Hybrid Quantum Mechanical/Molecular Mechanical (QM/MM) Studies of Enzyme Reaction Mechanisms: Building the Computational Engine

Alistair Rendell, Andrey Bliznyuk, Thomas Huber and Ajay Limaye
The Supercomputer Facility
The Australian National University
Canberra ACT 0200
AUSTRALIA

Email: Alistair.Rendell@anu.edu.au
Phone: 61-2-6249 3436
Fax: 61-2-6279 8199

and

Jill Gready, Stephen Greatbanks, and Rebecca Schmidt
The John Curtin School of Medical Research
The Australian National University
Canberra ACT 0200
AUSTRALIA

Email: Jill.Gready@anu.edu.au
Phone: 61-2-6279 8304
Fax: 61-2-6249 0415

Abstract- *One computational technique that is becoming increasingly important for the study of large chemical and biological systems is the so-called hybrid method. In this method a large system is divided into regions that are treated using different levels of theory. The different levels of theory used in hybrid calculations are often only available through the use of different codes. Furthermore, given the size of systems being studied it is important that each of these different codes is well optimised for the computational platform available. In this paper we describe some of our efforts to enhance the performance of Gaussian 94, Mopac93 and AMBER 5 for the Fujitsu VPP300 and AP3000 supercomputers, and the use of these packages and platforms in hybrid studies of enzyme reaction mechanisms.*

Key words: QM/MM, Enzyme reaction, Fujitsu Supercomputers, Computational Chemistry
Areas of Interest: Computational Chemistry and Life Sciences.

1. Introduction

Computer simulations have provided substantial insight into the physical behaviour of small molecules. An important goal of computational chemistry is to extend these techniques to larger systems that are more typical of naturally occurring biomolecules. Hybrid quantum mechanical and molecular mechanical (QM/MM) calculations represent one means of achieving this goal. In these methods, the central region of interest is modelled using quantum mechanics, while the environmental effects of the surrounding solvent, protein, etc are modelled using the less time consuming molecular mechanics (MM) method.

The QM/MM method can be used to address a number of critical research questions, many of which cannot be answered by experiment. Examples of potential applications include:

- characterization of the relative contribution of polarization effects to solvation energy

- calculation of the binding energies and optimal geometries of ion/ion, ion/molecule and molecule/molecule pairs (including hydrogen bonds)
- exploration of energetic differences among conformers, such as rotational energies about a dihedral angle
- calculation of proton affinities, deprotonation enthalpies, ionization potentials, dipole moments, and atomic charges
- investigation of the mechanism of a chemical reaction in solution or in the active site of an enzyme (relative energies of reactants, products, and transition states; rationalization of differing reactivities of a series of substrates; influence of different solvents or enzyme amino acids; optimization of substrate geometry; individual influence of each enzyme residue)
- characterization of photochemical reactions and electron transfer processes in solution and in enzymes
- determination of solvent effects on excited states and electronic spectroscopy
- energy minimizations, molecular dynamics (MD) or Monte Carlo (MC) simulations using QM/MM potentials

The first QM/MM implementation was reported by Levitt and Warshel in 1976, and was used to investigate the active site of the enzyme lysozyme[1]. In the QM region Levitt and Warshel used a semiempirical quantum chemical method (based on MNDO/2), while the MM region was represented by a polarizable force field for the protein and by a Langevin Dipoles (LD) method for the solvent. Although this paper first appeared over 20 years ago, hybrid methods did not gain widespread use until computer speeds increased. This is partcularly true for QM/MM approaches that use *ab initio* methods in the QM region. Thus while Singh and Kollman developed the hybrid QUEST program by combining *Gaussian 80* with AMBER [2] in 1986, the method did not include polarisation of the QM region by the MM and was used only to perform geometry optimisations on systems with a comparatively small QM region.

