FEATURE ARTICLE

Linear Scaling Density Functional Calculations with Gaussian Orbitals

Gustavo E. Scuseria

Department of Chemistry, Rice University, Houston, Texas 77005-1892

Received: February 22, 1999

Recent advances in linear scaling algorithms that circumvect the computational bottlenecks of large-scale electronic structure simulations make it possible to carry out density functional calculations with Gaussian orbitals on molecules containing more than 1000 atoms and 15 000 basis functions using current workstations and personal computers. This paper discusses the recent theoretical developments that have led to these advances and demonstrates in a series of benchmark calculations the present capabilities of state-of-the-art computational quantum chemistry programs for the prediction of molecular structure and properties.

Introduction

The fundamental goal of quantum chemistry is to solve the time-independent Schrödinger equation for a molecular system within the Born−Oppenheimer (fixed-nuclei) approximation. This is a differential equation for the eigenfunctions of the molecular Hamiltonian in $3N_e$ dimensions, where N_e is the number of electrons. The reasons for solving this equation should be obvious: equilibrium geometries, potential energy surfaces, activation barriers, heats of reaction, excitation energies, infrared and Raman spectra, electric properties such as dipole moments and polarizabilities, and magnetic properties such as chemical shifts, to name just a few, can all be calculated from first principles.

Since their introduction almost 50 years ago,[1] Gaussian orbitals have played a central role in the development of modern computational quantum chemistry. Even though Gaussian orbitals do not satisfy certain basic properties of the electronic wave functions, such as the electron−nuclear cusp and the exponential tail, they possess a fundamental advantage over other types of one-electron functions: if one expands the electronic wave functions as linear combinations of antisymmetrized Slater determinants built from molecular orbitals (MO) which in turn are constructed as linear combinations of Gaussian atomic orbitals (AO) and applies the variational principle, then all integrals that appear in the procedure can be solved analytically.

If the determinant expansion is chosen such that all excitations from occupied into unoccupied orbitals are included within a given atomic Gaussian basis (which in practice must be truncated to a finite size), the method yields exact eigenfunctions of the molecular Hamiltonian. Thus, ground and excited states of all symmetries can be so obtained with similar effort. Such a method is known as full configuration interaction (CI) or complete CI when done in terms of a complete set of molecular orbitals. The resulting expressions can be evaluated numerically using computers, and much effort has been devoted by quantum chemists to derive and implement very efficient algorithms for this purpose.

Thus, using Gaussian orbitals one can in principle calculate the eigenfunctions (and *all* of the properties mentioned above) of *any* molecular system. Unfortunately, the scaling of this computation with molecular size is roughly exponential, thus limiting the full CI approach to rather small molecules and basis sets, even with the most powerful computers. Evidently, approximations that preserve the basic physics in the system but alleviate the computational scaling are needed if one desires to carry out accurate calculations on realistic molecules.

The history of quantum chemistry is very much related to the quest for the best possible representation of the molecular wave function that will preserve conceptual simplicity and afford efficient computation. Many methods have been developed over the years to tackle the molecular electronic structure problem, but their review is outside the scope of this article. One should point out, however, that computational scaling has not been an important concern in this development, at least not until recently.

Some of the most popular quantum chemistry methods that include the effects of electron correlation, such as the second-order Moller−Plesset perturbation theory (MP2) and coupled cluster theory including all single, double, and perturbative triple excitations, CCSD(T), have $O(N^5)$ and $O(N^7)$ computational scaling, respectively, when formulated in terms of molecular orbitals. Here, N is a measure of molecular size, typically the number of basis functions in the calculation. Methods such as single-reference CCSD(T) became widely popular on the basis of their accuracy and speed[2] compared to alternative descriptions based on multireference formulations. Computational scaling is not an issue for small molecules because of the usually low prefactor that the steep scaling steps have. However, these steps take over and become fundamental bottlenecks for large enough N, severely limiting the applicability of accurate coupled cluster methods to large molecules.

Independent particle models such as Hartree−Fock (HF) or Kohn−Sham density functional theory (DFT)[3−5] are built upon a *single* Slater determinant wave function, and, even though they involve a self-consistent field (SCF), they have much better computational scaling between $O(N^2)$ and $O(N^3)$ depending on the size regime (i.e., depending on the actual value of N). There seems to have been much confusion about the scaling of HF and DFT in the literature where quotes of $O(N^4)$ and $O(N^3)$,

10.1021/jp990629s CCC: $18.00
Published on Web 06/03/1999

respectively, are still occassionally seen. Only for atoms or very small molecules are these scalings observed; scaling in large molecules is a different problem. As discussed below, HF and DFT methods based on Gaussian orbitals have effective scalings in most practical cases that are much better than quartic and cubic, respectively, and as shown in this article, can be brought down to the linear regime for large molecules.

Although the standard approach in "mainstream" correlation methods, such as MP2 and CCSD(T) mentioned above, has been to use a formulation based on canonical molecular orbitals (which are *delocalized*), much progress has recently been achieved in representations that are based on atomic (*localized*) orbitals,[6,7] especially for MP2 where asymptotic linear scaling has already been reported.[8] Note, however, the importance of the word "asymptotic" here; for small molecules, the canonical formulations remain more efficient.

Based solely on scaling considerations, independent particle models are the best candidates for calculations on large molecules. Of the two methods mentioned above, HF contains exact exchange but no correlation. On the other hand, DFT approximates both exchange and correlation.

DFT in the Kohn−Sham version[3] is an electron density theory where all quantum effects are assimilated into an exchange correlation functional whose existence can be proven[9] but whose exact form nobody knows. Over the years, physical and empirical approaches have been used to construct functionals. One could argue whether some of the resulting functionals are ab initio (i.e., first-principles) or not. The fundamental difference between mainstream wave function methods and DFT is that in the former one has a series of approximations that leads to the right answer (albeit with exponentially growing computational cost), whereas in DFT such a series of ever-improving approximations remains unknown. This is perhaps the price that one has to pay for taking the DFT "shortcut" to the many-body problem.

It is also worth mentioning that DFT is not restricted to ground state energies. Much progress has recently been accomplished in linear response time-dependent density functional theory for excited states within the adiabatic approximation.[10−12] Excitation energies and oscillator strengths can be obtained within this model accurately and efficiently, even for systems as large as C_{70}.[13] Unlike wave function methods, however, excited states via DFT also suffer from the lack of systematic improvement to the functional approximation mentioned for the ground state above.

On the other hand, it is now firmly established in the literature that DFT is much more accurate than HF. Furthermore, there is plenty of evidence that DFT is more accurate than MP2, although perhaps not systematically. Rotational barriers, dispersion forces, and activation barriers are examples of properties where MP2 usually gives better predictions than do the best current functionals.

"Mainstream" wave function methods are built upon the HF determinant and include the exchange interaction exactly but have to work very hard to recover the correlation energy. Current state-of-the-art functionals in DFT offer perhaps a more balanced although approximate treatment of exchange and correlation that yields surprisingly accurate results.

In summary, DFT provides an effective and efficient approach for the accurate calculation of molecular properties, structure, and spectra. In other words, DFT has a great performance/price ratio. The appearance of functionals remains unabated,[14−18,19,20] and there is hope that even better ones will be developed in the near future to correct deficiencies of state-of-the-art functionals and avoid their occassional failure.

