# IOWA STATE UNIVERSITY
## Digital Repository

2012

# Large-Scale MP2 Calculations on the Blue Gene Architecture Using the Fragment Molecular Orbital Method

Graham D. Fletcher
*Argonne National Laboratory*

Dmitri G. Fedorov
*National Institute of Advanced Industrial Science and Technology*

Spencer Pruitt
*Iowa State University*

Theresa Lynn Windus
*Iowa State University*, twindus@iastate.edu

Mark S. Gordon
*Iowa State University*, mgordon@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/chem_pubs

Part of the Chemistry Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/chem_pubs/572. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.

# Large-Scale MP2 Calculations on the Blue Gene Architecture Using the Fragment Molecular Orbital Method

**Abstract**

Benchmark timings are presented for the fragment molecular orbital method on a Blue Gene/P computer. Algorithmic modifications that lead to enhanced performance on the Blue Gene/P architecture include strategies for the storage of fragment density matrices by process subgroups in the global address space. The computation of the atomic forces for a system with more than 3000 atoms and 44 000 basis functions, using second order perturbation theory and an augmented and polarized double-$\zeta$ basis set, takes ~7 min on 131 072 cores.

**Disciplines**

Chemistry

**Comments**

# Large-Scale MP2 Calculations on the Blue Gene Architecture Using the Fragment Molecular Orbital Method

Graham D. Fletcher,[†] Dmitri G. Fedorov,[‡] Spencer R. Pruitt,[§] Theresa L. Windus,[§] and Mark S. Gordon*,[§]

[†]Argonne Leadership Computing Facility, Argonne, Illinois 60439, United States
[‡]National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan
[§]Iowa State University and Ames Laboratory, Ames, Iowa 50011, United States

**ABSTRACT:** Benchmark timings are presented for the fragment molecular orbital method on a Blue Gene/P computer. Algorithmic modifications that lead to enhanced performance on the Blue Gene/P architecture include strategies for the storage of fragment density matrices by process subgroups in the global address space. The computation of the atomic forces for a system with more than 3000 atoms and 44 000 basis functions, using second order perturbation theory and an augmented and polarized double-$\zeta$ basis set, takes $\sim$7 min on 131 072 cores.

## INTRODUCTION

In recent years, a breakdown in Dennard's scaling[1] has prevented CPU clock speeds from increasing significantly without giving rise to punitive power and cooling requirements. Fortunately, however, Moore's law[2] has continued to apply, and this has allowed processor designers to partially mitigate for the effect of plateauing clock speeds by having multiple compute units on a chip. This is evident in the mobile processor market where low power dual core processors are common all the way through to state of the art supercomputers equipped with massively multi-core processor nodes and special purpose accelerators. On the other hand, the aggregate floating point (or more general computational) performance of a chip/socket or even collection of chips/sockets in a node is expanding at a rate that is considerably greater than the speed at which data can be transferred between chips/sockets/nodes. This scenario significantly challenges the scalability of dense algebra problems, particularly the large, distributed matrix operations that are endemic in electronic structure algorithms.[3,4] Scientific application programmers have in turn responded by seeking ways to break down a problem into manageable parts, exploiting the locality of certain properties to yield multiple levels of model scaling in a given method, as well as multiple levels of parallelism, in order to better map the computation onto the architecture. One such method in electronic structure theory is the fragment molecular orbital (FMO)[5−7] method that is the focus of the work presented here.

While there are many other fragment-based methods,[8−19] a distinctive feature of the FMO method is that the electrostatic potential (ESP) that represents the entire system is included during the calculation of the energy of each individual fragment. Further, a many-body expansion is used to account for the interfragment interactions. The FMO approach has the chief advantages of scaling nearly linearly in computation cost with the problem size, the avoidance of any empirically fitted parameters, and compatibility with all quantum chemical methods. Thus, the FMO method offers considerable flexibility to mitigate the traditional bottlenecks of quantum chemistry in terms of cost, memory, and communication bandwidth. Parallel scalability derives from the concurrent and asynchronous execution of individual fragment calculations on distinct processor subgroups, provided that even load balancing can be achieved as it has been in this work.