Today *ab initio* QM/MM calculations are still performed only when optimising geometries and for relatively small QM regions. For semiempirical QM/MM, the situation is more advanced with a number of calculations being reported that used quite large QM regions [3] and, in some cases, complex dynamic simulations have been undertaken [4]. A brief summary of the history of QM/MM methods is given in Table 1, while for a detailed review of the QM/MM method in general the reader is referred to the review articles by Aqvist and Warshel [5] and Gao [6].

Our interest in QM/MM is motivated by the desire to study enzyme reaction mechanisms. The approach is attractive because a quantum method is required to correctly describe the bond breaking and forming processes that occur in enzyme reactions, but the sheer size of enzymes currently makes the exclusive use of QM methods impractical. Furthermore, while model studies on small fragment systems may be an alternative strategy for the study of some large systems, for enzymes the environment is usually of critical importance.

As indicated above, the basic requirements of the QM/MM method is a QM code, an MM code, and some "glue" to join the two together. In practice the QM code could be an *ab initio* code, a density functional code, or a semiempirical code, while the MM code could be based on one of a number of different force fields. Our current work uses either *Gaussian 94* [26] or Mopac [27] as the QM code and AMBER [28] as the MM code. For the glue we are currently using ChemShell [24]; this is a Tcl/Tk based "chemistry shell" that has been developed by Sherwood, specifically to enable QM/MM calculations to be performed by linking together third-party computational chemistry codes.

Even within the framework of the QM/MM model the computational cost of the calculations to be performed quickly becomes very large for enzymes. For example our current work is focused on the enzyme dihydrofolate reductase (DHFR), a system that contains approximately three thousand atoms. This enzyme is important in the biosynthesis of cellular DNA, and inhibition of its reaction mechanism has been exploited by several classes of cytotoxic drugs for treating diseases such as cancer. DHFR catalyses the reduction of folate to dihydrofolate and then to tetrahydrofolate in the presence of a nicotinamide dinucleotide phosphate (NADPH) cofactor (figure 1). At present several details of the catalytic mechanism of this reaction are in doubt despite many studies, both experimental and theoretical. Our current work aims to treat the reaction centre (i.e. the folate and NADPH) using QM methods while the bulk enzyme is modelled using the MM method. In Table 2 we show the number of atoms in the substrate and cofactor and the number of basis functions that are used for a variety of basis sets if these molecules are treated using an *ab initio* QM method.

Table 1: A potted history of QM/MM implementations

Year	Research Group	Program	Comments
1976	Warshel	QCFF/ALL (MINDO2/MM)[1]	first semiempirical QM; includes MM polarizability
1986	Kollman	QUEST (G80UCSF/AMBER)[2]	first coupled potential using *ab initio* QM
1989	Warshel	MOLARIS (EVB/MM)[7]	empirical valence bond
1990	Karplus	MOPAC/CHARMM[8]	semiempirical QM
1992	Gao	MCQUB (MOPAC/BOSS)[9]	semiempirical QM with Monte Carlo sampling
1992	Kim	EVB/MM[10]	empirical valence bond with dielectric continuum solvent
1992	Warshel	MOLARIS (MNDO/LD or SCAAS)[11]	refinement of early MO implementation
1993	Merz	deMon/AMBER[12]	density functional theory
1994	Gready	MOPS (MOPAC/AMBER)[13]	semiempirical QM
1994	Hypercube, Inc.	Hyperchem[14]	limited capabilities
1994	Thompson	ARGUS (AM1/SPCE)[15]	torsional term later included in AM1 for flexible molecules[16]
1994	Vasilyev	MNDO/OPLS[17]	semiempirical QM
1995	Liu	MOPAC/GROMOS[18]	semiempirical QM
1995	Merz	GAUSSIAN92/AMBER[19]	*ab initio* QM
1995	Merz	MOPAC/AMBER[20]	semiempirical QM
1995	Morokuma	GAUSSIAN92/MM2[21]	*ab initio* QM
1995	Thompson	ARGUS (QM/MMpol)[7]	Semiempirical QM with polarizable MM region
1995	Warshel	MOLARIS (GAMESS/SCAAS)[22]	*ab initio* QM
1996	Gao	3-21G/BOSS[23]	*ab initio* QM
1996	Sherwood	ChemShell[24]	generic QM/MM interface
1996	Thiel	MNDO/MM3[25]	semiempirical QM with polarizable MM region

