

ROAR

COPYRIGHT 1997-1999, Pennsylvania State University.
All rights reserved.

This software is covered by the AMBER license agreement which contains
restrictions on its use, disclosure and duplication.

Contents

1	History	4
2	Referencing	4
3	Overview	5
3.1	Summary of Available Features	5
3.2	Features not Available in ROAR	6
4	Theory	6
4.1	The Nosé-Hoover Chain Method	6
4.1.1	Constant T and P Methods	6
4.1.2	The Integrator	7
4.1.3	The ROAR Program	7
4.1.4	Performance	8
4.2	Ewald Sum	8
4.3	Long Range VDW Correction	9
4.4	Multiple time step method	10
4.5	Coupled Potential (QM/MM) Method	11
4.5.1	Basic Methodology	11
4.5.2	Practical Aspects	14
4.5.3	vdW Parameters for QM Atoms	15
4.5.4	Link Atoms in QM/MM Studies of Enzymes	16
4.5.5	Suggested QM/MM Terminology	16
5	Compiling	17
5.1	Supported Machines	17
5.2	General Instructions	17
5.3	Machine Specific Instructions	19
5.4	Porting to Other Machines	19
6	Input Parameters	19
6.1	General Initial Input	19
6.2	Minimization	20
6.3	Molecular Dynamics	20
6.4	Force Field	22
6.5	Nosé-Hoover Chain Temperature Coupling Parameters	24
6.6	Ewald Sum	28
6.7	Multiple Time Step	29
6.8	Water Cap	30
6.9	SHAKE Bond Constraints	31
6.10	Frozen or Restrained Atoms	31
6.11	Output	31
6.12	Coupled Potential	33
6.12.1	Coupled Potential Namelist Variables	34
6.12.2	Coupled Potential Formatted Input	35
6.12.3	Control Files for Coupled Potential Simulations	36
7	Files used	39

8 Demos	39
8.1 Water	39
8.2 Tripeptide in Water	41
8.3 Lipid Bilayer	42
8.4 Coupled Potential - Carbonic Anhydrase	43
8.4.1 Interpreting Coupled Potential Output	45
9 Questions?	46

1 History

ROAR has been developed over the last seven years in the laboratory of Dr. Kenneth M. Merz, Jr. at the Department of Chemistry, The Pennsylvania State University. The code is the result of the work of many different group members and original authors of the programs which were coupled together. Merz group contributors given in the chronology of their participation in the development of the code are:

Zhengwei Peng: Performed initial work on the AMBER 4.0/MOPAC 5.0 version.

Dave Hartsough: Completed initial working version of AMBER 4.0/MOPAC 5.0.

Robert Stanton: Wrote G92/AMBER4.0 versions and updated to AMBER 4.1.

Steve Dixon: Wrote MOPAC 7.0 version of code, also improved much of the FEP and PMF sections of the code, including the addition of constraints for the QM system.

Ailan Cheng: Incorporated Nosé-Hoover chain temperature coupling into AMBER 4.1, and wrote Multiple Time Step code.

K. V. Damodaran: Implemented Ewald sum and long range van der Waals corrections.

James J. Vincent: Extended parallel code to include Nosé-Hoover chain temperature coupling, also coded original MPI SANDER version.

Masayuki Mori: Implemented parallel Ewald sum.

Scott A. Best: Performed source code testing.

Arjan van der Vaart: Implemented Particle Mesh Ewald.

Gérald Monard: Debugged QM/MM routines.

Mireia Garcia: Implemented Limited memory BFGS minimizer.

Laura C. Van Zant: Performed source code testing.

Kenneth M. Merz Jr.: Kept everyone else on the right track and kept the project funded.

2 Referencing

ROAR presents an interesting referencing dilemma to the researcher because it is a combination of individually developed codes. Anyone who does not understand the appropriate usage requirements for the AMBER portion of the code and for MOPAC 7 should check into them prior to using this code. Three references are appropriate: the ROAR program, the AMBER 4.1 code, and one for the QM code used:

The ROAR program :

A. Cheng, R. S. Stanton, J. J. Vincent, A. van der Vaart, K. V. Damodaran, S. L. Dixon, D. S. Hartsough, M. Mori, S. A. Best, G. Monard, M. Garcia, L. C. Van Zant and K. M. Merz, Jr. (1999) ROAR 2.0, The Pennsylvania State University

The AMBER 4.1 program suite :

D. A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, Cheatham, III, T. E., D. M. Ferguson, G. L. Seibel, U. C. Singh, P. K. Weiner and P. A. Kollman (1995), AMBER 4.1, University of California, San Francisco.

MOPAC 7.0 :

MOPAC 7: A Public-Domain General Molecular Orbital Package (QCPE 455 7.0), J. J. P. Stewart, Quantum Chemistry Program Exchange, No 455, 13, 42-45 (1993).

3 Overview

ROAR is derived from the SANDER module of AMBER 4.1. Most importantly it differs from SANDER in the following ways:

3.1 Summary of Available Features

Nosé-Hoover Chain algorithms are combined with an explicit reversible integrator to solve Newton's equations of motion and to control the temperature and pressure during a simulation. Simulations can be performed under the following conditions:

- Constant temperature/constant volume (*i.e.*, NVT ensemble)

This option can be used to study the gas phase, solvated phases or the bulk phase of any molecular system (*i.e.*, a protein in water, bilayer, *etc.*). Up to two chains of thermostats can be used here. For gas phase simulations, only a single thermostat can be used with the default prep/link/edit procedures.

- Constant temperature/constant pressure (*i.e.*, NTP ensemble) with uniform dilation of the simulation box.

This option is for bulk phases only and a maximum of three chains of thermostats can be used. If bond constraints (*i.e.*, SHAKE) are used, all degrees of freedom as well as the simulation box must be coupled to the same chain. This restriction can be relaxed in the limit of a large barostat mass. However, it remains suggested that only one chain be used.

In the absence of bond constraints, one can use either two or three chains of thermostats. If two chains are used, one chain is coupled to the simulation box while the second chain is coupled to the entire system. If three chains are used, the first chain is coupled to the solute while the second and third chains are coupled to the solvent molecules and to the simulation box respectively.

- Constant temperature/constant pressure (*i.e.*, NTP ensemble) with a fully flexible simulation box.

This option should be used for bulk crystalline phases or liquid crystalline phases (for example, bilayers and solid simulations). Using this option for liquid phase simulations of proteins solvated by water (or other solvents) will lead to an unstable simulation box since there are no shear constraints to contain the system. The choice of the number of chains of thermostats is similar to the NTP case.

The Ewald sum or the Particle Mesh Ewald methods can be employed to treat long range electrostatic interactions for a periodic system. Long range corrections for van der Waals interactions, virial, and pressure have also been included.

Molecular Dynamics simulations using the Multiple Time Step Algorithm can be performed in order to speed-up simulations. At this time, this option is only supported when using Molecular Mechanics force fields.

The code is capable of running QM/MM calculations under constant temperature/constant volume (*i.e.*, NVT ensemble). At this time only provisions to carry out semiempirical based QM/MM calculations have been included.

The classical MD portion of the code runs in parallel. However, the QM/MM portion of the code runs only on serial platforms at this time.

Additional features which may affect data analysis:

- The velocity Verlet integrator is used to integrate the equations of motion. Thus, both the SHAKE and RATTLE algorithms are used whenever constraints are present.
- This code stores both coordinates and velocities at time t , which is different from the original SANDER (leap frog) where the coordinates are stored at time t and velocities at $t = dt/2$. This may affect your data analysis.
- Ewald sum is used only with the 12-6 and 12-10 potentials and with a constant dielectric function.
- The initial velocities are generated in ROAR by a new routine which removes the net momentum. This eliminates the need to periodically reposition the COM of the solute since the NHC algorithm conserves the net momentum during the simulation.
- In constant pressure simulations, the pressure is calculated based on the coordinate and velocity of each individual atom. This not only is required by the algorithm but also gives better statistics.
- Extra output files can be saved to monitor the performance of the MD simulation. The user has the option of either writing or not writing out these files.
- The belly option is supported in ROAR.

3.2 Features not Available in ROAR

- Berendsen's (original SANDER) temperature and pressure control methods have been replaced by the Nosé-Hoover Chain temperature and pressure control algorithms.
- The leap-frog algorithm has been replaced by a modified velocity Verlet/reversible integrator algorithm.
- Octahedral periodic boundary conditions are no longer available.
- The fast water option has been removed.
- NMR refinement has been eliminated.
- SGI_MP has been replaced by standard MPI which is now used on SGI platforms.
- The polarization option is no longer supported.

4 Theory

In the following, we have provided a brief overview of the major new theoretical tools that have been included in ROAR. This section is heavily referenced and the interested reader is encouraged to examine these references for further details.

4.1 The Nosé-Hoover Chain Method

4.1.1 Constant T and P Methods

Proper control of temperature and pressure is crucial in molecular dynamics simulations since most experiments are conducted at a specific temperature and under a specific pressure. Many methods have been developed to control the temperature and pressure in molecular dynamics simulations.¹ A simple, but crude, scaling method, or a modified scaling method can equilibrate the system to a new temperature or pressure efficiently.² However, the trajectory generated does not correspond to any existing ensemble.¹ Nosé introduced an extended system method which controls the temperature by coupling the system to a heat bath³ and Hoover⁴ demonstrated that

the equations of motion of Nosé dynamics can be derived from the Liouville theorem. Martyna *et al.* modified the equations of motion by introducing a chain of thermostats and formulated a Nosé-Hoover chain (NHC) algorithm.⁵ The trajectory generated using this approach corresponds to a canonical ensemble (NVT) and it was shown by these authors that the Nosé-Hoover chain approach samples the phase space more extensively. Anderson reported an extended system Lagrangian by including a PV term.⁶ Using this method, the volume of the system, an extra degree of freedom, is allowed to vary according to the equations of motion derived from Hamiltonian mechanics, thereby achieving the desired pressure. Parrinello and Rahman extended this method to allow both the volume and the shape of the simulation box to alter freely.⁷ Finally, Martyna *et al.*⁸ combined the Nosé-Hoover chain algorithm with the Anderson and Parrinello/Rahman methods and the resultant equations of motion generate the isobaric-isothermal ensemble (NTP). Importantly, the Hamiltonian for these extended system methods (both NVT and NTP) is a conserved quantity which can be used to monitor the performance of the simulation.

4.1.2 The Integrator

In addition to the temperature and pressure control methods, a stable integrator to solve Newton's equations of motion is required to carry out MD simulations successfully. Algorithms like Verlet, leap-frog (half-step) Verlet, velocity Verlet, and predictor-corrector algorithms are all widely used.¹ Recently, Tuckerman *et al.* developed a method to generate explicit reversible integrators.⁹ This method, in conjunction with the reference system propagator algorithm (RESPA),^{10,11} has been shown to increase simulation speed dramatically.¹² Martyna *et al.* applied the reversible integrator into Nosé-Hoover chain extended system methods and derived a modified velocity Verlet integrator.¹³

4.1.3 The ROAR Program

We have implemented the Nosé-Hoover chain constant temperature, Anderson constant pressure, Parrinello/Rahman constant pressure methods combined with the explicit reversible integrator algorithm into the SANDER module of AMBER 4.1.¹⁴ More specifically, we replaced the leap-frog algorithm originally used in SANDER by a modified velocity Verlet algorithm and replaced the original Berendsen temperature/pressure scaling methods by the Nosé-Hoover chain temperature/Anderson pressure controlling schemes. Multiple chains of thermostats were also implemented in order to control the temperature more efficiently.¹⁴ In addition to these methods, we added the Nosé-Hoover chain-Parrinello/Rahman fully flexible box to SANDER. This approach is very useful in the study of the structure and structural transformations present in crystalline phases. Details of the ROAR implementations can be obtained in a recent publication by Cheng and Merz.¹⁴

The following properties associated with the NHC algorithm are of interest to the users. For constant temperature simulations, one can calculate the conserved quantity extended Hamiltonian

$$H'(p, r, p_\xi, \xi) = U(r) + \sum_{i=1}^N \frac{p_i^2}{2m_i} + \sum_{i=1}^M \frac{p_{\xi_i}^2}{2Q_i} + N_f kT \xi_1 + \sum_{i=1}^M kT \xi_i \quad (1)$$

where the r 's and p 's are the positions and momenta, respectively, M is the number of thermostats on the chains, ξ and p_ξ are the thermostat variable and its conjugated momentum, Q_i is the mass of the i 'th thermostat, k is the Boltzmann constant and N_f is the number of degrees of freedom.

For the constant pressure and temperature simulations, similarly, there is a conserved quantity extended Hamiltonian H'

$$H' = U(r, \bar{h}) + \sum_{i=1}^N \frac{p_i^2}{2m_i} + \frac{p_\xi^2}{2Q} + \frac{1}{2W_g} Tr [\bar{p}_g^t \bar{p}_g] + P_{ext} det [\bar{h}] + (N_f + d^2) kT\xi \quad (2)$$

where \bar{h} is the box dimension tensor, p_g is the conjugate momentum, W_g is the barostat mass, P_{ext} is the desired pressure, and P_{int} is the pressure tensor where P_{int} is expressed as

$$(P_{int})_{\alpha\beta} = \frac{1}{V} \left[\sum_{i=1}^N \left(\frac{(p_i)_\alpha (p_i)_\beta}{m_i} + (F_i)_\alpha (r_i)_\beta \right) - (\bar{U}' \bar{h}^t)_{\alpha\beta} \right] \quad (3)$$

Simulations can be carried out with two types of box dilations: isotropic and fully flexible. For simulations pertaining to an isotropic box dilation, the box tensor reduces to a unit tensor multiplied by the box length, L , *i.e.* $\bar{h} = L\bar{I}$ and the pressure becomes

$$P = \frac{1}{3V} \left[\sum_{i=1}^N \left(\frac{p_i^2}{m_i} + F_i \cdot r_i \right) \right] \quad (4)$$

4.1.4 Performance

The NHC/reversible integrator does not require any extra computation time if constraints are not present. For simulations where the SHAKE and RATTLE procedures were used, a 10% increase in cpu time was observed for constant temperature simulations due to an extra call to RATTLE, while a 15% increase in cpu time was observed for constant temperature/pressure runs since SHAKE and RATTLE are both utilized. Nonetheless, with the advantages of the NHC method, we feel it is a superior algorithm for use in MD simulations of biomolecular systems.¹⁴

4.2 Ewald Sum

The Ewald sum method is a well known technique for calculating long range electrostatic interactions which are often neglected while using a finite non-bonded cutoff in molecular dynamics and Monte Carlo simulations.¹ This method was originally developed for crystal lattices. It is ideal for simulations involving infinitely repeating units through periodic boundary conditions, such as polymers and lipid bilayers, wherein solute-image- solute interactions pose no conceptual problem. The method has also been used in the simulation of neat liquids, although there are concerns that the use of this technique may introduce artificial periodicity in the liquid. However, this appears to only be a major concern in very small simulation cells.¹⁵ The Ewald sum method cannot be used for a non-periodic system such as a capped protein or gas phase system.