For parallel execution, the FMO implementation in GAMESS (General Atomic and Molecular Electronic Structure System)[20,21] can make use of the distributed data interface (DDI),[22,23] and its generalization to subgroups, the generalized DDI (GDDI).[24] GDDI allows processor subgroups to be created in such a way that, during an FMO calculation, they can access fragment data from other subgroups asynchronously in a "one-sided" fashion. Consequently, the FMO method is emerging as a highly effective means of harnessing modern supercomputing hardware to treat systems with thousands of atoms quantum-mechanically.

The FMO method has been applied to a broad range of large systems. Among many important examples of FMO applications are studies in protein folding[25] and drug design,[26−29] and an interface with the qualitative structure—activity relationship (QSAR).[30] The FMO method has also been applied to oligosaccharides,[31] zeolites,[32] nanowires,[33] and molecular clusters, in particular to the explicit treatment of solvents.[34,35]

One purpose of the present study is to demonstrate that the FMO method can make effective use of massively parallel computers that approach the petascale (i.e., the effective use of $\sim$100 000 or more compute cores) in both speed and resources. Second, this capability will facilitate future applications for the study of large, complex chemical problems that might otherwise be computationally intractable. Software development for massively parallel (i.e., peta and exascale) computers is frequently well behind the advances in hardware; therefore, efficient computational methods are needed to take advantage of the new computational architectures that are becoming available, as well as those that are anticipated in the near future. The FMO method discussed in the present work is one viable example of new high performance software in electronic structure theory.

## ■ METHODS AND COMPUTATIONAL DETAILS

The FMO method has been described extensively elsewhere;[6,7,36] therefore, only a brief description will be given here. The basic equation to obtain the total energy of the system divided into $N$ fragments is

$$E^{\mathrm{FMO2}} = \sum_I^N E'_I + \sum_{I>J}^N (E'_{IJ} - E'_I - E'_J) + \sum_{I>J}^N Tr(\Delta\mathbf{D}^{IJ}\mathbf{V}^{IJ}) \quad (1)$$

where $E'_I$ and $E'_{IJ}$ are the internal energies of fragments $I$ (monomers) and their pairs $IJ$ (dimers), polarized by the ESP field of all other fragments determined self-consistently. $\Delta\mathbf{D}^{IJ}$ is the difference between the density of the dimer $IJ$ and the sum of the densities of the monomers $I$ and $J$; $V^{IJ}$ is the ESP for dimer $IJ$. The FMO2 level of theory includes the explicit pair corrections shown in the second and third terms of eq 1, while FMO1 corresponds to the sum over monomers in the first term of the equation. The gradient is obtained by taking the fully analytic derivative of the FMO energy, recently developed by Nagata et al.[37,38] The FMO1 method, which was used for scalability tests in the previous papers, gives the internal energies of the fragments in the presence of the fields of all other fragments and also describes the many-body polarization, as demonstrated by the pair interaction energy decomposition analysis (PIEDA).[39] The individual fragment polarization energies $E_I^{PL}$ can be calculated as

$$\Delta E_I^{PL} = -(E'_I - E_I^0) \quad (2)$$

where $E_I^0$ is the internal energy of fragment $I$ (i.e., the energy computed without the field of the other fragments). All of the energies $E_I^0$ can be calculated in a single run and used to estimate the many-body polarization in large systems.

The key to achieving high parallel efficiency is the multilevel hierarchical approach, GDDI, that has so far been limited to two levels.[24,40] Specifically, computer nodes are divided into groups, and each group is assigned a particular fragment or fragment pair calculation to perform. The calculations within a group are performed without communication to other groups. However, at several points during the calculation some communications are required. The most important of these communications is to exchange fragment densities and to accumulate a total property, such as the energy. All workload balancing is done dynamically, at both the inter- and intragroup levels.

