# Linearscaling semiempirical quantum calculations for macromolecules

TaiSung Lee, Darrin M. York, and Weitao Yang

Citation: The Journal of Chemical Physics **105**, 2744 (1996); doi: 10.1063/1.472136
View online: http://dx.doi.org/10.1063/1.472136
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/105/7?ver=pdfcov
Published by the AIP Publishing

# Linear-scaling semiempirical quantum calculations for macromolecules

Tai-Sung Lee, Darrin M. York, and Weitao Yang
*Department of Chemistry, Duke University, Durham, North Carolina 27708*

(Received 9 April 1996; accepted 6 May 1996)

A linear-scaling method to carry out semiempirical quantum mechanical calculations for large systems has been developed based on the density matrix version of the divide-and-conquer approach. The method has been tested and demonstrated to be accurate and efficient. With this implementation, semiempirical quantum mechanical calculations are made possible for large molecules over 9000 atoms on a typical workstation. For biological macromolecules, solvent effects are included with a dielectric continuum model. © *1996 American Institute of Physics.* [S0021-9606(96)01231-7]

## INTRODUCTION

A quantum mechanical description of the electronic structure is important for many aspects of molecular modeling, including bond formation and cleavage in chemical reactions, polarization, and chemical bonding of metal ions. In these cases, it is difficult, if not impossible, to use the conventional molecular mechanical force fields. Quantum calculations are, however, expensive. The numerical effort of conventional electronic structure methods scales as $N^3$ or higher, where $N$ is the number of electrons. This cubic or higher order scaling is the ultimate bottleneck for the applications of quantum calculations to large molecules. Currently, only molecules with few hundred atoms can be treated in *ab initio* calculations,[1,2] while conventional semiempirical methods can handle about 1000 atoms with supercomputers.[3]

Much effort has been made in the development of linear scaling quantum calculations, i.e., methods that require computational effort proportional to the size of the system. Yang first proposed the divide-and-conquer approach and demonstrated that it is possible to attain a solution of linear scaling by localizing the electronic degrees of freedom.[4,5] Galli and Parrinello suggested a linear scaling algorithm and applied to tight-binding Hamiltonians.[6] Li, Nunes, and Vanderbilt introduced a variational method for obtaining the density matrix with cutoff in real space and showed linear scaling in computational effort.[7] Mauri, Galli, and Car,[8] and Ordejón *et al.*,[9] used unconstrained minimization methods combined with a local orbital formulation and were able to achieve linear scaling. Stechel, Williams, and Feibelman also proposed a block diagonalization process in subspace.[10] Goedecker and Colombo developed another linear scaling algorithm to replace the diagonalization process in the tight-binding calculations.[11] These linear-scaling methods eliminate the cubic-scaling step associated with the determination of the occupied electronic eigenstates in the quantum calculations. Nonetheless, application of these methods to macromolecule systems has not yet been demonstrated at the *ab initio* level. The main reason, in the case of linear combination of atomic orbitals, is that the matrix element evaluation is still the bottleneck, even though it has a computational scaling of only formally $O(N^2)$. Recent efforts have been directed to develop linear-scaling algorithms for the evaluation of the matrix elements and the long-range Coulomb interaction.[12–16] It is hopeful that work along this line will soon make first-principle calculations of the matrix elements efficient enough that the linear-scaling quantum mechanical algorithms can be effective.

The situation in semiempirical quantum mechanical methods is quite different.[17,18] The Hamiltonian matrix is constructed very efficiently from semiempirical approximations and hence the computational effort is in practice limited by the cubic-scaling diagonalization processes. This is just what the linear-scaling algorithms are designed to overcome. Stewart has proposed a local molecular orbital method for semiempirical calculations.[19] Yang and Lee[20] extended the divide-and-conquer approach[4] to accommodate density matrix description so that it can be applied to Hartree–Fock and semiempirical methods. Thus applying semiempirical quantum calculations to large systems becomes possible with this density-matrix version of the divide-and-conquer approach.

In this paper, we describe the implementation of the density matrix divide-and-conquer approach into the semiempirical MOPAC program.[21,22] The method is demonstrated to be accurate and efficient. Semiempirical quantum mechanical calculations are made possible for large molecules over 9000 atoms on a typical workstation. For description of biological macromolecules, solvent effects are included with a dielectric continuum model.

## THEORY

We briefly summarize the density-matrix divide-and-conquer approach below.[20] The one-electron density matrix can be used as the basic variable in various quantum mechanical calculations. Its matrix elements are given by

$$\rho_{ij} = \sum_m n_m C_{im} C_{jm}, \tag{1}$$

where $n_m$ is the occupation number of the $m$th molecular orbital and $\{C_{im}\}$ are the expansion coefficients of the molecular orbital over atomic basis functions. $\{C_{im}\}$ are determined by the algebraic eigenvalue equation

$$\mathbf{H}\mathbf{C}_m = \mathbf{S}\mathbf{C}_m \boldsymbol{\epsilon}_m, \tag{2}$$

 0021-9606/96/105(7)/2744/7/$10.00 

where $H$ is the molecular one-electron Hamiltonian matrix, $S$ is the overlap matrix, and $\epsilon_m$ is the eigenvalue corresponding to the eigenvector $\mathbf{C}_m$.