The current implementation in ROAR is only meant for net neutral unit cells. If the system is charged one must place the required number of counter ions in the computational box to make the total system neutral.

In the Ewald sum method, a Gaussian charge distribution of opposite sign is superimposed upon the original point charges, producing a screened charge distribution. The electrostatic interaction between the screened charges are short ranged. The original distribution is recovered by adding a second Gaussian charge distribution identical to the first, but of opposite sign (*i.e.* the same sign as the point charges). The interaction between these canceling distributions is calculated in reciprocal or k-space. For a more mathematical description, please see references 1 and 15.

The total electrostatic interaction is calculated as:

$$E(total) = E(real) + E(reciprocal) - E(self) - E(excl) \quad (5)$$

$E(real)$ is the screened interaction and $E(reciprocal)$ is due to the canceling Gaussian charge distribution. $E(self)$ corrects for the interaction of the canceling distribution with itself. In molecular systems we also must correct for the interaction of the canceling charge distribution between an atom and atoms in its excluded atom list.

In the current implementation, the various terms are calculated as follows:

$$E(real) = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \left(\sum_{|n|=0}^{\infty} z_i z_j \frac{erfc(\alpha|\mathbf{r}_{ij} + \mathbf{n}|)}{|\mathbf{r}_{ij} + \mathbf{n}|} \right) \quad (6)$$

In (6), the double summation runs over all atom-atom pairs and the third sum over n is over the cells. The $n = 0$ term refers to the primary computational box and this term does not involve pairs of an atom with its own image. In practice, however, the parameter α is set such that $E(real)$ is negligible for $r_{ij} > r_{cut}$, so that this term need be evaluated only over the nonbond pairs of an atom with $n = 0$. Without the complementary error function, this term is just the electrostatic energy using a nonbond cutoff of r_{cut} .

The reciprocal space contribution is calculated as:

$$E(reciprocal) = \frac{1}{2\pi V} \sum_{i=1}^N \sum_{j=1}^N \sum_{\mathbf{k} \neq 0} z_i z_j \left(\frac{4\pi^2}{k^2} \right) \exp\left(\frac{-k^2}{4\alpha^2}\right) \exp(i\mathbf{k} \cdot \mathbf{r}_{ij}) \quad (7)$$

where the summation is over a set of non-zero reciprocal vectors given by

$$(k_x, k_y, k_z) = 2\pi \left(\frac{l}{L_x}, \frac{m}{L_y}, \frac{n}{L_z} \right) \quad (8)$$

with the L 's being the respective box vectors. In fully flexible constant pressure simulations, where the box vectors are represented using the box dimension tensor \mathbf{h} , we use the general definition for the \mathbf{k} -vectors.

$$\mathbf{k} = 2\pi(\mathbf{h}^{-1})^t \begin{pmatrix} l \\ m \\ n \end{pmatrix} \quad (9)$$

Since the \mathbf{k} -space term is an even function of \mathbf{k} , the summation has to be done only over one half of the \mathbf{k} -vector sphere. The maximum extent of \mathbf{k} -vectors to be included in (7) is determined depending the convergence of the \mathbf{k} -space energy (for more information see below).

The two correction terms $E(self)$ and $E(excl)$ are calculated as,

$$E(self) = \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^N z_i^2 \quad (10)$$

$$E(excl) = \sum_{i=1}^N \sum_{j \in N_i^{excl}} \frac{z_i z_j erf(\alpha|\mathbf{r}_i - \mathbf{r}_j|)}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (11)$$

In equation (11), the second summation runs over the excluded atoms of atom i (*i.e.*, atoms that are linked through a bond, angle or torsion to atom i).

ROAR has the Ewald sum incorporated using the standard formulation,¹ as opposed to, for example, the particle mesh Ewald sum. We adopted this scheme because of its simplicity and clarity. The reciprocal space part of the code is also easily parallelized.

4.3 Long Range VDW Correction

Long range corrections have also been incorporated for the van der Waals (vdW) energy and its component of the virial. These corrections are used in conjunction with the Ewald sum method. These long range corrections should be used when employing the Ewald sum method in order

to balance out the long-range van der Waals and electrostatic contributions. The long range corrections for the van der Waals energy and its component of the virial are given by

$$E^{lrc}(vdw) = \frac{2\pi}{V} \sum_i \sum_{i \leq j} N_i N_j \left\langle \frac{A_{ij}}{9r_c^9} - \frac{C_{ij}}{3r_c^3} \right\rangle \quad (12)$$

$$VIR^{lrc}(vdw) = \frac{-2\pi}{V} \sum_i \sum_{i \leq j} N_i N_j \left\langle \frac{12A_{ij}}{9r_c^9} - \frac{6C_{ij}}{3r_c^3} \right\rangle \quad (13)$$

In (12) and (13), the summations are over the number of species of atoms in the system and N_i and correspond to the number of atoms in the i 'th species.

The concept is strictly applicable only to uniform systems (such as liquids), and is an approximation when used for a protein in a solvent, *etc.* For constant volume runs, this energy correction is a constant. In this case, the long range corrections do not have any effect on the structural properties or the dynamics. However, for constant pressure runs, the energy correction fluctuates with changing volume. Since the pressure is modified through $VIR^{lrc}(vdw)$, structural and dynamical properties are affected by the long range corrections.

4.4 Multiple time step method

Computer simulation of biomolecular systems is a rather computationally intensive undertaking even when performed on modern serial or parallel supercomputers due to the large size of biomolecules and the associated conformational space that must be sampled. In addition, the dynamics of a wide range of time scales are present due to the great complexity of biological systems. Generally, the dynamics of principal concern, such as those associated with typical biological transformations (e.g. catalysis), occur at a relatively long time scale. However, the fastest motions, such as bond stretching and angle bending, are the limiting factors that govern the size of the time step used during a molecular dynamics simulation study. The simulation time step has to be small enough so that the fast degrees of freedom are evolved properly, thereby ensuring numerical stability.

The force fields used in the simulation of biomolecular systems include the contributions due to bond, bond angle, dihedral angle, hydrogen-bonding, and non-bonded van der Waals and electrostatic interactions.^{21-23,55} When the AMBER force field for biomolecules^{22,55} is used, the potential energy of the system can be expressed as:

$$U = U_{\text{bond}} + U_{\text{angle}} + U_{\text{dihedral}} + U_{\text{LJ}} + U_{\text{el}} + U_{\text{H-bond}} \quad (14)$$

The frequency of bond vibration and angle bending is generally very high, while the dynamics due to the non-bonded interactions are relatively slow. Table 1 list the characteristic time scales for bond, angle, torsion, and non-bonded interaction. The non-bonded interactions are the most time consuming components in the force calculation. Therefore, the multiple time step method^{10-12,60} is very suitable under these circumstances, because one will be able to update the most expensive part of the energy and force calculation much less frequently than the faster part of the calculation (*i.e.*, bonds, angles, *etc.*). In addition to the time scale differences of the different components in the force field, the forces due to the non-bonded nearest neighbors varies significantly faster than those due to atoms which are further away. Thus, one can also separate the non-bonded interactions into long and short range components.

In ROAR the forces have been separated into three time scales: fast, intermediate and slow. The time step for updating the fast, intermediate and slow forces are dt/nm , dt/n , and dt . There are many ways to separate the forces; thus, our program is designed so that the user can select the separation scheme according to the system under investigation. When $n = 1$ or $m = 1$, it will reduce to a two time scale algorithm. When $n=m=1$, all forces will be updated at the same time interval, therefore, the multiple time step method becomes a standard integrator. Examples of three types of separation approaches are listed in Table 2. The first and second separation schemes were based on the components in the force field. For the electrostatic interactions

Table 1: Time Scales for the Lennard-Jones Interactions in the AMBER Force Field^a

Atom	ϵ (kcal/mol)	σ (Å)	t (ps)
Carbon	0.06	3.21	2.22
Polar hydrogen	0.02	1.78	0.62
Other hydrogen	0.01	2.74	1.34
Oxygen	0.20	2.85	1.25
Nitrogen	0.16	3.12	1.43

^a) Parm91 in particular.

Bond and angle frequencies and time scales

Bonds	freq (cm-1)	period (ps)
C - C	1000	0.033
C - H	3000	0.011
Angle	500	0.067
Torsion	200	0.16

using the Ewald sum, the reciprocal space part is mostly long range interactions, while the real space part involves short range interactions. Therefore, in MTS 2 the short range electrostatic interactions and van der Waals interactions (since those involved in hydrogen bonds are rather fast) were grouped into the intermediate time scale. For the third separation method (MTS 3) the non-bonded interactions were separated according to the distances between a pair of atoms as well, *i.e.*, van der Waals interactions and the real space part of the Ewald sum within a 7 Å cutoff were placed in the intermediate time scale, while those beyond 7 Å were placed in the slow time scales in order to further speed the simulation. Theoretical details and the performance of the multiple time step method are discussed elsewhere.⁶¹ Tables 3 and 4 list all the possible and recommended combinations of some critical parameters.

4.5 Coupled Potential (QM/MM) Method

4.5.1 Basic Methodology

The basic strategy for this approach was laid out in a seminal paper by Levitt and Warshel¹⁶ where a classical potential was combined with an early semiempirical quantum mechanical method (MINDO/2). Formally the method can be described as follows (independent of the QM or MM potential function used): The Hamiltonian used within the QM/MM formulation is taken to be an effective Hamiltonian (H_{eff}) which operates on the wavefunction (Ψ) of the system. The wavefunction is dependent on the position of the quantum mechanical nuclei, R_{QM} , the molecular mechanical nuclei, R_M , as well as the positions of the electrons, r .

$$H_{eff}\Psi(r, R_{QM}, R_M) = E(r, R_{QM}, R_M)\Psi(r, R_{QM}, R_M) \quad (15)$$

The effective Hamiltonian can be divided into three terms (equation 16), which are generated from the interactions which occur within and between the components of the system (see Scheme 1). The contributions considered here include the completely quantum mechanical interactions, H_{QM} (QM in Scheme 1), the purely molecular mechanical interactions, H_{MM} (MM in Scheme 1) and the interactions between the QM and MM portions of the system, $H_{QM/MM}$ (indicated by the arrow in Scheme 1).

Table 2: The Three Examples of Force Separation Schemes

MTS 1	
Time scale	Components of force
Fast	Bonds
Intermediate	Angles, torsions
Slow	Non-bonded van der Waals and electrostatic (<i>i.e.</i> all components of Ewald sum)

MTS 2	
Time scale	Components of force
Fast	Bonds
Intermediate	Angles, torsions, vdW and real space part of Ewald sum
Slow	Long range electrostatic (<i>i.e.</i> reciprocal part of Ewald sum)

MTS 3	
Time scale	Components of force
Fast	Bonds
Intermediate	Angles, torsions, vdW and real space part of Ewald sum within 7Å
Slow	Long range electrostatic (<i>i.e.</i> reciprocal part of Ewald sum) and vdW and real space part of Ewald sum beyond 7Å

$$H_{eff} = H_{QM} + H_{MM} + H_{QM/MM} \quad (16)$$

The total energy of the system can likewise be divided into three component parts.

$$E_{eff} = E_{QM} + E_{MM} + E_{QM/MM} \quad (17)$$

These component energies can be obtained by solving either the Roothaan-Hall equations¹⁷ (associated with Hartree-Fock based methodologies) or the Kohn-Sham equations¹⁸ (associated with density functional theory based methodologies) for H_{eff} (equation 1). Another way to express the total energy of the system is as the expectation value of H_{eff} . The purely MM term can be removed from the integral because it is independent of the electronic positions. Thus, the total energy of the system can be given as,

$$E_{eff} = \langle \Psi | H_{QM} + H_{QM/MM} | \Psi \rangle + E_{MM} \quad (18)$$

In equation 18, H_{QM} is the Hamiltonian given by either semiempirical,^{19,20} Hartree-Fock¹⁷ or density functional theory,¹⁸ while E_{MM} is an energy obtained using a classical force field of some type (see below for more discussion regarding the choices).²¹⁻²⁵ The key remaining term is $H_{QM/MM}$. This term represents the interaction of the MM atom "cores" with the electron cloud of the QM atoms when interacting with MM atoms, as well as the repulsion between the MM and QM atomic cores. Finally, it was found to be necessary to add a Lennard-Jones term to the QM atoms to obtain good interaction energies as well as good geometries for intermolecular interactions.²⁶ The form of this term is:

$$H_{QM/MM} = - \sum_{iM} \frac{q_M}{|r_{iM}|} + \sum_A \frac{q_M Z_A}{|R_{AM}|} + \sum_{AM} \left(\frac{A_{AM}}{R_{AM}^{12}} - \frac{B_{AM}}{R_{AM}^6} \right) \quad (19)$$

Where, q_M is the atomic point charge on the MM atom, r_{iM} is the QM electron to MM atom distance, Z_A is the core charge of QM atom A, R_{AM} is the QM atom A to MM atom M distance and A_{AM} and B_{AM} are the Lennard-Jones parameters for QM atom A interacting with MM atom M. The critical term that allows the QM region to "see" the MM environment is the first term in equation 19 where the summation is over all interactions between MM atoms and QM electrons. This represent the core-electron interaction between MM and QM atoms and is incorporated into the QM Hamiltonian explicitly. Thus, the QM electronic structure can respond to its environment through the interaction of its electrons with the surrounding solvent/protein. The last two terms of equation 19 are added on to the total energy once the electronic energy has been determined by a self-consistent field (SCF) procedure and does not affect the electronic distribution of the system directly, but does affect the geometry of the system through the computed gradients and, hence, the resulting electronic energy on subsequent SCF cycles.

