

Manual

AMBERPLUS – version 2008/A9.46-M2008

Masahiro Higashi and Donald G. Truhlar
Department of Chemistry and Supercomputing Institute
University of Minnesota, Minneapolis, Minnesota 55455

Program version: 2008
Program version date: July 28, 2008
Manual version date: April 17, 2009
Copyright 2008 (program) and 2008-2009 (manual)

ABSTRACT:

AMBERPLUS version 2008 is a patch that incorporates the following methods into one of the modules of AMBER, in particular *sander*, which carry out energy minimization, molecular dynamics (MD), and NMR refinements. AMBERPLUS introduces:

- the TINKER tapering function for long-range electrostatic interactions
- Umbrella sampling of a difference between two bond distances
- multi-configuration molecular mechanics (MCMM) or electrostatically embedded multi-configuration molecular mechanics (EE-MCMM) method as one of the methods for the quantum mechanical (QM) region of a combined QM/MM calculation; this is accomplished by using MC-TINKER program

The TINKER tapering function makes the electrostatic interactions become zero smoothly at a cutoff distance; it can be employed in order to conserve the total energy of the system with periodic boundary conditions without using Ewald summation. The umbrella sampling of a difference between two bond distances is often used to calculate the potential of mean force along a reaction coordinate. The MCMM method enables one to generate semiglobal potential energy surfaces of reactive systems efficiently in the gas phase. The EE-MCMM method is an extension of the MCMM method; it can generate semiglobal potential energy surfaces in the presence of an electrostatic potential. In AMBERPLUS the combined EE-MCMM and molecular mechanical (EE-MCMM/MM) method is available, where the electrostatically embedded QM energy in a conventional combined quantum mechanical and molecular mechanical (QM/MM) method is replaced by the EE-MCMM potential energy. To install the AMBERPLUS version 2008, one requires AMBER version 9 with bugfixes 1-46 and MC-TINKER version 2008.

CONTENTS

TITLE PAGE AND ABSTRACT.....	1
CONTENTS.....	2
1. REQUIRED REFERENCES	3
2. INTRODUCTION.....	4
2.1. Short description of the TINKER tapering function for long-range electrostatic interactions	6
2.2. Short description of the umbrella sampling of a difference between two bond distances	8
2.3. Short description of the multi-configuration molecular mechanics (MCMM) and electrostatically embedded multi-configuration molecular mechanics (EE-MCMM) methods	9
3. DISTRIBUTION.....	12
4. INSTALLATION.....	14
5. USAGE	16
5.1 File usage	16
5.2 How to use the TINKER tapering function for long-range electrostatic interactions... ..	18
5.3 How to carry out umbrella sampling along a reaction coordinate that is a difference between two bond distances.....	19
5.4 How to use the MCMM and EE-MCMM/MM methods	22
6. SAMPLE TEST RUNS.....	26
6.1. Test run 1	27
6.2. Test run 2	28
6.3. Test run 3	29
6.4. Test run 4	30
6.5. Test run 5	31
7. BIBLIOGRAPHY	33
8. COMPUTERS AND OPERATING SYSTEMS ON WHICH THE CODE HAS BEEN DEVELOPED AND TESTED.....	35
9. REVISION HISTORY.....	37
9.1. AMBERPLUS–version 2008/A9.46–M2008 (July 2008)	37

1. REQUIRED REFERENCES

Publications based on results obtained with this computer code should include the following references:

- 1 M. Higashi, and D. G. Truhlar, AMBERPLUS–version 2008, University of Minnesota, Minneapolis, MN, 2008.
- 2 D. A. Case, T. A. Darden, T. E. Cheatham, III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, K. M. Merz, D. A. Pearlman, M. Crowley, R. C. Walker, W. Zhang, B. Wang, S. Hayik, A. Roitberg, G. Seabra, K. F. Wong, F. Paesani, X. Wu, S. Brozell, V. Tsui, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, P. Beroza, D. H. Mathews, C. Schafmeister, W. S. Ross, and P. A. Kollman, AMBER 9, University of California, San Francisco, CA, 2006.
- 3 O. Tishchenko, M. Higashi, T. V. Albu, J. C. Corchado, Y. Kim, J. Villà, J. Xing, H. Lin, and D. G. Truhlar, MC-TINKER–version 2008, University of Minnesota, Minneapolis, MN, 2008.
- 4 J. W. Ponder, TINKER–version 3.5, Washington University, St. Louis, MO, 1997.