For calculations of biomolecules and other nanostructures to become routine, linear scaling of computational time with molecular size is required. Even quadratic scaling would not suffice. So, the challenge is to find new methods and algorithms that better reflect the physics of the molecular systems under study, yielding $O(N)$ scaling. One should also take into account the prefactor of linear scaling, which essentially determines the slope of the straight line and consequently the crossover between traditional and $O(N)$ approaches. There is no point in attaining linear scaling for extremely large molecules that are outside the realm of current computational capabilities. In other words, we are interested in achieving linear scaling of computational time as a function of molecular size with a small prefactor such that crossover with standard approaches occurs at modest sizes, typically 10 to 20 atoms. Furthermore, strictly linear scaling is not a must either; for all practical purposes, near-linear scaling is all that we need.

Linear Scaling DFT

We next discuss the steps required to carry out a traditional Gaussian-based DFT calculations and analyze their computational cost. First, one forms the molecular Hamiltonian, then solves for the MO coefficients, from which the density matrix can be straightforwardly evaluated. Both the Coulomb and exchange-correlation potentials depend on the electron density, so the whole procedure needs to be solved self-consistently.

The Hamiltonian is constructed in the atomic orbital basis and contains contributions from the kinetic energy, Coulomb and exchange-correlation potentials. The kinetic energy contribution to the Hamiltonian is a one-electron term which is very simple to handle. The number of nonvanishing kinetic energy integrals between AOs is in principle $O(N^2)$. In practice, however, the Gaussian product theorem eliminates most of them quickly, yielding $O(N)$ surviving terms in large molecules. This is simply because the product of two Gaussian orbitals (on the same electron) is another Gaussian function whose prefactor decays exponentially with the distance between orbital centers. Thus, most Gaussian products between distant centers yield zero integrals in the one-electron case. For the typical values of Gaussian exponents that one uses in quantum chemistry calculations. AOs that are separated by more than 10−20 Bohrs yield negligible contributions. In the electron−nuclei Coulomb term the integrand itself contains $O(N)$ nonvanishing terms yielding an $O(N^2)$ contribution which may be handled together with the two-electron Coulomb repulsion term (see below).

In general, there are three important bottlenecks for achieving linear scaling in Gaussian-based DFT calculations. These are the electronic Coulomb problem, the exchange-correlation quadrature, and the Hamiltonian diagonalization. These three bottlenecks are discussed separately below.

A. The Coulomb Problem and Fast Multipole Methods. The molecular Coulomb potential has three different components: electron−electron, electron−nuclei, and nuclei−nuclei. The last is just the nuclear repulsion energy. As mentioned above, the electron−nuclei term is asymptotically $O(N^2)$. The electron−electron Coulomb problem is also asymptotically $O(N^2)$ but it has a much larger prefactor than the electron−nuclei term (i.e., two-electron Coulomb integrals are computationally much more expensive than the one-electron integrals).

In the large molecule limit, the number of charge distributions (products of Gaussian orbitals) grows linearly with molecular size because of the Gaussian product theorem. All pairwise

interactions between these charge distributions have to be included in the electron−electron Coulomb potential evaluation because of the slowly decaying nature of $1/r$. This yields asymptotically an $O(N^2)$ algorithm. For rapidly decaying potentials, as for example $1/r^6$, one can efficiently screen individual interactions below a certain threshold and calculate the energy or potential to high accuracy in $O(N)$ operations.[8] Unfortunately, this is not true for $1/r$, unless r is huge. In small systems, the number of charge distributions grows as $O(N^2)$ which leads to $O(N^4)$ scaling for the so called two-electron integrals because all products of orbitals on the same electron are non-negligible and we deal with two electrons. However, as shown in the literature,[21,22] the $O(N^2)$ regime is achieved at fairly modest molecular sizes (tens of atoms). So for most practical purposes, the traditional Coulomb problem is $O(N^2)$ rather than $O(N^4)$.

Another important issue regarding the scaling of the Coulomb problem relates to the use of auxiliary basis sets for expanding the electron density.[23,24] This approach, also known as "resolution of the identity,"[25−29] reduces the four-center two-electron integrals to a maximum of three centers, so unless the auxiliary basis is huge, it is much faster than doing all of the Gaussian four-center integrals, especially for small molecules. In terms of Coulomb integrals, however, both methods yield asymptotic $O(N^2)$ scaling. Density expansion and resolution of the identity techniques require a matrix inversion which in principle scales as $O(N^3)$, so for very large systems the traditional approach of calculating four-center integrals may turn out to be more efficient.

In Hartree−Fock theory, all four-center two-electron integrals are required explicitly for evaluating the exchange interaction which is included exactly. Thus, density expansions are not useful either in HF calculations or for hybrid DFT functionals that contain a portion of HF exchange. In the case of "pure" DFT functionals (i.e., those that do not contain HF exchange), DFT exchange can be calculated by quadrature rather than explicit evaluation of all two-electron integrals which is $O(N^4)$ for small molecules. In this case, the $O(N^3)$ scaling of density expansions used to be championed as a functional advantage of DFT over HF. However, this overlooks the fact that two-electron integrals can be *mathematically* screened using exact bounds[21,30] and that only $O(N^2)$ of them survives.[22] In summary, the use of auxiliary basis sets for density expansion yields a substantially faster algorithm for small molecules, but the traditional approach has better scaling for large molecules. In any case, the development of $O(N)$ methods for the Coulomb problem, as discussed below, has made this scaling argument obsolete.

In our work, linear scaling for the electronic Coulomb problem is achieved by means of the fast multipole method (FMM)[31] developed for electronic structure calculations in our research group[32] and others.[33] Alternative formulations have also been proposed[34,35] as well as fast algorithms for the formation of the Coulomb matrix.[36,37] The FMM partitions all interactions between a near-field (NF) and a far-field (FF) portion of the Coulomb problem. In order to achieve high accuracy, NF interactions are done by analytic integration of Gaussians. The FF, which is composed of the vast majority of interactions in a large molecule, is done in a tree algorithm via multipole expansions. The molecule is divided in a hierarchy of boxes and multipole moments and potentials are calculated at these boxes and translated through the tree structure. In order to use FMM algorithms in electronic structure calculations, one needs to assign a finite range to the Gaussian charge distributions. Here again, the use of Gaussian orbitals is crucial because the mathematical bounds used in the Gaussian range definition which assigns a spatial extent to each charge distribution depends on the existence of analytic solutions for the Coulomb integrals between orbitals.[32,33] Our particular formulation of FMM for electronic structure calculations was named "Gaussian very Fast Multipole Method" (GvFMM).[32] In essence, if two charge distributions overlap (within the bounds imposed by their assigned spatial ranges which in turn depend on the desired accuracy), their interaction is done in the NF by analytic integration. The number of these interactions grows linearly with molecular size. In compact three-dimensional systems, the asymptotic linear regime occurs at much larger molecular sizes than, for example, in one-dimensional chains. On the other hand, nonoverlapping distributions (of which there are $O(N^2)$) are treated in the FF by multipoles in a tree algorithm that yields $O(N)$ scaling.[32]

The accuracy of the FMM (i.e., the error in the Coulomb energy or Coulomb contribution to the Hamiltonian obtained by FMM in comparison to the exact analytic solution) can be adjusted to machine double-precision and depends on three parameters: a tolerance factor *toler* (typically between 10^{-6} to 10^{-10} depending on the desired accuracy), *lmax*, the size of the multipole expansions representing the charge distributions in the far-field evaluation (typically between 12 to 20 depending on desired accuracy, too), and *boxlen*, the size of the FMM box at the finest mesh level which is related to the number of tiers in the tree algorithm. These parameters essentially determine the error associated with treating Coulomb integrals between Gaussian charge distributions by multipole expansions. This error is negligible for very well separated charge distributions (i.e., those whose edges are separated by a number of boxes) and can be controlled in FMM for all other cases.

