very little effect on $\rho(\boldsymbol{r})$ for a system with an appreciable HOMO/LUMO gap. Unfortunately, if every atom is assigned to buffer space, the computational savings afforded by a DAC strategy would be lost. The DAC approach will only increase the CPU time associated with matrix diagonalization by a factor equal to the number of subsystems as each subsystem calculation is equivalent to the global one. Buffer space must somehow be truncated. There are two commonly employed criteria for buffer space assignment. In some applications, the buffer space is assigned according to the number of covalent bonds separating a possible buffer atom from the nearest subsystem atom [2, 3]. For others, the actual distance between atoms is used to make this determination [14]. Both approaches have their merits. However, as we shall see, neither approach is completely satisfactory.

5.3 Benchmark Calculations

Benchmark calculations of the DAC approach to constructing the electronic density are performed on a series of glycine polypeptides. Calculations using 6-31G**

orbital bases within the local spin density approximation [15] are carried out with the DeFT program package [10]. The geometries of the polypeptides were previously optimized at the PM3 level using the Spartan 4.0 program package [16]. Each polypeptide is divided into terminal HCO- and $-NH_2$ subsystems, and a series of $-NHCH_2CO-$ subsystems. In these calculations, the recently developed DAC schemes are employed to fit the electronic density and the exchange-correlation potentials. These DAC methods are covered in Chapters 2 and 4, respectively. This is done to reduce the CPU time associated with the construction of the KS matrix. It is important to note that the DAC approach for constructing electronic density in the present study is not related to, and in no way affected by, these DAC fitting procedures. For each calculation, a cutoff of 5.0 Å is used for the fit of the electronic density while a cutoff of 7.0 Å is used for the fit of the exchange-correlation potentials. The combined error in the total energy introduced by these DAC fits is no greater than 0.005 kcal mol^{-1} for the molecules involved in this benchmarking procedure.

Benchmark calculations are carried out to establish the errors introduced by the DAC construction of $\rho(\boldsymbol{r})$. Their results will be discussed in the following sections. The effect of increasing the system size of these peptides on the errors generated will be studied. For a given peptide, we will also study how error fluctuates as the peptide adopts different conformations. Hopefully, these benchmark calculations can establish the sources of the errors introduced. This knowledge in turn can be used to improve the efficiency and reliability of these DAC calculations.

5.4 Establishing the Optimal Buffer Space Cutoff

Test calculations are performed on a series of extended glycine polypepeptides, ranging from the tetrapeptide (27 atoms) to the heptapeptide (48 atoms). Buffer atoms are assigned via a distance criterion. Note that for an extended conformation, using either the distance or the number of bonds separating an atom from a subsystem to assign buffer space is, for all intents and purposes, the same. Buffer space cutoffs ranging from 6.0 Å to 13.0 Å are tested on each polypeptide. The geometries of these polypeptides are such that very few atoms are added to subsystem calculations between the 10.0 Å and 11.0 Å cutoffs. Consequently, almost identical results are observed with these two cutoffs.

The first set of calculations investigates the effect of dropping all basis functions centered on atoms beyond the buffer space cutoff from the subsystem calculations. The results are listed in Table 5.1 alongside "all". The errors reported are relative to the results obtained with a conventional non-DAC approach to constructing the electronic density, but with DAC fitting procedures, however. As shown in Table 5.1, the error in the total energy decreases systematically as the buffer space cutoff is increased. In addition, the error increases systematically with system size. The increase in error with respect to system size has also been observed previously [3]. Our results also indicate that excluding all basis functions centered on atoms beyond the cutoff tends to overestimate the total energy, i.e., the DAC scheme always introduces a positive error. This can be easily explained. Each subsystem throws out those basis functions centered on distant atoms. Therefore, each subsystem calculation loses a certain amount of variational flexibility, and a higher energy results.

From Table 5.1, we also observe that the increase in the total energy's error is remarkably consistent with each additional glycine residue. For example, with a 9.0 Å cutoff, an increase of 0.052 kcal mol^{-1} error is observed when the system size is increased from the tetrapeptide to the pentapeptide. It increases by another 0.053 kcal mol^{-1} when the size is increased to the hexapeptide, and finally by another 0.052 kcal mol^{-1} when going to the heptapeptide. The same trend is also observed for other cutoffs, whether the errors are on the order of 0.1-1.7 kcal mol^{-1}, as with the 7.0 Å cutoff, or on the order of 0.001-0.02 kcal mol^{-1} as with the 12.0 Å cutoff. Obviously, using benchmark calculations for small molecules to estimate the expected error for larger systems is not appropriate if this systematic increase in error is not taken into consideration. Consider a tetrapeptide, where a 10.0 Å cutoff introduces a 0.013 kcal mol^{-1} error in the total energy. By extrapolation, using the same cutoff on a decapeptide will introduce an error that is a full order of magnitude larger. Obviously, this systematic increase is exacerbated by the use of the extended conformations and adding the same glycine residue repeatedly. However, this error in the total energy is not necessarily a great problem. In most instances, it is the energy differences between conformations that are of interest. Therefore, if the error remains roughly the same over the various regions of interest on that system's potential energy surface, its exact value is not so important.

To gain a deeper insight into the origin of these errors, four sets of calculations, each employing a different way of partially truncating basis sets beyond the buffer space cutoff, are performed. In these calculations, rather than completely throwing out a basis set, only a subset of the basis functions are excluded from the subsystem calculation. The

other basis functions are retained. A 6-31G** basis set can be divided into four sets of basis functions [13]:

(I) contractions of six primitives that describe the core electrons of heavy atoms (core);

(II) contractions of three primitives that describe the inner regions of the valence orbitals (inner valence);

(III) single uncontracted primitives that describe the outer regions of the valence orbitals (outer valence);

(IV) single uncontracted primitives that allow for polarization of the electronic density (polarization).

For a 6-31G** basis set, a heavy atom from the second period has one (s), four (s, p_x, p_y, p_z), four (s, p_x, p_y, p_z), and six (d_{xx}, d_{xy}, d_{xz}, d_{yy}, d_{yz}, d_{zz} – the Cartesian d functions) basis functions in each category, respectively. A hydrogen atom has zero, one (s), one (s), and three (p_x, p_y, p_z), respectively.