In general, the number of groups is chosen to balance the losses due to synchronization. For example, when a group finishes early and has to wait for the others to finish, having fewer groups is more efficient. On the other hand, when fewer nodes are assigned to a group, the parallelization within a group (e.g., an RHF or MP2 calculation of a given fragment or dimer) is more efficient, so having more groups is preferred. It has been shown previously[24] how varying the group size affects the synchronization and data exchange timings, as well as the general performance. For water clusters with a uniform fragment size, the load balancing is simpler than for proteins,[24] for which other techniques such as doing the larger jobs first[24] and employing semidynamic load balancing[40] (i.e., static load balancing on large GDDI groups for a few large fragments and dynamic load balancing on small GDDI groups for the rest) have been found to be necessary. Because the number of tasks is different for monomers vs dimers, a different grouping strategy is often employed for these two different components of a calculation, with the regrouping performed at the end of the monomer step. Guidelines for the grouping of CPU cores have been given elsewhere.[9] In general, it is recommended that each

group does several calculations for more even load balancing, especially when fragments are different sizes. As an example, consider FMO2 calculations on 4096 water molecules at the RHF/6-31G(d) level of theory, with each fragment defined to be two water molecules. This means there are 2 098 128 dimers. Most of these dimers are far enough apart to be treated using the ESP approximation, but 24 411 dimers must still be treated with quantum mechanics. If one uses 2048 groups for the dimer calculations, there will be ~12 dimers/group. The total wall clock time for this set of calculations is 11.5 min. Doubling the number of dimer groups to 4096 increases the wall clock time to 14.6 min. So, there is clearly a dependence on the number of groups that is used in each step of a calculation. In the present work, the default FMO options are used, except that all ESPs are computed using the one-electron approximation.

Prior to this work, FMO calculations were performed on computer clusters with dozens, or at most hundreds, of CPU cores with local disks attached to each node. In such cases, the I/O required to store and access the fragment densities is negligible compared to the total times. However, on supercomputers containing thousands to tens of thousands of cores, the I/O constitutes a major bottleneck. Due to recent multicore CPU development, the CPU core count continues to increase while the number and efficiency of storage devices typically lags far behind. Additionally, many modern massively parallel computers such as the K computer[41] or the Blue Gene/P computer[42] have no local storage. The I/O system that is usually provided on such computers generally does not meet the I/O demands in the FMO production code, due to the required access to fragment densities in order to calculate ESPs.

In previous FMO implementations, the master process of each group created a direct-access file in which the densities of all fragments are stored. The new approach, employed here, is based on a large array containing all fragment densities created in shared memory distributed among nodes. The standard DDI functionality described elsewhere is used.[23] The fragment densities are stored on data servers and sent on demand to compute nodes directly, with these communications sometimes involving intergroup operations. In addition, the RAMDISK feature on the Blue Gene/P was found to be effective in reducing I/O to hard disks by causing scratch files to be written to the memory of the nearest I/O node, thereby greatly improving performance. The main difference between the density file and other scratch files is that the latter are local to each group calculation and need not be made accessible to other groups, while the former must be made either fully global or synchronized between groups at some points. The underlying MP2 gradient calculations also use DDI memory.[22,43]

As an example of the consequent gain in efficiency, consider 1024 water molecules whose energy was calculated using FMO2 with MP2 and the 6-31G(d,p) basis set, both with the previous disk-based implementation ("FMOd") and the new implementation ("FMOm"). Each calculation was run on 1024 nodes (4096 cores) on a Blue Gene/P computer. The wall time required for the FMOd calculation was 335.4 min, whereas the corresponding FMOm wall time was 10.7 min. This is a dramatic improvement in efficiency. This 31-fold speed-up demonstrates that the DDI-based density storage is paramount to running FMO calculations effectively on large-scale parallel computers.

The hardware platform used for all calculations was the Blue Gene/P computer (*Intrepid*) at the Argonne Leadership Computing Facility (ALCF). A Blue Gene/P node consists of a quad-core

**Table 1. The Performance of FMO2-MP2 Force Calculations on the Blue Gene/P[a]**

|  |  | racks: | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|---|---|
|  |  | cores: | 4096 | 8192 | 16 384 | 32 768 | 65 536 | 131 072 |
| waters | atoms | basis functions | wall time (min) | | | | | |
| 128 | 384 | 2432 | 0.5 | 0.4 |  |  |  |  |
| 256 | 768 | 4864 | 1.1 | 0.7 | 0.5 |  |  |  |
| 512 | 1536 | 9728 | 3.6 | 2.0 | 1.2 | 0.8 | 0.7 |  |
| 1024 | 3072 | 19 456 | 10.7 | 6.3 | 3.4 | 2.1 | 1.5 | 1.2 |
| 2048 | 6144 | 38 912 | 34.6 | 18.5 | 11.1 | 6.1 | 3.7 | 2.6 |

[a] The atomic basis set is 6-31G(d).