The original 1976 paper describing this method by Warshel and Levitt was clearly ahead of its time, as this approach was not widely used again until the late 1980's and early 1990's when it was re-examined by several research groups using a number of different quantum mechanical methods, including: semiempirical (S),²⁶⁻³² density functional theory (DFT)³³⁻³⁶ and Hartree-Fock (HF) *ab initio* methods.³⁷⁻³⁹ Warshel has also pioneered an alternative strategy to study reactive processes in solution and in enzymes which he has termed the empirical valence bond (EVB) method.⁴⁰ This approach has been applied to a wide range of problems⁴⁰ and has some advantages and disadvantages relative to QM/MM methods that use molecular orbital or density functional QM methodologies.⁴⁰ We will not comment on this approach further, but the interested reader is directed towards Warshel's recent book on this subject.⁴⁰ In addition to the various potentials a number of different sampling techniques have been used in conjunction with QM/MM studies including energy minimizations,^{26,37} Monte Carlo (MC) simulations²⁷ and molecular dynamics (MD) simulations.^{33,38}

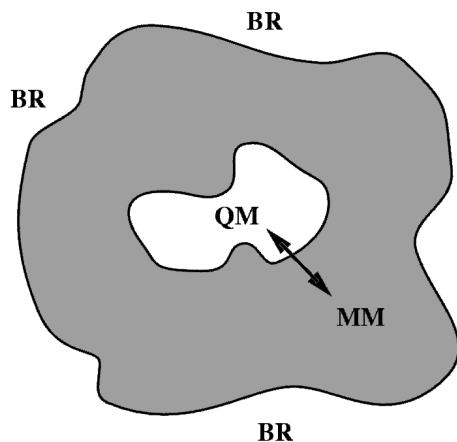


Figure 1: Division of a QM/MM system

4.5.2 Practical Aspects

There are a number of decisions required when designing and carrying out a QM/MM study. The first is the way in which conformational sampling is to be done. In many ways the choice of sampling technique and choice of model Hamiltonian are interdependent. For example, using a purely *ab initio* approach on a large biomolecular system is too computationally expensive to effectively use a sampling technique like MD, but is more compatible with an energy minimization approach. However, it must be kept in mind that the energy minimization techniques, while less computationally demanding since only a few thousand energy and gradient evaluations are involved, are prone to being trapped in local minima that are near the initial starting configuration. Thus, care must be exercised when using energy minimization techniques when one is interested in sampling the very rich conformational space of a protein. In order to get effective sampling of a protein system an MC or MD approach is preferred to minimization, and because of the generality of MD techniques they are more typically used in QM/MM studies. MC methods can be very powerful tools to study solution phase systems,⁴¹ but they are not as powerful as MD techniques when it comes to the study of enzymatic systems,⁴⁰ even though Jorgensen and co-workers have made several recent advances in their application to protein systems. MD and MC methods are much more computationally demanding than energy minimization because, by their very nature, they require the evaluation of the energy and forces (MD only) of a system many thousands of times. For example, in a protein system, to sample 100 ps using an MD simulation requires at least the evaluation of the energy and forces 100,000 times with a typical time step of 1 fs. Indeed, we have found that in some cases a timestep of 0.5 fs or less may be more appropriate in some QM/MM MD studies of protein systems, which further increases the number of times the energy and gradient must be evaluated. Early studies using the basic QM/MM approach as applied to enzymes utilized energy minimization techniques,^{16,42} while more recent studies have begun to utilize MD simulation techniques.⁴³⁻⁴⁵ However, in all cases, the QM part of the system utilized a semiempirical method because of the expense associated with a fully *ab initio* or DFT method. Indeed, at the present time, it is unlikely that a "long" MD simulation using an *ab initio* or DFT Hamiltonian will be carried out on a protein system given the inherent computational expense of this type of calculation. Thus, we are at present limited to MD simulations of proteins using semiempirical models and using energy minimization (or even so-called single point calculations where the energy is evaluated for a single geometric configuration) with *ab initio* or DFT Hamiltonians. Energy minimization, MD and single point

calculations are all possible within ROAR, but only for a semiempirical Hamiltonian. Future releases will include DFT and *ab initio* capabilities.

The next necessary choice is the selection of the quantum mechanical level to be employed. The choice of MM model will be discussed briefly below, but it should be pointed out that, because the QM portion of a QM/MM calculation so dominates the MM portion in computational expense and energy, the critical computational choice which will impact the speed of the algorithm has to do with selecting the QM potential and not the MM model. However it must be noted that recent efforts to incorporate a polarized MM model into a QM/MM calculation increases the expense of the MM portion of the calculation significantly.³⁰ Semiempirical QM models (e.g., MNDO,^{46,47} AM1²⁰ and PM3⁴⁸⁻⁵⁰) formally scale as $O(N^3)$ (where due to semiempirical approximations N is typically no greater than 4 for a "heavy" atom like carbon), while local density approximation(LDA) DFT methods also scale as $O(N^3)$,¹⁸ but N is typically much greater than that found in semiempirical theory (e.g., for a typical good quality basis set like 6-31G**¹⁷ contracted basis functions are used to represent a carbon atom¹⁷). Hartree-Fock *ab initio* methods formally scale $O(N^4)$ and correlated Hartree-Fock approaches can scale as $O(N^5)$ or higher.¹⁷ Thus, in terms of computational expense, semiempirical methods are the cheapest followed by LDA-DFT, Hartree-Fock, and finally correlated Hartree-Fock. However, the accuracy of these methods follow roughly the reverse order. It is important to note, with the advent of new linear-scaling methodologies⁵¹⁻⁵³ which introduce improved algorithms for Fock matrix generation and solution the scaling properties for QM calculations are changing, but for our purposes the above analysis will suffice.

While in principle one would prefer to use the most accurate level of theory in all cases, this is not practical at this point in time. Thus, using contemporary computers and current numerical approaches it is only possible to carry out "long" (e.g., 100 ps or more) MD simulations using semiempirical Hamiltonians on a typical protein active site (e.g., about 20-30 heavy atoms), while it is possible to carry out energy minimization studies using all QM levels except for highly correlated Hartree-Fock approaches. Within ROAR we include only semiempirical capabilities at this time.

Typically, a simple MM potential is used in QM/MM studies.^{21-23,54,55} These potentials contain harmonic bonds and angles along with a truncated Fourier series expansion to represent the torsion potential. Non-bonded interactions (*i.e.*, those interactions beyond the 1 and 4 positions along a polymer chain) are typically represented with a Lennard-Jones "6-12" potential function, while electrostatic interactions are handled using atom centered charges. Examples of force fields of this type include OPLS,²³ AMBER^{22,54,55} and CHARMM.²¹ This class of potential function has been quite successful in modeling protein structure and dynamics. The major deficiency in these models is the lack of explicit polarization. This typically leads to an unbalanced model in QM/MM studies because the QM region is polarized while the MM region is not. However, recent work has gone into incorporating polarization effects into the MM region of a QM/MM study (so-called QM/MMpol approach).^{30,31} This is a promising approach, but for now will be confined to explicit solvent studies as opposed to studies of enzymatic systems.^{30,31} ROAR at this time uses an unpolarized MM model in QM/MM studies.

4.5.3 vdW Parameters for QM Atoms

One of the more difficult decisions which needs to be made when setting up a QM/MM simulation is the proper value for the van der Waals parameters to be used on the QM atoms for their interaction with MM atoms (term 3 of equation 19). While one of the strengths of QM/MM methods is their ability to accurately model a system without extensive reparameterization of either the QM or MM methodology, it has been noted empirically that a 5-10% scaling^{26,33,38,56,57} of the vdW parameters normally used within a classical force field can greatly improve the accuracy of the free energies of solvation calculated using QM/MM methods. Some researchers will also scale the MM core charges seen by the QM partition of the system (terms 1 and 2 of equation 19) to improve calculated radial distribution functions and solvation free energies calculated for a solute. While researchers⁴¹ have worked to establish definitively the vdW pa-

rameters to be used in QM/MM calculations through in depth parameterization on test systems, it must be noted that the necessary scaling factor is highly method dependent (*i.e.* S, DF, or HF) and even basis set dependent. This is unfortunate, as it requires individual parameterization for any particular combination of MM and QM levels of theory or acceptance of the 5-10% scaling factor approximation.

4.5.4 Link Atoms in QM/MM Studies of Enzymes

In solution phase reactions the boundary between the solute (usually QM) and the solvent (usually MM) is very clear, and typically it is possible to avoid introducing the QM/MM interface between atoms that are covalently linked. However, in the case of enzymes this is not the case and it becomes necessary to introduce the concept of link atoms that covalently connect the QM and MM regions used in the representation of the protein. This interfacial region can be quite arbitrary and should be chosen with care. For example, consider modeling a glutamic acid residue within the QM/MM framework (see Scheme 2). The first consideration is where to make the QM and MM "cut" such that it does not adversely affect the electronic structure that we associate with a glutamate ion. If we treat the carboxylate as the formate anion this will greatly alter the pKa relative to a typical carboxylate anion (4.0 versus 4.5!). Thus, while formate is computationally convenient, it is electronically the incorrect choice. A better choice is to add an extra carbon atom to generate the acetate anion, which has a better pKa match with the glutamate anion.

The next step necessary when the QM/MM interface falls at a covalent bond is the introduction of an extra QM atom, for example H_{QM} in Scheme 2. This is necessary to cap the exposed valence at the carbon atom such that a closed shell calculation is done. There are several reasons for doing this, not the least of which being that the free radical of the carboxylate anion is not what we are attempting to model. Another important consideration is that closed-shell calculations are, from a technical aspect, far easier to carry out than are open-shell calculations. This so-called link atom does not see any atoms within the MM region because it is an extra QM atom solely included to satisfy valence considerations that normally would not be present in the real system. The final MM carbon atom of the Glu residue is then attached to the final QM carbon atom by a MM-type harmonic bond. This bond keeps the MM and QM regions at the appropriate distance from one another. The remaining angles and dihedrals present across the QM/MM boundary are typically represented using MM terms (*i.e.*, a harmonic bond angle and a Fourier torsion potential) and care should be taken in selecting these such that the torsional properties around the bond at the QM/MM boundary are as accurate as possible (e.g., based on the appropriate QM or MM calculations).

The concept of a link atom is an approximation inherent in the basic QM/MM approach of which one needs to be aware when treating protein systems or any system that requires the use of a covalent QM/MM boundary region. However, through careful selection of the location of this region, gross errors can be avoided. New and better ways need to be formulated to treat the link atom region and, indeed, some work has proceeded along these lines. In particular, the work of Rivail and co-workers^{32,39} using localized orbitals to represent the QM/MM boundary region looks promising, but needs to be explored further to see how it performs in a variety of situations. At this time, however, ROAR only includes the capabilities to use the link atom approximation.

4.5.5 Suggested QM/MM Terminology

Unfortunately, at this point there is no universally accepted nomenclature that can be used to describe QM/MM methods in a concise manner. Indeed, there are several competing nomenclatures present in the literature which tend to confuse the uninitiated about the precise differences between various methods. Herein we have used the nomenclature outlined in Table 3. This table also contains some approximate information regarding the accuracy of the various methods and also summarizes other abbreviations that we have used in this brief introduction to QM/MM

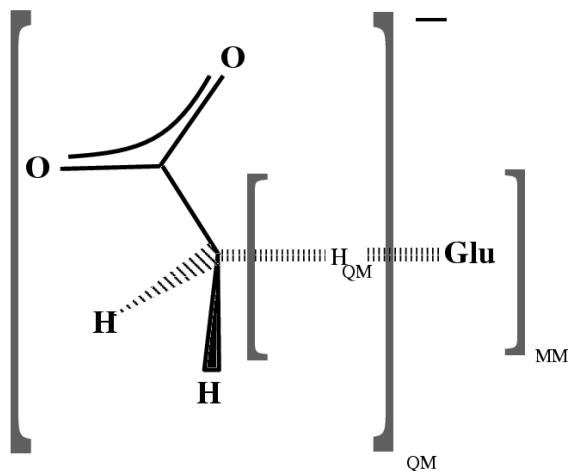


Figure 2: Position of the link atom at a QM/MM frontier.

methods.

5 Compiling

5.1 Supported Machines

The ROAR code has been tested on the following serial Unix machines:

Cray C90 DEC Alpha Hewlet Packard IBM RS6000 SGI Sun Linux

Cray T3D, T3E IBM SP2

NOTE: The coupled potential version of ROAR is not supported on any parallel machines at this time.

5.2 General Instructions

1. Edit the Makefile

Macro definitions for each supported machine type are listed at the top of the Makefile. Find the machine you intend to compile on and uncomment all lines under that machines heading by removing the leading # from the lines.

2. Compile by typing make

To create the ROAR program simply type make. This will create an executable called roar which you can then move to an appropriate place.

To create the coupled potential version of ROAR type:

```
make roar-cp
```

When making the coupled potential version, a number of files are always recompiled, even if you have previously compiled regular ROAR. This is to ensure that code specific to the coupled potential version is included.

Table 3: Summary of QM/MM Methods: Definition of Abbreviations and the Approximate Accuracy.

Abbreviation	Meaning	Accuracy ^a
QM/MM	Quantum Mechanical/Molecular Mechanical. Term used to define this class of methods.	N/A
MM	Molecular Mechanical. Term used to broadly define force fields based on classical mechanics. Only three widely used examples are given below.	N/A
AMBER	Assisted Model Building and Energy Refinement. ²² Optimized for proteins and DNA.	Depends on system, but generally good.
OPLS ^b	Optimized Potentials for Liquid Simulations. Optimized for proteins and liquids. ²³	Depends on system, but generally good.
CHARMm	Chemistry at Harvard Molecular Mechanics. ²¹ Optimized for proteins and DNA.	Depends on system, but generally good.
S/MM	Semiempirical/Molecular Mechanical. Term used to generically define the class of semiempirical QM/MM methods.	N/A
MNDO/MM	Modified Neglect of Differential Overlap/Molecular Mechanical. MNDO is a semiempirical Hamiltonian. Hydrogen bonding poorly handled, not generally useful. ⁴⁶	Depends on system $\pm 3\text{-}\pm 7$ kcal/mol
AM1/MM	Austin Model 1/Molecular Mechanical. AM1 is a semiempirical Hamiltonian. Hydrogen bonding treated, but geometric details poorly reproduced. ²⁰	Depends on system $\pm 3\text{-}\pm 6$ kcal/mol
PM3/MM	Parametric Model 3/Molecular Mechanical. PM3 is a semiempirical Hamiltonian. Best method for hydrogen bonding, therefore, this is the most versatile method. ⁴⁸	Depends on system $\pm 3\text{-}\pm 5$ kcal/mol
DFT/MM	Density Functional Theory/Molecular Mechanical. Generic term defining the class of density functional theory QM/MM methods.	N/A
LDA/Basis/MM	Local Density Approximation/Basis³/Molecular Mechanical.	Depends on system $\pm 3\text{-}\pm 6$ kcal/mol
GC/Basis/MM	Gradient Corrected/Basis^c/Molecular Mechanical.	Depends on system $\pm 2\text{-}\pm 4$ kcal/mol
HF/MM	Hartree Fock/Molecular Mechanical. Generic term for Hartree-Fock QM/MM methods.	N/A
HF/Basis/MM	Hartree Fock/Basis^c/Molecular Mechanical.	Depends on system $\pm 2\text{-}\pm 5$ kcal/mol
MPX/Basis/MM	Møller Plesset X (X=2 or 4 typically)/Basis^c/Molecular Mechanical.	Depends on system $\pm 1\text{-}\pm 3$ kcal/mol

a) These numbers **qualitatively** define the accuracy of the methods.

b) OPLS includes the TIP3P and TIP4P water models of Jorgensen.⁵⁸

c) For a description of basis sets (e.g. 6-31G**, *etc.*) and Møller-Plesset theory see.¹⁷

If you compile the coupled potential version and then wish to remake the regular version of ROAR you must make clean first.

5.3 Machine Specific Instructions

Cray Parallel Machines: Directory paths for MPI include files and libraries are handled by modules in current versions of Unicos. You must load the appropriate MPI module to compile the parallel version. See your system administrator for assistance.