As indicated in Table 5.1, these four groups of basis functions do not contribute evenly to the error in the total energy. Omitting the polarization functions from the subsystem calculations at any given cutoff has the least effect on the total energy. Omitting the outer valence functions turns out to have the greatest effect. For a cutoff value between 6.0-8.0 Å, the errors introduced by omitting both the core and inner valence functions beyond the cutoff are nearly as large as those introduced by excluding only the outer valence functions. However, once the cutoff is extended to 9.0 Å, the error associated with dropping either the core or inner valence functions is much smaller. In particularly, dropping the core functions beyond 9.0 Å from subsystem calculations has

very little effect on the total energy. The results also show that the errors arising from the partial truncation of basis functions increases with system size. When the errors become reasonably small, the sum of all errors associated with each set of functions approximately equals the error that arises when all four sets of functions are eliminated simultaneously. Taking the heptapeptide as an example, with a 10.0 Å cutoff, the error is 0.62 kcal mol^{-1} when all four sets of functions are dropped simultaneously. If each set of functions is excluded individually, the sum of their errors is 0.57 kcal mol^{-1}. In fact, the sum of the errors arising from the individual set is systematically smaller than the error associated with dropping all four sets simultaneously. This observation is consistent with the notion of losing variational flexibility. If only a portion of the basis set is truncated, the remaining basis functions associated with an atom beyond the buffer space cutoff can compensate somewhat for those functions in the set that have been dropped from the subsystem calculation. But, no such compensation can occur if all four sets are simultaneously omitted. It is important to remember that these observations only apply to a reasonably long cutoff (roughly 9.0 Å and beyond). There is another source of error associated with smaller cutoffs. This will be discussed shortly.

In general, excluding polarization functions from subsystem calculations contributes the least to the error in the total energy. This is most likely because polarization functions are not absolutely required to provide a reasonable, qualitative, description of the electronic structure. The core, inner valence, and outer valence, together, constitute a split valence basis set. However, if the inner and outer valence functions are contracted into a single set, a minimal basis set is created. The ability of such minimal and split-valence bases to provide qualitatively accurate results is well

documented [13]. Excluding polarization functions from subsystem calculations will only result in a loss of variational flexibility. If this happens, each and every subsystem MO is, at the very least, qualitatively well described by a split-valence basis.

It is apparent that the spatial extent of the basis functions plays an important role. In these four groups of basis functions, the outer valence functions are the most diffuse, while the core functions are the least. The inner valence and polarization functions possess roughly the same spatial extent. Neglecting the polarization functions at sufficiently long cutoffs, we see that there is a clear correspondence between the spatial extent of the functions being dropped and the error resulting from these functions being dropped. Diffuse functions contribute most to the error. The tails of diffuse functions from distant atoms can be used to improve the description of an MO in the vicinity of the actual subsystem atoms. When a cutoff is imposed on subsystem calculations, these diffuse functions will be excluded. The variational flexibility provided by these diffuse functions is lost. Therefore, a higher total energy results.

However, with shorter cutoffs, this simple relationship does not always holds. In fact, the effect of dropping core functions becomes more important than dropping the more diffuse inner valence functions. Therefore, there must be another factor that plays a more important role when shorter cutoffs are employed. Recall that, within a given subsystem calculation, any two subsystem molecular orbitals are orthogonal to one another. However, these subsystems' molecular orbitals are not strictly orthogonal to the molecular orbitals obtained from other subsystem calculations. Also, they are not orthogonal to the global molecular orbitals obtained from a conventional calculation. By omitting basis functions necessary for the proper description of those molecular orbitals

surrounding the subsystem in question, the MOs in the subsystem are satisfying orthogonality constraints that are very much unlike those in the global, non-DAC, approach. Note that these constraints are merely *different*, and not necessarily any easier or harder to satisfy. Consequently, the sign on the total energy error could be either positive or negative. This is indeed what we observe. For instance, removing core functions beyond an 8.0 Å cutoff, the DAC $\rho(\boldsymbol{r})$ yields a total energy that is too low. In most instances, however, the loss of variational flexibility is still the dominating factor. Therefore, the DAC energies still tend to be too high. The negative errors that can arise from the improper orthogonality constraints indicate that, overall, the DAC approach is not variational. Hence, there is no guarantee that the error would systematically become smaller each time the buffer space cutoff is set to a larger value.

5.5 Multiple Buffer Space Cutoffs

In the previous section, we have established that the various types of basis functions do not contribute equally to the DAC errors. Further, at sufficiently large cutoffs (9.0 Å and beyond), the errors increase linearly with system size. Recall that the objective of truncating the buffer space in a DAC calculation is to reduce the overall computational time. Therefore, if different cutoffs can be used for different basis functions, significant CPU time could be saved without sacrificing too much in accuracy.

To test the feasibility of using different buffer space cutoffs for different types of basis functions, calculations are carried out on the extended conformation of the octapeptide (55 atoms). In this benchmark calculation, we arbitrarily set our error

tolerance at 0.1 kcal mol^{-1}. The problem is further simplified by our arbitrarily insisting that no one single set of basis functions will contribute more than 0.025 kcal mol^{-1} to the error. By inspection of Table 5.1, and taking the linear increase in error with system size into account, a 9.0 Å cutoff is used for the core, inner valence and polarization functions, and a cutoff of 12.0 Å is used for the outer valence functions. With these cutoffs, we should expect the errors arising from these four sets of basis functions to be approximately 0.007, 0.018, 0.010, and 0.015 kcal mol^{-1}, respectively. The sum of these predicted values is 0.050 kcal mol^{-1}. Two test calculations, one using the conventional, non-DAC, approach and the other using a DAC approach with multiple cutoff values are performed on the octapeptide. The total energy obtained by the DAC calculation is found to be 0.036 kcal mol^{-1} higher than that obtained from the conventional calculation. This is very close to our predicted value. This multiple cutoff approach makes for a far more efficient DAC calculation. For a general molecular system extending in all three dimensions, a 12.0 Å sphere encloses a volume that is more than double that enclosed by a 9.0 Å sphere. In our test calculations with a 6-31G** basis, 11 of the 15 functions are excluded within the shell between 9.0 Å and 12.0 Å. Even though atoms located between the 9.0 Å and 12.0 Å spheres are part of "buffer space", within this shell, only the four outer valence functions are retained. Recall that it is the number of basis functions, and not the number of atoms that determines the cost of a subsystem calculation. Therefore, any modification of the DAC approach that can significantly cut down the number of basis functions in a subsystem calculation will be of great importance. Multiple cutoffs are one such approach. Use of multiple cutoffs will also become far more important when performing calculations with higher quality basis sets. A better basis set will most

often contain even more diffuse functions. Consequently, the discrepancy in the spatial extents of the "tight" and the diffuse functions will become even more prominent. The need for multiple cutoffs will become even more crucial in achieving greater efficiency within a DAC scheme.