**Table 2. The Performance of FMO2-MP2 Force Calculations on the Blue Gene/P[a]**

|  |  | racks: | 1 | 2 | 4 | 8 | 16 | 32 |
|---|---|---|---|---|---|---|---|---|
|  |  | cores: | 4096 | 8192 | 16 384 | 32 768 | 65 536 | 131 072 |
| waters | atoms | basis functions | wall time (min) | | | | | |
| 128 | 384 | 5504 | 8.6 | 4.8 | 2.7 | 1.8 |  |  |
| 256 | 768 | 11 008 | 19.8 | 10.5 | 5.8 | 3.4 | 2.2 |  |
| 512 | 1536 | 22 016 |  | 28.9 | 15.4 | 8.6 | 4.9 | 3.2 |
| 1024 | 3072 | 44 032 |  |  | 41.1 | 22.0 | 12.2 | 7.1 |

[a] The atomic basis set is aug-cc-pVDZ.

PPC450 chip, running at 850 MHz, with 2 GB of DRAM. Each board contains 32 compute nodes and has two dedicated I/O nodes to handle access to disk. The relatively low clock rate and node memory combine to give the architecture a very desirable FLOP/Watt ratio of 360 MFLOPS/Watt. Nodes are linked via three networks, one for collective communications with a bandwidth of 6.8 GB/s, a point-to-point interconnect of 3D torus topology with a bandwidth of 3.4 GB/s, and a 10 GB/s Ethernet link for I/O. Intrepid has 40 "racks" of 1024 nodes each, giving it a total of 163 840 processor cores.

### ■ RESULTS

Water clusters were used as the test systems for the benchmarks that are presented here, as they permit a convenient and systematic series of problem sizes. The following procedure was used to construct the water clusters used in this work. First, the oxygen atoms were fixed to grid points with an increment equivalent to the O—O separation of a typical hydrogen bond, taken to be 2.98 Å. The hydrogen atoms were fixed at the MP2/aug-cc-pVDZ O—H bond length and angle, matching the intended level of theory that will be subsequently applied to the entire cluster. The directional orientation of this fixed geometry was chosen at random. Water molecules were first arranged in cubes containing eight waters. This arrangement yields fragments containing 1, 2, 4, and 8 waters in a "droplet" configuration rather than a "chain", since the former is more favorable to the convergence of both the local and global FMO-RHF equations, and since bulk water contains primarily droplet-like clusters. A sufficient number of cubes is then created in order to yield the desired "slab" of water, itself as close to cubic as possible.

To survey a range of cluster sizes, calculations were performed on clusters containing 128, 256, 512, 1024, 2048, and 4096 waters. To assess scalability across a range of processor partitions, 1, 2, 4, 8, 16, and 32 racks of Intrepid were used. In all calculations, the FMO series is taken to second-order (FMO2),

sufficient for kcal/mol agreement of the energies with full calculations. The choice of two water molecules per fragment was found to give kcal/mol accuracy as well as the best overall performance, scalability, and execution time. The number of processor groups was chosen to be equal to the number of fragments. Larger choices for the number of groups, as might be done in an attempt to align that number with the number of so-called "self-consistent" dimers, were found to adversely increase the communication overhead and give a sharp decline in performance. The level of theory was chosen to be MP2, because (despite its limitations) MP2 has been found to give results in good agreement with coupled cluster, CCSD(T), calculations for water clusters.[44]

Single-point energies and gradients were computed for all clusters. The gradients will be used in future dynamics simulations. Two series of calculations were performed spanning the cluster sizes and processor partitions noted above: the first series uses the 6-31G(d) atomic basis set[45] (see Table 1); the second series uses the aug-cc-pVDZ basis set[46] (see Table 2). The next basis set in the "Dunning" series, aug-cc-pVTZ,[46] is more than twice as large (105 Cartesian basis functions for water) as the aug-cc-pVDZ basis set (43 Cartesian functions), so only a short-range of small clusters could be computed in a reasonable time with the current hardware. Calculations in a given partition series (row) of the tables begin below 1 h in duration and continue to either the point of "diminishing returns" regarding scalability or the maximum partition, whichever comes first. The scalability of the largest calculation, that of 4096 waters (having 12 288 atoms) at the MP2/6-31G(d) level of theory, is depicted in Figure 1, where it may be seen that the computations scale rather well with a system size up to the full complement of more than 131 000 cores.