5.4 Porting to Other Machines

Macro definitions for generic Unix machines are included in the Makefile. Edit these as appropriate for your environment. System specific code is located in sys.F. Calls to timers and file open routines are included there.

6 Input Parameters

REMINDER: Only namelist input is supported in ROAR. The parameters listed here are the only ones supported by ROAR. Since ROAR is derived from the SANDER module of AMBER, most variables have the same meaning as in SANDER. Those related to the temperature and pressure control now refer to the Nosé-Hoover parameters. Almost all input parameters have default values which will keep the program running, but the results and the simulation conditions may not be what is desired. Required input parameters are so noted.

6.1 General Initial Input

TIMLIM: Time limit, in seconds, for the job. Default 999999.

IG: The seed for the random number generator.
The starting velocities of the atoms and that of the thermostats of an MD run depend on the random number generator seed. Default: 71277.

KFORM: Format of topology file from PARM
= 0 Binary (not used in double precision version)
= 1 Formatted (default)

VLIMIT: Limit for the velocities in a MD simulation.
Optional input parameter. The default is set to 0.0.
If VLIMIT .NE. 0.0, then any component of the velocity that is greater than $\text{abs}(\text{VLIMIT})$ will be reduced to VLIMIT (preserving the sign). This can be used to avoid occasional instabilities in molecular dynamics runs. VLIMIT should be set (if at all) to a value like 20.0, which is well above the most probable velocity in a Maxwellian-Boltzmann distribution at room temperature. A warning message will be printed whenever the velocities are modified. Runs that have more than a few warnings should be carefully examined.

NTU: Flag for the unit of energy. Default = 1, and should always set to 1.
= 1 Energy is reported in kcal/mol
= 2 Energy is reported in kJ/mol (not currently supported)

6.2 Minimization

IMIN:	Flag to run minimization = 0 No minimization (only do molecular dynamics; default) = 1 Perform minimization (and no molecular dynamics)
MAXCYC:	Maximum number of cycles of minimization. Default 1.
NCYC:	After NCYC cycles the method of minimization is switched from steepest descent to the conjugate gradient method. Default 10.
NTMIN:	Flag for the method of minimization. = 0 Full conjugate gradient minimization. The first 10 cycles are steepest descent at the start of the run and after every nonbonded pairlist update = 1 For NCYC cycles the steepest descent method is used then conjugate gradient is switched on (default). = 2 Only steepest descent method is used. = 3 Only Limited Memory BFGS method is used. This method usually performs faster than conjugate gradient minimization (see Liu and Nocedal, Mathematical Programming 45 (1989) 503-528).
DX0:	The initial step length. (Note: it is DX zero not DX letter O) If the initial step length is big then the minimizer will try to leap across the energy surface and sometimes the first few cycles will give a huge energy, however the minimizer is generally smart enough to adjust itself. Default 0.01.
DXM:	The maximum step length allowed. Default 0.5.
DRMS:	Convergence criterion for the energy gradient: minimization will halt when the root-mean-square of the Cartesian elements of the gradient is less than DRMS. Default 1.0E-4 kcal/mole Å

6.3 Molecular Dynamics

NRUN:	Number of MD-runs of NSTLIM steps to be performed. Since the restart coordinates are written only at the end of each "run", it is sometimes advisable to break a long MD calculations into several "runs". The number of picoseconds of molecular dynamics is equal to the product of NRUN x NSTLIM x DT. Default 1
NSTLIM:	Number of MD-steps per NRUN to be performed. Default 1.
NDFMIN:	Number of degrees of freedom that will be subtracted from the total number of degrees of freedom. If either NTCM or NSCM. NE. 0 then this option should be set equal to 6. Otherwise, NDFMIN should be 0. NDFMIN, NTCM, and NSCM are ignored for belly dynamics. Default 0.
NTCM:	Flag for the removal of translational and rotational motion at the beginning of the simulation. = 0 The translational and rotational motion about the center of mass is not removed (default) = 1 The above motion is removed one time at the beginning of the simulation.

NSCM: Flag for the removal of translational and rotational motion at regular intervals.
 After every NSCM steps, translational and rotational motion will be removed. This flag is ignored for both belly and periodic simulations. Default 0.

INIT: Flag for different starting procedures.
 If option NTX is less than 4, INIT should be equal to 3. If option NTX is greater than or equal to 4, this option should be equal to 4.
 = 3 Starting velocities are assigned from a Maxwellian distribution at TEMPI K. If input TEMPI = 0, TEMPI is set to 10 K. Net momentum of the entire system will be removed.
 = 4 Input velocities (NTX = 4) are read from the "inpcrd" file.
 Note: The Nosé-Hoover chain/reversible algorithm used in ROAR conserves the linear momentum of the system. When restarting a run with velocities and if the net momentum is not zero, it will be conserved. If one wants to have zero net momentum during the simulation, one should restart a run using ROAR without velocities, then ROAR will choose the initial velocities and remove the net momentum.

T: The time at the start (ps). This is for your own reference and is not critical. Start time is taken from the coordinate input file. Default 0.0.

DT: The time step (ps).
 Recommended MAXIMUM is .002 if SHAKE is used, or .001 if it isn't. Note that for temperatures above 300K, the step size should be reduced since greater temperatures mean increased velocities and longer distances traveled between each force evaluation, which can lead to anomalously high energies and temperatures. Default 0.001.

NTB: Flag indicates if periodic boundary condition is used.
 Required input parameter.
 If NTB = 0 then a boundary is NOT applied regardless of any boundary condition information in the topology file. The value of NTB specifies whether constant volume or constant pressure dynamics will be used. Options for constant pressure are described in the NHC section.
 = 0 no periodicity is applied
 = 1 constant volume simulation
 = 2 constant pressure simulation
 Note: If NTB .NE. 0, there must be a periodic boundary in the topology file. Constant pressure is not used in minimization (IMIN=1, above).
 For a periodic system, constant pressure is the only way to equilibrate the density of the system if the starting state is not correct. For example, the solvent packing scheme used in EDIT can result in voids being generated when solvent molecules are subtracted from the system. Another consideration is the shrinkage of the box during a constant pressure MD simulation which can result in solvent molecules 'seeing' parts of the solute in opposite directions if the initial box size was chosen too small. The remedy for this is to build a large enough box when beginning a constant pressure MD simulation.

- NTX:** Flag to specify reading options for restart coordinate/velocity file
- = 1 formatted, X is read with no V (default)
 - = 2 unformatted, X is read with no V
 - = 4 unformatted, X and V are read
 - = 5 formatted, X and V are read
 - = 6 unformatted, X, V and BOX(1..3) are read
 - = 7 formatted, X, V and BOX(1..3) are read
 - = 11 formatted, X, V and NHC thermostat X and V are read
 - = 12 unformatted, X, V and NHC thermostat X and V are read
 - = 13 formatted, X, V, NHC thermostat X and V, and BOX are read
 - = 14 unformatted, X, V, NHC thermostat X and V, and BOX are read
- IMGSLT:** Control solute-solvent imaging in calculations with periodic boundary conditions.
- = 0 Solute is imaged with solvent. Solute is allowed to interact with solvent images (it they are within CUT). Default.
 - = 1 No solute-solvent imaging. Solute does not see image solvent.
- This assumes that the solute is centered in the periodic system, and is not free to migrate. Do not use this with mobile solutes. This option is mainly useful for large solutes. This option is not recommended since ROAR does not re-position the center of the solute at the center of the simulation box.
- IFTRES:** Flag to remove the nonbonded cutoff from the solute in simulations with periodic boundary conditions
- =0 ALL intramolecular solute - solute nonbonded interactions are calculated regardless of whether the interatomic distance is greater than the nonbonded cutoff. Solute-solute imaging is turned off.
 - =1 Nonbonded interactions are evaluated normally (default).
- This option is recommended when the Ewald sum method is used.
- Note: For simulation of highly charged solutes in a water bath, it can be useful to calculate ALL solute-solute nonbonded interactions in order to reduce electrostatic problems. This is especially important for highly charged systems like nucleic acids. Note that this option is intended for small solutes, and will generate many more nonbonded pairs than the normal method if the solute is large. Counterions added in EDIT are considered part of the solute.

6.4 Force Field

- NTF:** Force evaluation. Note: If SHAKE is used (see NTC), it is not necessary to calculate forces for the constrained bonds.
- = 1 complete interaction is calculated (default)
 - = 2 bond interactions involving H-atoms omitted (use with NTC = 2)
 - = 3 all the bond interactions are omitted (use with NTC = 3)
 - = 4 angle interactions involving H-atoms and all bonds are omitted
 - = 5 all bond and angle interactions are omitted

- = 6 dihedrals involving H-atoms and all bonds and all angle interactions are omitted
 - = 7 all bond, angle and dihedral angle interactions are omitted
 - = 8 all bond, angle, dihedral and non-bonded interactions are omitted
- IDIEL:** Type of dielectric function to be used in calculating the electrostatic energy.
- = 0 distance dependent dielectric function.
- This is used to mimic the presence of a high dielectric solvent, typically for simulating a protein when no explicit water is present.
- = 1 constant dielectric function. This is the default.
- This is used when there is explicit solvent (e.g., water).
- DIELC:** Dielectric multiplicative constant for the electrostatic interactions.
- If DIELC .le. 0.0 then DIELC= 1.0.
- DIELC and IDIEL are coupled. For example to obtain a dielectric constant of 4 r_{ij} set DIELC=4 and IDIEL=0. Default 1.0.
- CUT:** The primary cutoff distance for non-bonded interactions.
- CUT should be no more than half the shortest BOX dimension in order to maintain spherical symmetry in the nonbonded potential. Note that ROAR uses both residue-based and atom-based cutoffs. For the former this means that if any atom of one residue is within CUT of any atom of another residue, every atom in each residue will see every atom of the other residue. This is done in order to avoid splitting the residue dipoles. The average effective cutoff is thus increased by the average residue diameter. If Ewald sum is used, the pairlist is generated using the atom-based cutoff. Default = 9.0Å.
- NTNB:** Non-bonded pairlist.
- = 0 no pairlist will be generated and no non-bonded interactions are calculated.
 - = 1 normal behavior (default; recommended).
- NSNB:** The non-bonded pairlist will be updated every NSNB steps.
- It is recommended that the pairlist be updated every 10 timesteps, but for very mobile systems or when short cutoffs are used it may be necessary to update the pairlist more frequently. If the non-bonded cutoff is larger than the system size (*i.e.*, no cutoff), you should set NSNB to a large value so that the pairlist is only constructed once. Default 10.
- NTID:** Water pairlist method.
- = 0 In periodic systems the water pairlist is generated from oxygen coordinates. Default and recommended.
 - = 86 pairlist generated on residue basis from atom coordinates. *i.e.* if any pair of atoms in different waters is within the cutoff, all the interactions between the 2 waters are used. This yields an effective cutoff distance that is somewhat longer than that specified in CUT. This option may be extremely slow and is provided only for comparison to old runs.
- SCNB:** 1-4 vdW interactions are divided by SCNB. Default 2.0.

- SCEE:** 1-4 electrostatic interactions are divided by SCEE; the 1991 and previous force fields used 2.0, while the 1994 force field uses 1.2. No default; Required input parameter.
- CUT2ND:** An optional secondary cutoff.
If $CUT2ND > 0.0$, then at every nonbonded pairlist update (every NSNB steps), the energies and forces due to interactions in the range $CUT < R_{ij} \leq CUT2ND$ will be determined. These energies and forces will be added to the non-bonded interactions within CUT distance at every timestep. The idea is that long-range interactions change more slowly than short range interactions, and thus this dual cutoff method allows one to include longer-range information at only a moderate additional cost. Default 0.0.
- ICHDNA:** Option to modify the charge of end hydrogens.
This is useful for "in vacuo" simulations of RNA and DNA. Without this option, energy minimization calculations on nucleotides will result in bonding between the 5' and 3' hydrogens and the corresponding phosphate groups. This option transfers the charge from H5' to O5' so that the hydrogen on the end is neutral.
= 0 no charge modification (default)
= 1 modify charge

6.5 Nosé-Hoover Chain Temperature Coupling Parameters

- NTORP:** Flag for the type of NHC simulations to run
Required input parameter
= 2 constant temperature and volume (*i.e.* NTV ensemble)
= 3 constant temperature and pressure (*i.e.* NTP ensemble)
The box must be cubic and will be dilated isotropically
= 4 constant temperature and pressure (*i.e.* NTP ensemble)
Fully flexible box dilation
Note: NTORP must be consistent with NTB.

IF NTB = 0 or 1, NTORP must be 2
IF NTB = 2, NTORP must be 3 or 4
- NCHAIN:** Number of thermostat chains to be used. Required input parameter
In the current version, the maximum for NCHAIN is set to 3. Multiple chains can help alleviate "cold solute/hot solvent" problems. This parameter must be consistent with NTC, NTT and NTORP
= 1 One chain for entire system
 1. If NTC = 2 or 3 (SHAKE) and NTORP > 2 (Constant-P),
 2. For gas phase simulations, only single thermostat can be used with the default prep/link/edit procedures. With some special manipulations, one can use two thermostats.
= 2 Two chains. This option should be used:

1. If $NTORP = 2$
For MD: one chain for solute, one chain for solvent
For CP: one chain for QM region, one chain for MM region
2. If $NTORP = 3$ or 4 and $NTC = 1$ (i.e, no SHAKE),
For MD: one chain for the entire system, one for MD box

= 3 Three chains. This option should be used:

1. If $NTORP = 3$ or 4 and $NTC = 1$ (i.e, no SHAKE),
For MD: one chain for solute, one for solvent, one for MD box

MTSTAT(I): Flag indicates the number of thermostats on I'th chain. Required input parameter. Example:

$MSTAT(1) = 2, MSTAT(2) = 2$

A number between 1 and 10 can be used although anything more than 3 makes little difference in the results.

NY, NC: Parameters related to the reversible NHC integrator. Optional input parameter

NY: Allowed values are 3 and 5, default value = 5

NC: Should be set to 1.

