PARALLELIZED HARTREE-FOCK CODE FOR SCALABLE STRUCTURAL AND ELECTRONIC SIMULATION OF LARGE NANOSCALE MOLECULES

DAVID GOODE
HARVARD UNIVERSITY, BROOKHAVEN NATIONAL LABORATORY

1. Problem and Motivation

All molecules in our world can be characterized as being composed of 3 relatively immutable particles at the energy scales in question: protons, neutrons, and electrons. The protons and neutrons are bound together by something called the strong force into the nucleus, about which the electrons, which are relatively free by comparison, settle into an energetically favorable probability distribution given the coulomb forces exerted by the nucleii and by other electrons. The strong force is in fact so strong that the internal structure of nucleii can be ignored in almost all chemical applications, and it can simply be treated as a single, massive, charged particle. The mass is also large enough that the nucleii can even be treated somewhat classically as a point particle with definite velocity and position. This is most unlike the electrons, which are small enough that they must be described, as dictated by the laws of quantum mechanics, by a probability distribution corresponding to a superposition of basis position states; they do not have a definite position or classical "orbit" about the nucleii.

Schrodinger's equation is a very complex linear differential equation that accurately describes the behavior of small particles, such as the electrons and nuclei that appear in atomic nucleii, and governs their evolution in time. It is very general, and gives extremely accurate predictions for electron densities and total energies in most cases relevant to the science of chemistry and molecular mechanics. Knowing such information has many consequences. Finding the minimum energy state of a molecule corresponds to the state it will usually be in in nature (at minimal temperature). Properties such as electron affinities, polarizations, and reaction energies can be predicted based upon the changes in energies observed in the modeled system.

Schrodinger's equation is not completely correct; more accurate and complicated theories exist, such as Quantum Field theories, that deal with nuclear physics and relativistic particles with a higher degree of accuracy. Even the Schrodinger equation itself though is too difficult to solve in general for all but the tiniest molecules, so there are also approximations to it that sacrifice some accuracy in order to make computational solutions more feasible.

One such approximation is the Hartree-Fock method, which approximates the interactions between electrons with a mean-field interaction, or an averaged overall repulsion, rather than a point-to-point effect on each electron probability distribution. It is a law of quantum mechanics that can be derived from quantum field theories that electrons, being fermions, must be antisymmetric under exchange within the wavefunction. This means roughly that only two electrons of opposite spins can ever occupy the exact same space, and this effect can often be thought of as an actual repulsive force between same-spin electrons, which significantly affects the behavior of a multielectron system surrounding an atom. It is often said that the fermionic nature of electrons is what is responsible for the entire science of chemistry, and thus life itself as we know it; were electrons all able to occupy the same point, they would all reach the lowest orbital of an atom or molecule, and no interesting chemical bonding behavior or interactions would result. The Hartree-Fock method respects this requirement, and uses the simplest possible antisymmetrized wavefunctions to comply with it exactly.

There are many different computational chemistry codes in existence, many of which use this approximation in some fashion. At Brookhaven, the NWChem suite is the primary one used. However, a number of chemists at Brookhaven are currently attempting to synthesize nanoparticles for use in solar power collection. Molecules that can be so synthesized are very large, on the order of thousands of atoms, and have important behaviors that depend on this larger global structure. Existing codes were not fast enough to run such large molecules, thus the goal was to develop a streamlined, faster, less flexible code that could handle the simulation of such molecules in a massively parallelized fashion, taking advantage of the 1000s of processors on current supercomputers, as well as possibly the 10,000s or 100,000s of processors on future supercomputers.

2. Background and Related Work

Quantum mechanics is described similarly to the Hamiltonian formulation of classical mechanics, whereby the time evolution of a system is encapsulated in the Hamiltonian operator or equation. The Hartree Fock method is founded upon the use of a mean-field approximation to the multi-electron Hamiltonian for a given molecule. This avoids the insoluble multi-body problem and reduces the solution of Schrodingers equation from a linear differential equation in a theoretically infinite basis with infinite solutions corresponding to all excited energies to a non-linear iterative optimization problem in a finite basis of reasonable basis orbitals that usually correspond to individual atomic orbitals.