In the density matrix formalism of the divide-and-conquer approach, the density matrix is divided into subsystem contributions by the use of the symmetric weight matrices $P^\alpha$

$$\rho_{ij}=\sum_\alpha P_{ij}^\alpha \rho_{ij}\equiv\sum_\alpha \rho_{ij}^\alpha, \tag{3}$$

$$\sum_\alpha P_{ij}^\alpha=1, \quad \forall i,j=1\cdots M, \tag{4}$$

where $\alpha$ is index of subsystems and $M$ is the size of the basis sets. Currently, the Mulliken-type weight matrix is used[23]

$$\begin{aligned} P_{ij}^\alpha &= 1 \quad \text{if } i\in\alpha \text{ and } j\in\alpha \\ &= 1/2 \quad \text{if } i\in\alpha \text{ and } j\notin\alpha. \\ &= 0 \quad \text{if } i\notin\alpha \text{ and } j\notin\alpha \end{aligned} \tag{5}$$

Because of the local nature of the density matrix in real space, the density matrix projected into each subsystem can be approximated by solving the expansion coefficients locally; namely,

$$\rho_{ij}^\alpha\cong P_{ij}^\alpha\sum_m n_m^\alpha C_{im}^\alpha C_{jm}^\alpha, \tag{6}$$

where $n_m^\alpha$ and $C_{im}^\alpha$ are the occupation number and the eigenvector of the $m$th molecular orbital in the $\alpha$th subsystem, respectively. The local eigenvectors for the $\alpha$th subsystem are determined by the subsystem eigenvalue equation

$$\mathbf{H}^\alpha\mathbf{C}_m^\alpha=\mathbf{S}^\alpha\mathbf{C}_m^\alpha\epsilon_m^\alpha, \tag{7}$$

where $H^\alpha$ is the molecular one-electron Hamiltonian matrix, $S^\alpha$ is the overlap matrix, and $\{\epsilon_m^\alpha\}$ are the eigenvalues for the $\alpha$th subsystem.

The occupation number $n_m^\alpha$ is approximated by the Fermi function $f_\beta(\mu-\epsilon_m^\alpha)$, with $f_\beta(x)=[1+\exp(-\beta x)]^{-1}$, where $\beta$ is the inverse temperature, $\mu$ is the chemical potential, $\mu$ is chosen so that normalization of the density is maintained

$$N=\sum_\alpha\sum_{ij}\rho_{ij}^\alpha S_{ij}^\alpha, \tag{8}$$

$$\rho_{ij}^\alpha=P_{ij}^\alpha\sum_m n_m^\alpha C_{im}^\alpha C_{jm}^\alpha\cong 2P_{ij}^\alpha\sum_m f_\beta(\mu-\epsilon_m^\alpha)C_{im}^\alpha C_{jm}^\alpha, \tag{9}$$

where $N$ is the total number of electrons and the factor 2 accounts for the spin degrees of freedom. The foregoing description of the density-matrix divide-and-conquer approach applies to general electronic structure calculations. The construction of density matrix from fragment contributions in the manner of Eqs. (3)–(5) based on the Mulliken population analysis has also been employed by Walker and Mezey,[24] and Massa, Huang and Karle.[25]

In semiempirical calculations, the electronic energy is expressed by

$$E=\tfrac{1}{2}\sum_{ij}\rho_{ij}(H_{ij}^{\text{core}}+F_{ij}), \tag{10}$$

where $H^{\text{core}}$ is the one-electron core Hamiltonian matrix and $F$ is the Fock matrix. It can be easily expressed in the divide-and-conquer approach

$$E=\tfrac{1}{2}\sum_\alpha\sum_{ij}\rho_{ij}^\alpha(H_{ij}^{\text{core}}+F_{ij}). \tag{11}$$

The energy gradient expressions for the divide-and-conquer approach have been derived and shown to be accurate.[20,26] In the MOPAC package, the energy gradients are calculated with the frozen density approximation.[21] With this approximation, the divide-and-conquer energy gradient with respect to the $\alpha$th nucleus position $R_\alpha$ is expressed by

$$\boldsymbol{\nabla}_{R_\alpha}E=\tfrac{1}{2}\sum_\alpha\sum_{ij}\rho_{ij}^\alpha\boldsymbol{\nabla}_{R_\alpha}(H_{ij}^{\text{core}}+F_{ij}). \tag{12}$$

The gradients can be calculated by analytical methods[27] or by the finite difference method.[21] Except that the total density matrix is approximated by the divide-and-conquer approach, other procedures to calculate gradients are the same as in MOPAC. The BFGS optimization procedure is used in the original MOPAC package for geometry optimization.[28] This procedure requires constructing the Hessian matrix which has an $O(N^2)$ scaling requirement of memory usage; it cannot be used for large molecules. Instead, we chose the commonly used conjugate gradient method for geometry optimization.[29]

For solution phase calculations, the COSMO model was used.[21,30–33] This model treats the solvent as a conductorlike dielectric continuum. The solute charge distribution is represented by a set of atomic charges, dipole moments, and quadrupole moments, that induces a reaction field charge density on the solvent accessible surface of the solute. The solvation energy can be written as the reaction energy between those charges

$$E_{\text{sol}}=\mathbf{q}^T\mathbf{B}\mathbf{Q}+\tfrac{1}{2}\mathbf{q}^T\mathbf{A}\mathbf{q}, \tag{13}$$

where $\{Q_i\}$ are the atomic multipoles of solute and $\{q_j\}$ are the induced charges on the solvent accessible surface. The atomic multipoles of the $\alpha$th atom of solute molecule, $Q_\alpha^i$, can be expressed in terms of the density matrix $\rho$ and the core charge $Q_\alpha^{\text{core}}$, which is the sum of the nuclear charge and the core electron charge, by the following:

$$Q_\alpha^{SS}=Q_\alpha^{\text{core}}-\rho_\alpha^{SS}; \tag{14}$$

$$Q_\alpha^{SM}=-\rho_\alpha^{SM}; \tag{15}$$

$$Q_\alpha^{MN}=-\rho_\alpha^{MN}, \quad (MN=X,Y,Z), \tag{16}$$

where $\rho_\alpha^{SS}$, and $\rho_\alpha^{SM}$, and $\rho_\alpha^{MN}$ are the matrix elements corresponding to the $S-S$ orbital pair, the $S-P_M$ orbital pair, and the $P_M-P_N$ orbital pair of the $\alpha$th atom, respectively. For hydrogen atom, only $\rho_\alpha^{SS}$ is needed. **A** and **B** are the matrices of interaction between those charges. If $R_\alpha$ is the position of the $\alpha$th atom and $r_i$ and $r_j$ are the position of the charge $q_i$ and $q_j$, then the matrix elements can be written as

$$A_{ij}=|r_i-r_j|^{-1}, \quad A_{ii}=3.8|S_i|^{1/2}, \tag{17}$$

$$B_{i,(\alpha,MN)}=\frac{1}{|r_{i\alpha}|} \quad \text{for } MN=SS,$$

$$=\frac{1}{|r_{i\alpha}|}-\frac{T_\alpha}{|r_{i\alpha}|^3}+\frac{3(r_{i\alpha})_K{}^2 T_\alpha}{|r_{i\alpha}|^5}$$

$$\text{for } MN=KK, K=(X,Y,Z)$$

$$=\frac{6(r_{i\alpha})_K(r_{i\alpha})_{K'}T_\alpha}{|r_{i\alpha}|^5}$$

$$\text{for } MN=KK', K,K'=(X,Y,Z), K\neq K'$$

$$=\frac{(r_{i\alpha})_K d_\alpha}{|r_{i\alpha}|^3} \quad \text{for } MN=SK, K=(X,Y,Z), \tag{18}$$

where $S_j$ is the surface area associated to the charge $q_j$ (in unit of e.s.u.), $r_{i\alpha}=r_i-R_\alpha$, $(r_{i\alpha})_K$ is the $K$-direction component of $r_{i\alpha}$, and $d_\alpha$ and $T_\alpha$ are parameters of the $\alpha$th atom.

Normally, $\{q_j\}$ are obtained by minimizing Eq. (13) and than scaled appropriately to finite dielectric.[34] In MOPAC, the minimization process is done through inversion of the **A** matrix. This method is fast for small systems because only one matrix inversion operation is needed for the whole SCF procedure. However, it cannot be applied to large systems because of the $O(N^3)$ computational effort for matrix inversion and the $O(N^2)$ requirement of the computer memory to store **A** or $\mathbf{A}^{-1}$.

In our implementation, a preconditioned conjugate gradient method was used to minimize $E_{\text{sol}}$ in Eq. (13) to obtain the charge set $\{q_j\}$.[33] This method needs iterative operations in the conjugate gradient minimization and leads to an additional iterative procedure within a single SCF cycle. It requires only moderate overhead relative to gas phase calculations for large molecules because it does not need the CPU intensive matrix inversion procedure.

The COSMO gradient term is given by

$$\nabla_{R_\alpha} E_{\text{sol}}=-\mathbf{q}^T(\nabla_{R_a}\mathbf{B})\mathbf{Q}+\tfrac{1}{2}\mathbf{q}^T(\nabla_{R_a}\mathbf{A})\mathbf{q}. \tag{19}$$

Because only the Coulomb type interactions are involved, the gradients of matrices **A** and **B** with respect to the $a$th nucleus position $R_\alpha$ can be easily calculated.

## IMPLEMENTATION

In the divide-and-conquer approach, each subsystem is described by a set of local basis functions, instead of the entire set of atomic orbitals. The accuracy of the description is enhanced by the use of basis functions of neighboring atoms. These neighboring atoms are called the buffer atoms.[5] We here select buffer atoms by a distance criterion, $R_b$: If an atom is within a distance $R_b$ of a subsystem, this atom will be included as a buffer atom for that subsystem. The diagonalization for a subsystem is performed with atomic basis functions on the subsystem atoms and buffer atoms, and the computational effort scales as $N_\alpha^3$ where $N_\alpha$ is the number of basis functions in the $\alpha$th subsystem and its buffer region. Previous studies using density functional theory have shown the buffer region size needed for a given accuracy is independent of the size of the whole molecule.[5,20] Hence, one can choose $N_\alpha$ to become roughly a constant; for example, each subsystem consists of a single amino acid and its buffer region includes all nearby atoms within a 6.0 Å cutoff ($R_b$=6.0 Å). Thus overall linear scaling can be reached with roughly fixed-size buffer region for each subsystem.

While the divide-and-conquer method overcomes the $O(N^3)$ scaling problem in the diagonalization process, the $O(N^2)$ or higher order scaling of the computer memory needs to be addressed. Since most of matrix elements in quantum calculations are negligibly small for large molecules, sparse matrix storage methods can be employed. For density matrix, because of its locality in real space, we can truncate the matrix elements using a distance criterion, $R_h$. Only the matrix elements corresponding to atom pairs with interatomic distance less than $R_h$ are evaluated and stored. This cutoff makes the memory storage of the density matrix proportional to the size of the molecules and also significantly reduces the CPU time used for matrix element evaluation. The one-electron core Hamiltonian and Fock matrices are treated similarly.

In the original MOPAC program, the two-electron integrals are calculated once and stored. While this is faster, the memory needed for the two-electron integrals is roughly $400N^2$ bytes where N is the total number of heavy atoms. It becomes impossible to store these integrals when the molecule is very large. Therefore, in our implementation they are calculated on the fly when needed. No significant slowdown was observed by doing this.