NHCPAR, QMASS, VMASS, QFACTOR, WFACTOR

When using the NHC algorithm the system is coupled to a heat bath through chain(s) of thermostats. The thermostats are part of the extended system. The dynamics of each thermostat is described by a variable (here called coordinate) and its conjugate momentum. They evolve during the simulation like other degrees of freedom. The thermostat mass controls the strength of the coupling between the system and the heat bath. In a constant pressure simulation, the volume and shape of the simulation box are allowed to vary during the simulation. Similarly the barostat is used to control the pressure during a simulation and the barostat variable and its conjugate momentum can evolve during the simulation. The barostat mass regulates the dynamics of the simulation box: A smaller mass results in stronger coupling which can disrupt the dynamics of the system if it is too small, while weaker coupling leads to inefficient control of the temperature and pressure, thus, longer simulation times are needed to reach equilibrium. Of course, the proper choice of the thermostat and barostat masses are essential to achieve an efficient sampling of phase space. The dynamics of the system are independent of the thermostat or barostat mass. These values do not have to be exact and a wide range of values will work. In the original papers, the authors discussed how to choose these parameters.^{13,59} The thermostat mass is proportional to the number of degrees of freedom (N_f) connected to it and the temperature of the simulation. It is also inversely proportional to the characteristic frequency (ω_t) of the system. Similarly the barostat mass is related to the frequency of the volume fluctuation ω_b .

$$\begin{aligned}
 Q_{mass}(1) &= \frac{N_f kT}{\omega_t^2} \\
 Q_{mass}(2) &= \frac{kT}{\omega_t^2} \\
 Q_{mass}(MTSTAT) &= \frac{2kT}{\omega_t^2} \\
 V_{mass} &= \frac{N_f kT}{\omega_b^2}
 \end{aligned}
 \tag{20}$$

Thus, for a biomolecular system, typically the dynamics cover a wide frequency range. It is not very straightforward to select the masses for the thermostats. Also the frequency depends on the specific simulation condition such as if SHAKE is used, *etc.*

For the convenience of the users of ROAR, we developed a scheme which sets the thermostat and barostat masses automatically after reading in the system information and the simulation conditions. This scheme has been tested on several types of systems such as a protein in the gas phase, a protein solvated by water, a membrane bilayer, and pure water and methanol. Many systems of biological application will have similar spectra of dynamics. Thus, the parameters determined this way should be suitable for a wide range of systems. However, this scheme may fail for some systems with special properties. We also provide a remedy should this occur.

Note: The dimension of QMASS and VMASS are (mass*length)**2. The values are in reduced units. In another words, they are consistent with the AMBER unit convention (*i.e.*, energy (kcal/mol), time (ps/20.45), length(Å)).

NHCPAR: Parameter indicates how the thermostat and barostat masses are set Default value = 0. Required input parameter.

= 1 Input of QMASS and VMASS are required. This option is for those who wish to experiment with their own parameters

= 0 Input of QMASS and VMASS are not required. They will be set automatically.

The values will be printed in the mdout file.

If the program halts and complains about a SHAKE and RATTLE failure, it is possible that the thermostat/barostat masses are not appropriate. Of course, there are many other reasons the program will halt. Check all other possibilities. If it still prints out the same error message, try the following.

For a constant temperature run:

1. Instead of using default QFACTOR (see following), double its value, and put it in to the input file
2. run ROAR again, if it fails again, repeat step (1), but triple the default QFACTOR value. Keep increasing the value until the simulation runs.

For a constant pressure run, one needs to find out whether QFACTOR or WFACTOR (or both) is(are) inappropriate. First run a constant temperature simulation, make sure QFACTOR is OK. Then follow similar steps to adjust WFACTOR.

QFACTOR: Scaling factor for the thermostat masses

Not required unless the default value fails

QMASS(I,J): Mass of I'th thermostat on J'th chain. Required for both NTV and NTP simulations

Data type: real two dimensional array

Required input if NHCPAR = 1

Note: For a restart run initiated from a restart file written by ROAR, QMASS will be read in from the restart file. Input from "mdin" is not required. However, if you do wish to use a different QMASS, which is not recommended, you may do so by setting NTX < 7.

Note: this line is repeated for each thermostat on each chain. For example if there are 2 chains with 3 thermostats each, the line will be repeated 6 times to define each thermostat.

Note: QMASS strongly depends on the characteristic frequency of the system. A set of working parameters may not work if one changes from the NTC=2 to NTC=3 option, for example.

- WFACTOR:** Scaling factor for the barostat mass. For constant pressure run only.
Data type: Real. Default value = 0.3
Not required unless the default value fails.
- VMASS:** Barostat mass. Required for constant pressure simulations only.
For a restart run initiated from a restart file written by ROAR, VMASS will be read in from the restart file. Input is not required.
- PEXT:** The desired pressure for the system to maintain (in units of bar, where 1 bar = 1 atm)
Required input if NTORP = 3 or 4. Default = 1 bar.
- TEMP0:** The desired temperature for the system to maintain (in units of K)
Required input for both NTV and NTP simulations. Default = 300 K.
Note: TEMP zero not letter O.
- TEMPI:** Initial temperature.
For INIT = 3, the initial velocities are assigned from a Maxwellian distribution at TEMPI K. If TEMPI = 0, the initial velocities will be assigned at 10 K. TEMPI has no effect if INIT = 4. Default = 10 K.
- NTT:** Flag for temperature control. This parameter should be consistent with NCHAIN and NTORP.
The function of this flag is to notify the program to adjust the degree of freedom "count" according to the temperature controlling method which is specified by NCHAIN. This parameter should be consistent with NTC and NCHAIN
- = 1 this option is used
1. If NCHAIN = 1 and NTORP = 2, 3, 4
i.e., entire system (all atoms) are connected to the same thermostat chain
 2. If NCHAIN = 2 and NTORP = 3 or 4
i.e., all atoms are connected to one chain and the box to another chain
- = 5 This option is used
1. If NCHAIN = 2 and NTORP = 2
i.e., solute (or QM region) are connected to one thermostat chain, and solvent (or MM region) to another thermostat
 2. If NCHAIN = 3, NTORP = 3 or 4 and NTC = 1.
i.e., solute is connected to one thermostat chain solvent to another thermostat chain and the simulation box to the third chain
- Note: If NTT = 5, one can specify the variable ISOLVP to redefine the last_solute_atom pointer. See ISOLVP.

ISOLVP: Last_solute_atom pointer to be used with temperature controlling methods when separate thermostat chains have been requested (NCHAIN > 1 and NTT = 5).

By default (ISOLVP = 0), the last non-TIP3P water molecule in the system is generally taken as the last_solute_atom. For example, the counterions are considered part of the "solute" by default. One could re-define the counterions to be part of the solvent by setting ISOLVP.

6.6 Ewald Sum

KMAX: Input parameter for Ewald sum.

Required input parameter.

= 0 No Ewald sum will be used.

This option has to be used for gas phase simulations. The program will halt if KMAX > 0

> 0 Ewald sum will be used. This value will be used as KMAX

Default = 4.

ALPHA: This is the decay parameter in the complementary error function used in the real space part of the total electrostatic energy.

A non-zero alpha input will override the default value.

The default value for alpha will be used if no alpha input is given. (default = 0.35)

When using Ewald sum KMAX should be set for each system the following way.

1. Determine the cutoff distance you are going to use. Since long range corrections are also included, 9Å should be sufficient. If your system size permits, use a larger (or smaller) value.
2. Start with a lower value (for example, KMAX = 4). Set NSTLIM = 0 and get the zero step energy output. Note down the EE_K value. Do this by increasing KMAX in steps (4,5,6,7,8,9,...). You will notice that EE_K is converging. You can use the MINIMUM KMAX required to obtain convergence. Again, it is up to the user to determine what degree of convergence is desirable.
3. Increasing KMAX naturally increases the number of k-vectors to be used for calculating EE_K. The number of k-vectors increases dramatically if your box is "severely" non-cubic. Hence, going for convergence up to the 4th decimal place may severely slow down the run compared to 3rd decimal place convergence.
4. The default value for the decay parameter α in equation 6 has been set such that $\text{erfc}(\alpha r)/r$ is $\sim 10^{-5}$ at $r = r_{cut}$. Simulation of very large systems with the half box length significantly greater than the nonbond cutoff may require $\text{KMAX} \gg 8$ (the default value) in order to get a convergence in $\text{EE}_K \sim 10^{-4}$. If you require only a lesser degree of accuracy, you may input an alpha lower than the default value. This will of course, result in a higher residual real space interaction at the cutoff distance and a lower k-space contribution to the electrostatic energy. However, it will converge at lower KMAX.
5. Currently the Ewald sum part is setup for maxtom=5000. If your system is larger, change maxatom in ewald.h located in ROAR and recompile. Similarly, the run will stop if the required number of k-vectors is larger than the specified limit (maxnk=5000), or if you have asked for a larger KMAX than the limit (maxkv=25). The actual values of these variables used for your system are printed in the output file. IF YOU FIND THAT YOU

ARE USING NUMBERS MUCH LOWER THAN THE SET LIMIT, edit ewald.h and reduce the limits to values closer to your requirement and recompile the code. IT IS WORTH THE EFFORT since the code with the lower memory requirement runs more efficiently, particularly on serial machines.

6.7 Multiple Time Step

- NRESPA:** Flag indicates if multiple time step method option is used.
 = 1 No multiple time step method is used. (default if coupled potential is enabled)
 = 3 Three scale multiple time step method is used. (default without coupled potential)
 Fast, intermediate, and slow forces are assigned to scale 3, 2, and 1, respectively
- NRESP2:** Parameter controls the size of time step for the intermediate timescale if multiple time step method is used.
 (i.e. $dt_2 = dt/NRESP2$).
- NRESP3:** Parameter controls the size of time step for the fast timescale if multiple time step method is used.
 (i.e. $dt_3 = dt/NRESP2/NRESP3$).
- NHCOP:** Parameter controls the timescale of Nosé-Hoover chain variables.
 Default = 1: NHC variables are upgraded in the slow scale.
- NSCBH,NSCBA:** Parameters indicate the time scale of forces due to X-H bonds and all other bonds, respectively.
 Default = 3: the interactions are in the fast scale.
- NSCAH,NSCAA:** Parameters indicate the time scale of forces due to angles involving hydrogen atoms and all other angles, respectively.
 Default = 2: the interactions are in the intermediate scale.
- NSCPHH,NSCPHA:** Parameters indicate the time scale of forces due to torsion angles involving hydrogen atoms and all other torsion angles, respectively.
 Default = 2: the interactions are in the intermediate scale.
- NBLONG:** Parameter indicates the time scale of long range electrostatic interactions (i.e. the reciprocal part of the Ewald sum or PME). Default: 1.
- NMTS:** Parameter indicates the number of time scales for the non-bonded van der Waals and real space part of Ewald/PME interactions.
 = 1: the interactions are in the same timescale
 = 2: multiple time scales are applied to the non-bonded interactions according to the distance to the interested atom. i.e. interactions within certain cutoff distance are in a shorter timescale and the interactions beyond this cutoff are in a longer timescale. (default)
- RCUTMTS(i):** Cutoff distances for the multiple time scale of non-bonded interactions. This parameter should be consistent with NMTS.
 Default $RCUTMTS(1) = CUT$ when $NMTS = 1$.
 Default $RCUTMTS(1) = 7.0\text{\AA}$ and $RCUTMTS(2) = CUT$ for $NMTS = 2$.

- NBMTS(i):** Time scale for the corresponding component of non-bonded interactions if NMTS = 2.
Default NBMTS(1) = 2, and NBMTS(2) = 1.
- SWIDTH:** Switching function is used when the multiple time step method is turned on. This parameter is the distance over which the forces are tapered to zero.
Default = 0.4Å.
- RCUSHN:** Buffer region for the pairlist.
Default = 1.0Å

	Gas phase	Solvated phase			
		No MTS			MTS
		NVT	NPT	NTP	NVT
NTB	0	1	2	2	1
NTORP	2	2	3	4	2
NRESPA	1	1	1	1	3
NTC	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3	1
KMAX	0				
IFPME	0				
NMTS	1	N/A	N/A	N/A	
NTT	1				
NCHAIN	1				
Box shape	N/A	Rectangular	Cubic	Flexible	Rectangular

Table 4: All the allowed options for ROAR 2.0 (List only some critical parameters)

6.8 Water Cap

- IVCAP:** Flag to control Cap Option.
The Cap refers to a spherical portion of water centered on a point in the solute and restrained by a soft half-harmonic potential. Caps are constructed using EDIT.
= 0 Cap will be in effect if it is passed from the parm module (default)
= 1 Cap will be activated except that the Cap atom pointer will be modified
= 2 Cap will be inactivated
- MATCAP:** The Cap atom pointer.
This is the last Non-Cap atom number. If IVCAP = 1 then the pointer passed from the PARM module will be overwritten by this number. PARM passes the NATCAP parameter which is replaced by the value in MATCAP. Default 0.
- FCAP:** The Force Constant for the Cap restraint potential.
A value of 0.0 for FCAP will result in the default force constant of 1.5.

6.9 SHAKE Bond Constraints

- NTC:** Flag for SHAKE and RATTLE to perform bond length constraints.
= 1 SHAKE and RATTLE are not performed (default)
= 2 bonds involving hydrogen are constrained
= 3 all bonds are constrained
- The size of the MD timestep is determined by the fastest motions in the system. SHAKE/RATTLE can be used to constrain the bonds in a system to remove the high frequency motions associated with the bond potential. This will allow the use of a larger timestep. It is not recommended that SHAKE be used in an energy minimization. The SHAKE algorithm is derived for MD integrators, and therefore, the equations for the coordinate corrections are not suitable for energy minimization.
- TOL:** Geometric tolerance for bond constraints in SHAKE/RATTLE. Recommended maximum: $<0.0005 \text{ \AA}$. Default 0.0005.

6.10 Frozen or Restrained Atoms

- IBELLY:** Flag for belly type dynamics.
Belly dynamics means a subset of the atoms in the system will be allowed to move, and the coordinates of the rest will be frozen. The moving atoms are specified in GROUP format at the end of all other input parameters from file "mdin". GROUP input is described in the Appendix.
= 0 No belly run (default).
= 1 Belly run.
- NTR:** Flag for restraining specified atoms in Cartesian space using a harmonic potential.
= 0 No position restraints (default).
= 1 MD with restraint of specified atoms.
- Note: the restrained atoms are read in GROUP format after the other parameters from file "mdin" - see Appendices for GROUP. The coordinates are read in "restr" format from the "refc" file (see NTRX, above).
- NTRX:** Format of the Cartesian coordinates for restraint from file "refc".
= 0 Unformatted (binary) form.
= 1 Formatted (default) form.
- Note: the program expects file "refc" to contain coordinates for all the atoms in the system. A subset for the actual restraints is selected by the GROUP input which follows.

6.11 Output

- NTX0:** Format of the final coordinates, velocities, and box size (if constant pressure run) written to file "restr". (Note: NTX zero not letter O).
= 0 Unformatted.
= 1 Formatted (default).

- NTPR:** Every NTPR steps energy information will be printed in human- readable form to files "mdout" and "mdinfo".
"mdinfo" is closed and reopened each time, so it always contains the most recent energy and temperature. Default 50.
- NTWX:** Every NTWX steps the coordinates will be written to file "mdcrd".
NTWX=0 inhibits all output. Default 0.
- NTWV:** Every NTWV steps the velocities will be written to file "mdvel".
NTWV=0 inhibits all output. Default 0.
- NTWE:** Every NTWE steps the energies and temperatures will be written to file "mden" in compact form.
NTWE=0 inhibits all output. Default 0.
- NTWXM:** The maximum number of steps that NTWX is active.
At this number of steps no more trajectories will be written to file "mdcrd". Set this to 0 to disable the limit. Default = 999999.
- NTWVM:** Analogous to NTWXM for velocities. 0 to disable. Default = 999999.
- NTWEM:** Analogous to NTWXM for energies. 0 to disable. Default = 999999.
- IOUTFM:** Format of velocity, coordinate, and energy sets.
= 0 Formatted (default).
= 1 Binary.
- NTWPRT:** Coordinate/velocity archive limit flag.
This flag can be used to decrease the size of the coordinate/velocity archive files, by only including that portion of the system of greatest interest. (E.g. one can print only the solute and not the solvent, if so desired).
Coordinate/velocity archives will include:
= 0 all atoms of the system (default).
< 0 only the solute atoms.
> 0 only atoms 1->NTWPRT.
- NHCPRNT:** Flag to print NHC related properties to output files every NTPR steps.
= 0 no output files printed. (Default).
= 1 output files printed.