The solution space is the set of all linear combinations of a finite number of chosen basis functions. In theoretical quantum mechanics, the basis would generally be complete and usually infinite, in order to span the entire solution space. This is obviously computationally impractical, so a finite set of basis functions is chosen so that linear combinations therein should well approximate the ideal solution. This process is extremely analogous to that of fourier analysis, whereby the periodic functions form a complete basis for all suitably continuous functions in L^2, but a very good approximation can be generated by just constructing a function out of a few dozen of these basis functions, not an infinite number.

The solutions are basically occupation numbers for each basis element. The basis functions are chosen such that integrals between combinations of them, corresponding to coulomb repulsion, electron exchange interactions, and probability densities, can be efficiently and accurately calculated. The problem is thus reduced to one of iteratively minimizing the energy associated with the occupations of these orbitals (for a constant total occupation number corresponding to the number of electrons), by first finding the minimum occupancies associated with the Hamiltonian of the current solution, using these as the new occupancies and finding the new approximate Hamiltonian, and then repeating until convergence.

An important aspect of this calculation is determining the contributing elements of the Hamiltonian corresponding to repulsion between the given basis functions. These basis functions correspond to probability distributions in space, and these integrals among them can be thought of as representing the coulomb repulsion and same-spin exchange interactions between electrons. The integrals relate to four basis functions simultaneously:

$$\int_{\mathbb{R}^3}\int_{\mathbb{R}^3}\frac{\Psi(\mathbf{x}_1)\Phi(\mathbf{x}_1)\Lambda(\mathbf{x}_2)\Omega(\mathbf{x}_2)}{|\mathbf{x}_1-\mathbf{x}_2|}d\mathbf{x}_1 d\mathbf{x}_2$$

The evaluation of these integrals is one of the most computationally-intensive parts of the algorithm, scaling with $O(N^4)$. However, they must only be done once and can be reused for subsequent iterations. Additionally, the use of the L2 norm makes use of the Hilbert space over these well-behaved, generally Schwartz class functions that make up the basis elements, and allows for the use of the Cauchy-Schwarz inequality to forgo calculations of integrals below a certain threshold in magnitude. Because the chosen basis functions decay exponentially with radius, in large molecules this reduces the number of integrals that must be calculated to $O(N^2)$, as is empirically confirmed in the results.

The actual iterative calculation can be solved using a generalized Hermitian eigenvalue problem over this basis in terms of both spins:

$$\mathbf{F}^\alpha\mathbf{C}^\alpha = \mathbf{S}\mathbf{C}^\alpha\epsilon^\alpha$$

The analogous eigenvalue problem for the other spin is solved simultaneously. Here, S is the overlap matrix between the basis functions in use (in an ideal theoretical case this is usually assumed to be the identity if the basis functions are orthogonal), C is the coefficients of the electron orbitals, and F is the Fock approximation to the Hamiltonian, of which epsilon is the eigenvalues when acting upon C, giving the energies of the orbitals. The F and P matrices for one spin are:

$$F^\alpha_{\mu\nu} = H^{core}_{\mu\nu} + \sum_\lambda\sum_\sigma P^\alpha_{\lambda\sigma}\left[(\mu\nu|\sigma\lambda) - (\mu\lambda|\sigma\nu)\right] + P^\beta_{\lambda\sigma}(\mu\nu|\sigma\lambda)$$

$$P^\alpha_{\mu\nu} = \sum_a^{N^\alpha} C^\alpha_{\mu a}(C^\alpha_{\mu a})*$$

Here, P is the density matrix associated with the current orbital coefficients, which are calculated in the generalized Hermitian eigenvalue problem. Thus, F and C are simultaneously dependent on each other, and the two sets of linear equations must be solved iteratively until convergence in a resulting non-linear optimization

problem. The four-index parenthesis represent the 2-electron integrals previously described, which give the "overlap" of different basis elements. The second term of Fs equality incorporates the mean-field coulomb interaction and exchange interaction of the electrons; the first is the kinetic energy and nuclear attraction components of the Born-Oppenheimer approximated Hamiltonian for the molecular system in the chosen basis.