For example: "The calculations were carried out using the AMBERPLUS¹ computer program, which incorporates some methods into the AMBER² program and interfaces the AMBER² and MC-TINKER³ programs. MC-TINKER³ is built on single-configuration molecular mechanics subroutines from the TINKER⁴ computer program."

The references are required; the wording is optional.

2. INTRODUCTION

The AMBERPLUS program is a patch that incorporates some methods into one of modules of AMBER [1], in particular *sander*, which carry out energy minimization, molecular dynamics (MD), and NMR refinements, and that interfaces the *sander* program with the MC-TINKER [2] program, which provides the potential energy surface.

One of the methods incorporated by the AMBERPLUS program is the TINKER [3] tapering method for long-range electrostatic interactions. In the original *sander* program, unless the particle mesh Ewald [4] or original Ewald method is employed, the electrostatic interactions are sharply truncated at a cutoff distance, which prevents the conservation of the total energy. In the TINKER tapering method, which is similar to the force switching method in Ref. [5], a two polynomial multiplicative-additive shifted energy switch is applied to make the electrostatic interactions become zero smoothly at a cutoff distance; this enables one to conserve the total energy.

Another capability in the AMBERPLUS program is to impose a harmonic potential on a difference between two bond distances. The AMBER version 9 program itself can impose a harmonic potential on a bond distance, an angle, or a torsional angle, but the AMBER version 9 program cannot impose such a potential on a difference between two bond distances. A harmonic potential on a difference between two bond distances can be employed to evaluate the potential of mean force by the umbrella sampling method in an S_N2 reaction [6], a proton transfer reaction [7], or hydride transfer reaction [8] because the formation and breaking of chemical bonds take place simultaneously.

The MC-TINKER program is a composite of the computer program TINKER, which is program for single-configuration molecular mechanics calculations, and the MCT module, which is a module for multi-configuration molecular mechanics calculations. The MC-TINKER program can carry out four types of calculations: calculations based on single-configuration molecular mechanics (MM or, for emphasis, SCMM), as in TINKER itself, or calculations based on multi-configuration molecular mechanics (MCMM) [9-13], calculations based on electrostatically embedded MM or SCMM (which may be called either EE-MM or EE-SCMM) [14,15] (also see Sec. 2.4 in the MC-TINKER manual) in which MM is applied to systems in the presence of an electrostatic potential, and calculations based on electrostatically embedded MCMM (EE-MCMM) [16] in which MCMM is applied to systems in the presence of an electrostatic potential. (Throughout this

manual, we use MM and SCMM interchangeably as synonyms.) In the AMBERPLUS program, the potential energy calculated by the MC-TINKER program can be employed as one of quantum mechanical (QM) methods in a QM or combined QM and MM (QM/MM) system. By replacing the electrostatically embedded QM energy in the QM/MM system with the EE-MM or EE-MCMM energy, a procedure that can be called EE-MM/MM or EE-MCMM/MM, the QM/MM potential energy surface with high-level QM calculations can be generated with low computational cost [17]. Energy minimization (geometry optimization) and molecular dynamics simulation can be carried out with the potential energy surface calculated by the MC-TINKER program.

2.1. Short description of the TINKER tapering function for long-range electrostatic interactions

In the original *sander* program, unless the particle mesh Ewald or original Ewald method is employed, the charge-charge electrostatic interaction energy V_{ab} between atom a and b is sharply truncated at a cutoff distance,

$$V_{ab}(r_{ab}) = \begin{cases} \frac{Q_a Q_b}{r_{ab}} & r_{ab} < r_{\text{cut}} \\ 0 & r_{ab} \geq r_{\text{cut}} \end{cases} \quad (1)$$

where r_{ab} is the distance between atoms a and b , Q_a and Q_b are the atomic charges on atom a and b , and r_{cut} is a cutoff distance. This interaction energy V_{ab} is not a continuous function at $r_{ab} = r_{\text{cut}}$, and the discontinuity prevents the conservation of the total energy during a molecular dynamics (MD) simulation.