Figure 1: Folate, dihydrofolate, tetrahydrofolate and NADPH

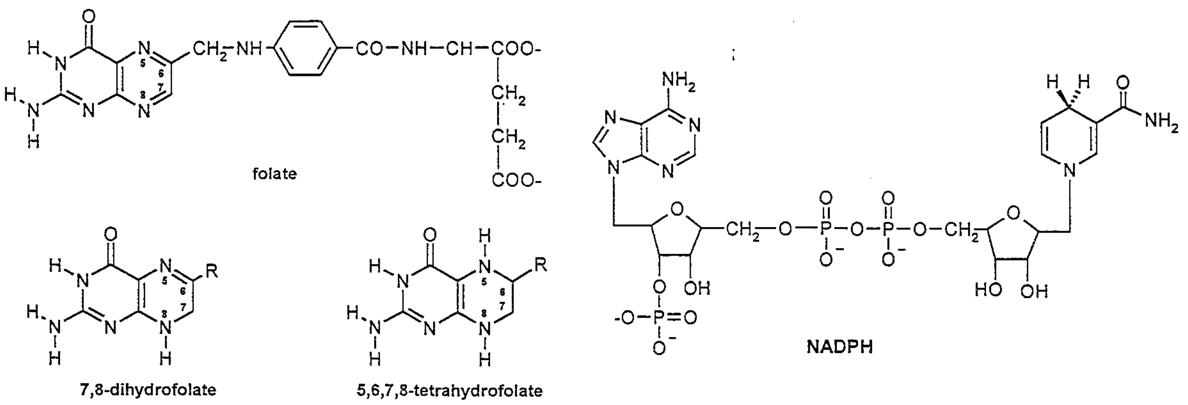

Table 2: Number of functions required to treat folate and NADPH using a variety of different basis sets

System	Atoms	Basis set	Basis Functions
Folate	49	3-21G	322
		6-31G*	514
		6-31+G*	642
NADPH	74	3-21G	496
		6-31G*	784
		6-31+G*	976

Clearly from Table 2 a prerequisite for our QM/MM studies is to have well tuned versions of the core *ab initio*, semiempirical, and force field codes on our intended computer platforms - the Fujitsu VPP300 vector parallel and the Fujitsu AP3000 scalar parallel supercomputers. In the following three sections we briefly outline some of our efforts to enhance the performance of *Gaussian 94*, Mopac, and AMBER 5.0 on these platforms. First, however, we note the key characteristics of the Fujitsu VPP300 and AP3000 supercomputers.

2. The Fujitsu VPP300 and AP3000 Parallel Supercomputers

The VPP300 at the Australian National University is from the current range of vector parallel supercomputers manufactured by Fujitsu. Each processor of the VPP is a vector processor with a clock speed of 7nsec and a peak performance of 2.2Gflops (a slightly enhanced 6.5nsec/2.4Gflops VPP-E model is now available). Peak performance is obtained by having vector pipes that can perform 8 operations per clock cycle, and by chaining the multiply and addition pipes together. Each processor can have 0.5, 1.0 or 2GBytes of local memory, and is linked to other processors by a crossbar switch supporting bidirectional communications at 570Mbytes/sec (614Mbytes/sec on the VPP-E). The VPP range can be configured with between 1 and 512 processors. By using advanced CMOS technology the cost-performance and physical characteristics of the machine are much improved compared with the previous VPP500 generation. Further information is given in ref. 29.

The Fujitsu AP3000 is a distributed memory parallel server consisting of multiple workstations connected via a high-speed communication network - the AP-Net. Each node uses the Sun UltraSPARC processor and runs the Solaris operating system. Since each node is effectively a standalone workstation, the latest UltraSPARC processor can readily be incorporated into the system, consequently the series has seen a steady migration from using 140MHz, to 167MHz, to 200MHz etc processors. The AP-Net has a peak performance of 200MBytes/sec bidirectional. Further information is given in ref. 30.