To further save memory, the eigenvectors obtained in solving local diagonalization problem are not entirely stored. Instead, based on the step-function-like character of Fermi function, only those eigenvectors affecting the determination of the chemical potential are stored. For other eigenvectors, the contributions to the density matrix are calculated before they are discarded.

All results were performed on an IBM RS/6000 workstation with a 67 MHz POWER2 CPU, 256 MB memory, and 512 MB swapping disk space. The PM3 Hamiltonian was used for all calculations.[35,36] All the divide-and-conquer calculations have the same parameter setting (i.e., the same KEYWORD) as MOPAC default values. The program has the ability to divide any molecule into roughly equal size subsystems. However, currently a subsystem is defined as one amino acid residue for protein molecules, and one nucleotide unit for DNA molecules.

## RESULTS

### Accuracy

We chose a small protein, RP71955, which consists of 280 atoms, to test the accuracy of our implementation. The structure of this molecule is an NMR structure proposed by Fréchet *et al.*[37]

Gradients can be calculated analytically or by the finite difference method in semiempirical calculations. Because the formulas to calculate matrix elements are much simpler than

TABLE I. The rms values of differences in Cartesian gradients by different methods in gas phase and solution phase (numbers in parentheses) calculations for the RP 71955 molecule. ''MOPAC'' means original MOPAC calculation, ''DC'' means the divide-and-conquer calculation, ''A'' is analytic gradient calculation, and ''F'' is the finite difference method. All entries are in unit of kcal/mol/Å.

| | MOPAC,A | MOPAC,F | DC,A | DC,F |
|---|---|---|---|---|
| MOPAC,A | 0 | 2.92*E*-2 | 8.97*E*-2 | 9.46*E*-2 |
| | (0) | (2.89*E*-2) | (4.50*E*-2) | (5.23*E*-2) |
| MOPAC,F | | 0 | 9.40*E*-2 | 9.07*E*-2 |
| | | (0) | (5.46*E*-2) | (4.25*E*-2) |
| DC,A | | | 0 | 2.89*E*-2 |
| | | | (0) | (2.89*E*-2) |
| DC,F | | | | 0 |
| | | | | (0) |

those to calculate derivatives of matrix elements, the finite difference method is faster than the analytical method. Table I shows the root-mean-square (rms) values of Cartesian gradient calculations by different methods for the RP71955 molecule. The finite difference method gives very close results compared to the analytic method, thus we chose the finite difference method to calculate gradients, as is used in MOPAC package.[21]

Table II shows the accuracy for different matrix cutoff distance, $R_h$, and different buffer size, $R_b$. We define an accuracy criterion as $5\times10^{-3}$ kcal/mol per atom in energy calculations and 0.1 kcal/mol/Angstrom in gradient calculations. We found buffer size should be no less than 6.0 Å to meet this accuracy criterion. This agrees with previous density-functional divide-and-conquer studies.[20,26] For the cutoff distance in density and Hamiltonian matrices, a value of 7.0 Å is appropriate to save the memory usage and CPU time without sacrificing accuracy. Hence $R_b$ is set to be 6.0 Å and $R_h$ to be 7.0 Å, for subsequent calculations.

The inverse temperature $\beta$ used in Eq. (9) has been tested with the values of 20, 40, 100, and 1000 $eV^{-1}$. The maximum difference between values of SCF heat of formation is less than 0.02 kcal/mol. Subsequently, we chose $\beta=39.3$ $eV^{-1}$, which corresponds to 298 K.

## Geometry optimization

We tested the geometry optimization in solution for the RP71955 molecule (with which the original MOPAC program failed to finish the optimization process). The optimization process stops when the following criterion is met: The maximum Cartesian gradient is less than 1.0 kcal/mol/Å. This is comparable to the default value set in the MOPAC package, which is 1.0 kcal/mol/Å for the rms value of gradients.

Figure 1 shows the decreasing of the maximum Cartesian gradient, the rms value of the Cartesian gradients, and the SCF heat of formation versus the number of iterations in the geometry optimization. That the total energy (heat of formation) decreases exponentially with the number of iteration shows the conjugate gradient method performs well in searching the energy minimum. At the final iteration, which is at the 144th geometry optimization iteration and the 324th SCF calculation, the maximum gradient and the rms value of the gradients are 0.87 and 0.19 kcal/mol/Å, respectively.

TABLE II. The self-consistent calculation results of divide-and-conquer method for the RP71955 molecule in gas phase and solution phase (numbers in parentheses). $E$(dc) is the heat of formation of divide-and-conquer calculation. $E$(mopac), the heat of formation of the original MOPAC calculation, has value of −743.136 04 kcal/mol for gas phase and −972.662 54 kcal/mol for solution phase. $N$ is the total number of atoms. $G$(dc) and $G$(mopac) are the Cartesian gradients from finite difference in the divide-and-conquer calculation and the original MOPAC calculation, respectively. $R_h$ is the distance used to cutoff the matrices and $R_b$ is the distance used to define the buffer atoms. All energies in kcal/mol, gradients are in kcal/mol/Å, and $R$'s are in Å.