5.6 Distance and Topology Buffer Space Criteria

Since buffer atoms must be included in any DAC approach, defining the buffer space is a crucial aspect of the DAC calculation. Two buffer space criteria, distance and topology, are considered in this study. The topology criterion uses the number of bonds separating a potential buffer atom from the subsystem of interest to determine whether or not that atom is to be assigned to the buffer space of that subsystem. An n bond cutoff assigns an atom to a subsystem's buffer space if n or less covalent bonds separate that atom from any one subsystem atom.

Up to now, we have established that the errors introduced by a DAC approach to constructing $\rho(\boldsymbol{r})$ can be substantial. As a result, the total energy obtained is not reliable. However, when scanning the potential energy surface of a molecule, it is relative energies that are of utmost interest, e.g., the energy difference between two different conformations of the same molecule. When employing an approximate method to improve the efficiency of a calculation (such as the DAC approach), it is important to establish whether the errors introduced are systematic throughout the potential energy surface. A classic example of systematic errors introduced by a methodology is the Hartree-Fock method. The correlated motion of the electrons is not adequately described

in a Hatree-Fock calculation. This typically introduces a roughly1 eV error in to the binding energy for a pair of electrons in a well localized orbital. Hence, the total energy of a molecule obtained within a Hartree-Fock calculation is not accurate. However, this inadequacy is experienced by all the conformations of that molecule. When calculating the relative energy, we can rely on a systematic cancellation of errors. This yields acceptable relative energies, though they still are not as accurate as those obtained by correlated methods (such as the MP2 method). Similarly, we would like the errors introduced by our DAC construction of $\rho(r)$ to be systematic throughout the potential energy surface. To establish if this is the case, we will perform benchmark calculations on three different conformers of the glycine pentapeptide. Test calculations are performed using both the distance and buffer space criteria with the same basis sets and methods outlined above. The three chosen conformations are the fully extended conformer (Figure 5.1), the α helix conformation (Figure 5.2), and the conformation containing a series of four seven-member hydrogen-bonded (C_7 [17]) rings (Figure 5.3). Among these conformations, the α helix has the most compact structure, followed by the C_7 conformer. The first set of calculations uses a distance criterion to assign buffer atoms. All basis functions associated with an atom beyond the cutoff are omitted from a subsystem calculation. The second set of calculation uses the topology criterion. In this approach, all conformers will have the same set of buffer atoms for each subsystem calculation as the connectivity is not altered when adopting new conformations. For the α helix conformer, use of the distance criterion places the largest number of buffer atoms in each subsystem calculation because it is the most compact structure. The extended

conformer always has the least. The results of these test calculations are presented in Table 5.2.

Using a distance cutoff, the α helix conformer always has the smallest error while the extended conformer always has the largest. Actually, since no atom is further than 7.0 Å from any one subsystem in the α helix conformer, no error exists when the buffer space is extended to 7.0 Å or beyond in the α helix conformer. However, with a 7.0 Å cutoff, the extended conformer still has an error just over a full kcal mol^{-1}. Obviously, assigning buffer space via a distance criterion will always artificially prefer a compact structure to a more extended structure. No systematic cancellation of errors will occur between these two conformers' calculations. Hence, relative energy differences between the two will be unreliable. With a very short cutoff, such as 4.0 Å, the issue of improper orthogonalization dominates and negative errors in the total energy are observed in the extended and α helix conformers.

Assigning buffer atoms via a topology criterion will force all calculations to have the same number of buffer atoms within the subsystem calculations. It is hoped that this will eliminate the artificial preference for more compact structures that is observed when using a distance criterion. As indicated in Table 5.2, with any cutoff less than 11 bonds, the DAC calculations provide grossly inaccurate results for the α helix conformer. This is totally understandable. The hydrogen bonds in the α helix are between a carbonyl oxygen and an amide hydrogen in a subsystem eleven bonds away. These two atoms are not included in each other's buffer space if a cutoff of 10 bonds or less is used. The electronic structure surrounding these hydrogen bonds are greatly perturbed in DAC calculations when such cutoffs are employed. Since a negative error cannot result from a

loss in variational flexibility, this suggests that the orthogonality constraints being satisfied by the subsystem MO's are very much unlike those in the true global set. The total energies generated are far too low. Even when the cutoff is extended to 13 bonds, an appreciable error still exists for the α helix since some atoms as little as 3.5 Å away are still being excluded from the terminal subsystem calculations (the buffer space of the central -$NHCH_2CO$- subsystems enclose the entire polypeptide). With a cutoff of fourteen bonds, each subsystem's buffer space spans the entire molecule. As a result, the true, non-DAC, results are recovered.

As for the C_7 conformer, the results are very bad if a cutoff of four bonds or less is used. Hydrogen bonds in this conformer occur within the seven-membered O=C-N-C_α-C-N-H rings (Figure 5.3). The carbonyl oxygen and the amide hydrogen are separated from each other's subsystem by five bonds. Therefore, a five-bond cutoff provides a reasonable DAC energy. A four-bond cutoff does not. It is evident from Table 5.2 that the error produced could be either positive (as with the four bond cutoff) or negative (as with the three bond cutoff).