3. Approach and Uniqueness

A fully new computational chemistry program was implemented, written in C++. It uses the standard MPI supercomputing communication interface, an implementation of which is available on BlueGene/L. The code is fairly cross-platform compatible, also working on the OpenMPI implementation distributed with Ubuntu Linux. The program makes heavy use of the LaPACK and ScalaPACK serial and parallelized, respectively, linear algebra libraries, which have implementations for both BlueGene and regular Linux. These are very standard, open-source, well-supported libraries for many standard linear algebra routines. ScalaPACK, the parallelized version, utilizes MPI through the BLACS communication library. Most existing quantum chemistry programs do not use the linear algebra package, and thus this presents a novel approach that may have different, perhaps advantageous scaling properties, both with system size and with processor grid size. All the current algorithmic code is written by David Goode; some of the support code, and a reduced serialized version of the code, were written by Nicholas D'Imperio.

The Hartree-Fock method employed makes use of contracted Gaussian basis functions for easy analytic integration and reasonable accuracy as compared to observed and calculated single atom electron orbitals. Differentiation based generating functions are used for higher, non-symmetric orbitals up to d-type, which consist of spherically symmetric Gaussians multiplied by asymmetric monomials in X, Y, and Z, all in position space.

The parallelization is achieved through 2D block-cyclic distribution of most $O(N^2)$ matrices involved in eigenvector calculations, such as the coefficients matrix, the Fock matrix, and the density matrix. The program supports open spin configurations through simultaneously solving one set of density matrices per spin, as outlined in Modern Quantum Chemistry by Szabo and Ostlund (1996) and other texts. Higher orbitals up to D, corresponding physically to more angular momentum and mathematically to lack of spherical symmetry, are supported with derivative-based generating functions for Cartesian Gaussians and their integrals as well as linear equations for polynomials in the basis of the Hermite polynomials, which are easier to integrate analytically. The specific equations were derived in Mathematica, and the Hermite Gaussian integration methods were described in a paper by McMurchie and Davidson:

$$[NLM|r_{12}^{-1}|N'L'M'] = \lambda(-1)^{N'+L'+M'}\left(\frac{\partial}{\partial a}\right)^{N+N'}\left(\frac{\partial}{\partial b}\right)^{L+L'}\left(\frac{\partial}{\partial c}\right)^{M+M'}\int_0^1 e^{-\alpha(a^2+b^2+c^2)u^2}du$$

Here, N,L, and M correspond to indices of Hermite polynomials for each dimension, one set for each pair of basis functions being evaluated. This is the form of one element of the two electron integrals.

Many optimizations were implemented in parallel. Two-electron integrals are prescreened and ignored based on the Cauchy Schwarz bound, and they are calculated in an order that maximizes the utility of this bound. This saves over 90% of computation time and storage in a 700 atom test molecule. The electron density matrix is broadcast and reused for energy calculations as well as Fock matrix calculations. The coefficients matrix is shared in a novel checkerboard pattern that provides each processor with the transpose row it needs for calculations while balancing 2 communications per step across every processor. This algorithm was achieved through use of symmetric block sizes that make transpose locations uniform across every block stored cyclicly on a given node.

The specific communication topologies and parallelized algorithms that were derived and coded for this program were not taken from any references, although it is likely that similar ideas have been used before. One example is the parallelization of the Fock or F matrix calculation, which determines the approximate Hamiltonian to be used to find the energies during the subsequent iteration of the process. The interprocessor scheme in blue, below, was employed. The red squares correspond to parts of the density or P matrix stored by a single processor. To calculate its part of the F matrix, it must iterate over the whole P matrix. It does this processor by processor. When it gets to the green one, it needs the 2-electron integrals it has (the ones with lambda and sigma on one side of the parenthesis in the F equation) as well as the ones that other processes have, namely the orange one about itself and about the yellow processor (each processor stores $O(N^2)$ integrals

naively, although most are trivial). It sends and receives this data as well as the requisite P matrix data to complete its part of the calculation, as do all the other processors simultaneously.