| ($E$(dc)-$E$(mopac))/$N$ | $R_h$ | | | | |
|---|---|---|---|---|---|
| $R_b$ | 6.0 | 7.0 | 8.0 | 9.0 | 10.0 |
| 4.0 | 7.33*E*-03 | −1.75*E*-01 | −3.87*E*-01 | −7.43*E*-01 | −8.18*E*-01 |
| | (2.24*E*-02) | (−2.70*E*-01) | (−7.67*E*-01) | (−1.51*E*+00) | (−1.84*E*+00) |
| 5.0 | 1.38*E*-02 | −2.10*E*-02 | −4.28*E*-02 | −4.62*E*-02 | −8.76*E*-02 |
| | (1.62*E*-02) | (1.54*E*-02) | (1.22*E*-02) | (−1.23*E*-01) | (−4.01*E*-01) |
| 6.0 | 4.23*E*-03 | 3.20*E*-03 | 3.11*E*-03 | 3.10*E*-03 | 3.09*E*-03 |
| | (5.41*E*-03) | (4.47*E*-03) | (4.40*E*-03) | (4.40*E*-03) | (4.40*E*-03) |
| 7.0 | 2.01*E*-03 | 7.69*E*-04 | 6.39*E*-04 | 6.23*E*-04 | 6.22*E*-04 |
| | (2.14*E*-03) | (9.44*E*-04) | (8.20*E*-04) | (8.07*E*-04) | (8.06*E*-04) |
| 8.0 | 1.78*E*-03 | 5.02*E*-04 | 3.56*E*-04 | 3.37*E*-04 | 3.31*E*-04 |
| | (1.86*E*-03) | (6.13*E*-04) | (4.77*E*-04) | (4.61*E*-04) | (4.60*E*-04) |
| rms($G$(dc)-$G$(mopac)) | | | | | |
| 4.0 | 8.80*E*-01 | 2.14*E*+00 | 2.12*E*+00 | 2.11*E*+00 | 2.10*E*+00 |
| | (8.79*E*-01) | (7.03*E*-01) | (7.39*E*-01) | (7.68*E*-01) | (8.05*E*-01) |
| 5.0 | 2.34*E*-01 | 2.40*E*-01 | 2.06*E*-01 | 3.04*E*-01 | 3.09*E*-01 |
| | (2.13*E*-01) | (1.44*E*-01) | (1.44*E*-01) | (1.44*E*-01) | (1.80*E*-01) |
| 6.0 | 8.95*E*-02 | 9.07*E*-02 | 9.08*E*-02 | 9.08*E*-02 | 9.08*E*-02 |
| | (4.15*E*-02) | (4.25*E*-02) | (4.13*E*-02) | (4.14*E*-02) | (4.15*E*-02) |
| 7.0 | 1.76*E*-02 | 1.30*E*-02 | 1.29*E*-02 | 1.29*E*-02 | 1.29*E*-02 |
| | (1.67*E*-02) | (2.09*E*-02) | (1.68*E*-02) | (1.67*E*-02) | (1.67*E*-02) |
| 8.0 | 1.38*E*-02 | 6.97*E*-03 | 6.77*E*-03 | 6.77*E*-03 | 6.74*E*-03 |
| | (1.63*E*-02) | (2.04*E*-02) | (1.65*E*-02) | (1.63*E*-02) | (1.63*E*-02) |

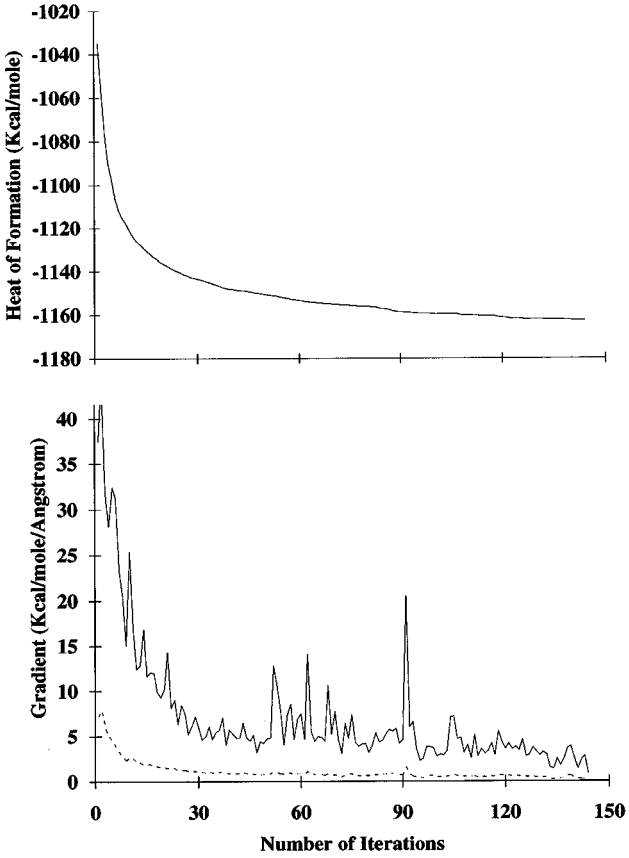


FIG. 1. The change of SCF energy (heat of formation) and gradients in the geometry optimization process. The upper is SCF energy in kcal/mol while the lower is the maximum Cartesian gradient (solid line) and the rms gradient (dash line) in kcal/mol/Å.

The total CPU time for the whole optimization is 118 251.64 s which is about 33 h. The average number of iterations used to get a converged result in each SCF calculation is 6.8. The optimized geometry has been taken as the input geometry for original MOPAC to calculate the gradients and gives values of 0.74 and 0.20 for the maximum gradient and the rms of gradients, respectively; those values already meet the default geometry optimization criterion in the MOPAC package. This confirms that the divide-and-conquer energy minimum is also a converged minimum in the original MOPAC program.