In the case of the extended conformer, no cutoff value in this study provides grossly inaccurate results. The hydrogen bonding in the extended conformer is occurring between a carbonyl oxygen and amide hydrogen within the same –$NHCH_2CO$- subsystem. Therefore, a reasonable description of these hydrogen bonds is provided with any cutoff value. Provided that the buffer space cutoff allows for a proper description of the hydrogen bonding in the C_7 conformer (i.e., beyond a four bond cutoff), the errors for the extended and C_7 conformers would be of approximately equal magnitude. If these errors were to be divided by the total number of atoms, the errors obtained are slightly

smaller than those reported by Yang and Lee [3]. They implemented the density formulation of divide and conquer approach within a DF method similar to DMol [18]. Using a double-zeta + polarization basis set, they found that the error obtained for the pentapetide with a 7 bond buffer space cutoff is 0.0064 kcal/mol per atom. Our DAC errors for the extended and the C_7 conformers are 0.0035 and 0.0031 kcal/mol per atom. respectively.

5.7 Conclusion

An adequate buffer space must be assigned to each subsystem within any DAC calculation. A criterion based solely on the distance or the number of bonds separating a potential buffer atom from a subsystem will generally fail to provide adequate buffer spaces. A distance approach will artificially favour compact structures. A topology approach will artificially favour extended structures. It is important to note that reliable results can only be obtained if we are confident that a DAC scheme will reproduce the true, non-DAC, total energy. We can never rely on a systematic cancellation of errors when a DAC scheme is used to construct $\rho(\boldsymbol{r})$. The relative energies between two very different conformers are not reliable. Using a distance criterion to assign buffer space can provide reliable total energies if the cutoff is sufficiently large that the errors in the total energies are of the same order as the errors in the relative energies that we are willing to tolerate. In the case of small extended glycine polypeptides, 0.01 kcal mol^{-1} accuracy requires a 12.0 to13.0 Å cutoff, while a cutoff of 9.0-10.0 Å pprovides 0.1 kcal mol^{-1} accuracy. Errors in relative energies can be minimized if the geometry changes are kept as small as possible. For example, the errors in the total energies of the extended

and C_7 conformers are far more similar to one another than they are to the errors of the α-helix conformer. Nevertheless, we must always take into account that these errors will rise steadily with system size.

It has been established that DAC calculations with short and long cutoff are governed by two different sources of errors. With longer cutoffs, the primary source of error is the loss of variational flexibility in the individual subsystem calculations. Hence, the DAC energies are too high. Though the loss of variational flexibility still contributes to the errors when shorter cutoffs are used, the issue of improper orthogonalization of the subsystem MO's to the true global MO's dominates. When this occurs, the DAC energy might become either too high or too low. However, we feel that if a chosen cutoff is so short that the latter source of error dominates, then that cutoff is far too short and it is highly unlikely that it can provide any meaningful results.

It has also been shown that some basis functions contribute more to the error than others. Based on this fact, a more efficient DAC approach can be developed. Shorter cutoffs are used for those basis functions that contribute the least to the DAC error. Longer cutoffs are used for those functions that contribute the most. The test calculation on the extended glycine octapeptide has shown that a reliable result can be obtained via a multiple cutoff approach. Furthermore, analysis of benchmark calculations on smaller systems can be used to reliably predict the error in the DAC total energy for a larger related system. The multiple cutoff approach allows for a far more efficient DAC calculation, especially when working with compact globular structures.

In this study, the issue of CPU time is not addressed. The DAC approach was introduced as an alternative to solving the global KS or HF equations of very large

systems. The test calculations in our study are on relatively small systems, however. Therefore, the DAC calculations still require more CPU time than their conventional counterparts. Although the DAC scheme foregoes the need of handling the larger global matrix and deals only with smaller matrices, the total sum of the CPU times associated with the diagonalization of the subsystem matrices is greater than the CPU time associated with the diagonalization of the single, global, matrix. However, for sufficiently large systems, the DAC method will eventually scale linearly and become more efficient than the conventional approach which scales cubically. This linear scaling behaviour has been observed in semi-empirical calculations [8, 9] on much larger systems. Our results suggest that linear scaling is far easier to achieve with a semi-empirical calculation since a minimal basis is used and no diffuse functions exit. The buffer space cutoffs within semi-empirical DAC calculations can therefore be much shorter. Linear scaling would be even harder to achieve if larger basis sets containing even more diffuse functions than those found in a 6-31 G** basis were to be employed for DAC calculations. Fortunately, in the foreseeable future, it is unlikely that bases more complete than the split valence plus polarization bases studied here will be used for calculations on very large systems. However, clearly, the use of minimal bases in benchmark DAC calculations can be misleading when trying to establish the usefulness of DAC approaches with reasonably flexible basis sets such as the 6-31G** basis.

Table 5.1

Errors in the 6-31G total energy (kcal mol^{-1}) introduced by the DAC construction of $\rho(r)$ for a series of extended glycine polypeptides.**

Functions dropped	Cutoff	Tetrapeptide	Pentapeptide	Hexapeptide	Heptapeptide
all	6.0	+1.292	+1.933	+2.571	+3.208
	7.0	+0.656	+1.007	+1.361	+1.711
	8.0	+0.298	+0.481	+0.664	+0.846
	9.0	+0.044	+0.096	+0.149	+0.201
	10.0	+0.013	+0.029	+0.046	+0.062
	11.0	+0.013	+0.030	+0.046	+0.062
	12.0	+0.003	+0.007	+0.012	+0.016
	13.0	+0.000	+0.001	+0.002	+0.002
core	6.0	+0.475	+0.735	+0.994	+1.254
	7.0	+0.294	+0.461	+0.628	+0.795
	8.0	-0.069	-0.108	-0.148	-0.184
	9.0	+0.000	+0.003	+0.005	+0.006
	10.0	+0.000	+0.002	+0.004	+0.005
	11.0	+0.000	+0.002	+0.004	+0.005
	12.0	-0.001	-0.001	-0.002	-0.002
	13.0	+0.000	+0.000	+0.000	+0.000
inner valence	6.0	+0.348	+0.522	+0.707	+0.884
	7.0	+0.131	+0.198	+0.264	+0.331
	8.0	+0.040	+0.061	+0.084	+0.106
	9.0	+0.004	+0.007	+0.010	+0.014
	10.0	+0.003	+0.005	+0.008	+0.010
	11.0	+0.003	+0.005	+0.008	+0.010
	12.0	+0.001	+0.002	+0.003	+0.004
	13.0	+0.000	+0.000	+0.000	+0.001
Outer valence	6.0	+0.871	+1.320	+1.771	+2.219
	7.0	+0.346	+0.528	+0.715	+0.896
	8.0	+0.125	+0.200	+0.276	+0.351
	9.0	+0.017	+0.036	+0.056	+0.075
	10.0	+0.008	+0.019	+0.030	+0.040
	11.0	+0.008	+0.019	+0.029	+0.040
	12.0	+0.002	+0.005	+0.009	+0.012
	13.0	+0.000	+0.001	+0.001	+0.002
polarization	6.0	+0.114	+0.176	+0.239	+0.301
	7.0	+0.023	+0.037	+0.050	+0.064
	8.0	+0.025	+0.041	+0.057	+0.072
	9.0	+0.003	+0.005	+0.006	+0.008
	10.0	+0.000	+0.001	+0.001	+0.002
	11.0	+0.000	+0.001	+0.002	+0.002
	12.0	+0.000	+0.001	+0.001	+0.001
	13.0	+0.000	+0.000	+0.000	+0.000