Linear scaling of computational time with molecular size has been demonstrated in the literature for FMM in a series of benchmark cases.[32] Furthermore, FMM has also been implemented in connection with first and second energy derivatives to eliminate the evaluation of Coulomb integral derivatives in the far-field.[38,39] Crossover of FMM with state-of-the-art Gaussian integration occurs at very modest molecular sizes, typically between 10 to 20 atoms, depending on the basis set, desired accuracy, and particular characteristics of the system under consideration, especially its dimensionality. In summary, FMM is a powerful tool for circumvecting the explicit evaluation of two-electron integrals. The accuracy can be easily adjusted, it scales linearly with molecular size, and has a small prefactor which yields early crossover with traditional integral evaluation. Thus, the "integral bottleneck" that characterized quantum chemistry calculations for many years has clearly been defeated.

B. Fast Linear Scaling Quadratures for Exchange and Correlation. Because of its complicated dependence on the electron density and its derivatives, the matrix elements of the DFT exchange correlation (XC) potential and energy cannot be obtained by analytic integration even when using Gaussian orbitals. Thus, one resorts to numerical quadrature for calculating them. Furthermore, in order to achieve high accuracy, especially required for precise analytic differentiation of the energy with respect to nuclear displacements (analytic forces and frequencies), sophisticated multicenter quadrature schemes which include large numbers of grid points are used in Gaussian DFT. The integrals are partitioned over atomic centers using a weight scheme,[40,41] and a further decomposition into radial and angular components of each atomic contribution is introduced.[42−44]

In essence, every atomic region contributes to XC integrals

at any given grid point, and the value of the weight depends on all pairs of atoms. This yields $O(N^3)$ scaling for the weight evaluation. Other steps in the quadrature also yield $O(N^3)$ scaling simply because the number of atomic-based grid points grows linearly with molecular size and their contributions need to be evaluated over every pair of basis functions. Although the feasibility of achieving linear scaling for the XC quadrature has been recognized,[45,46] many of the existing numerical integration routines take advantage only of the fast-decaying nature of Gaussian basis functions yielding scalings of $O(N^2)$.

Exploiting the localized nature of the XC potential (which decays much more rapidly than $1/r$), one can confine its contributions at a given grid point to a relatively small region around it with negligible loss of accuracy. We have developed an atomic weight scheme (SS weights)[47] whose numerical evaluation is much less costly than other popular schemes widely used.[40] By design, the occurrence of the weight value 1 or 0 is much more frequent in our scheme than in other methods. This is achieved by constructing a smooth polynomial connection between 0 and 1 over a *narrow* spatial range. The fundamental advantage of our weight scheme is that by determining in advance whether a weight value will have the numerical value of 1 or 0, which occurs every time a grid point is very close or far from a particular atom, respectively, we avoid the actual numerical weight computation which involves several floating point operations.[47] In other schemes, there are a large number of weights whose values turn out to be for all practical purposes equivalent to 1 or 0, but whose precise numerical value is nevertheless evaluated.

We have also introduced the concept of "microbatches" of grid points which share a common set or relevant (local) basis functions. The combination of SS weights with microbatching yields an accurate quadrature scheme with $O(N)$ scaling behavior at small molecular sizes.[47] In this paper, we will occassionally refer to this quadrature scheme as LinXC. The performance of LinXC has been documented in the literature both for energy[47] and energy second derivative calculations.[48] In this paper, additional benchmarks on even larger systems are presented.

C. Avoiding Diagonalization. The Hamiltonian diagonalization is intrinsically $O(N^3)$, and there are no easy or simple ways to reduce this scaling. It should be pointed out, however, that diagonalization has such a small prefactor that its computational cost is very low up to around 2000−3000 basis functions. Of course the use of fast quadratures and of FMM for the Coulomb problem pushes the point where diagonalization becomes time-dominant to smaller molecular sizes. For molecules containing around 2000 to 3000 basis functions, we have documented in the literature[32,47,48] speed-up factors between one and two orders of magnitude (depending on molecular size) for the linear scaling Coulomb and quadrature steps compared to previous algorithms. These advances point toward diagonalization as becoming the major obstacle for density functional calculations on very large molecules.

An additional complication in the diagonalization step is its $O(N^2)$ memory requirements. Standard diagonalization routines normally require a full copy of the matrix to be diagonalized in core memory. Depending on machine architecture and resources, and the particular molecular system under consideration, one can easily get memory-bound before getting CPU-bound.

The best solution to the diagonalization bottleneck is to avoid it and replace it with some alternative. Several methods have been proposed in the literature for this purpose. The reader is refered to an excellent recent review by Goedecker[49] for details on alternatives to diagonalization. We have developed and implemented several of these methods and benchmarked them with semiempirical Hamiltonians.[50−53] In our experience, conjugate-gradient density matrix search (CGDMS) is one of the best alternatives to diagonalization[53] in semiempirical calculations. The corresponding work for DFT is still under progress, and it is not completely clear yet which particular alternative (or combination of methods) may be most efficient in Gaussian-based DFT calculations. Because the DFT Hamiltonian and density matrices are normally much less sparse than the semiempirical ones, conclusions obtained for semiempirical methods may not be directly extrapolated to the DFT case.

The results presented in this paper are obtained using our particular formulation of the CGDMS method.[50,51,54] This is an approach based on ideas by Li-Nunes-Vanderbilt (LNV)[55] and generalized by us to non-orthogonal bases.[54] CGDMS is essentially a conjugate gradient density matrix search algorithm that minimizes an energy expression which depends on the density matrix subject to N_e representability conditions. Three or four conjugate gradient iterations are usually enough to obtain the density matrix to good accuracy at any given cycle of the SCF procedure. Only the constraint on the number of electrons is explicitly imposed through a Lagrange multiplier (chemical potential). The idempotency constraint (i.e., the fact that occupation numbers are strictly 0 or 1 in independent particle models) is imposed indirectly as an energy penalty through McWeeny's purification transformation.[56] Of course, conventional quantum chemistry methods that go beyond the independent particle approximation (i.e., self-consistent field) do not solely depend on the one-particle density matrix. In other words, CGDMS is a practical alternative to the diagonalization step in SCF but does not provide an alternative to, for example, the Hamiltonian diagonalization involved in configuration interaction.