In Table III, the geometry obtained is compared with the input geometry which is an experimental NMR minimized average structure. The rms values show the bond lengths and bond angles from our calculation are very close to the experimental data, while the dihedral angles have a somewhat larger deviation. The rms positional deviation is 0.35 Å, indicating only a small deviation from the NMR structure.

TABLE III. The rms values of the differences of geometry parameters obtained from the divide-and-conquer calculation and original MOPAC calculation for the RP71955 protein molecule.

| | rms value |
|---|---|
| Bond length (Å) | 0.026 |
| Bond angle (degree) | 2.83 |
| Dihedral angle (degree) | 14.65 |
| rms positional deviation (Å) | 0.35 |
| rms positional deviation without H atoms (Å) | 0.32 |

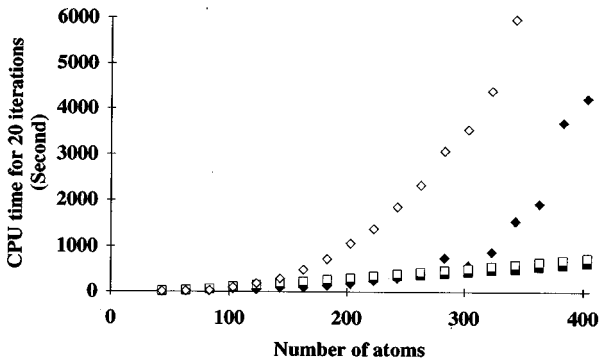


FIG. 2. The CPU time comparison between the divide-and-conquer implementation and the original MOPAC in gas phase (solid diamond), the original MOPAC in solution phase (unfilled diamond), the divide-and-conquer gas phase calculation (solid square), and the divide-and-conquer solution phase calculation (unfilled square).

## CPU time and memory usage

First, the scaling behavior of CPU times for both the original MOPAC and our implementation were compared by testing a series of alanine polypeptides which were constructed by SYBYL in linear alpha helix form and have size of 13 atoms to 403 atoms. Figure 2 shows the average CPU time used in a 20-iteration energy calculation test in both gas and solution phase calculations. The cubic scaling behavior of original MOPAC method is clearly shown. It also demonstrates the limitation of the maximum size treatable in MOPAC on a typical workstation. In this test, the divide-and-conquer method is faster than original MOPAC when the size of molecule is greater than 263 atoms in gas phase calculation, and when the size of molecule is greater than 123 atoms in solution phase calculation. For space-packed 3-D molecules, we find it is already faster when calculating RP71955 molecule of 280 atom; the average CPU time for one iteration in a SCF calculation is 39.93 s using the original MOPAC and is 29.72 s using the divide-and-conquer method in gas phase calculations. In solution phase calculations, the values become 296.89 s for original MOPAC and 44.74 s for the divide-and-conquer method. The main reason for this significant speed-up in solution calculations is that the $O(N^3)$ matrix inversion process is replaced by a faster conjugate gradient method.[33]

Several molecules with size of 256 atoms to 9378 atoms (Table IV) were chosen to test the CPU time needed in the divide-and-conquer method. For each molecule, one iteration of energy calculation was performed, followed by the gradient calculation. Figure 3 shows the CPU time used for the energy calculations for those molecules in gas phase and solution phase; Fig. 4 is the CPU time for the gradient calculations. A nearly linear-scaling behavior in the energy calculations is demonstrated. The HIV protease tetramer and hexamer molecules need significantly smaller CPU time than

TABLE IV. The molecules chosen for calculations in Fig. 3 and Fig. 4 and the memory needed for one density matrix in original MOPAC and the divide-and-conquer implementation with $R_h$=7.0 Å. The unit is million bytes (Mbyte). The structures are from various schemes and indicated in the footnotes.

| Molecule | Number of Atoms | Memory needed | |
|---|---|---|---|
| | | MOPAC | Divide-and-Conquer |
| 4 C–G pairs A–DNA[a] | 256 | 2.24 | 0.62 |
| RP71955[b] | 280 | 2.13 | 0.61 |
| 8 C–G pairs A–DNA[a] | 508 | 8.91 | 1.36 |
| Crambin[c] | 642 | 10.54 | 1.66 |
| BPTI[d] | 892 | 20.33 | 2.34 |
| Crambin dimer[c] | 1284 | 42.16 | 3.40 |
| HIV protease[e] | 1563 | 58.91 | 3.76 |
| lysozyme[f] | 1960 | 98.55 | 5.72 |
| P21[g] | 2662 | 181.04 | 7.79 |
| HIV protease dimer[e] | 3126 | 235.59 | 8.25 |
| Superoxide dismutase[j] | 4380 | 488.63 | 12.93 |
| Alcohol dehydrogenase[i] | 5639 | 783.38 | 16.86 |
| HIV protease 4-mer[e] | 6252 | 942.31 | 15.42 |
| Restriction endonuclease bamhi[k] | 7115 | 1315.44 | 22.67 |
| Acetylcholinesterase[h] | 8408 | 1807.02 | 26.78 |
| HIV protease 6-mer[e] | 9378 | 2120.14 | 24.57 |

[a]SYBYL minimization.
[b]NMR (Ref. 37).
[c]Crystal (Ref. 38), then SYBYL minimization.
[d]Crystal (Ref. 39), then MM simulation (Ref. 40).
[e]Crystal (Ref. 41), then MM simulation (Ref. 42).
[f]Crystal (Ref. 43), then AMBER minimization.
[g]Crystal (Ref. 44), then AMBER minimization.
[h]Crystal (Ref. 45), then AMBER minimization.
[i]Crystal (Ref. 46), then AMBER minimization.
[j]Crystal (Ref. 47), then AMBER minimization.
[k]Crystal (Ref. 48), then AMBER minimization.

other molecules with similar size because they are relatively less space packed, which means that less atoms are included as buffer atoms and the CPU time is reduced as a result. However, in solution phase calculations, these two molecules have more solvent accessible surface area and need more CPU time. The gradient calculations give an $O(N^2)$ curve, which we will discuss in the next section.