Table 5.2

Errors in the 6-31G total energy (kcal mol^{-1}) introduced by the DAC construction of $\rho(r)$ for three conformers of the glycine pentapetide.**

Cutoff	extended	C_7	α helix
4.0 Å distance cutoff	-6.542	+1.485	-0.731
5.0 Å distance cutoff	+2.833	+1.605	+0.135
6.0 Å distance cutoff	+1.933	+0.034	+0.063
7.0 Å distance cutoff	+1.007	+0.037	+0.000
8.0 Å distance cutoff	+0.481	+0.007	+0.000
9.0 Å distance cutoff	+0.096	+0.002	+0.000
10.0 Å distance cutoff	+0.029	+0.000	+0.000
3 bond cutoff	-4.381	-64.236	-22.414
4 bond cutoff	+2.883	+40.689	-21.590
5 bond cutoff	+1.000	+1.394	-29.574
6 bond cutoff	+0.494	+0.199	-30.030
7 bond cutoff	+0.119	+0.104	-9.586
8 bond cutoff	+0.030	+0.010	-27.008
9 bond cutoff	+0.010	+0.005	-32.809
10 bond cutoff	+0.001	+0.002	-35.697
11 bond cutoff	+0.000	+0.000	+0.727
12 bond cutoff	+0.000	+0.000	+1.165
13 bond cutoff	+0.000	+0.000	+0.575
14 bond cutoff	+0.000	+0.000	+0.000

Figure 5.1

The fully extended conformation of the glycine pentapeptide.

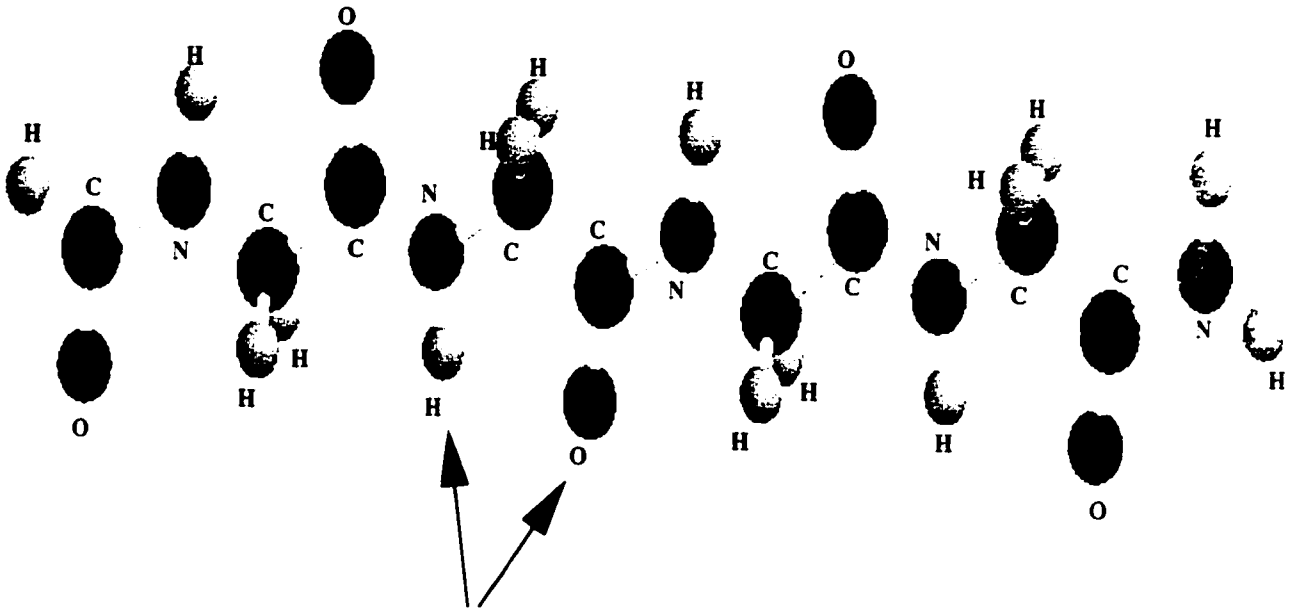


Figure 5.2

The α helix conformation of the glycine pentapeptide.