The philosophy behind CGDMS is to exploit the fact that the one-particle density matrix is the fundamental variable in independent particle models like HF or DFT. Eigenvectors of the effective Hamiltonian which are obtained through the diagonalization step are in practice only needed to construct the density matrix. However, this is not the only way of obtaining the density matrix, and one can instead adopt direct search methods like CGDMS. If desired, orbitals can always be obtained from the density matrix using techniques like Lanczos that yield $O(N)$ scaling as long as one is interested in a fixed (independent of molecular size) number of orbitals. Unfortunately, the number of orbitals required to obtain the density matrix itself is proportional to molecular size (i.e., the number of electrons), so Lanczos and similar techniques have $O(N^3)$ scaling as an alternative to diagonalization.[54]

The key to linear scaling performance in all alternatives to diagonalization resides in exploiting sparse matrix multiplication techniques. Multiplication of dense matrices is an $O(N^3)$ procedure. In CGDMS, one essentially replaces diagonalization by an algorithm that involves matrix multiplication. Matrix multiplication of sparse matrices is an $O(N)$ procedure, so sparsity of the effective Hamiltonian and density matrices is a fundamental ingredient for achieving linear scaling. The intrinsic nature of the molecular system under consideration and molecular size are important factors in determining sparsity. In this regard, the size of the HOMO−LUMO gap is connected to "localization" and, consequently, sparsity in the system. It is well known that systems showing metallic character (i.e., small HOMO−LUMO gap) yield denser Hamiltonians and density matrices than insulators (large HOMO−LUMO gaps).[54,57] In practical implementations of CGDMS, one sets a neglect

threshold below which matrix elements are not stored, so sparsity is also a function of the desired accuracy.

The original LNV method[55] was formulated for orthogonal bases like those used in tight-binding and semiempirical calculations.[50,51] For non-orthogonal bases we have proposed[54] a transformation of the non-orthogonal problem to an orthogonal basis using the Cholesky decomposition of the overlap matrix. Because the overlap matrix between Gaussian AOs becomes sparser with larger molecular size, the Cholesky decomposition can be done in $O(N)$ operations. The inverse Cholesky decomposition which is also required in our method has more demanding scaling requirements. It has such a small prefactor, however, that its scaling is not a problem unless one deals with huge size systems. We have also analyzed alternatives based on symmetric orthonormalization which involves the square root and inverse square root of the overlap matrix and concluded that Cholesky is a much better alternative.[54]

We next discuss the concept of "progressive convergence" which is employed in the benchmarks presented in this paper and is introduced in this work for the first time.

Progressive Convergence

DFT calculations need to be carried out in a self-consistent fashion because the density matrix, which is the fundamental quantity determining the energy, also determines the Coulomb and XC potentials. The changes in the density matrix during the SCF procedure are more important during the early SCF iterations when relaxing the guess density matrix (obtained by an AM1 semiempirical calculation in our case) which is usually much more sparse than the DFT converged matrix. What this means in practice is that during the initial stages of the SCF procedure there is no point in accurately calculating the density matrix by CGDMS. The density matrix will change significantly during the SCF procedure because of the relaxation of the potential. Furthermore, even the Coulomb and XC potentials themselves do not need to be accurately determined during the initial stages because of their dependence on the density matrix.

In practical calculations, one has a target accuracy for the total energy and density matrix. This target accuracy pretty much determines the *final* values of neglect thresholds and parameters to be used in the calculation. The specific values of these thresholds and parameters determine the accuracy of CGDMS, FMM, and LinXC (they will be discussed in more detail below).

The fundamental idea in progressive convergence (PC) is that during the initial stages of the SCF procedure one can use looser values for thresholds and parameters without affecting the final answer as long as these thresholds and parameters get tightened up as convergence progresses and eventually reach the final values that yield the desired accuracy. Because the $O(N)$ methods used in large-scale calculations scale steeply with neglect thresholds and accuracy parameters, the savings resulting from using looser values at the early stages of the SCF procedure are very important. The computational savings of using PC for CGDMS are actually much larger than those resulting from FMM or LinXC and are quantified below. One can argue that convergence may be hurt (i.e., more SCF cycles needed to converge) when using PC. However, our practical experience so far is that the overhead in terms of number of SCF cycles is minimal, if any at all.

Our PC scheme works in practice by dynamically setting thresholds and parameters on the basis of the root mean square deviation between the density matrices of two consecutive SCF cycles. This quantity will be denoted RMSDP in the following. The CGDMS neglect threshold is simply determined by a multiplicative factor, typically around 0.1 to 0.001, applied to RMSDP. Very agressive CGDMS neglect thresholds usually lead to loss of idempotency and nonconvergence, so a minimum starting CGDMS threshold of 10^{-3}, as used in this work, is normally advised.

As mentioned above, the FMM procedure involves intrinsic parameters that determine accuracy (*lmax, boxlen,* and *toler*) and other thresholds normally used in Gaussian codes to neglect charge distributions which are far apart. The latter also affects the accuracy of the NF integrals. The accuracy in the quadrature is also affected by other thresholds, and more importantly, by the size of the numerical grid used in the calculation and its pruning. Under normal circumstances the values of all these parameters, thresholds, and grids are determined by the desired target accuracy and the specific application; for example, whether the user is interested in an energy calculation or a force calculation. In PC, the target accuracy determines the *final* values of all thresholds, parameters, and grids, but their specific values are allowed to change from looser to tighter, and are dynamically adjusted as convergence proceeds.

In this paper, we demonstrate the current capabilities of our recently developed programs in a series of benchmark DFT calculations presented in the next section. Even though these calculations represent the state-of-the-art at the current time, there is no doubt in our minds that continuing developments in linear scaling methods will render these benchmarks obsolete in the not too distant future.

Benchmark Calculations

For the purpose of benchmarking the linear scaling capabilities of our methods and algorithms, we have designed a series of water clusters in two and three dimensions. The molecules are built from a basic cubic box of roughly 3.5 Å side that contains eight water molecules in its interior. This cubic box is then replicated in two and three dimensions using an $m \times m$ and $n \times n \times n$ scheme. The largest m and n used in this work are 12 and 5, respectively. This yields a maximum of 1152 water molecules in a two-dimensional (2D) 12×12 cluster and 1000 water molecules in a more compact three-dimensional (3D) $5 \times 5 \times 5$ structure.

The specific DFT functionals used in this work are the local density approximation (LSDA),[4,5] the Becke-88 exchange functional combined with Lee−Yang−Parr correlation (BLYP),[58,59] and the Perdew−Burke−Erznehof (PBE)[14] generalized gradient approximation to exchange and correlation.

Hybrid functionals like B3LYP[60] or PBE1PBE[19,61] are not benchmarked in this paper because they involve a portion of HF exchange. Although much progress has recently been achieved in linearizing the computational scaling of HF exchange,[57,62,63] it is true that its presence makes DFT calculations more involved, especially in molecules with small HOMO−LUMO gaps. Besides, the most recent developments in XC functionals[17,16,64,20] seem to indicate that alternative formulations based on the kinetic energy density τ are every bit as accurate as methods like B3LYP, if not more accurate for certain problems. For these reasons, we have decided not to include a discussion of the scaling of hybrid functionals in this work and limit its scope to "pure" DFT functionals.

All calculations were carried out using a development version of the *Gaussian* suite of programs.[65] Two types of basis sets were used: 3-21G and 6-31G**. The former is of double-zeta quality and the latter contains polarization functions both on H (p orbitals) and first-row atoms (d orbitals). The largest

TABLE 1: Energies (Hartrees) and CPU Times (min) per SCF Cycle Obtained by Conjugate Gradient Density Matrix Search (CGDMS) and Diagonalization in a Series of Two-Dimensional Water Cluster Calculations at the LSDA/3-21G Level of Theory

					CPU times (min) per cycle	
cluster	molecule	Nbf[a]	energy	ΔE[b]	CGDMS	diagonalization[c]
3×3	$(H_2O)_{72}$	936	−5445.35917	5×10^{-5}	2	2
4×4	$(H_2O)_{128}$	1644	−9680.71549	7×10^{-5}	3	11
5×5	$(H_2O)_{200}$	2600	−15126.19784	5×10^{-5}	6	53
6×6	$(H_2O)_{288}$	3744	−21781.80277	2×10^{-5}	8	162
7×7	$(H_2O)_{392}$	5096	−29647.53523		10	409
8×8	$(H_2O)_{512}$	6656	−38723.39133		17	910
9×9	$(H_2O)_{648}$	8424	−49009.37589		22	1845
10×10	$(H_2O)_{800}$	10400	−60505.48460		29	3472
12×12	$(H_2O)_{1152}$	14976	−87128.08452		40	10368

[a] Number of contracted Gaussian basis functions. [b] CGDMS energy error (compared to diagonalization) obtained with *thresh* $= 10^{-7}$ and SCF convergence criteria set to RMSDP $= 10^{-7}$ and MAXDP $= 10^{-5}$. [c] Extrapolated values for clusters larger than 6×6.

calculations carried out in this work include up to around 15 000 basis functions.