There are five NHC output files and their names are set in the program. If one want to save the files, one should rename them after each run since they will be overwritten during the next run. These files are very specialized and most users will not be interested in looking at them. Therefore, the default option is not to print them out.

The following is a description of the NHC files

- nhcpress:** This file stores the instantaneous values (bar) of the pressure and its components.
The pressure components include P_{kin} , $P_{nonbond}$, P_{1-4} , P_{bond} and P_{const}
- nhcvolum:** This file stores the instantaneous values of the enthalpy, volume, box length, pressure and box temperature

nhchprim:	This file stores the instantaneous value of the extended Hamiltonian and its components (H_0 and $H_{nosé}$)
nhcaverag:	This file stores the average value (following every 10*NTPR steps) of the pressure as well as the components of pressure, box temperature, box enthalpy, box volume and box length whenever applicable
nhcrattle:	Lagrangian multipliers from SHAKE and RATTLE

6.12 Coupled Potential

NOTE: In addition to the coupled potential namelist variables listed here there are several additional lines of "GROUP type" input which must be specified after the namelist, but before any BELLY atoms have been defined. It is also possible to activate several options present in the coupled potential version of the code by including specific files in the directory being used to run the job. These will also be described here.

Three separate types of PMF/FEP runs are possible within the ROAR code. Each type of FEP/PMF run is specified by namelist flags:

1. FEP run in which the QM-MM charge interactions are scaled out (to calculate the solvation free energy).
2. Removal of the vdW cavity left by the QM atoms in a solvation free energy calculation (this could also be done with GIBBS).
3. A windows PMF run for separating two molecular systems.

IMPORTANT NOTES ON PMF's and FEP's:

- ONLY the coupled potential code currently will run PMF's and FEP's within ROAR.
- To do a windows FEP run to scale out QM-MM charge interactions, set IDOCHG=1, IDOVDW=0, and IDOPMF=0. In this case, the charge scaling factor CHGLAM will be read in, as well as the sampling step size DCHG. The FEP windows simulation is done with double-wide sampling, so the allowed values of CHGLAM are 0.0, DCHG, 2*DCHG,..., 1.0.
- To do a slow growth FEP run for zeroing out QM vdW parameters AFTER MM charges have been scaled out, set IDOCHG=1, IDOVDW=1, IDOPMF=0, and CHGLAM=0.0. The value of DCHG is ignored. The initial and final lambda values (VLAMBI and VLAMBF) should be specified for the vdW slow growth run. Normally these would be 1.0 and 0.0, but if a run has to be restarted the initial value might not be 1.0. Since no electrostatic coupling exists between the QM and MM systems, no QM SCF calculations will be done in this case. A dummy QM solute is maintained by holding the QM solute in a rigid conformation throughout the course of the simulation.
- A slow growth FEP run to zero out QM vdW parameters BEFORE MM charges have been scaled out may be accomplished in either of two equivalent ways. The user can set IDOCHG=0, IDOVDW=1, and IDOPMF=0. CHGLAM should be left at its default value of zero. Alternatively, the user can set IDOCHG=1, IDOVDW=1, and IDOPMF=0. Initial and final lambda values should be specified for the slow growth but can be left at their default values. Since electrostatic coupling exists, QM SCF calculations will be performed. If one intends to eventually zero out the MM charges, it is recommended that this option not be used, but rather option (2) listed above followed by option (3).
- To do a windows PMF run for separating two molecular systems, set IDOCHG=0, IDOVDW=0, and IDOPMF=1. Additional control data for the PMF will be read in formatted form (see below).

6.12.1 Coupled Potential Namelist Variables

- IFQT:** Flag for coupled potential run, if set must also define NQT and MODCHG.
Several additional formatted input lines must also be included after the namelist input along with several separate files within the directory where the job is running. These files and formatted lines are described below.
0 = normal run: no QM atoms (default).
1 = coupled potential run.
- NQT:** Number of quantum mechanical atoms. No default must be specified if IFQT is set to one.
- MODCHG:** Number of MM atoms that are going to have their charges modified.
Typically MM atoms around the link atoms in a QM/MM simulation have the default value of the charge changed. This variable defines how many such atoms are included in the calculation as these will be read in during the formatted input section.
Default = 0.
- IDOCHG:** Couples MM charges to lambda so that their influence can be removed during a simulation. Normally used as the first step in an "electrostatically decoupled" free energy perturbation calculation.
0 = normal run no coupling (default).
1 = option activated charges coupled to lambda.
- IDOVDW:** Couples QM vdW parameters to lambda so that their contribution to the energy of a system can be calculated. Normally used as the second step in a two step "electrostatically decoupled" free energy of "solvation" type calculation.
0 = normal run no coupling (default).
1 = option activated charges coupled to lambda.
- IDOPMF:** Flag for PMF run. If IDOPMF has been set to 1 then the definition of the PMF system is read in during the formatted reads (described below). The atoms defined are the ones moved slowly apart over the run. They are moved in the direction specifically defined in this same section of formatted input.
0 = normal run: no PMF (default).
1 = option activated (program will look for PMF input)
- CHGLAM:** Initial charge scaling factor used with IDOCHG. This value is changed at each step by DCHG.
Default = 1.0
- DCHG:** Value by which CHGLAM is changed at each step during the FEP simulation.
Default = 0.0
- VLAMBI:** Initial value of lambda for slow growth runs (only available for charge and vdW scaling and not available for PMFs).
Default = 1.0

- VLAMBF:** Final value of lambda for slow growth runs (only available for charge and vdW scaling and not available for PMFs).
Default = 0.0
- NPERT:** The number of groups of atoms that will be moved in the PMF. Must be specified if a PMF run is being done.
- NEQUIL:** Number of equilibration steps. Must be specified for any coupled potential run. NOTE: The variable NSTLIM should be set larger than NEQUIL + NSAMPL. Also important is that coupled potential jobs should normally be done within one "run" (NRUN=1). While this is not an absolute requirement it is suggested.
- NSAMPL:** Number of sampling steps. Must be specified for any run where IDOVDW, IDOCHG, or IDOPMF is equal to one.
Default = 0.

6.12.2 Coupled Potential Formatted Input

*****Lines specific to PMF runs*****

The next five lines are repeated NPERT times to define each PMF group if a PMF run is being done. If this is not the case, then these lines should be omitted.

Line #FEP-1 1) IABOND

- IABOND:** If the PMF is for a bond, then this should be the atom in group i that participates in the bond. If the PMF is for the movement of the center of mass, then this should be zero.

Line #FEP-2 1) SXCM 2) SYCM 3) SZCM

Data type: real, free format

- SXCM:** X component of starting center of mass for group i, or, if IABOND(i)≠0, the starting coordinates of the atom IABOND(i).
- SYCM:** Y component of COM.
- SZCM:** Z component of COM.

Line #FEP-3 1) PX 2) PY 3) PZ

Data type: real, free format

- PX:** X component of step for moving group i. It is assumed that this corresponds to separating the groups.
- PY:** Y component of step.
- PZ:** Z component of step.

Line #FEP-4 1) IATMS(1) 2) IATMS(2) 3) etc.

Data type: real, free format

IATM: The next call reads in the list of atoms in group i. The list may take up as many lines as necessary and ranges of atoms may be indicated with a slash, e.g., 1/5 means 1,2,3,4,5. Terminate list with a blank line.

Line #FEP-5 Blank line Format->columns 1-5 ='blank'

The above 4 lines must be followed by a "blank" line. Due to unfortunate machine dependence problems, the line must actually say "blank" instead of just having nothing on it. The word blank must appear just as it does here, "blank", with no capital letters and appearing in columns 1-5.

*****General formatted lines for all coupled potential runs*****

QM atoms are defined in the next series of lines along with the identities of MM atoms which are to have their charges modified.

Line #CP-1 1) LABELS(1) 2) LABELS(2) 3) etc. Format-> Real, Free Format

LABELS: the QM atom numbers.

Line #CP-2 1) ICHG 2) TMPCHG Format-> 1 integer and 1 real in Free Format

This line is repeated MODCHG times.

ICHG: atom number for which charge is to be changed.

TMPCHG: new charge.

NOTE: Any belly atom input follows directly after the last line of input (CP-2) in the normal AMBER group format.

6.12.3 Control Files for Coupled Potential Simulations

FILENAME	DESCRIPTION	REQUIREMENTS
keyword	MOPAC specific commands	Essential

To run a coupled potential job, it is necessary to include a file named "keyword" within the directory in which the job is running. This file contains MOPAC specific commands which would normally be included on the MOPAC command line. Below we list some of the commands which can be included along with those that MUST be included (these are clearly marked). This file should contain one line with the MOPAC commands listed sequentially. It is strongly suggested that the user examine the example in the roar/demos/coupled directory and modify as needed.

Available keywords:

GEO-OK: overrides normal MOPAC geometry checks. Required
ISCF: do only one SCF cycle for every AMBER geometry. Required
GRAD: calculate gradients even though only one SCF. Required
ANALYT: calculate analytical gradients. Required
PRECISE: determine accuracy of SCF calculation. Suggested
CHARGE: defines charge of QM portion of system.

PM3: use PM3 Hamiltonian.
MNDO: use MNDO Hamiltonian. Default
AM1: use AM1 Hamiltonian.
MULLIK: calculate Mulliken, CM1, and CM2 charges. Optional

FILENAME	DESCRIPTION	REQUIREMENTS
cntrl.dat	ESP charge calculation	Optional

It is possible to calculate the ESP charges for the QM atoms within the field of charges of the MM atoms. The only input necessary to cause the calculation of ESP charges is the inclusion of the file "cntrl.dat" within the directory in which the job is running. The calculation of ESP charges is fairly computationally intensive depending on the size of the system and speed of the workstation used. For a large QM system with many MM atoms on an average workstation, several hours of computational time might be expected. The file "cntrl.dat" needs only to include two integers on the first line of the file (free format). These integers correspond to:

INESP: First MD step on which to calculate ESP charges. Required.
ISTESP: Number of MD steps between ESP charge calculation. Required.

FILENAME	DESCRIPTION	REQUIREMENTS
move1.dat	Internal coordinate constraint algorithm	Optional

The presence of this file acts as little more than a flag as it need not contain any data. If the file "move1.dat" is present in the directory in which the job is being run, then the internal coordinate constraints will be enforced by moving only the first atom in each internal coordinate list. This is in contrast to the normal algorithm which strives to achieve a constraint value by moving both ends of the internal coordinate equally.

FILENAME	DESCRIPTION	REQUIREMENTS
cutoff.dat	Secondary cutoffs for QM atoms	Optional

It is possible to define a secondary cutoff for QM atoms within a coupled potential run. As discussed in the theory section of the manual, the energy of the QM portion of the system is highly dependent on the cutoff used. The energy can also be greatly affected by residues moving in and out of a residue based cutoff (particularly charged residues). It is therefore often advisable to use a longer cutoff for the QM residues to minimize this effect while not greatly increasing the cost of the calculation. To specify a secondary cutoff the file "cutoff.dat" should be included in the directory in which the job is being run. This file has free format lines for each residue for which there is to be a secondary cutoff. Each line should contain the residue number and the new cutoff (*i.e.*, ICUTI,RCUTI).

ICUTI: Residue number for which there is to be an increased cutoff.
RCUTI: New cutoff for that residue.

FILENAME	DESCRIPTION	REQUIREMENTS
constraint.dat	QM constraints	Optional

Geometry constraints can be imposed on the QM portion of the system. This is done by reading in geometric constraint data for the QM portion of the system from the file 'constraint.dat'.

If the file is not present, then no QM constraints are assumed. The routine returns with `ierror=0` if constraint data is in the proper format. Otherwise `ierror=1`. A check is also made for the file 'move1.dat'. If this file is present (it need not contain any data), then the internal coordinate constraints will be enforced by moving only the first atom in each internal coordinate list.

Each constraint takes up one line in 'constraint.dat'. For the I 'th constraint, the I 'th line in the file should have one of the three following (free-field) formats:

```
(IABCD(J,I),J=1,4), ICODE(I)      (ICODE(I)=0)
or
(IABCD(J,I),J=1,4), ICODE(I), QCNSTR(I)      (ICODE(I)=1)
or
(IABCD(J,I),J=1,4), ICODE(I), (IABCD0(J,I),J=1,4)      (ICODE(I)>1)
```

Here, an internal coordinate is defined by the atoms IABCD(1,I), IABCD(2,I), IABCD(3,I), AND IABCD(4,I):

```
IABCD(1,I)-IABCD(2,I) = bond length specifier
IABCD(1,I)-IABCD(2,I)-IABCD(3,I) = bond angle specifier
IABCD(1,I)-IABCD(2,I)-IABCD(3,I)-IABCD(4,I) = Dihedral angle specifier
```

If the internal coordinate is a bond length or bond angle, then zeroes should be entered for IABCD(3,I) and/or IABCD(4,I).

The value of ICODE(I) determines whether the first, second, or third format is used. If ICODE(I)=0 then the internal coordinate is constrained to be equal to its initial value for the duration of the simulation. If ICODE(I)=1, then the coordinate is constrained to be equal to CONSTR(I). If ICODE(I) is greater than one, the coordinate is constrained according to a reference coordinate defined by atoms (IABCD0(J,I),I=1,4). Here again, the manner in which the constraint is defined depends on the value of ICODE(I):

```
ICODE(I) = 2 —> IABCD COORDINATE = IABCD0 COORDINATE
ICODE(I) = 3 —> IABCD DIHEDRAL = NEGATIVE OF IABCD0 DIHEDRAL
ICODE(I) = 4 —> IABCD DIHEDRAL = IABCD0 DIHEDRAL + 180 DEGREES
ICODE(I) = 5 —> IABCD DIHEDRAL = IABCD0 DIHEDRAL + 120 DEGREES
ICODE(I) = 6 —> IABCD DIHEDRAL = IABCD0 DIHEDRAL - 120 DEGREES
```

NOTE: that the atom numbering should correspond to the AMBER convention and bond lengths are specified in angstroms, while bond angles and dihedrals are in degrees.