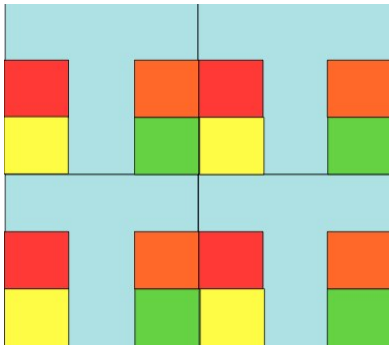

FIGURE 1. Fock or F matrix calculation

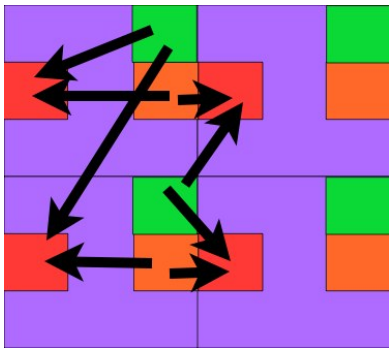

FIGURE 2. Coefficients or C matrix calculation

Another parallelization, in purple, involved calculating the density or P matrix in terms of the coefficients or C matrix, which was calculated in the generalized eigenproblem and distributed across the process grid as well. This step determines the total occupation of each basis function in the system, and gives all the information at each step about the probabilities of finding electrons at any point in space around the molecule. This, in turn, allows calculations of molecular properties such as polarization and charge distribution. For this calculation, each value depends on the entire row of the value and the row of the transpose of the value, as seen in the equation presented in the theory section. To perform this parallelization, the processor grid was required to be square, which allowed the red processor to request information from the orange one and the green one and get the relevant C matrix entries for all its blocks. This $O(N^3)$ algorithm is illustrated for one of the $O(N^2)$ steps for a single processor on the right. Note how the processor grid lines up to allow this calculation on the red processor with only the information sent from the orange and green. This same operation is repeated on all other processors simultaneously, and the sending is initiated such that each processor does 2 sends and 2 receives on every cycle. The operation on the right would also be repeated with the column on the right of the green and orange processors transmitted coefficients matrix to complete this step.

4. Results and Contributions

Overall scaling results were good but could certainly be improved, and both more work and more testing remain. Shown is a graph of the scaling performance up to 1024 processors, one of the scaling of individual components. All tests were performed on the BlueGene/L system at Brookhaven National Laboratory. Eventually, tests on much larger systems are planned on many more processors, but there hasn't yet been enough human or computer time to attempt all these envisioned studies. The F matrix calculation scaling was quite good, but the two electron scaling was not as performant as expected; it should improve with a more complex basis elements and larger systems, however.

On the other hand, the Cauchy-Schwarz bound for the two electron integral calculations was quite precise, only returning a small number of "false positives" whereby the bound was weak and the integral could matter

but, upon calculation, was found to be too small. The bound is theoretically guaranteed to never return "false negatives." On a test molecule of 100 atoms, the bound correctly threw out 93% of integrals, calculated the remainder, and found 90% of those calculated to in fact be relevant. This discounts any urgent need for a stronger bound or heuristic to avoid more integrals. The ScalaPACK eigenvector calculation scaled poorly on the systems in question, which involved relatively small matrices. The scaling of this package is studied more extensively elsewhere, and is not of immediate concern. Improved scaling, improved convergence, better accuracy, and extended output functionalities are all areas for future work. Another intriguing result involved