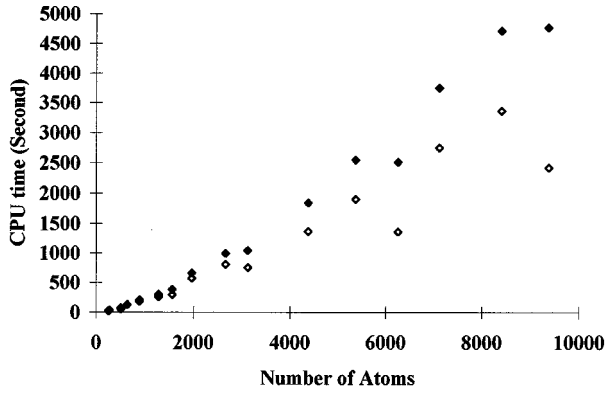


FIG. 3. The CPU time needed for one iteration in energy calculations for molecules listed in Table IV in-gas phase (unfilled) and solution phase (filled).

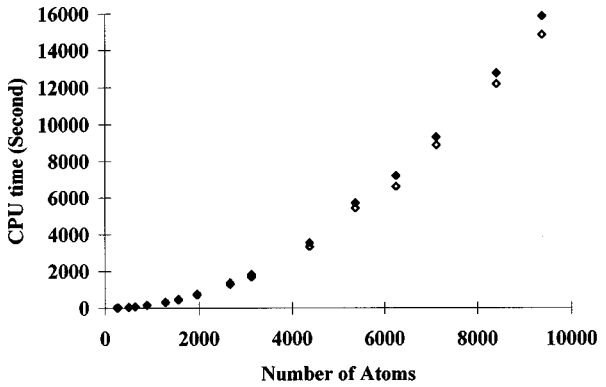


FIG. 4. The CPU time needed for the gradient calculations for molecules listed in Table IV in gas phase (unfilled) and solution phase (filled).

To compare the memory usage in our implementation with original MOPAC, we calculated memory usage of the density matrix (The one-electron core Hamiltonian matrix and the Fock matrix have the same size as the density matrix.) The results are listed in Table IV. The total memory needed in the divide-and-conquer implementation is roughly eight times the size of the density matrix, while MOPAC needs more than twenty times.

## DISCUSSIONS AND CONCLUSIONS

We would like to address several points associated with the linear-scaling curve obtained in one-iteration energy calculations. First, the total CPU time needed to get a converged SCF energy may not necessarily be $O(N)$, because the number of iterations to get converged results for large molecules may vary. We currently use a simple mixing scheme to ensure and speed up the convergence, and find that the numbers of iterations needed to get converged energies for different molecules are almost unpredictable. For molecules we tested, they fell into a range of 20 to 90, and are independent to the size of molecules. Thus a detailed study of convergence is needed.

Second, because there is no diagonalization process in the gradient calculations, the divide-and-conquer method cannot afford any advantage to reduce the order of scaling. The $O(N^2)$ curve, clearly shown in Fig. 4, mainly roots in that the gradients are evaluated in an atom-pairwise way. To calculate the energy gradients with respect to a certain nucleus position, contributions from all other atoms must be considered, since the long-range Coulomb interaction cannot be ignored for any atom pair. This $O(N^2)$ scaling, however, is not the main problem for the molecules presented here. For example, in the worst case, the HIV protease hexamer molecule with 9378 atoms, the CPU time to calculate all the gradients is about three times as the CPU time for one iteration in the SCF solution phase energy calculation. As we mentioned, typically it needs 20–90 iterations to get converged SCF results and for each SCF calculation only one gradient calculation is performed. Thus the CPU time for gradient calculations is still not a major part of the total CPU time. Although we did not clearly observe the effect of

$O(N^2)$ scaling from the Coulomb interaction in the energy calculations, we expect it will appear for larger molecules and fast multipole methods will be needed to overcome this problem.[16,38,39]

Third, we have extended the capability of semiempirical quantum mechanical methods to treat molecules of over 9000 atoms on a typical workstation. Many properties, such as the Mulliken charges, the molecular dipole moments, the solvation energy and the electrostatic potential surface, can be calculated. However, for large molecules, large amount of CPU time is needed to perform a full geometry optimization, which is necessary in some types of problems; for example, a molecule with 3000 atoms will need more than ten days to finish a full geometry optimization if assuming that the numbers of SCF calculations needed is about 320. The solution to this problem will be either using more sophisticate algorithms in the optimization process or that the geometry optimization process must be broken into local geometry optimization processes, i.e., only optimizing a small region while keeping the rest part frozen.

In addition, as proposed by Yang and Lee,[20] the density matrix version of the divide-and-conquer method can be applied to density-functional and Hartree–Fock calculations. We realize that the current computer power is still difficult to handle large molecules at high level *ab initio* calculations, however, the linear scaling of quantum calculations at this level will eventually become necessary.

In conclusion, we demonstrated the $O(N^3)$ of the diagonalization process has been circumvented by incorporating the divide-and-conquer method with semiempirical approximation. The nearly $O(N)$ scaling makes semiempirical methods applicable to large molecules such as proteins and enzymes. With this method, quantum calculations of many interesting macromolecules can be performed and a better understanding of those systems at the electronic level will be realized.