7 Files used

Usage: `roar -i mdin -o mdout -p prmtop -c inpcrd -r restrt -ref refc -x mdcrd -v mdvel -e mden -inf mdinfo`

The following is a list of all files used or created by ROAR:

file	unit	in/out	purpose
mdin	5	in	control data for the min/md run
prmtop	8	in	molecular topology, force field, periodic box type, atom and residue names
inpcrd	9	in	initial coordinates and (optionally) velocities and periodic box size
refc	10	in	(optional) reference coordinates for position constraint
mdout	6	out	user readable state information and diagnostics
mdinfo	7	out	latest mdout-format energy information
restrt	16	out	final coordinates, velocity, and box dimensions if any - for restarting run
mdcrd	12	out	coordinate sets saved over trajectory
mdvel	13	out	velocity sets saved over trajectory
mden	15	out	extensive energy data over trajectory
keyword		in	coupled potential MOPAC input
cutoff.dat		in	coupled potential secondary cutoff
move1.dat		in	coupled potential constraint
constraint.dat		in	coupled potential constraint
etot.out		out	coupled potential rollin, rollout and total energies
nhcpress	20	out	instantaneous values of pressure and its components
nhcvolum	21	out	instantaneous value of enthalpy, volume, box length, pressure and box temperature
nhchprim	22	out	instantaneous value of extended Hamiltonian
nhcaverag	23	out	average values of some thermodynamic properties
nhcrattle	24	out	Lagrangian multipliers from SHAKE and RATTLE

8 Demos

There are four demos that are included with ROAR and are all located in the directory `roar/demos` which include a water box, tripeptide in water, lipid bilayer, and a coupled potential simulation. All of the files required to run these demos are included in each of the four appropriately labeled directories. Also included in each directory are script files to run each particular demo. Once the source code has been compiled, all that is required to run each demo is to type the script's name. For example, to run the water demo, type `run_water`. Output files are provided for each demo which have already been run using an SGI with an IRIX 6.3 operating system. These files are included so that the user will have output files to directly compare their results to those provided with each demo. It should be noted that the results may differ slightly depending on the type of machine in which the calculations are being run (*i.e.* the results from running on an SGI may vary slightly when compared to those run on an IBM). It is recommended that the user go through each demo to become more familiar with the options available in ROAR.

8.1 Water

This demo contains a fairly simple example of running a molecular dynamics (MD) simulation with ROAR. In this example, a pure liquid simulation is carried out on a system consisting of a box of TIP3P water molecules. This demo is designed to demonstrate some of the basic options available when running MD using ROAR.

In this simulation, there are a few important points to note:

- A box of TIP3P water was utilized through the solvation of a dummy particle.
- The liquid simulation was carried out using constant temperature and volume to illustrate the Nosé-Hoover Chain (NHC) approach for maintaining the temperature of the system.
- The simulation was carried out using the SHAKE algorithm to constrain all bonds.
- The simulation incorporated the Ewald sum method for calculation of long range electrostatic interactions. Also included are the long range vdW corrections for the energy, virial, and pressure.
- The simulation is run for 1000 steps with energy information being written to the output file md.out every 50 steps. The box of waters was initially minimized, however the results are not shown in this demo. Coordinates and velocities are written only once at the end of the simulation.

In order to carry out the MD liquid simulation as just described, a number of variables in the input file unique to this type of calculation need to be set. A listing of these parameters and their specific values for this job are listed below. Only those parameters which differ from the default values are discussed.

- NTORP = 2:** The flag indicates that the simulation is a constant temperature and volume simulation.
- NCHAIN = 1:** One thermostat chain is used in this simulation to maintain the temperature. This chain is for the solvent only since there is no solute in the box.
- MTSTAT(1) = 3:** These are the number of thermostats for each chain.
- NHCPAR = 0:** The values for QMASS are to be set automatically. If NHCPAR = 1, values for QMASS have to be set manually.
- NTT = 1:** This value is set since there is only one thermostat chain and the entire system is connected to this same thermostat chain.
- NSTLIM = 1000:** The number of MD-steps per NRUN to be performed. The length of the simulation is NSTLIM*NRUN*DT.
- NTB = 1:** A constant volume simulation is performed.
- TEMP0 = 300:** This flag indicates that the temperature of the system is to remain 300K for the duration of the simulation.

The following two flags correspond to the Ewald sum method:

- KMAX = 4:** This flag is set when the Ewald sum method is to be performed. If $KMAX > 0$, an Ewald sum calculation is performed. Refer to the manual for setting of the appropriate value for this variable.
- ALPHA = 0.35:** If no Ewald sum calculation is performed, set ALPHA = 0. Otherwise use the default value of 0.35.

The input files which are supplied in this demonstration are md.inNVT, parm.top, and parm.crd. The only output file that is included is md.out.

8.2 Tripeptide in Water

This demo contains a slightly more complex example of running a MD simulation within ROAR. In this example, molecular dynamics is used as a tool for studying the molecular level details of simulating a small solute in a box of TIP3P water. The system consists of the tripeptide Asp-Thr-Ala and demonstrates many of the available options which can be applied when running MD using ROAR. This example can be applied to much larger macromolecular systems such as proteins and enzymes as well.

In this simulation, there are a few important points to note:

- The system consists of the tripeptide Asp-Thr-Ala solvated by 261 TIP3P water molecules.
- The AMBER all atom force field was used for the representation of the model.⁵⁵
- The liquid simulation was carried out using constant temperature and pressure to illustrate the Nosé-Hoover Chain (NHC) approach for maintaining the temperature and pressure of the system.
- The simulation was performed using the SHAKE algorithm which constrains only X-H bonds in the system.
- The system utilized periodic boundary conditions and was allowed to dilate isotropically.
- The simulation incorporated the Ewald sum method for calculation of long range electrostatic interactions. Also included are long range corrections for the energy, virial, and pressure.
- The simulation is run for 1000 steps with energy information being written to the output file md.out every 50 steps. The system was initially minimized, however the results are not shown in this demo. Coordinates and velocities are written only once at the end of the simulation.

In order to carry out this aqueous MD simulation of the tripeptide as just described, a number of variables in the input file unique to this type of calculation need to be set. A listing of these parameters and their specific values for this job are listed below. Only those parameters which differ from the default values are discussed.

NTORP = 3: The flag indicates that the simulation is a constant temperature and pressure with isotropic box dilation.

NCHAIN = 1: One thermostat chain is used in this simulation to maintain the temperature, pressure, and box dimension.

MTSTAT(1) = 3: These are the number of thermostats for each chain.

NHCPAR = 1: The values for QMASS are required and must be explicitly specified in the input file. Normally this is set to the default value of 0 which sets QMASS and VMASS automatically.

QMASS(1,1) = 100.0:

QMASS(2,1) = 0.100:

QMASS(3,1) = 0.20: These are the thermostat masses which are required for all simulations. Three values are needed since MTSTAT(1) = 3. It is recommended that these values be set automatically by fixing the value NHCPAR = 0.

VMASS = 1000: This is the barostat mass and is required for all constant pressure simulations. It is recommended to set this value automatically by fixing the value NHCPAR = 0.

- PEXT = 1.0:** The reference pressure at which the simulation is maintained.
- NTT = 1:** This value is set since there is only one thermostat chain and the entire system is connected to this same thermostat chain.
- NSTLIM = 1000:** The number of MD-steps per NRUN to be performed. The length of the simulation is NSTLIM*NRUN*DT.
- NTB = 2:** A constant pressure simulation is performed.
- TEMPO = 300:** This flag indicates that the temperature of the system is to remain 300K for the duration of the simulation.
- NTC = 2:** This flag is for SHAKE to perform bond length constraints. In this demo, all X-H bonds within the system are constrained.

The following two flags correspond to the Ewald sum method:

- KMAX = 7:** This flag is set when the Ewald sum method is to be used. If KMAX > 0, an Ewald sum calculation is performed. Refer to the manual for setting of the appropriate value for this variable.
- ALPHA = 0.35:** If no Ewald sum calculation is performed, set ALPHA = 0. Otherwise use the default value of 0.35.

The input files which are supplied in this demonstration are md.in, parm.top and pep.crd. The only output file that is included is md.out.

8.3 Lipid Bilayer

This demo contains a more complex example of running a molecular dynamics (MD) simulation within ROAR. In this example a bulk phase lipid bilayer simulation is carried out in order to gain understanding of the structure and dynamics of this system. This demo is designed to provide further examples of the options within ROAR.

In this simulation, there are a few important points to note:

- The bilayer water system consists of dilauroylphosphatidylethanolamine (DLPE) lipids and SPC/E water molecules in a liquid crystalline phase, thus making it an ideal system to demonstrate the use of the NHC fully flexible box model.
- The AMBER united atom force field was used in the representation of the system.⁵⁵
- The bulk phase simulation was carried out using constant temperature and pressure to illustrate the Nosé-Hoover Chain (NHC) approach for maintaining the temperature, pressure, and box dimension of the system.
- The simulation was carried out using the SHAKE algorithm to constrain all bonds.
- The simulation incorporated the Ewald sum method for calculation of long range electrostatic interactions. Also included are long range corrections for the energy, virial, and pressure. The Ewald sum method is ideal for simulations involving infinitely repeating units through periodic boundary conditions such as lipid bilayers.
- The simulation is run for 500 steps with energy information being written to the output file md.out every 20 steps. The box was initially minimized and then equilibrated at constant volume, resulting in the starting structure which is used in this example. Coordinates and velocities are written only once at the end of the simulation.
- Since this simulation utilizes a fully flexible box dilation, the initial shape of the box can be rectangular rather than cubic.

In order to carry out the MD bulk phase simulation as just described, a number of variables in the input file unique to this type of calculation need to be set. A listing of these parameters and their specific values for this job are listed below. Only those parameters which differ from the default values are discussed.

- NTORP = 4:** The flag indicates that the run is a constant temperature and pressure, fully flexible box simulation.
- NCHAIN = 1:** One thermostat chain is used in this simulation to maintain the temperature, pressure, and box dimension. Also, since SHAKE is used to constrain all bonds, NCHAIN must be set to 1.
- MTSTAT(1) = 3:** These are the number of thermostats for each chain.
- NHCPAR = 0:** The values for QMASS are set automatically. If NHCPAR = 1, values for QMASS are set manually.
- NTT = 1:** This value is set since there is only one thermostat chain and the entire system is connected to this same thermostat chain.
- NSTLIM = 500:** The number of MD-steps per NRUN to be performed. The length of the simulation is NSTLIM*NRUN*DT.
- NTB = 2:** A constant pressure simulation is performed.
- TEMPO = 300:** This flag indicates that the temperature of the system is to remain 300K for the duration of the simulation.

The following two flags correspond to the Ewald sum method:

- KMAX = 7:** This flag is set when the Ewald sum method is to be performed. If $KMAX > 0$, an Ewald sum calculation is performed. Refer to the manual to determine the appropriate value for this variable.
- ALPHA = 0.35:** If no Ewald sum calculation is performed, set ALPHA = 0. Otherwise use the default value of 0.35.

The input files which are supplied in this demonstration are md.in, parm.top, and parm.crd. The only output file that is included is md.out.

8.4 Coupled Potential - Carbonic Anhydrase

This demo contains a fairly complicated example of running a coupled potential job within ROAR. In this example the QM/MM method is used as a tool to understanding the molecular level details in the catalytic cycle of the enzyme human carbonic anhydrase II (HCAII). Although the input of most coupled potential jobs will be considerably simpler, we are attempting here to demonstrate many of the available options simultaneously.

In this simulation there are several important points to note:

- TIP3P water molecules were added using the AMBER CAP option to solvate the active site and are retained there using a harmonic force through out the simulation.
- Only a select number of residues were simulated quantum mechanically due to the expense incurred with each additional QM residue. The QM residues chosen in this system were the imidazoles complexed to the zinc (coming from His-94, -96, and -119), the active site zinc itself, and the hydroxide which sits in the fourth ligand site. The PM3 Hamiltonian was used for the QM portion of the system.

- All of the remaining MM atoms were simulated using the AMBER united atom force field. While this may seem unusual, it is mostly for historical reasons, as our group has accumulated many simulations on this system using the united atom force field. The all atom force field is completely supported within the coupled potential code and is recommended over the united atom force field.
- The junction between the QM and MM regions was made between the C- β and C- γ of the His residues and capped with hydrogen atoms so that a closed shell calculation is done.
- The belly option was applied to all atoms within 15Å of the active site so that only atoms within this radius were allowed to move during the MD simulations.
- The temperature of the system was maintained using the Nosé-Hoover Chain (NHC) method coupling one thermostat to the QM portion of the system and one thermostat to the MM region.
- One window of a PMF is being carried out in this simulation with the quantum mechanical hydroxide being moved away from the zinc atom.

In order to carry out the coupled potential PMF simulation as just described, a number of variables in the input file unique to this type of calculation need to be set. A listing of these parameters and their specific values for this job are listed below. Only those parameters which do not contain default values are discussed. The proper settings for the NHC temperature coupling parameters are also discussed.

- NTORP = 2:** Only simulations with constant temperature can be used with the coupled potential calculations (NO CONSTANT PRESSURE SIMULATIONS WITH QM/MM at this time).
- NCHAIN = 2:** One thermostat pertains to the MM portion of the simulation while the second thermostat is associated with the QM region. The degree of freedom assignments are done automatically.
- MTSTAT(1) = 3:** These are the number of thermostats for each chain.
- MTSTAT(2) = 3:**
- NHCPAR = 0:** The values for QMASS are set automatically (see explanation as described earlier in the manual).
- NTT = 5:** This value is set since there are two thermostat chains being incorporated in the simulation (one for the MM region and one for the QM region).

The second set of parameters which are important in this simulation are those which dictate the coupled potential portion of the simulation. These are as listed below as well.

- IFQT = 1:** This flag indicates that a coupled potential calculation is performed.
- NQT = 30:** There are 30 atoms in this simulation which will be treated quantum mechanically. These atoms are listed later in the input file following the PMF information. This number does NOT include the number of link atoms.
- MODCHG = 9:** There are nine MM atoms located around the link atoms which undergo a charge modification. These are also listed later in the input file and follow after the listing of the QM atoms.
- IDOPMF = 1:** This signals that a PMF calculation is to be performed.

NEQUIL = 200: The calculation is to be run with 200 steps of equilibration.

NSAMPL = 200: The calculation is to be run with 200 steps of sampling.

Since a PMF calculation is carried out in this simulation, information concerning the atoms or groups of atoms which are being moved is necessary. This input directly follows the namelist variables in the input file as shown in the file `qmmm.in` of this demo. In this calculation, the atom being displaced is the hydroxide ion. The initial starting coordinates of this atom as well as the distance the atom is moved are input. The remainder of the file consists of a listing of the QM atoms, followed by those atoms whose charges are changed from their default AMBER values due to their proximity to the QM atoms. The final group of residues listed in the input concern the belly option which will not be described here (see AMBER 5.0 for a detailed explanation).