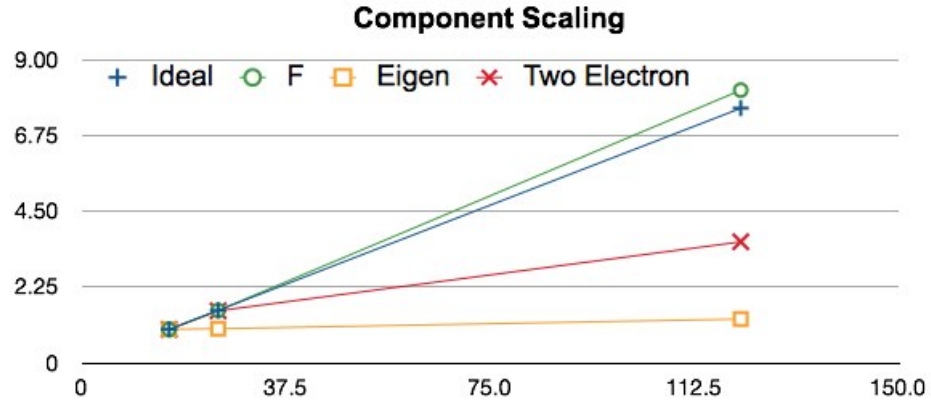

FIGURE 3. Component Scaling

the dependence of convergence upon localization of the basis. Although not discussed here, the iterations are not always found to converge in practice, and various methods such as finite temperature are employed to aid this process. In testing, it was found that more localized basis functions were much more likely to converge on large systems, and that these coefficients could then be used as a good starting point to slowly expand the basis functions in space. This is a result that could be used to inform features of this and other quantum chemistry programs in the future. Overall, computational chemistry is a very major aspect of parallelized computation.

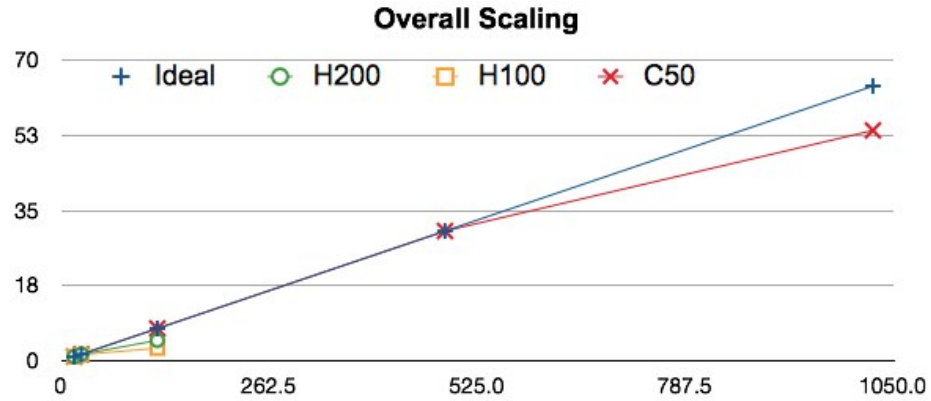

FIGURE 4. Scaling Performance

It is used frequently to inform experiments and synthesis operations in research and industry. Any performance or capability improvements are thus very important, and with this project we began studying the possibilities for scaling of a simplified algorithm, as well as the applications of standard packages such as ScaLAPACK for use in this field.

Research was performed at Brookhaven National Laboratory by David Goode, a Physics, Math, and Computer Science student at Harvard University, under the mentorship of Dr. Michael McGuigan and Nicholas D'Imperio.

5. References

(1) Szabo, Attila, & Ostlund, Neil S. (1996). Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory. Mineola, NY: Dover.

(2) Cook, David B. (2005). Handbook of Computational Quantum Chemistry. Mineola, NY: Dover.

(3) Pacheco, Peter S. (1997). Parallel Programming with MPI. San Francisco, CA: Morgan Kaufmann Publishers.

(4) Gill, Peter M. W., Johnson, Benny G., & Pople ,John A. (1993). A simple yet powerful upper bound for Coulomb integrals. Chemical Physical Letters, Volume 217.

(5) Hunt, William J., & Goddard, William A. (1969). Excited states of H2O using improved virtual orbitals. Chemical Physical Letters, Volume 3.

(6) McMurchie, Larry E., & Davidson, Ernest R. (1978). One- and Two-Electron Integrals over Cartesian Gaussian Functions. Journal of Computational Physics 26.

(7) Wolfram Research, Inc., Mathematica, Version 7.0, Champaign, IL (2008).

(8) Schuchardt, K.L., Didier, B.T., Elsethagen, T., Sun, L., Gurumoorthi, V., Chase, J., Li, J., & Windus, T.L. (2007). Basis Set Exchange: A Community Database for Computational Sciences. J. Chem. Inf. Model., 47(3), 1045-1052, 2007, doi:10.1021/ci600510j.

(9) S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, & R. Whaley. ScaLAPACK: A Linear Algebra Library for Message-passing Computers. In Proceedings of the 1997 SIAM Conference on Parallel Processing, May 1997