The scaling of the cost of the calculation with problem size may be assessed by examining the columns in Tables 1 and 2. In Table 1, with the smaller basis set, doubling the size of the
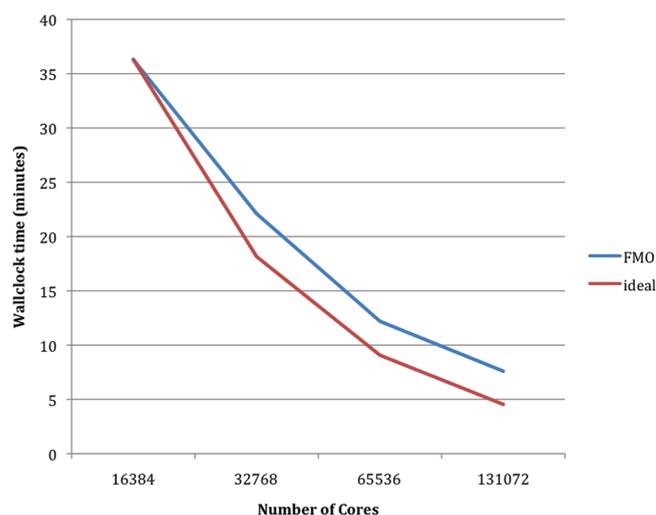
**Figure 1.** FMO2-MP2/6-31G(d) forces calculation of 12 288 atoms on Blue Gene/P.

system increases the cost by roughly a factor of 3. For the larger basis set, the corresponding factor is a bit smaller, between 2 and 3. This is not as good as linear scaling, but considerably better than quadratic scaling. The deviation from the linear regime is mainly caused by the increasing fraction of the far separated dimers computed with the electrostatic approximation. As the cluster size increases, the parallel scalability with the number of cores improves, as one would expect. The aug-cc-pVDZ calculations are on the order of 10 times more expensive than the corresponding 6-31G(d) calculations. This is consistent with the basis set roughly doubling in size on going from the smaller (Table 1) to the larger (Table 2) basis set, combined with the cubic increase in the fragment cost with the number of basis functions. Of great interest with regard to the use of the FMO method are the prospects for *ab initio* MP2 molecular dynamics (MD) simulations of bulk water and other fluids. The approximate linear scaling exhibited in the tables makes such calculations feasible. The present benchmarks provide reference points for the estimation of costs within the current hardware constraints. For example, MD step times on the order of 1 min for clusters 512−1024 water molecules are highly encouraging.

## ■ SUMMARY

This work has demonstrated that the FMO code in GAMESS can efficiently utilize "petascale" processor counts. Specifically, the GAMESS/FMO method can use up to 131 072 processor cores of a Blue Gene/P supercomputer to obtain the MP2 atomic forces for a system with more than 4000 atoms, with the 6-31G (d) basis set, in approximately 7 min. Of course, the Blue Gene/P is not the only supercomputer architecture capable of achieving petascale computing. However, the group division of processors chosen for this work is unique to the Blue Gene/P architecture due to the great number of variables required for optimal efficiency during electronic structure calculations. These variables include fragment size (number of basis functions per fragment and homogeneity of fragment size), level of theory (e.g., HF, MP2 etc), and most importantly the computer architecture being used. While the first two variables can affect the accuracy of the FMO calculation, the last variable can also affect these choices due to computational resources available.

Likewise, depending on the level of theory, memory requirements may necessitate larger GDDI groups.