There are additional input files which are required when running a coupled potential simulation. Examples of those files which are used in this demo are illustrated below.

keyword: The keyword file contains MOPAC commands for running the QM portion of the simulation. The commands used in this simulation are fairly general and can be used with most simulations.

constraint.dat: The `constraint.dat` file contains geometry constraints which can be imposed on the QM portion of the simulation. In this demo, one of the atoms constrained is the zinc atom in the active site.

cutoff.dat: The `cutoff.dat` control file allows for secondary cutoffs for QM atoms in order to minimize the number of residues moving in and out of the residue based cutoffs. Without this secondary cutoff, large effects in the energy associated with the QM region may arise.

move1.dat: this is included for this particular PMF run as we only want the hydroxide coordinate moved and not both hydroxide and zinc moved equally.

8.4.1 Interpreting Coupled Potential Output

The output in "mdout" from a coupled potential run will appear much like the output from any other simulation done using ROAR. There are, however, several additional lines of output in the beginning of "mdout" which describe the input options used in the coupled potential run. These include any constraints used as well as information concerning a PMF if this type of simulation is being run. Double checking this output against what you believe the job should be doing according to your input is highly suggested. Due to the complicated nature of the coupled potential input, mistakes can be easily made and this simple precaution can greatly reduce the amount of computer time wasted on errors in the input variables. Another value which is printed in "mdout" is the SCF energy at every geometry step. This is done to give the user a feeling for the behavior of this sensitive quantity. An examination of the SCF energy at the end of a simulation is encouraged in order to check for inconsistencies.

If a PMF, VDW, or charge transformation run is performed, an additional output file will be found in the directory from which the job was started. This file (`etot.out`) contains the rollin, rollout and total energies from each of the sampling steps. This data can be used to estimate a free energy for the window being simulated in that run. The free energy is calculated using a small utility program entitled `pmf.f`, which is included in the `~roar/demos/coupled` directory. It will calculate free energies from the data contained in `etot.out`. However, it is a simple matter to write either a PEARL script, a C program, or import the data into a spreadsheet program such as Excel, so that any of these methods might also be used to calculate the change in free energy from the data.

The program `pmf` can be compiled using the FORTRAN compiler on the machine being used (*i.e.* `f77`, `xlf`, `f90`, *etc.*) and a command such as: `f77 -o pmf pmf.f`. This will compile the file `pmf.f` and create an executable program named "pmf". The program `pmf` can then be run by

typing pmf. The utility program will query for the number of steps/points/lines in this file (*i.e.* the number of sampling steps). The output will be displayed in the current window and consists of the free energies for the forward and back sampling in this window.

It is important to remember that the number obtained from the pmf.f program is just the forward and backward free energies for one window corresponding to $\lambda + \delta\lambda$ and $\lambda - \delta\lambda$. To obtain a true free energy for an entire simulation or PMF, windows corresponding to all the values of λ between 0 and 1 need to be simulated and this procedure repeated for each window. For a more complete description of PMF's and FEP's please see the AMBER manual section on GIBBS.

The input files which are supplied in this demonstration are qmmm.in, qmmm.top, and qmmm.crd. Control files which are supplied are keyword, constraint.dat, and cutoff.dat. The output files that are included are qmmm.out and etot.out. It should be noted that ROAR must be recompiled in order to include the coupled potential portion of the code (see compiling section of this manual).

9 Questions?

Send questions concerning compiling, running, and interpreting results from ROAR to:

roar@psu.edu

Send questions about running MD in general or the AMBER force field to:

amber@cgl.ucsf.edu

This is the general amber help address. Questions specific to ROAR will be routed to the ROAR address.

General questions appropriate for the entire AMBER community can be sent to the AMBER mail reflector:

amber-request@cgl.ucsf.edu

URL: <http://roar.chem.psu.edu>

References

- [1] Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Clarendon Press: Oxford, 1987.
- [2] Berendsen, H. J. C.; Potsma, J. P. M.; van Gunsteren, W. F.; DiNola, A. D.; Haak, J. R. *J. Chem. Phys.* **1984**, *81*, 3684-3690.
- [3] Nosé, S. *J. Chem. Phys.* **1984**, *81*, 511-519.
- [4] Hoover, W. G. *Phys. Rev. A* **1985**, , 1695-1697.
- [5] Martyna, G. J.; Klein, M. L. *J. Chem. Phys.* **1992**, *97*, 2635-2643.
- [6] Andersen, H. C. *J. Chem. Phys.* **1980**, *72*, 2384.
- [7] Parrinello, M.; Rahman, A. *Phys. Rev. Lett.* **1980**, *45*, 1196-1199.
- [8] Martyna, G. J.; Tobias, D. J.; Klein, M. L. *J. Chem. Phys.* **1994**, *101*, 4177-4189.
- [9] Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990-2001.
- [10] Tuckerman, M. E.; Berne, B. J.; Rossi, A. *J. Chem. Phys.* **1991**, *94*, 1465-1469.

- [11] Tuckerman, M. E.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1991**, *94*, 6811-6815.
- [12] Tuckerman, M. E.; Martyna, G. J.; Berne, B. J. *J. Chem. Phys.* **1990**, *93*, 1287-1291.
- [13] Martyna, G. J.; Tuckerman, M. E.; Tobias, D. J.; Klein, M. L. *Mol. Phys.* **1996**, *87*, 1117 - 1157.
- [14] Cheng, A.; Merz, K. M., Jr. *J. Phys. Chem.* **1996**, *100*, 1927-1937.
- [15] Smith, P. E.; Pettitt, B. M. *J. Chem. Phys.* **1996**, *105*, 4289-4293.
- [16] Warshel, A.; Levitt, M. *J. Mol. Biol.* **1976**, *103*, 227-249.
- [17] Hehre, W. J.; Radom, L.; Schleyer, P. v. R.; Pople, J. A. *Ab Initio Molecular Orbital Theory*; John Wiley & Sons: New York, 1986.
- [18] Parr, R. G. In *Annual Review of Physical Chemistry*; B. S. Rabinovitch, J. M. Schurr and H. L. Strauss, Ed.; Annual Reviews Inc.: Palo Alto, 1983; pp 631.
- [19] Pople, J. A.; Beveridge, D. L. *Approximate Molecular Orbital Theory*; McGraw-Hill: New York, 1970.
- [20] Dewar, M. J. S.; Zoebisch, E. G.; Healy, E. F.; Stewart, J. J. P. *J. Am. Chem. Soc.* **1985**, *107*, 3902-3909.
- [21] Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. *J. Comput. Chem.* **1983**, *4*, 187-217.
- [22] Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M., Jr.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*, 5179-5197.
- [23] Jorgensen, W. L.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1988**, *110*, 1657-1666.
- [24] Bowen, J. P.; Allinger, N. L. *Reviews in Computational Chemistry* **1991**, *2*, 81-98.
- [25] Burkhardt, U.; Allinger, N. L. *Molecular Mechanics*; American Chemical Society: Washington, D.C., 1982.
- [26] Field, M. J.; Bash, P. A.; Karplus, M. *J. Comp. Chem.* **1990**, *11*, 700-733.
- [27] Gao, J.; Xio, X. *Science* **1992**, *258*, 631-635.
- [28] Luzhkov, V.; Warshel, A. *J. Comput. Chem.* **1992**, *13*, 199-213.
- [29] Thompson, M. A. *J. Am. Chem. Soc.* **1995**, *117*, 11341-11344.
- [30] Thompson, M. A. *J. Phys. Chem.* **1996**, *100*, 14492-14507.
- [31] Bakowies, D.; Thiel, W. *J. Phys. Chem.* **1996**, *100*, 10580-10594.
- [32] Monard, G.; Loos, M.; Théry, V.; Baka, K.; Rivail, J.-L. *Int. J. Quant. Chem.* **1996**, *58*, 153-159.
- [33] Stanton, R. V.; Hartsough, D. S.; Merz, K. M., Jr. *J. Comput. Chem.* **1994**, *16*, 113-128.
- [34] Wesolowski, T. A.; Warshel, A. *J. Phys. Chem.* **1993**, *97*, 8050-8053.
- [35] Wei, D.; Salahub, D. R. *J. Chem. Phys.* **1994**, *101*, 7633-7642.
- [36] Tuñón, I.; Martins-Costa, M. T. C.; Milloy, C.; Ruiz-López, M. F.; Rivail, J.-L. *J. Comput. Chem.* **1996**, *17*, 19-29.

- [37] Singh, U. C.; Kollman, P. A. *J. Comput. Chem.* **1986**, *7*, 718-730.
- [38] Stanton, R. V.; Little, L. R.; Merz, K. M., Jr. *J. Phys. Chem.* **1995**, *99*, 17344-17348.
- [39] Assfeld, X.; Rivail, J.-L. *Chem. Phys. Lett.* **1996**, *263*, 100-106.
- [40] Warshel, A. *Computer Modelling of Chemical Reactions in Enzymes and Solutions*; John Wiley & Sons, Inc.: New York, 1991, pp 236.
- [41] Gao, J. In *Reviews in Computational Chemistry*; K. B. Lipkowitz and D. B. Boyd, Ed.; VCH Press: New York, 1995; Vol. 6.
- [42] Bash, P. A.; Field, M. J.; Davenport, R. C.; Petsko, G. A.; Ringe, D.; Karplus, M. *Biochemistry* **1991**, *30*, 5826-5832.
- [43] Merz, K. M., Jr.; Banci, L. *J. Phys. Chem.* **1996**, *100*, 17414-17420.
- [44] Merz, K. M., Jr.; Banci, L. *J. Am. Chem. Soc.* **1997**, *119*, 863-871.
- [45] Hartsough, D. S.; Merz, K. M., Jr. *J. Phys. Chem.* **1995**, *99*, 11266-11275.
- [46] Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4899-4907.
- [47] Dewar, M. J. S.; Thiel, W. *J. Am. Chem. Soc.* **1977**, *99*, 4907-4917.
- [48] Stewart, J. J. P. *J. Comp. Chem.* **1989**, *10*, 209-220.
- [49] Stewart, J. J. P. *J. Comp. Chem.* **1989**, *10*, 221-264.
- [50] Stewart, J. J. P. *J. Comp. Chem.* **1991**, *12*, 320-341.
- [51] Yang, W.; Lee, T.-S. *J. Chem. Phys.* **1995**, *103*, 5674-5678.
- [52] Dixon, S. L.; Merz, K. M., Jr. *J. Chem. Phys.* **1996**, *104*, 6643-6649.
- [53] White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. *Chem. Phys. Lett.* **1994**, *230*, 8-16.
- [54] Weiner, S. J.; Kollman, P. A.; Nguyen, D. T.; Case, D. A. *J. Comput. Chem.* **1986**, *7*, 230-252.
- [55] Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P. *J. Am. Chem. Soc.* **1984**, *106*, 765-784.
- [56] Gao, J. *J. Phys. Chem.* **1992**, *96*, 537-540.
- [57] Stanton, R. V.; Hartsough, D. S.; Merz, K. M., Jr. *J. Phys. Chem.* **1993**, *97*, 11868-11870.
- [58] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926.
- [59] Nosé, S.; Klein, M. L. *Mol. Phys.* **1983**, *50*, 1055-1076.
- [60] Tuckerman, M. E.; Berne, B. J. *J. Chem. Phys.* **1991**, *95*, 8362-8364.
- [61] Cheng, A.; K. M. Merz, J. *J. Phys. Chem. B* **1999**, *103*, 5396-5405.

Index

1SCF, 38

ALPHA, 29
ALPHA = 0.35, 42, 44, 45
AM1, 38
ANALYT, 38

CHARGE, 38
CHGLAM, 36
constraint.dat, 47
CUT, 24
CUT2ND, 24
cutoff.dat, 47

DCHG, 36
DIELC, 23
DRMS, 20
DT, 21
DX0, 20
DXM, 20

FCAP, 31

GEO-OK, 38
GRAD, 38

IABOND, 36
IATM, 37
IBELLY, 32
ICHDNA, 24
ICHG, 37
ICUTI, 39
IDIEL, 23
IDOCHG, 35
IDOPMF, 35
IDOPMF = 1, 47
IDOVDW, 35
IFQT, 35
IFQT = 1, 47
IFTRES, 23
IG, 19
IMGSLT, 22
IMIN, 20
INESP, 38
INIT, 21
IOUTFM, 33
ISOLVP, 28
ISTESP, 38
IVCAP, 31

keyword, 47
KFORM, 19

KMAX, 29
KMAX = 4, 42
KMAX = 7, 44, 45

LABELS, 37

MATCAP, 31
MAXCYC, 20
MNDO, 38
MODCHG, 35
MODCHG = 9, 47
move1.dat, 47
MTSTAT(1) = 3, 42, 43, 45, 46
MTSTAT(2) = 3, 46
MTSTAT(I), 25
MULLIK, 38

NBLONG, 30
NBMTS(i), 31
NC, 26
NCHAIN, 25
NCHAIN = 1, 42, 43, 45
NCHAIN = 2, 46
NCYC, 20
NDFMIN, 21
NEQUIL, 36
NEQUIL = 200, 47
nhcaverag, 34
nhchprim, 34
NHCOP, 30
NHCPAR, 27
NHCPAR = 0, 42, 45, 46
NHCPAR = 1, 43
nhcpress, 34
NHCPRNT, 33
nhcrattle, 34
nhcvolum, 34
NMTS, 30
NPERT, 36
NQT, 35
NQT = 30, 47
NRESP2, 30
NRESP3, 30
NRESPA, 30
NRUN, 21
NSAMPL, 36
NSAMPL = 200, 47
NSCAH,NSCAA, 30
NSCBH,NSCBA, 30
NSCM, 21
NSCPHH,NSCPHA, 30
NSNB, 24

NSTLIM, 21
NSTLIM = 1000, 42, 44
NSTLIM = 500, 45
NTB, 22
NTB = 1, 42
NTB = 2, 44, 45
NTC, 32
NTC = 2, 44
NTCM, 21
NTF, 23
NTID, 24
NTMIN, 20
NTNB, 24
NTORP, 25
NTORP = 2, 42, 46
NTORP = 3, 43
NTORP = 4, 45
NTPR, 33
NTR, 32
NTRX, 32
NTT, 28
NTT = 1, 42, 44, 45
NTT = 5, 46
NTU, 20
NTWE, 33
NTWEM, 33
NTWPRT, 33
NTWV, 33
NTWVM, 33
NTWX, 33
NTWXM, 33
NTX, 22
NTX0, 32
NY, 26
NY, NC, 26

PEXT, 28
PEXT = 1.0, 44
PM3, 38
PRECISE, 38
PX, 37
PY, 37
PZ, 37

QFACTOR, 27
QMASS(1,1) = 100.0, 43
QMASS(2,1) = 0.100, 43
QMASS(3,1) = 0.20, 43
QMASS(I,J), 27

RCUSHN, 31
RCUTI, 39
RCUTMTS(i), 30

SCEE, 24

SCNB, 24
SWIDTH, 31
SXCM, 36
SYCM, 36
SZCM, 37

T, 21
TEMPO, 28
TEMPO = 300, 42, 44, 45
TEMPI, 28
TIMLIM, 19
TMPCHG, 37
TOL, 32

VLAMBF, 36
VLAMBI, 36
VLIMIT, 20
VMASS, 27
VMASS = 1000, 44

WFACTOR, 27