All CPU times reported in this paper were obtained on a single processor of an SGI Origin-2000 machine with the 195 MHz R10000 MIPS processor with 4 Mb Level 2 cache. The largest calculations in this work required less than 10 gigabytes of disk space and a maximum of 180 megawords of memory. It is worth mentioning that in our programs, memory and disk space requirements scale linearly with molecular size. Most of the calculations reported herein can be carried out with just a few gigabytes of disk space and less than 60 megawords of memory. These are modest hardware requirements, easily accessible nowadays in individual workstations or even personal computers.

Table 1 presents converged energies at the LSDA/3-21G level of theory for the two-dimensional water clusters. The FMM parameters used in these calculations are *lmax* $= 12$, *toler* $= 6$, and *boxlen* $= 2$ Bohrs. The CGDMS neglect threshold is *thresh* $= 10^{-7}$. The final quadrature grid is a pruned 50×194. All calculations reported in this paper were carried out using *Gaussian* standard neglect thresholds for NF integrals and screening of negligible charge distributions which are default for energy point calculations.

As discussed above, when using progressive convergence, the values of parameters and thresholds are adjusted dynamically and the values quoted above are *final* values, except for *boxlen* which is kept fixed throughout the entire calculations. The SCF convergence criteria are set to RMSDP $= 10^{-7}$ and MAXDP $= 10^{-5}$ (the maximum absolute value change for individual density matrix elements between two successive SCF cycles).

In Table 1, we quote the differences in energy between the $O(N)$ calculations and the diagonalization results. With the parameters and thresholds used for Table 1, we roughly obtain five decimals of accuracy in the total energy as compared to diagonalization. This energy examination is made for clusters up to 6×6; the steep computational cost and memory requirements of diagonalization makes further comparison impractical. CPU times for CGDMS and diagonalization are also presented in Table 1. The CGDMS figures are *average* values over all SCF cycles. As explained above, the computational cost of an SCF cycle varies significantly when using PC; the early SCF cycles are inexpensive and CPU time goes up as one reaches SCF convergence. The computational cost increase with SCF cycles is much more pronounced for CGDMS than FMM or LinXC. For the results in Table 1, the ratios between the last and first SCF cycles for the largest benchmark case (12×12) are 2, 3, and 10, for the Coulomb, quadrature, and CGDMS steps, respectively.

The speed-ups obtained for the Coulomb step using FMM and for the quadrature using LinXC, compared to state-of-the-art traditional algorithms, are well documented in the literature and do not need to be repeated here.[32,38,39,47,48] On the other hand, our previous CGDMS timings for DFT calculations reported in the literature[54] correspond to a preliminary version of our code which was far from being optimized and did not include progressive convergence. Thus, to demonstrate the capabilities of our current algorithms and programs, we also present in Table 1 a comparison between CGDMS and diagonalization CPU times for the set of 2D water benchmark clusters with LSDA/3-21G.

The diagonalization CPU times per SCF cycle quoted in Table 1 are from actual calculations up to 3744 basis functions (6×6) and are then extrapolated using $O(N^3)$ scaling. It is nevertheless important to note that for the largest system in Table 1, the 12×12 water cluster, PC CGDMS is about 250 times faster than the diagonalization estimate; the average CPU time for CGDMS is less than 1 hour, whereas diagonalization would have taken roughly 1 week of CPU time per SCF cycle. Furthermore, diagonalization of the roughly $15\,000 \times 15\,000$ Hamiltonian matrix for the 12×12 case in Table 1 roughly requires 450 megawords of memory (assuming that two full matrices are needed), whereas using sparse matrix techniques storage is much less demanding (just a few megawords for each sparse matrix) given that only 1% of Hamiltonian matrix elements are actually larger than the 10^{-7} neglect threshold in absolute value. In addition to the speed-ups furnished by FMM and LinXC, the substantial improvement provided by PC CGDMS over diagonalization makes an impractical calculation become fairly routine.

CPU times for all steps in the DFT calculation obtained with 3-21G and 6-31G** basis sets at the LSDA level of theory in 2D and 3D water clusters are presented in Figures 1−4. All of these benchmarks were obtained with thresholds and parameters identical to those quoted above for the results in Table 1. Results in Figures 1−4 were carried out using PC as well; thus, the quoted CPU times are averages over the entire SCF procedure. The calculations in Figure 1 were obtained with SCF convergence criteria identical to those of Table 1, whereas in Figures 2−4, the SCF convergence criteria were set to RMSDP $= 10^{-6}$ and MAXDP $= 10^{-4}$, with the expectation of four decimals accuracy in the total energy (i.e., errors less than 0.1 kcal/mol) which is sufficient for most practical total energy calculations.

Several comments regarding the results in Figures 1−4 are pertinent. A quick glance at these figures indicates that in most cases we do not obtain perfectly straight lines when plotting CPU time per SCF cycle as a function of molecular size. There are several reasons for this. First, we do not claim perfect linear scaling for all of the steps in the calculation. Most importantly, the plotted times are averaged over the number of SCF cycles

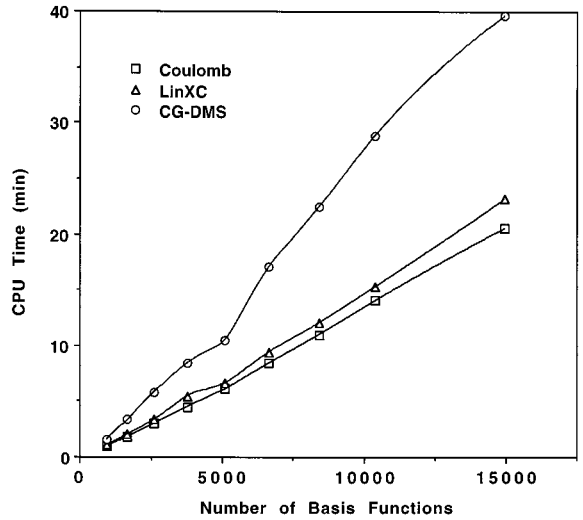

Figure 1. Average CPU times per SCF cycle as a function of molecular size for two-dimensional water clusters at the LSDA/3-21G DFT level of theory.