The number of cores per node, amount of memory per core, network speed, and CPU clock rate as well as the choice of communication layer (e.g, MPI, ARMCI) can all affect how the user chooses the GDDI group division. The relatively small node size (four cores per node) of the Blue Gene/P architecture makes it particularly flexible in terms of the GDDI group size. However, other architectures, such as the Cray XE6 and SGI Altix, typically consist of 12−16 cores per node with significantly higher clock rates. These numbers are only likely to increase with advances in microprocessor technology, leading to an increase in the minimum GDDI group size. The obvious solution to this issue is to simply increase the fragment size in order to maintain a high level of computational efficiency, with the byproduct being an increase in the accuracy of the FMO calculation. This exemplifies the importance of GDDI group choice by showing the importance of proper group size for the appropriate fragment size.

Problem sizes in the thousands-of-atoms range represent a "critical mass" in the applicability of electronic structure theory to chemistry at which many issues of national strategic importance (for instance, renewable energy and medicine), including systems on the biological scale (e.g., proteins), become accessible with predictive capability and detailed understanding. The FMO method is clearly able to address such problems with both efficiency and accuracy. Further improvements in performance will necessitate addressing several remaining bottlenecks. Among these are load balancing, the communications (e.g., input/output) overhead, and the use of outdated programming paradigms, especially in legacy codes. All of these issues are being addressed by the authors and colleagues and will continue to be addressed in the future.

## ■ AUTHOR INFORMATION

**Corresponding Author**
*E-mail: mark@si.msg.chem.iastate.edu.

**Notes**
The authors declare no competing financial interest.

## ■ ACKNOWLEDGMENT

## ■ REFERENCES

(1) The Impact of Dennard's Scaling Theory. *IEEE Solid-State Circuits Society News*, **2007**, *12* (1).

78

dx.doi.org/10.1021/ct200548v |*J. Chem. Theory Comput.* 2012, 8, 75–79

(2) Moore, G. E. Cramming More Components Onto Integrated Circuits. *Electronics* **1965**, *38*, 114–117.

(3) Ishimura, K.; Kuramoto, K.; Ikuta, Y.; Hyodo, S. MPI/OpenMP Hybrid Parallel Algorithm for Hartree–Fock Calculations. *J. Chem. Theory Comput.* **2010**, *6*, 1075–1080.

(4) Aprà, E.; Harrison, R. J.; Shelton, W. A.; Tipparaju, V.; Vázquez-Mayagoitia, A. Computational chemistry at the petascale: are we there yet? *J. Phys. Conf. Ser.* **2009**, *180*, 012027.

(5) Kitaura, K.; Ikeo, E.; Asada, T.; Nakano, T.; Uebayasi, M. Fragment molecular orbital method: an approximate computational method for large molecules. *Chem. Phys. Lett.* **1999**, *313*, 701–706.

(6) Fedorov, D. G.; Kitaura, K. Extending the Power of Quantum Chemistry to Large Systems with the Fragment Molecular Orbital Method. *J. Phys. Chem. A* **2007**, *111*, 6904–6914.

(7) Fedorov, D. G.; Kitaura, K. *The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems*; CRC Press: Boca Raton, FL, 2009.

(8) Otto, P. Investigation of the stability of oligonucleotides and oligodinucleotides. *THEOCHEM* **1989**, *188*, 277–288.

(9) Gao, J. L. Toward a molecular orbital derived empirical potential for liquid simulations. *J. Phys. Chem. B* **1997**, *101*, 657–663.

(10) Sorkin, A.; Dahlke, E. E.; Truhlar, D. G. Application of the Electrostatically Embedded Many-Body Expansion to Microsolvation of Ammonia in Water Clusters. *J. Chem. Theory Comput.* **2008**, *4*, 683–688.

(11) Söderhjelm, P.; Ryde, U. How Accurate Can a Force Field Become? A Polarizable Multipole Model Combined with Fragment-wise Quantum-Mechanical Calculations. *J. Phys. Chem. A* **2009**, *113*, 617–627.

(12) Gordon, M. S.; Mullin, J. M.; Pruitt, S. R.; Roskop, L. B.; Slipchenko, L. V.; Boatz, J. A. Accurate Methods for Large Molecular Systems. *J. Phys. Chem. B* **2009**, *113*, 9646–9663.