3. Enhancing the Performance of QM Codes on Vector Processors

In direct self-consistent field (SCF) and density functional (DFT) calculations the bulk of the computational time is spent computing integrals, multiplying these by density matrix elements and then summing the result into the so called Fock matrix. While many of these operations can be vectorised and made to run very well on vector processors, an exception is the final scatter-add of contributions to the Fock matrix. This operation is illustrated in eqn.1 and does not vectorise as it is genuinely recursive.

$$F(index(i))=F(index(i))+X(i) \qquad (1)$$

In eqn.1 F is the Fock matrix, X is the contribution to be added and *index* is some arbitrary index vector. The dimension of the Fock matrix scales as the number of basis functions squared, and for our calculations it is real symmetric so only the lower half is stored. Thus, for NADPH using a 6-31G* basis set there are 784 functions (Table 2) resulting in a Fock matrix of size 784*(784+1)/2, taking up 2.4 Mbytes of memory. The number of contributions to the Fock matrix varies from iteration to iteration in the direct SCF (or DFT) procedure, but in this case there are typically 60-70 million elements per iteration which are computed in batches of a few hundred thousand elements of X per batch. On the VPP300 this scatter-add operation was found to take roughly 70% of the total computation time for an SCF calculation on folate or NADPH. Given the dominance of this process we have devoted some time to investigating strategies for vectorising this process. In particular, two approaches which remove the recursion and enable vectorisation have been investigated, (i) use of replicated Fock matrices, and (ii) presorting of the contributions.

3.1 Replicated Fock Matrices

In this approach additional memory is used to hold multiple copies of the Fock matrix. Successive elements of X are then scattered into the different Fock matrices in a round robin fashion. Since the memory associated with each Fock matrix is unique, the scatter-add into the different Fock matrices can be vectorised with a vector length given by the number of Fock matrices used. The disadvantage of this method is that the size of the Fock matrix can be quite large (eg 10Mbytes for our combined folate+NADPH calculations using the 6-31+G* basis set) and it may not be possible to store many copies, thus limiting the vector length. It should also be noted that this method requires that the multiple Fock matrices are first zeroed and then summed together to form the final "real" Fock matrix when all elements of *X* have been scattered. In practice, however, this is not a severe limitation since both operations are fully vectorised.

3.2 Presorting of Contributions

The recursion present in eqn.1 can be removed if we presort the elements of *X* and the corresponding *index* into buckets or bins, such that elements in the different bins are guaranteed not to point to the same element of the Fock matrix. Vectorisation can then occur by picking elements from different bins, with a vector length given by the number of bins. For example, the elements of *X* could be sorted into two bins, those for which the corresponding index vector points to an odd element of the Fock matrix and those that point to an even element. We can then take one element from the odd bin and one from the even bin and scatter these two elements in a vector operation that is not recursive, but only has a vector length of two. Obviously more than two bins are required!

Sorting of elements into bins based on the value of one of the bits of their binary representation can easily be vectorised, and forms the basis of a binary radix sort. In contrast to a full binary radix sort we only need to sort on as many bits as we want bins, specifically the number of bins is given by 2^n where n is the number of bits sorted. Since the number of bins is the vector length in the final scatter-add there is some advantage in maximising the number of bins. This advantage, however, must be weighed against the fact that each bit sort is associated with one pass through all the elements of *X/index* and, therefore, scales as the number of elements in *X/index*. In contrast with the replicated Fock scheme the sorting of *X* can be buffered, giving much more modest memory requirements.

3.3 Performance Comparison

To test the two schemes outlined above we have produced a small test code that scatters several million random elements into a matrix of varying dimensions. In Table 3 we present the comparative timings obtained using a standard scalar loop, using the replicated matrix scheme, and using the presorting method. The number of elements scattered is varied as is the dimension of the matrix being summed into. We also vary the number of replicated matrices and bins used. The results show both methods are faster than the original code, but clearly using replicated matrices is the best solution. Based on this result we have tested the use of a replicated Fock matrix scheme in the *Gaussian 94* program. The resulting performance as a function of the number of Fock matrices used in a test calculation on the valinomycin molecule with 789 basis functions is shown in Figure 2. This shows that with just 30 Fock matrices a substantial performance improvement is obtained.