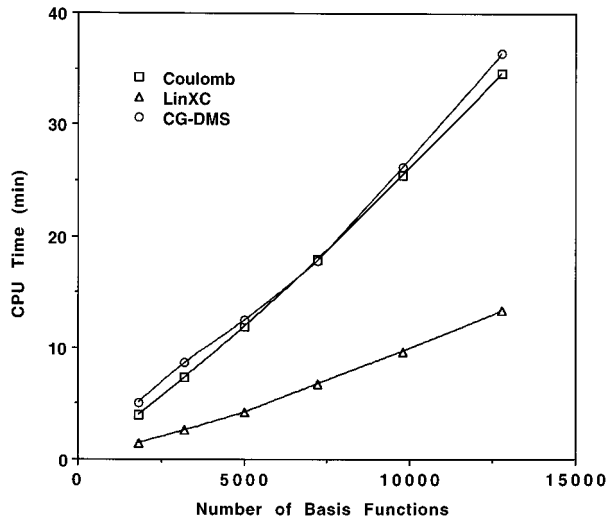

Figure 2. Average CPU times per SCF cycle as a function of molecular size for three-dimensional water clusters at the LSDA/3-21G DFT level of theory.

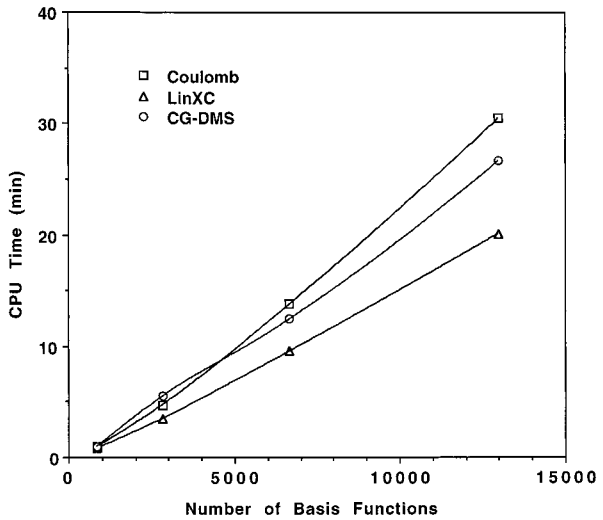

Figure 3. Average CPU times per SCF cycle as a function of molecular size for two-dimensional water clusters at the LSDA/6-31G** DFT level of theory.

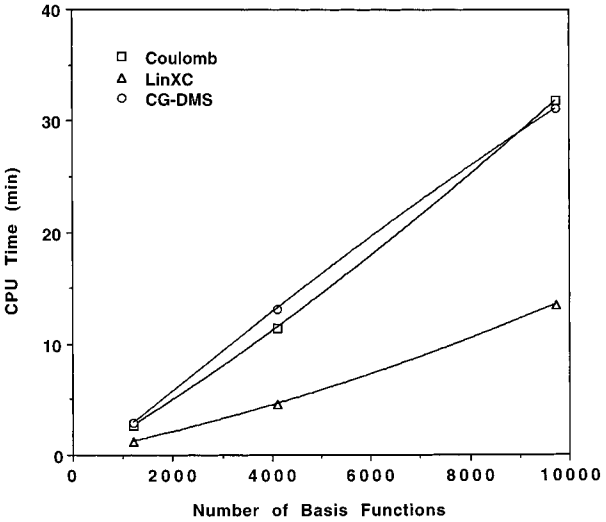

Figure 4. Average CPU times per SCF cycle as a function of molecular size for three-dimensional water clusters at the LSDA/6-31G** DFT level of theory.

which may vary somewhat arbitrarily from molecule to molecule, with a slight increase of a few extra cycles for the largest cases. Because the last SCF cycles are the most costly, an additional SCF cycle may substantially increase the average times, especially for CGDMS. Nevertheless, the reported curves in Figures 1−4 are indicative of linear scaling and are substantial improvements over the computational scalings of the traditional methods and algorithms.

The Coulomb times quoted in Figures 1−4 involve both FMM and NF integral evaluation plus the contraction step with the density matrix. Roughly speaking, these two steps contribute each about 50% of the CPU time. In other words, with the choice of FMM parameters described above and for the particular systems under consideration, our Coulomb CPU times are well balanced between FF and NF. Further improvements to these CPU times could be expected from advances in the NF contraction[36] which we have not implemented yet.

We next discuss the relative weights of the Coulomb, LinXC quadrature, and CGDMS steps among the carried out benchmarks. For the 2D case with a 3-21G basis (Table 1 and Figure 1), the CGDMS step is the most costly. This is almost always the case although in the other figures the computational cost of the Coulomb step approaches that of CGDMS and actually surpasses it in a few instances. The quadrature step is in all cases the least computationally expensive.

Basis set affects computational cost significantly. For a given molecule, if we change the basis from 3-21G to 6-31G**, the number of basis functions increases substantially, and so does the CPU time. However, for a given number of basis functions, a 6-31G** calculation on a given system is more expensive than a 3-21G calculation with identical number of basis functions (and consequently on a larger similar molecule) because polarized basis sets normally yield much less sparse matrices than unpolarized basis sets. Note that a straight comparison of the results between Figures 1 and 2 is misleading because they were carried out using different SCF convergence criteria. On the other hand, the point above is well illustrated by the results in Figures 3 and 4 for the 3D water clusters: for a given number of basis functions, the 6-31G** results are more computationally demanding than the 3-21G results.

We have also carried out BLYP and PBE calculations on all clusters studied in this work. The results obtained are very much the same for all molecules and basis sets studied: the Coulomb and CGDMS CPU times are fairly insensitive to the particular

functional used, whereas the LinXC quadrature time increases by roughly a factor of 2 in all cases (2D and 3D with or without polarization basis functions). When this factor 2× for the LinXC quadrature step is incorporated into Figures 1−4, we obtain plots for 3-21G (2D and 3D clusters) where LinXC becomes the most expensive computational step. The effect is less pronounced for polarized basis sets (Figures 2 and 4) where an additional factor 2× on LinXC still yields plots where Coulomb and CGDMS are more computationally expensive.

Concluding Remarks

The objectives of this feature article are to discuss some of the recent advances in linear scaling DFT methods, present benchmark calculations on model systems, and attempt to give the reader a feeling for where computational quantum chemistry is currently heading.

It is fair to ask, however, whether the benchmark calculations presented in this paper are representative of other molecular systems. This a difficult question because quantum chemists are normally interested in a wide variety of molecules which may or may not be represented by the benchmark water clusters. The most significant elements to be taken into account for these potential comparisons are dimensionality, HOMO−LUMO gap, desired accuracy, and basis set. The more compact a molecular system is, the less sparse all matrices are, and the more demanding the $O(N)$ DFT calculation will turn out to be. With the current set of methods and algorithms, linear scaling for self-similar systems is not a problem. The prefactor (i.e., the slope of the straight line relating CPU time to molecular size) is strongly dependent on the particular characteristics of the system under consideration, and most importantly, sparsity, which in turn depends on basis set, HOMO-LUMO gap, and selected thresholds for a target accuracy.

In order to give the reader a useful comparison, we have carried out an LSDA/3-21G energy calculation on an RNA fragment that contains 1026 atoms and 6767 basis functions. The system is depicted in Figure 5 and the coordinates were obtained from the PDB database. Using PC, thresholds and parameters identical to those of Table 1 and Figure 1, and SCF convergence criteria set to RMSDP $= 10^{-7}$ and MAXDP $= 10^{-5}$, the average CPU times per SCF step are 54, 17, and 172 CPU minutes for the Coulomb, LinXC, and CGDMS steps, respectively. The energy calculation converges in 27 SCF cycles which is more than usual but may simply reflect some intrinsic SCF convergence difficulty of the chosen RNA fragment. If the SCF convergence criteria are set to the looser values RMSDP $= 10^{-6}$ and MAXDP $= 10^{-4}$, the average CPU times per SCF step are reduced to 48, 14, and 109 CPU minutes for the Coulomb, LinXC, and CGDMS steps, respectively. The calculation converges in 23 SCF cycles thus yielding shorter average CPU times, as discussed above.