(13) Mata, R. A.; Stoll, H.; Cabral, B. J. C. A Simple One-Body Approach to the Calculation of the First Electronic Absorption Band of Water. *J. Chem. Theory Comput.* **2009**, *5*, 1829–1837.

(14) Weiss, S. N.; Huang, L.; Massa, L. A generalized higher order kernel energy approximation method. *J. Comput. Chem.* **2010**, *31*, 2889–2899.

(15) Hua, S. G.; Hua, W. J.; Li, S. H. An Efficient Implementation of the Generalized Energy-Based Fragmentation Approach for General Large Molecules. *J. Phys. Chem. A* **2010**, *114*, 8126–8134.

(16) Řezáč, J.; Salahub, D. R. Multilevel Fragment-Based Approach (MFBA): A Novel Hybrid Computational Method for the Study of Large Molecules. *J. Chem. Theory Comput.* **2010**, *6*, 91–99.

(17) Yeole, S. D.; Gadre, S. R. On the applicability of fragmentation methods to conjugated $\pi$ systems within density functional framework. *J. Chem. Phys.* **2010**, *132*, 094102.

(18) He, X.; Merz, K. M., Jr. Divide and Conquer Hartree–Fock Calculations on Proteins. *J. Chem. Theory Comput.* **2010**, *6*, 405–411.

(19) Kobayashi, M.; Kunisada, T.; Akama, T.; Sakura, D.; Nakai, H. Reconsidering an analytical gradient expression within a divide-and-conquer self-consistent field approach: Exact formula and its approximate treatment. *J. Chem. Phys.* **2011**, *134*, 034105.

(20) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. General atomic and molecular electronic structure system. *J. Comput. Chem.* **1993**, *14*, 1347–1363. http://www.msg.ameslab.gov/gamess/index.html (accessed Dec. 2011).

(21) Gordon, M. S.; Schmidt, M. W. Advances in electronic structure theory: GAMESS a decade later. In *Theory and Applications of Computational Chemistry, the first forty years*; Elsevier: Amsterdam, 2005; pp 1167–1189.

(22) Fletcher, G. D.; Gordon, M. S.; Schmidt, M. W. Developments in Parallel Electronic Structure Theory. *Adv. Chem. Phys.* **1999**, *110*, 267.

(23) Fletcher, G. D.; Schmidt, M. W.; Bode, B. M.; Gordon, M. S. The Distributed Data Interface in GAMESS. *Comput. Phys. Commun.* **2000**, *128*, 190–200.

(24) Fedorov, D. G.; Olson, R. M.; Kitaura, K.; Gordon, M. S.; Koseki, S. A new hierarchical parallelization scheme: generalized distributed data interface (GDDI), and an application to the fragment molecular orbital method (FMO). *J. Comput. Chem.* **2004**, *25*, 872–880.

(25) He, X.; Fusti-Molnar, L.; Cui, G.; Merz, K. M., Jr. Importance of dispersion and electron correlation in ab initio protein folding. *J. Phys. Chem. B* **2009**, *113*, 5290–5300.

(26) Sawada, T.; Fedorov, D. G.; Kitaura, K. Role of the key mutation in the selective binding of avian and human influenza hemagglutinin to sialosides revealed by quantum-mechanical calculations. *J. Am. Chem. Soc.* **2010**, *132*, 16862–16872.

(27) Fujimura, K.; Sasabuchi, Y. The Role of Fluorine Atoms in a Fluorinated Prostaglandin Agonist. *Chem. Med. Chem.* **2010**, *5*, 1254–1257.

(28) Ohno, K.; Mori, K.; Orita, M.; Takeuchi, M. Computational Insights into Binding of Bisphosphates to Farnesyl Pyrophosphate Synthase. *Curr. Med. Chem.* **2011**, *18*, 220–233.

(29) Fedorov, D. G.; Alexeev, Y.; Kitaura, K. Geometry Optimization of the Active Site of a Large System with the Fragment Molecular Orbital Method. *J. Phys. Chem. Lett.* **2011**, *2*, 282–288.