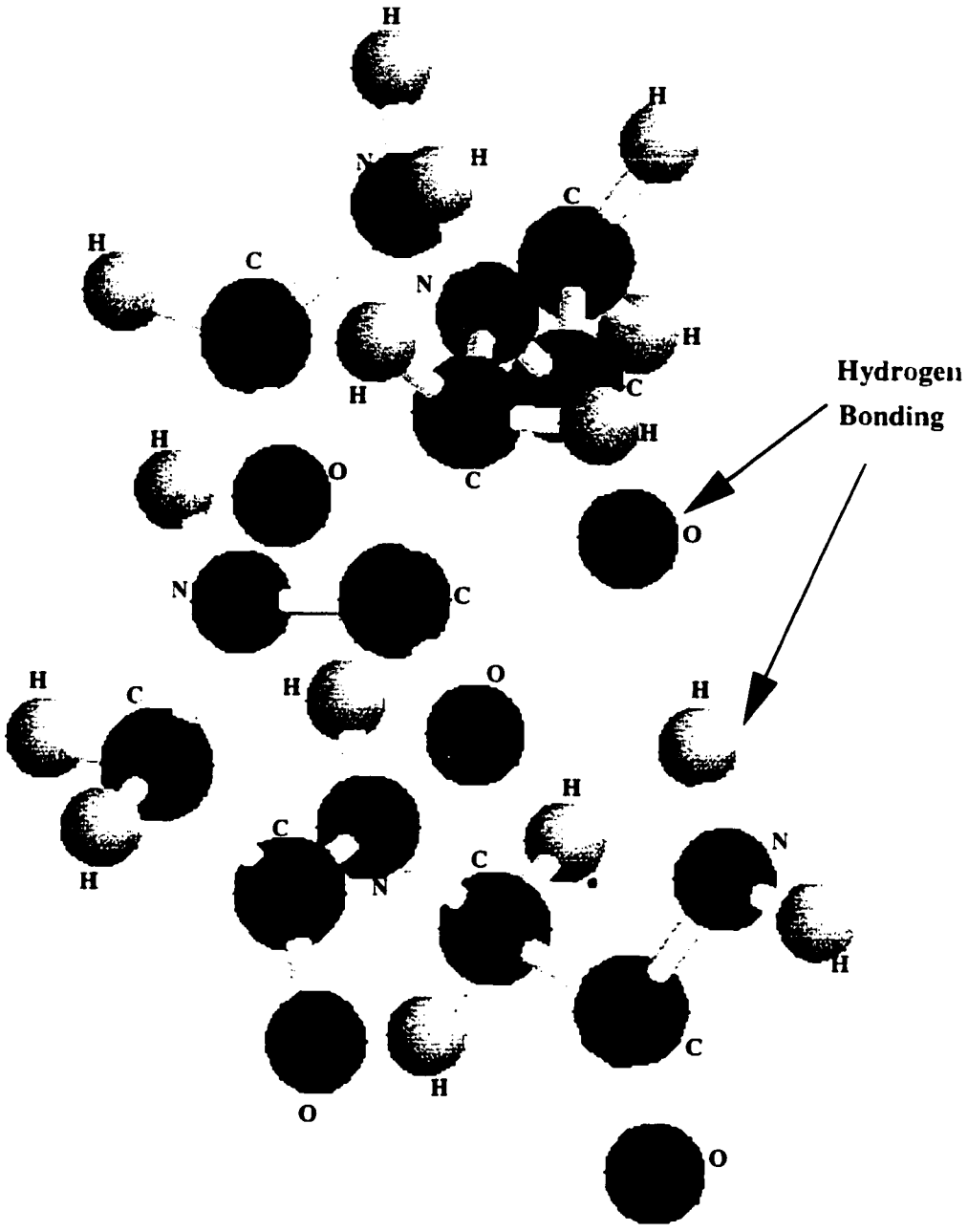
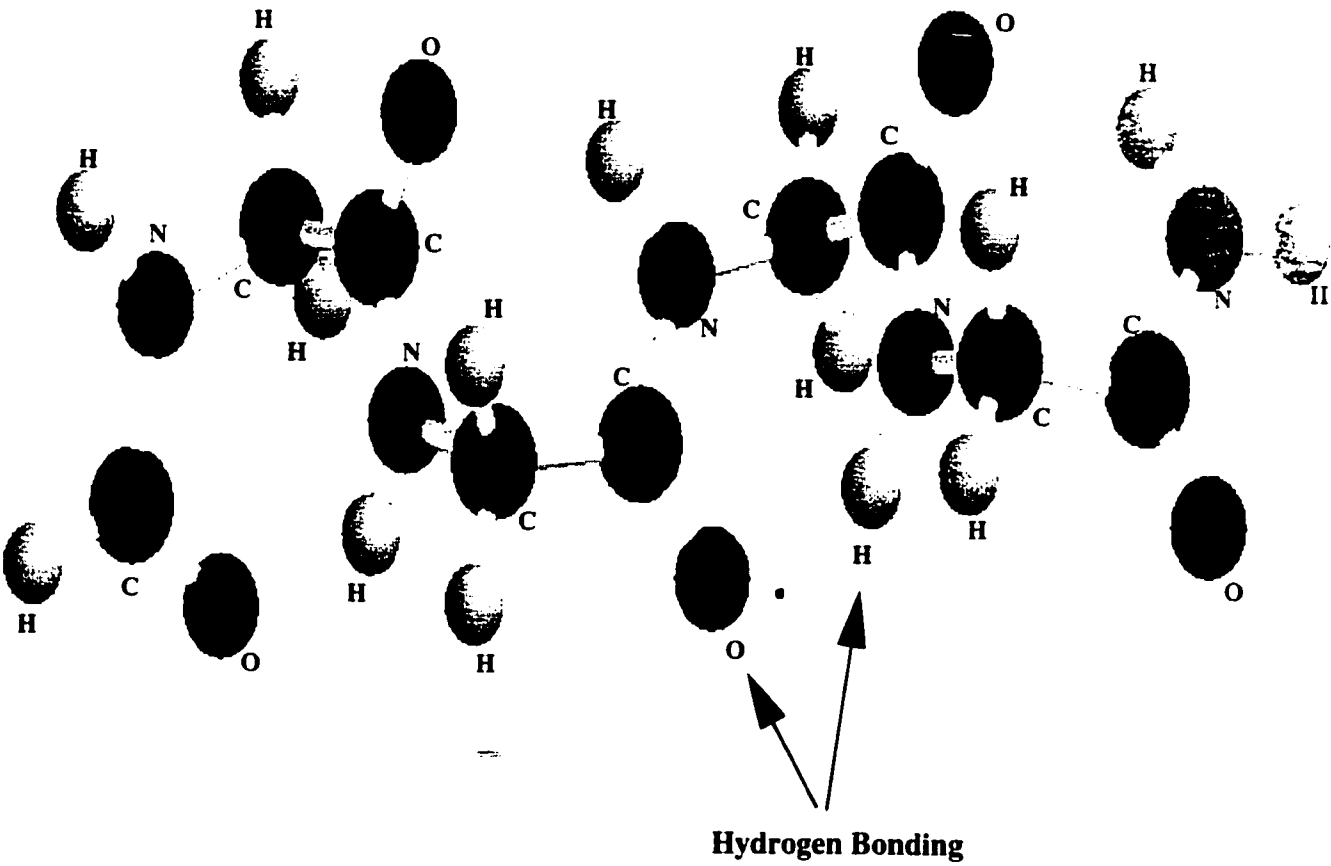


Figure 5.3

The C_7 conformation of the glycine pentapeptide.