Figure 2: Relative time for an SCF energy calculation with 789 basis functions obtained on the Fujitsu VPP300 as a function of the number of Fock matrices used.

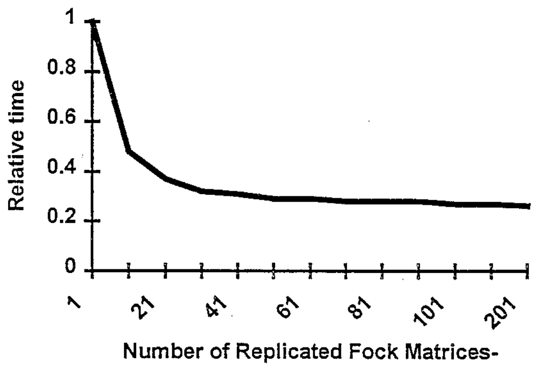

Table 3: Comparative times (sec) for different scatter-add algorithms obtained on the Fujitsu VPP300 using a small test code

		Original	———Replicated———				———Presort———			
Dimensions		Scalar	Number of Matrices				Number of Bins			
Fock	X	Code	8	32	128	512	8	32	128	512
10^5	10^7	3.8	1.8	0.5	0.3	0.3	2.2	1.2	1.2	1.8
10^5	10^8	37.9	16.8	4.3	1.5	1.0	21.6	11.5	12.3	17.9
10^6	10^8	38.8	17.9	5.4	2.7	-	20.1	11.5	11.9	-
10^6	10^9	388.4	167.8	43.4	15.0	-	201.4	119.0	123.1	-

4. Gradient evaluations in semiempirical calculations

The dominant step in performing semiempirical QM/MM simulations is the evaluation of the semiempirical energy and gradient (force on each atom). In recent work we have been investigating a novel approach for evaluating the gradient for semiempirical wavefunctions [31] that is two times faster than the standard implementation.

The standard method for evaluating the gradient of a semiempirical wavefunction is to use a finite difference approach, and although some attempts have been made to implement analytic formulae this has always proved too computationally expensive [32]. In Mopac, the numerical implementation effectively takes each pair of atoms in the molecule, evaluates the energy at the equilibrium separation, and then at geometries corresponding to a small displacement in the *x*, *y* or *z* Cartesian coordinates of one of the atoms. The resulting energies are then used in a one-sided finite difference expression to give a contribution to the total force on each atom. (In Mopac use of the *PRECISE* keyword results in the numerical differentiation being carried out using a two-sided finite difference scheme with displacements being made in both the positive and negative direction along each Cartesian axes).

The key to our new scheme is the observation that the basic numerical displacement originally performed in the *x*, *y* and *z* Cartesian directions can be replaced by a single displacement along the internuclear coordinate of the two atoms. (This is a consequence of the fact that for semiempirical methods based on the ZDO approximation, i.e. all methods in Mopac, the total energy can be calculated solely as a sum of at most two-centre interaction terms). Expressions for the energies at the Cartesian displaced geometries can then be approximated using a Taylor series expansion. The result is a reduction by a factor of two in the number of displaced computations required, so that instead of 4 evaluations at r_0, $r_0+\delta x$, $r_0+\delta y$, $r_0+\delta z$, only 2 at r_0 and $r_0+\delta r$ are needed.

Table 4: Time (sec) required for the gradient evaluation portion of Mopac on single 167MHz processor of the Fujitsu AP3000 using the original and new algorithm

Molecule	Number of Atoms	Original	New
Valinomycin	168	4.37	1.92
NADPH	78	1.06	0.44
Leu-enkephalin	77	1.24	0.55
Methotrexate	53	0.58	0.20

The new gradient scheme is being tested in Mopac. In Table 4 we show the computation time required on a single processor of the AP3000 (167MHz UltraSPARC) for the gradient evaluation portion of Mopac calculations performed on a variety of test molecules. The results clearly show the twofold decrease in the gradient evaluation time.