Comparing these results with those in Figures 1 and 2 for similar number of basis functions, one concludes that all three DFT steps for the RNA piece are computationally more expensive than those for the water clusters, especially CGDMS which is about 5× more costly than the 3D cluster case. These results simply indicate that typical biomolecules may have density matrices and Hamiltonians which are denser than 3D water clusters, but they are still amenable to efficient treatment by the methods and algorithms discussed in this work. All of the algorithms discussed in this paper are amenable to efficient implementation in parallel machines, which will significantly reduce elapsed CPU times. The combination of ever decreasing computer prices with faster processor performance can only help make DFT calculations on very large molecules more efficient and widely available.

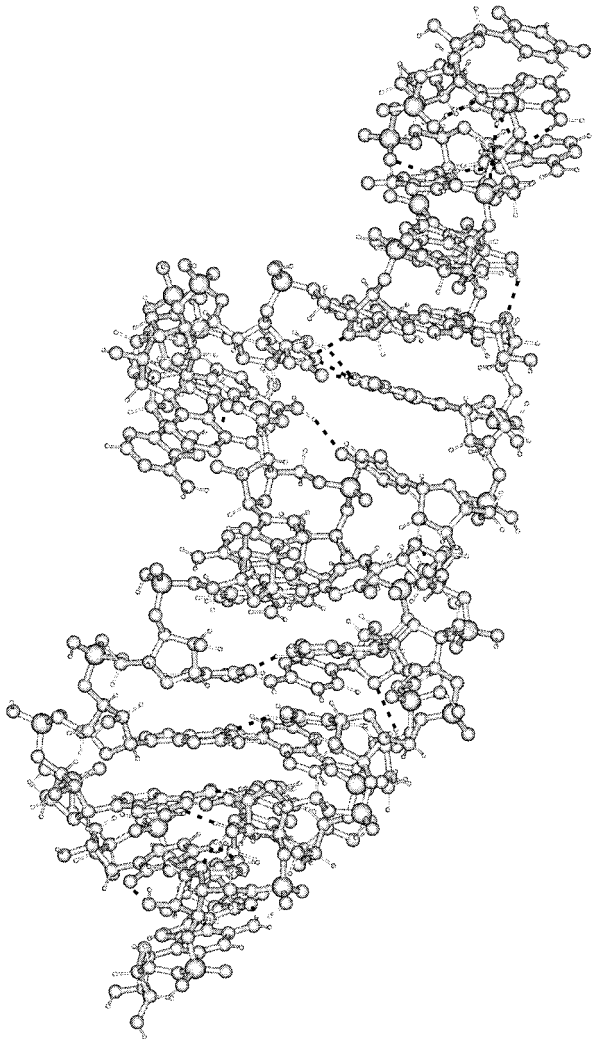

Figure 5. 1026 atom RNA fragment calculated at the LSDA/3-21G DFT level of theory.

We should also point out that analytic energy gradients with respect to nuclear displacements, which are used in geometry optimizations, can be straightforwardly computed for independent particle methods in $O(N)$ operations. This result follows from the analytic energy gradient expression which is a simple function of the density matrix. We have already accomplished this goal in practical calculations.[66] Linear scaling of computational time with molecular size for second derivatives with respect to nuclear positions (i.e., frequencies) or linear response methods like those used in computing excitation energies is also feasible.[66,67]

Finally, we would like to speculate on whether $O(N)$ DFT methods are the end of the road in quantum chemistry calculations. Even if the prefactor of linear scaling DFT methods can be significantly reduced from its current stand and computer hardware becomes faster and cheaper, the answer to the question above very much depends on our ability to develop a next generation of functionals whose predictions, when compared to experiment, will turn out to be much more accurate than those currently feasible. We would like the next generation of functionals to correct most of the deficiencies that current functionals suffer (e.g., activation barriers, dispersion forces, and charge-transfer complexes) and, for example, to yield atomization energies an order of magnitude more accurate than

it is now possible. It remains to be seen whether this goal is achievable. Many research groups (including ourselves) are actively seeking new functionals on some promising grounds.

It will be hard for "mainstream" wave function methods like coupled-cluster theory to catch up with the computational speed of $O(N)$ Gaussian DFT for large systems. On the other hand, it is true that large basis set coupled cluster calculations almost always "get the right answer for the right reason".[2] Current functionals in DFT yield significantly more accurate results than Hartree−Fock theory, but they are still short of providing final results. This is evident in small molecular systems where the accuracy of wave function methods can provide definitive answers and is of great aid in sorting out experimental questions. One may thus forecast that if DFT fails to deliver a next generation of significantly more accurate functionals, it would then be reasonable to assume that much work will be devoted to developing fast (i.e., small prefactor) $O(N)$ wave function methods, an area where some research groups are already active.[6−8]

Acknowledgment. This paper benefited from the contributions of my students, postdocs, and collaborators: Andrew D. Daniels, Konstantin Kudin, John M. Millam, R. Eric Stratmann, and Mike Frisch. Different portions of our linear scaling DFT work have been supported by the National Science Foundation (CHE-9618323), the Air Force Office of Scientific Research (Grant No. F49620-98-1-0280), NASA grant number NAG2-1112, the Welch Foundation, Gaussian, Inc., and the IBM corporation. The referee is thanked for constructive comments.