References

(1) W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

(2) W. Yang, *J. Mol. Struct. (THEOCHEM)* **225**, 461 (1992).

(3) W. Yang and T. S. Lee, *J. Chem. Phys.* **103**, 5674 (1995).

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

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

(6) M. Challacombe, E. Schwegler and J. Almlof, *J. Chem. Phys.* **104**, 4685 (1996).

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

(8) S. L. Dixon and K. M. Merz, *J. Chem. Phys.* **104**, 6643 (1996).

(9) T. S. Lee, D. M. York and W. Yang, *J. Chem. Phys.* **105**, 2744 (1996).

(10) A. St-Amant, the latest version of DeFT can be downloaded from (http://www.chem.uottawa.ca/DeFT.html).

(11) B. I. Dunlap, J. W. D. Connolly and J. R. Sabin, *J. Chem. Phys.* **71**, 3396 (1979).

(12) J. Andzelm and E. Wimmer, *J. Chem. Phys.* **96**, 1280 (1992).

(13) W. J. Hehre, L. Radom, P. v. R. Schleyer and J. A. Pople, *Ab Initio Molecular Orbital Theory* (Wiley-Interscience; New York; 1986).

(14) Q. Zhao and J. B. Nicholas, *J. Chem. Phys* **104**, 767 (1996).

(15) R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules* (Oxford University Press; New York; 1989).

(16) Spartan SGI/R4K Release 4.0.3.c (Wavefunction Inc., Irvine, 1995).

(17) T. Head-Gordon, M. Head-Gordon, M. J. Frisch, C. L. Brooks and J. A. Pople, *J. Am. Chem. Soc.* **113**, 5989 (1991).

(18) B. Delley, *J. Chem. Phys.* **92**, 508 (1990).

Chapter 6

Conclusion and Future Work

Conclusion

Over the years, the utility of quantum mechanical methods in studies of chemical and biochemical systems has been well established. However, the computational cost of these methods generally has a high order dependence on system size. This severely limits the size of the molecular systems that can be studied. Therefore, to realistically model truly large systems, this unfavorable scaling must be addressed. Recently, much effort has been made in developing methods that scale linearly with system size so as to make accurate calculations on larger systems become a reality. Yang's divide and conquer philosophy is one such promising approach. It allows us to forego the need to ever treat the global system. Instead, the properties of the global system are obtained from a series of subsystem calculations. The approaches taken to achieve linear scaling within DeFT are inspired by this DAC philosophy, and they have been outlined in earlier chapters.

In Chapter 2, we confirmed that a DAC fit of the density does scale linearly with system size. Though it is possible to achieve linear scaling for the evaluation of the Coulomb component of the HF and KS matrices without the use of a fitted density, we established that a fitted density can dramatically reduce the number of floating operations needed to be performed. If a fitted density is not sufficiently accurate, we demonstrated that it could be replaced by a more accurate hybrid DAC density. This hybrid DAC density is constructed by mixing true and fitted subsystem densities obtained from within a DAC scheme. The quality of these hybrid DAC densities can be systematically improved by extending the true density regions. For large systems, evaluating the

electrostatic potential on a grid with a hybrid DAC density has been shown to exhibit near linear scaling behaviour.

Chapter 3 described an efficient algorithm for carrying out a DAC calculation on massively parallel computers. This algorithm exploits the coarse grain parallelism provided by the DAC philosophy to achieve high efficiency on both small and large numbers of processors. The precise method of choice depends on the number of processors and subsystems available. Benchmarks calculations have been performed on a T3E computer with up to 48 processors. Their findings are as follows. With a small number of processors, the simple fine grain approach is most efficient. As more processors are added, however, the poor scalability of the straightforward fine grain approach becomes apparent. The combined coarse/fine grain scheme thus becomes a better approach. A single pass combined coarse/fine grain parallel scheme performs all subsystem calculations in unison with more processors assigned to longer subsystem calculations. Load balancing is ensured by assigning each subsystem to a fraction of the total number of processors. This fraction is as similar as possible to the fraction of the total CPU time required by that subsystem. Optimal load balancing can only be achieved when the ratio of processors to subsystems is large enough. If the processor to subsystem ratio is relatively small, multiple passes can be performed so that this ratio within an individual pass is greatly enhanced. However, once more processors become available, it is best to reduce the number of passes so that each subsystem gets fewer processors. This reduces the impact of Amdahl's law on any individual subsystem calculation. Hence, the efficiency within each subsystem calculation increases. Allowing the algorithm to adapt

itself to the processor to subsystem ratio, near linear speedups are observed up to 48 processors.

The modification of exchange and correlation modules within DeFT were outlined in Chapter 4. These were required to achieve near linear scaling in DeFT's XC procedures. The major change made was the implementation of a DAC approach within the XC fitting procedure. Errors arising from the DAC XC fitting procedures can be reduced to under 0.01 kcal mol^{-1} when the buffer space cutoff is extended to 7.0 Å. This error slowly, but systematically, increases with system size. Analytic integration of the fitted XC terms on the final SCF cycle on an augmented grid yields an XC energy that is less susceptible to grid noise and yields a reliable energy gradient. It has been established that a far smaller number of grid points are required for the fits of XC terms throughout the course of the SCF to maintain the same level of precision in the energy. For large enough systems (in our test case, systems with at least 69 atoms), the CPU time required for each XC step increases near linearly with system size. None of the XC steps scales worse than $N^{1.4}$ beyond this point.