(30) Mazanetz, M. P.; Ichihara, O.; Law, R. J.; Whittaker, M. Prediction of cyclin-dependent kinase 2 inhibitor potency using the fragment molecular orbital method. *J. Cheminformatics* **2011**, *3*, 2.

(31) Sawada, T.; Fedorov, D. G.; Kitaura, K. Structural and interaction analysis of helical heparin oligosaccharides with the fragment molecular orbital method. *Int. J. Quantum Chem.* **2009**, *109*, 2033–2045.

(32) Fedorov, D. G.; Jensen, J. H.; Deka, R. C.; Kitaura, K. Covalent Bond Fragmentation Suitable To Describe Solids in the Fragment Molecular Orbital Method. *J. Phys. Chem. A* **2008**, *112*, 11808–11816.

(33) Fedorov, D. G.; Avramov, P. V.; Jensen, J. H.; Kitaura, K. Analytic gradient for the adaptive frozen orbital bond detachment in the fragment molecular orbital method. *Chem. Phys. Lett.* **2009**, *477*, 169–175.

(34) Sato, M.; Yamataka, H.; Komeiji, Y.; Mochizuki, Y.; Ishikawa, T.; Nakano, T. How Does an $S_N2$ Reaction Take Place in Solution? Full Ab Initio MD Simulations for the Hydrolysis of the Methyl Diazonium Ion. *J. Am. Chem. Soc.* **2008**, *130*, 2396–2397.

(35) Kistler, K. A.; Matsika, S. Solvatochromic Shifts of Uracil and Cytosine Using a Combined Multireference Configuration Interaction/Molecular Dynamics Approach and the Fragment Molecular Orbital Method. *J. Phys. Chem. A* **2009**, *113*, 12396–12403.

(36) Nagata, T.; Fedorov, D. G.; Kitaura, K. In *Linear-Scaling Techniques in Computational Chemistry and Physics*; Zalesny, R., Papadopoulos, M. G., Mezey, P. G., Leszczynski, J., Eds.; Springer: Berlin, 2011; Chapter 2.

(37) Nagata, T.; Fedorov, D. G.; Kitaura, K. Derivatives of the approximated electrostatic potentials in the fragment molecular orbital method. *Chem. Phys. Lett.* **2009**, *475*, 124–131.

(38) Nagata, T.; Fedorov, D. G.; Kitaura, K. Importance of the hybrid orbital operator derivative term for the energy gradient in the fragment molecular orbital method. *Chem. Phys. Lett.* **2010**, *492*, 302–308.

(39) Fedorov, D. G.; Kitaura, K. Pair interaction energy decomposition analysis. *J. Comput. Chem.* **2007**, *28*, 222–237.

(40) Ikegami, T.; Ishida, T.; Fedorov, D. G.; Kitaura, K.; Inadomi, Y.; Umeda, H.; Yokokawa, M.; Sekiguchi, S. Full Electron Calculation Beyond 20,000 Atoms: Ground Electronic State of Photosynthetic Proteins. *Proc. of Supercomputing 2005*; IEEE Computer Society: Seattle, 2005.

(41) See, for example, http://www.fujitsu.com/global/news/pr/archives/month/2011/20110620-02.html.

(42) See, for example, http://en.wikipedia.org/wiki/Blue_Gene#Blue_Gene.2FP (accessed Dec. 2011).

(43) Fletcher, G. D.; Rendell, A. P.; Sherwood, P. A Parallel Second-Order Møller-Plesset Gradient. *Mol. Phys.* **1997**, *91*, 431–438.

(44) Xantheas, S. S.; Burnham, C. J.; Harrison, R. J. *J. Chem. Phys.* **2002**, *116*, 1493–1499. Olson, R. M.; Bentz, J. L.; Kendall, R. A.; Schmidt, M. W.; Gordon, M. S. *J. Comput. Theor. Chem.* **2007**, *3*, 1312–1328.

(45) Hehre, W. J.; Ditchfield, R.; Pople, J. A. *J. Chem. Phys.* **1972**, *56*, 2257–2261.

(46) Dunning, T. H., Jr. *J. Chem. Phys.* **1989**, *90*, 1007–1023.

79

dx.doi.org/10.1021/ct200548v |*J. Chem. Theory Comput.* 2012, 8, 75–79