References and Notes

(1) Boys, S. F. *Proc. R. Soc. Lond. A* **1950**, *200*, 542.
(2) Lee, T. J.; Scuseria, G. E. In *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy*; Langhoff, S. R., Ed.; Kluwer Academic Publishers: Dordrecht, 1995; p 47.
(3) Kohn, W.; Sham, L. J. *Phys. Rev. A* **1965**, *140*, 1133.
(4) Parr, R. G.; Yang, W. *Density-Functional Theory of Atoms and Molecules*; Oxford University Press: Oxford, 1989.
(5) Dreizler, R. M.; Gross, E. K. U. *Density Functional Theory*; Springer Verlag: Berlin, 1990.
(6) Saebo, S.; Pulay, P. *Annu. Rev. Phys. Chem.* **1993**, *44*, 213.
(7) Hampel, C.; Werner, H. J. *J. Chem. Phys.* **1996**, *104*, 6286.
(8) Ayala, P. Y.; Scuseria, G. E. *J. Chem. Phys.* **1999**, *110*, 3660.
(9) Hohenberg, P.; Kohn, W. *Phys. Rev. B* **1964**, *136*, 864.
(10) Petersilka, M.; Gossmann, U. J.; Gross, E. K. U. *Phys. Rev. Lett.* **1996**, *76*, 1212.
(11) Jamorski, C.; Casida, M. E.; Salahub, D. *J. Chem. Phys.* **1996**, *104*, 5134.
(12) Bauernschmitt, R.; Ahlrichs, R. *Chem. Phys. Lett.* **1996**, *256*, 454.
(13) Stratmann, R. E.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **1998**, *109*, 8218.
(14) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865; Perdew, J. P.; Burke, K.; Ernzerhof, M. **1997**, *78*, 1396 (E).
(15) Becke, A. D. *J. Chem. Phys.* **1998**, *109*, 2092.
(16) Filatov, M.; Thiel, W. *Phys. Rev. A* **1998**, *57*, 189.
(17) Van Voorhis, T.; Scuseria, G. E. *J. Chem. Phys.* **1998**, *109*, 400.
(18) Hamprecht, F. A.; Cohen, A. J.; Tozer, D. J.; Handy, N. C. *J. Chem. Phys.* **1998**, *109*, 6264.
(19) Ernzerhof, M.; Scuseria, G. E. *J. Chem. Phys.* **1999**, *110*, 5029.
(20) Perdew, J. P.; Kurth, S.; Zupan, A.; Blaha, P. *Phys. Rev. Lett.* **1999**, *82*, 2544.
(21) Häser, M.; Ahlrichs, R. *J. Comput. Chem.* **1989**, *10*, 104.
(22) Strout, D. L.; Scuseria, G. E. *J. Chem. Phys.* **1995**, *102*, 8448.
(23) Sambe, H.; Felton, R. H. *J. Chem. Phys.* **1975**, *62*, 1122.
(24) Dunlap, B. I.; Rösch, N. *J. Chim. Phys. Phys.-Chim. Biol.* **1989**, *86*, 671.
(25) Beebe, N. H. F.; Linderberg, J. *Int. J. Quantum Chem.* **1977**, *12*, 683.
(26) van Alsenoy, C. *J. Comput. Chem.* **1988**, *9*, 620.
(27) Feyereisen, M.; Fitzgerald, G.; Komornicki, A. *Chem. Phys. Lett.* **1993**, *208*, 359.
(28) Vahtras, O.; Almlöf, J.; Fereisen, M. W. *Chem. Phys. Lett.* **1993**, *213*, 514.
(29) Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. *Chem. Phys. Lett.* **1995**, *240*, 283.
(30) Häser, M. *Theor. Chim. Acta* **1993**, *87*, 147.
(31) Greengard, L.; Rokhlin, V. *J. Comput. Phys.* **1987**, *73*, 325.
(32) Strain, M. C.; Scuseria, G. E.; Frisch, M. J. *Science* **1996**, *271*, 51.
(33) White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1994**, *101*, 6593. White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. *Chem. Phys. Lett.* **1994**, *230*, 8.
(34) Kutteh, R.; Apra, E.; Nichols, J. *Chem. Phys. Lett.* **1995**, *238*, 173.
(35) Perez-Jorda, J. M.; Yang, W. *Chem. Phys. Lett.* **1995**, *247*, 484; *J. Chem. Phys.*, **1997**, *107*, 1218.
(36) White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1996**, *104*, 2620.
(37) Challacombe, M.; Schwegler, E.; Almlöf, J. *J. Chem. Phys.* **1996**, *104*, 4685.
(38) Burant, J. C.; Strain, M. C.; Scuseria, G. E.; Frisch, M. J. *Chem. Phys. Lett.* **1996**, *248*, 43.
(39) Burant, J. C.; Strain, M. C.; Scuseria, G. E.; Frisch, M. J. *Chem. Phys. Lett.* **1996**, *258*, 45.
(40) Becke, A. D. *J. Chem. Phys.* **1988**, *88*, 2547.
(41) Delley, B. *J. Chem. Phys.* **1990**, *92*, 508.
(42) Gill, P. M. W.; Johnson, B. G.; Pople, J. A. *Chem. Phys. Lett.* **1993**, *209*, 506.
(43) Murray, C. W.; Handy, N. C.; Laming, G. J. *Mol. Phys.* **1993**, *78*, 997.
(44) Treutler, O.; Ahlrichs, R. *J. Chem. Phys.* **1995**, *102*, 346.
(45) van Wullen, C. *Chem. Phys. Lett.* **1994**, *219*, 8.
(46) Perez-Jorda, J. M.; Yang, W. *Chem. Phys. Lett.* **1995**, *241*, 469.
(47) Stratmann, R. E.; Scuseria, G. E.; Frisch, M. J. *Chem. Phys. Lett.* **1996**, *257*, 213.
(48) Stratmann, R. E.; Burant, J. C.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **1997**, *106*, 10175.
(49) Goedecker, S. *Rev. Mod. Phys.* 1999, in press.
(50) Xu, C. H.; Scuseria, G. E. *Chem. Phys. Lett.* **1996**, *262*, 219.
(51) Daniels, A. D.; Millam, J. M.; Scuseria, G. E. *J. Chem. Phys.* **1997**, *107*, 425.
(52) Bates, K. R.; Daniels, A. D.; Scuseria, G. E. *J. Chem. Phys.* **1998**, *109*, 3008.
(53) Daniels, A. D.; Scuseria, G. E. *J. Chem. Phys.* **1998**, *110*, 1321.
(54) Millam, J. M.; Scuseria, G. E. *J. Chem. Phys.* **1997**, *106*, 5569.
(55) Li, X. P.; Nunes, R. W.; Vanderbilt, D. *Phys. Rev. B* **1993**, *47*, 10891.
(56) McWeeny, R. *Rev. Mod. Phys.* **1960**, *32*, 335.
(57) Burant, J. C.; Scuseria, G. E.; Frisch, M. J. *J. Chem. Phys.* **1996**, *105*, 8969.
(58) Becke, A. D. *Phys. Rev. A* **1988**, *38*, 3098.
(59) Lee, C.; Yang, W.; Parr, R. G. *Phys. Rev. B* **1988**, *37*, 785.
(60) Becke, A. D. *J. Chem. Phys.* **1993**, *98*, 5648.
(61) Perdew, J. P.; Ernzerhof, M.; Burke, K. *J. Chem. Phys.* **1996**, *105*, 9982.
(62) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1998** *109*, 1663.
(63) Schwegler, E.; Challacombe, M. *J. Chem. Phys.* **1996**, *105*, 2726.
(64) Ernzerhof, M.; Scuseria, G. E. *J. Chem. Phys.* in press.
(65) Gaussian 99, Development Version (Revision A.5); Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Zakrzewski, V. G.; Montgomery, J. A., Jr.; Stratmann, R. E.; Burant, J. C.; Dapprich, S.; Millam, J. M.; Daniels, A. D.; Kudin, K. N.; Strain, M. C.; Farkas, O.; Tomasi, J.; Barone, V.; Cossi, M.; Cammi, R.; Mennucci, B.; Pomelli, C.; Adamo, C.; Clifford, S.; Ochterski, J.; Petersson, G. A.; Ayala, P. Y.; Cui, Q.; Morokuma, K.; Malick, D. K.; Rabuck, A. D.; Raghavachari, K.; Foresman, J. B.; Cioslowski, J.; Ortiz, J. V.; Stefanov, B. B.; Liu, G.; Liashenko, A.; Piskorz, P.; Komaromi, I.; Gomperts, R.; Martin, R. L.; Fox, D. J.; Keith, T.; Al-Laham, M. A.; Peng, C. Y.; Nanayakkara, A.; Gonzalez, C.; Challacombe, M.; Gill, P. M. W.; Johnson, B.; Chen, W.; Wong, M. W.; Andres, J. L.; Gonzalez, C.; Head-Gordon, M.; Replogle, E. S.; Pople, J. A. Gaussian Inc.: Pittsburgh, PA, 1998.
(66) Scuseria, G. E., to be published.
(67) Ochsenfeld, C.; Head-Gordon, M. *Chem. Phys. Lett.* **1997**, *270*, 399.