5. The AMBER molecular dynamics code on the Fujitsu machines

Our interest in AMBER is twofold, firstly as the MM part of a QM/MM calculation, and secondly for performing simulation calculations to obtain starting structures for use in QM/MM calculations. Indeed the computational efficiency of AMBER is far more crucial to the latter, as in most QM/MM calculations the time spent in the MM part of the calculation is irrelevant compared with that spent in the QM part.

Initial porting of AMBER to the AP3000 and VPP300 revealed somewhat disappointing performance on the VPP300. In Table 5 we present the times obtained for one of the standard AMBER 5 benchmarks on the two Fujitsu machines, an SGI Power Challenge also available locally, and some other data given on the AMBER 5 benchmarking WWW site (http://www.AMBER.ucsf.edu/amber/bench50.html). The benchmark calculation runs 100 time steps of molecular dynamics on plastocyanin in water, and contains a total of 11585 atoms. The calculation uses a non-bonded interaction cutoff of 12Å and updates the pairlist after every 20 time steps.

Table 5: AMBER 5 times for 100 time steps of MD on plastocyanin (CPU, excluding setup time) recorded on a single processor of various computers

Machine	Time(sec)
VPP300 (tuned)	65.9
VPP300 (original)	116.8
AP3000 167MHz	286.2
SGI-Power Challenge (195MHZ)	214.5
*Pentium Pro 200MHz	510.8
*Cray T90	72
*Cray C90	117
*DEC 5/440	224
*HP 8000 180 MHz	245

* time has been taken from AMBER 5 benchmarking www site

The performance bottleneck on the VPP300 was traced to two problems in the code:

- The computation of the pairlist, though occurring only every 20 time steps was taking a disproportionately large fraction of the total time
- The computation of the nonbonded interactions

For the pairlist the problem was a result of very short vector lengths associated with the number of atoms in a residue (typically 10). The code was modified to coalesce the problem loop with an outer loop that effectively defines a large number of residues. This resulted in much larger vector lengths, speeding up this routine by a factor of 5. For the nonbonded force interaction vector lengths were reasonable (200-500), but performance was comparatively poor. The problem here was due to memory bank conflicts that arise from repeated access to the same interaction potential parameters. Specifically, given two interacting atoms the code determines the type of atoms (carbon, hydrogen, oxygen etc) and then does a lookup of the force field parameters for these atom types. Naturally, there are many atoms of the same type in a simulation giving rise to a lookup that repeatedly accesses the same memory address. To remedy this problem we have chosen to store the parameters of the non-bonded pair interaction along with the pairlist. While this adds to the overall memory requirement, for most cases on the VPP300 memory is not a limiting problem. The performance of the resulting tuned code is also given in Table 5. The new code is nearly twice as fast as the original, and substantially faster on the VPP300 compared with the single 167MHz AP3000 processor.

6. Conclusions

The computational study of enzyme reaction mechanism presents many theoretical and computational challenges. Hybrid QM/MM methods offer one means for studying such systems. Even then, however, it is important that key computational chemistry packages are tuned for the computational platforms being used. In this paper we have outlined our efforts to enhance the performance of well known *ab initio*, semiempirical and molecular dynamics codes for the Fujitsu VPP300 and AP3000 supercomputers. In all cases substantial performance enhancements have been obtained.

7. Acknowledgments

This work was partially supported by Fujitsu Japan. Helpful discussions with Roger Brown and David Singleton are gratefully acknowledged.