Chapter 5 discussed the efficiency and reliability of a DAC approach to constructing the electronic density. Benchmark calculations were performed on a series of glycine polypeptides using a 6-31G** basis set. It has been found that assigning the buffer space via a distance criterion favors a more compact structure. On the other hand, a topological criterion favors extended structures. No systematic cancellation of errors between two very different conformations of the polypeptide is possible, and hence relative energies are not reliable. Reliable results can only be obtained if we are confident that the DAC scheme can reproduce the true non-DAC energy. It has also been

established that DAC errors can be reduced systematically by increasing the buffer space cutoff. For an extended polypeptide of roughly 40 atoms, a buffer space cutoff of at least 12.0Å is required to achieve 0.01 kcal mol^{-1} accuracy. Errors in relative energies can be minimized if the changes in the geometry are kept as small as possible. These errors increase systematically with system size. This fact must always taken into account when estimating errors for larger systems. We have established that DAC calculations with short and long cutoffs are governed by two very different sources of errors. With longer cutoffs, a loss in variational flexibility dominates the errors. A higher DAC energy results. With shorter cutoffs, the improper orthogonalization of the subsystem molecular orbitals to the true global molecular orbitals dominates the errors. This causes the errors associated with shorter cutoffs to be either positive or negative. We also found that different types of basis functions contribute to different extents to the error. Therefore, a multiple cutoff scheme may be used to save considerable amounts of CPU time.

Future Work

Near linear scaling for the fits of the density and the exchange and correlation terms have been achieved. Within DeFT, diagonalization and evaluation of the Coulomb interactions still need to be modified before linear scaling within the entire program is achieved.

Coulomb interactions between charge distributions can be classified as either short- or long- range interactions. A long-range interaction is an interaction between two charge distributions that do not overlap. Conversely, a short-range interaction is an

interaction between two charge distributions that do overlap. Since charge distributions have a finite extent, overlap with other distributions are limited to their immediate vicinity. For a large molecule, increasing system size increases the number of charge distributions but the number of appreciable overlaps for a given charge distribution remains constant. The total number of short-range interactions thus increases linearly with system size.

To achieve linear scaling for Coulomb interactions, the continuous fast multipole method has been established to be a very efficient approach. In this approach, charge distributions are sorted into lowest level boxes according to their extents and the spatial location of their centers. Charge distributions that do overlap with others are placed within a special level where interactions between them are evaluated explicitly. Interactions between charge distributions that do not overlap can be accurately approximated by multipole expansions. The CPU time required to evaluate these long-range Coulomb interactions is trivial compared to evaluate the integrals explicitly for short-range interactions. Therefore, the total CPU time required to calculate Coulomb interactions is dominated by the number of integrals that require explicit evaluations. As established earlier, the number of short-range interactions increases linearly with system size. Therefore, the CPU time for explicit evaluation of the integrals grows linearly with system size, and the total CPU time for evaluating Coulomb interactions increases linearly. The continuous fast multipole method will thus be implemented within DeFT in the near future.

The simplest approach to solving the HF or KS equations is by diagonalization. Diagonalization, however, scales cubically with system size. To achieve linear scaling,

an alternative procedure must be sought. It has been found that the HF or KS equations can also be solved by annihilating all interactions through the Fock or KS operator between the occupied and virtual MO's. Annihilation is done by a Jacobian rotation. For any molecular system, the number of occupied MO's increases as N, where N is related to system size. The computational cost of a rotation between two MO's is proportional to the number of basis functions. This is also proportional to N. The interaction between an occupied MO and each virtual MO must be annihilated. The number of virtual orbitals also increases linearly with N. Therefore, if nothing else is done, this approach scales as N^3. However, if all the MO's are localized, the size of each localized MO will become independent of system size. Also, we need only worry about annihilating occupied/virtual interactions between orbitals that are located near each other. For a large enough system, the number of neighbouring molecular orbitals becomes a constant. The interaction between occupied localized MO's with unoccupied localized MO's separated by a large distance will automatically be zero. Therefore, for large enough systems, the CPU time spent annihilating interactions between localized MO's only rises linearly with system size. This approach has recently been implemented within DeFT. Benchmarking this procedure is underway.

Claims to Original Research

The works described in this thesis have been published as the following refereed journal articles:

1. Sor Koon Goh and Alain St-Amant, "Using a Fitted Electronic Density to Improve the Efficiency of a Linear Combination of Gaussian-Type Orbitals Calculation", *Chem. Phys. Lett.* **264**, 9 (1997).

   In this paper, we demonstrate that a divide and conquer fit of the electronic density can be done in a fashion that scales linearly with system size. We also propose the use of a hybrid divide and conquer density for application that require a more accurate density than simply the fitted density.

2. Sor Koon Goh and Alain St-Amant, "Improving the Efficiency and Reliability of the Divide and Conquer Approach to Constructing the Electronic Density", *Chem. Phys. Lett.* **274**, 429 (1997).

   A detail investigation of the errors introduced by a divide and conquer construction of the electronic density is performed. It has been found that different types of basis functions contribute to varying degrees to the error in the total energy. This fact can be exploited to define a multiple buffer space cutoff approach that will improve the efficiency of a divide and conquer calculation. Topology and distance criteria for buffer space cutoffs were investigated. Neither is found to be satisfactory.

3. Sor Koon Goh, Carlos P. Sosa, and Alain St-Amant, "A Scalable Divide and Conquer Algorithm Combining Coarse and Fine Grain Parallelization", *Theor. Chem. Acc.* **99**, 197 (1998).

   An efficient algorithm for a divide and conquer calculation on massively parallel computers is described. Near linear speedups are observed up to 48 processors on a Cray T3E computer. Optimal efficiency is maintained up to an even larger number of processors by allowing the algorithm to adapt to the number of processors available and the number of subsystems to be treated.

4. Sor Koon Goh, Roger T. Gallant, and Alain St-Amant, "Towards Linear Scaling with Fitted Exchange-Correlation Terms in the LCGTO-DF Method via a Divide and Conquer Approach", *Int. J. Quantum Chem.* **69**, 405 (1998).

   The divide and conquer fits of the XC terms are described. The necessary modifications made to achieve near-linear scaling for every XC procedure within DeFT is discussed. The new grid implemented within DeFT to achieve higher numerical precision is presented and tested.