## ACKNOWLEDGMENTS

Financial support from the NIH Parallel Computing Resource for Structural Biology at the University of North Carolina-Chapel Hill, the National Science Foundation, and the U.S. Environmental Protection Agency is gratefully acknowledged. This work has also been partially supported by the Exxon Education Foundation. D.Y. acknowledges partial funding support through an NSF postdoctoral fellowship jointly funded by the North Carolina supercomputing Center. W.Y. is a Alfred P. Sloan Research Fellow.

[1] J. P. Lewis, O. F. Sankey, and P. J. Ordejón, Phys. Rev. B (submitted).
[2] L. Y. Zhang and R. A. Friesner, J. Phys. Chem. **99**, 16479 (1995).
[3] D. Bakowies, M. Bühl, and W. Thiel, J. Am. Chem. Soc. **117**, 10113 (1995).
[4] W. Yang, Phys. Rev. Lett. **66**, 438 (1991).
[5] W. Yang, Phys. Rev. A **44**, 7823 (1991).
[6] G. Galli and M. Parrinello, Phys. Rev. Lett. **69**, 3547 (1992).
[7] X.-P. Li, R. W. Nunes, and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).
[8] F. Mauri, G. Galli, and R. Car, Phys. Rev. B **47**, 9973 (1993).
[9] P. Ordejón *et al.*, Phys. Rev. B **48**, 14646 (1993).
[10] E. B. Stechel, A. P. Williams, and P. J. Feibelman, Phys. Rev. B **49**, 3898 (1993).
[11] S. Goedecker and L. Colombo, Phys. Rev. B **73**, 122 (1994).
[12] J. M. Pérez-Jordá and W. Yang, Chem. Phys. Lett. **247**, 484 (1995).
[13] J. M. Pérez-Jordá and W. Yang, Chem. Phys. Lett. **241**, 469 (1995).
[14] J. M. Pérez-Jordá and W. Yang, J. Chem. Phys. **104**, 8003 (1996).
[15] C. A. White *et al.*, Chem. Phys. Lett. **230**, 8 (1994).
[16] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).
[17] J. J. P. Stewart, in *Review in Computational Chemistry*, Vol. 1, edited by K. B. Lipkowitz and D. B. Boyd (VCH, New York, 1990), Chap. 2.
[18] W. Thiel, Tetrahedron **44**, 7393 (1988).
[19] J. J. P. Stewart, Abstract of the 211th ACS National meeting, Division of Computers in Chemistry.
[20] W. Yang and T.-S. Lee, J. Chem. Phys. **103**, 5674 (1995).
[21] J. J. P. Stewart, *MOPAC7 Version 2 Manual* (QCPE, Bloomington, 1993).
[22] J. J. P. Stewart, J. Comput.-aided Mol. Design **4**, 1 (1990).
[23] R. S. Mulliken, J. Chem. Phys. **23**, 1833 (1955).
[24] P. D. Walker and P. G. Mezey, J. Am. Chem. Soc. **115**, 12423 (1993).
[25] L. Massa, L. Huang, and J. Karle, J. Quantum Chem. **29**, 371 (1995).
[26] Q. Zhao and W. Yang, J. Chem. Phys. **102**, 9598 (1995).
[27] M. J. S. Dewar and V. Yamaguch, Comput. Chem. **2**, 25 (1978).
[28] J. D. Head and M. C. Zerner, Chem. Phys. Lett. **122**, 264 (1985).
[29] W. H. Press *et al.* Numerical Recipes in Fortran, 2nd ed. (Cambridge University Press, New York, 1992), p. 413.
[30] A. Klamt and G. Schüürmann, Perkin Trans. **2**, 799 (1993).
[31] T. N. Troung and E. V. Stephanovich, J. Chem. Phys. **103**, 3709 (1995).
[32] J. Andzelm, C. Kölmel, and A. Klamant, J. Chem. Phys. **103**, 9312 (1995).
[33] D. York, T.-S. Lee, and W. Yang, Chem. Phys. Lett. (submitted).
[34] J. D. Jackson, *Classical Electrodynamics* (Wiley, New York, 1975).
[35] J. J. P. Stewart, J. Comput. Chem. **10**, 209 (1989).
[36] J. J. P. Stewart, J. Comput. Chem. **10**, 221 (1989).
[37] D. Fréchet *et al.*, Biochemistry **33**, 42 (1994).
[38] M. M. Teeter, S. M. Roe, and N. H. Heo, J. Mol. Biol. **230**, 292 (1993).
[39] J. Deisenhofer *et al.*, Acta Crystallograph. B **31**, 238 (1975).
[40] D. York *et al.*, Proc. Natl. Acad. Sci. **91**, 7815 (1994).
[41] A. Woldawer, M. Miller, and M. Jaskólski, Science **245**, 616 (1989).
[42] D. York, T. Darden, and L. Pedersen, J. Chem. Phys. **99**, 8345 (1993).
[43] K. Harata, Acta Crystallograph. B **50**, 250 (1994).
[44] E. F. Pai *et al.*, EMBO J. **9**, 2351 (1990).
[45] J. L. Sussman *et al.*, Science **253**, 872 (1991).
[46] H. Eklund, J.-P. Samama, and T. A. Jones, Biochemistry **23**, 5982 (1984).
[47] H.-X. Deng *et al.*, Science **261**, 1047 (1993).
[48] M. Newman *et al.*, Science **269**, 656 (1995).
[49] L. Greengard, Science **265**, 909 (1994).
[50] C. A. White *et al.*, Chem. Phys. Lett. **230**, 8 (1994).