8. References

1. A. Warshel and M. Levitt, *Theoretical Studies of Enzymic Reactions: Dielectric, Electrostatic, and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme.* J. Mol. Biol., 1976. **103**: p. 227.
2. U.C. Singh and P.A. Kollman, *A Combined Ab Initio Quantum Mechanical and Molecular Mechanical Method for Carrying Out Simulations on Complex Molecular Systems: Applications to the CH_3Cl + Cl- Exchange Reaction and Gas Phase Protonation of Polyethers.* J. Comput. Chem., 1986. 7: p. 718-730.
3. M.A. Thompson and G.K. Schenter, *Excited States of the Bacteriochlorophyll b Dimer of Rhodopseudomonas viridis: A QM/MM Study of the Photosynthetic Reaction Center That Includes MM Polarization.* JACS, 1995. **99**: p. 6374-6386.
4. P.L. Cummins and J.E. Gready *A molecular dynamics and free energyperturbation (MD/FEP) study of the hydride-ion transfer step in dihydrofolate reductase using a combined quantum and molecular mechanical (QM/MM) model.* J. Comput. Chem., 1998, in press.
5. J. Aqvist and A. Warshel, *Simulation of Enzyme Reactions Using Valence Bond Force Fields and Other Hybrid Quantum/Classical Approaches.* Chem. Rev., 1993. **93**: p. 2523-2544.
6. J.L. Gao, *Methods and Applications of Combined Quantum Mechanical and Molecular Mechanical Potentials,* in *Reviews in Computational Chemistry 7.* 1996, VCH Publishers, Inc.: New York. p. 119-185.
7. A. Warshel and S. Creighton, *MOLARIS,* in *Computer Simulation of Biomolecular Systems,* W.F. van Gunsteren and P.K. Weiner, Editors. 1989, ESCOM: Leiden. p. 120.
8. M.J. Field, P.A. Bash, and M.K. Karplus, *A Combined Quantum Mechanical and Molecular Potential for Molecular Dynamics Simulations.* J. Comp. Chem., 1990. **11**(6): p. 700-733.
9. J.L. Gao and X. Xia, *A Priori Evaluation of Aqueous Polarization Effects Through Monte Carlo QM-MM Simulations.* Science, 1992. **258**: p. 631-635.
10. H.J. Kim and J.T. Hynes, *A Theoretical Model for SN1 Ionic Dissociation in Solution. 1. Activation Free Energetics and Transition-State Structure.* JACS, 1992. **114**: p. 10508.
11. V. Luzhkov and A. Warshel, *Microscopic Models for Quantum Mechanical Calculations of Chemical Processes in Solutions: LD/AMPAC and SCAAS/AMPAC Calculations of Solvation Energies.* J. Comp. Chem., 1992. **13**(2): p. 199-213.
12. R.V. Stanton, D.S. Hartsough, and K.M. Merz, *Calculation of Solvation Free Energies Using a Density Functional/Molecular Dynamics Coupled Potential.* J. Phys. Chem., 1993. **97**: p. 11868-11870.
13. J.E. Gready, S. Ranganathan, and P.L. Cummins. *Reaction Mechanisms in Enzyme Active Sites Using a "Divide and Conquer" Approach: Mixed Quantum and Molecular Mechanics (QM/MM).* in Advances in Computational Life Sciences, Vol. 2: Humans to Proteins, Michaelowicz, M.T., ed., CSIRO Publishing, Melbourne, p.201-212.
14. *HyperChem Users Manual: Computational Chemistry.* 1994, Waterloo, Ontario, Canada: Hypercube, Inc.
15. M.A. Thompson, E.D. Glendening, and D. Feller, *The*

Nature of K+/Crown Ether Interactions: A Hybrid Quantum Mechanical-Molecular Mechanical Study. J. Phys. Chem., 1994. **98:** p. 10465-10476.

16. M.A. Thompson, *Hybrid Quantum Mechanical/Molecular Mechanical Force Field Development for Large Flexible Molecules: A Molecular Dynamics Study of 18-Crown-6.* J. Phys. Chem., 1995. **99**: p. 4794-4804.
17. V.V. Vasilyev, *Tetrahedral Intermediate Formation in the Acylation Step of Acetylcholinesterases: A Combined Quantum Chemical and Molecular Mechanical Model.* J. Mol. Struct. (Theochem), 1994. **304**: p. 129-141.
18. H. Liu, F. Muller-Plathe, and W.V. van Gunsteren, *A Molecular Dynamics Simulation Study with a Combined Quantum Mechanical and Molecular Mechanical Potential Energy Function: Solvation Effects of the Conformational Equilibrium of Dimethoxyethane.* J. Chem. Phys., 1995. **102**: p. 1722.
19. R.V. Stanton, L.R. Little, and K.M. Merz, *An Examination of a Hartree-Fock/Molecular Mechanical Coupled Potential.* J. Phys. Chem., 1995. **99**: p. 17344-17348.
20. D.S. Hartsough and K.M. Merz, *Dynamic Force Field Models: Molecular Dynamics Simulations of Human Carbonic Anhydrase II Using a Quantum Mechanical/Molecular Mechanical Coupled Potential.* J. Phys. Chem., 1995. **99**: p. 11266-11275.
21. F. Maseras and K. Morokuma, *IMMOM: A New Integrated Ab Initio + Molecular Mechanics Geometry Optimization Scheme of Equilibrium Structures and Transition States*, J. Comp. Chem., 1995, **9**, p.1170-1179.
22. R.P. Muller and A. Warshel, *Ab Initio Calculations of Free Energy Barriers for Chemical Reactions in Solution.* J. Phys. Chem., 1995. **99**: p. 17516-17524.
23. J. Freindorf and J.L. Gao, *Optimization of the Lennard-Jones Parameters for a Combined ab initio Quantum Mechanical and Molecular Mechanical Potential Using 3-21G Basis Set.* J. Comp. Chem., 1996. **17**: p. 386-395.
24. P. Sherwood, *personal communication to A. Rendell*, 1996.
25. D. Bakowies and W. Thiel, *Hybrid Models for Combined Quantum Mechanical and Molecular Mechanical Approaches.* J. Phys. Chem., 1996. **100**: p. 10580-10594.
26. M.J. Frisch, G.W. Trucks, H.B. Schlegel, P.M.W. Gill, B.G. Johnson, M.A. Robb, J.R. Cheeseman, T.A. Keith, G.A. Peterson, J.A. Montgomery, K. Raghavachari, M.A. Al-Laham, V.G. Zakrzewski, J.V. Ortiz, J.B. Foresman, J. Cioslowski, B.B. Stefanov, A. Nanayakkara, M. Challacombe, C.Y. Peng, P.Y. Ayala, W. Chen, M.W. Wong, J.L. Andres, E.S. Replogle, R. Gomperts, R.L. Martin, D.J. Fox, J.S. Binkley, D.J. Defrees, J. Baker, J.P. Stewart, M. Head-Gordon, C. Gonzalez, and J.A. Pople, *Gaussian 94 (Revision C.2)*, Gaussian, Inc., Pittsburgh P.A., 1995.
27. J.J.P. Stewart, *Mopac93, revision 2*, QCPE 1997.
28. D.A. Case, D.A. Pearlman, J.W. Caldwell, T.E. Cheatham III, W.S. Ross, C.L. Simmerling, T.A. Darden, K.M. Merz, R.V. Stanton, A.L. Cheng, J.J. Vincent, M. Crowly, D.M. Ferguson, R.J. Radmer, G.L. Seibel, U.C. Singh, P.K. Weiner, and P.A. Kollman, 1997, *AMBER 5*, University of California, San Francisco.
29. N. Uchida, *Hardware of the VX/VPP300/VPP700 Series of Vector-Parallel Supercomputer Systems*, Fujitsu SCI. Tech. J. 1997. **33**. p6.
30. H Ishibata, M. Takahashi, snd H Sato, *Hardware of AP3000 Scalar Parallel Server*, Fujitsu Sci. Tech. J. 1997. **33**. p24.
31. A.A. Bliznyuk and A.P. Rendell, *New methods for computation of gradients in semiempirical methods*, (in preparation).
32. S. Patchkovski, University of Zurich, *Private communication to A. Bliznyuk*, 1997.