To conserve the total energy during a MD simulation, the potential energy function must be continuous and differentiable. Many shifted or switched functions have been developed to treat long-range electrostatic interactions [5]. In AMBERPLUS, the option of TINKER [3] tapering method for long-range electrostatic interactions is made available. In the TINKER [3] tapering method, which is similar to the force switching method in Ref. [5], the charge-charge electrostatic potential is given by

$$V_{ab}(r_{ab}) = \begin{cases} \frac{Q_a Q_b}{r_{ab}} - \frac{Q_a Q_b}{r_c} & r_{ab} \leq r_{\text{tap}} \\ \sum_{k=0}^5 c_k r_{ab}^k \left(\frac{Q_a Q_b}{r_{ab}} - \frac{Q_a Q_b}{r_c} \right) + Q_a Q_b \sum_{k=0}^7 f_k r_{ab}^k & r_{\text{tap}} < r_{ab} < r_{\text{cut}} \\ 0 & r_{\text{cut}} \leq r_{ab} \end{cases} \quad (2)$$

where $r_{\text{tap}} (< r_{\text{cut}})$ is a tapering distance, the beginning of the tapering window, $r_c = \frac{1}{2}(r_{\text{tap}} + r_{\text{cut}})$, and c_k and f_k are coefficients calculated from r_{tap} and r_{cut} and determined to connect V_{ab} at $r_{ab} = r_{\text{tap}}$ and $r_{ab} = r_{\text{cut}}$ smoothly. This potential energy function is continuous and differentiable in the entire range of r_{ab} , and it has continuous

second derivatives. Therefore, the total energy can be conserved during a MD simulation with this function.

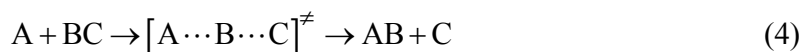
2.2. Short description of the umbrella sampling of a difference between two bond distances

One of methods to calculate potential of mean force along a reaction coordinate is umbrella sampling method [18-21]. In the umbrella sampling method, an restraint potential V_{us} , here assumed to be harmonic, is added to the system to sample a reaction coordinate in the vicinity of a user-supplied value:

$$V_{\text{us}}(s) = \frac{1}{2}k(s - s_0)^2 \quad (3)$$

where s is a reaction coordinate, s_0 is the user-supplied value, and k is a constant. During the biased simulation, the distribution of the reaction coordinate is stored. By sampling the distributions with various s_0 values, the potential of mean force can be computed by various methods such as the weighted histogram analysis method (WHAM) [19-21].

In the original *sander* program, a distance, angle, or torsional coordinate can be chosen as the reaction coordinate s of the harmonic potential. However, a difference between two distances cannot be chosen. The difference between two distances can be employed as the reaction coordinate (e.g., Ref. [6-8]). For example, in case of an S_N2 reaction such as



it is convenient to take the reaction coordinate as the difference between the BC distance and the AB distance. The AMBERPLUS patch provides the capability to add a harmonic potential as a function of a difference between two distances.

2.3. Short description of the multi-configuration molecular mechanics (MCMM) and electrostatically embedded multi-configuration molecular mechanics (EE-MCMM) methods

Since details of the MCMM and EE-MCMM method are presented in Refs. [9-13,16] and the MC-TINKER manual, we describe the methods only briefly here. The MCMM method can generate semiglobal potential energy surfaces (PES) of gas-phase reactions. In the MCMM method, the Born-Oppenheimer potential energy at geometry \mathbf{q} is represented as the lowest eigenvalue of the 2×2 diabatic Hamiltonian matrix:

$$\mathbf{U}^{\text{MCMM}}(\mathbf{q}) = \begin{pmatrix} U_{11}(\mathbf{q}) & U_{12}(\mathbf{q}) \\ U_{12}(\mathbf{q}) & U_{22}(\mathbf{q}) \end{pmatrix} \quad (5)$$

where the diagonal elements, U_{11} and U_{22} , are MM energy functions [22] that describe reactants and products, respectively. The off-diagonal element, U_{12} , is the key feature of the MCMM algorithm. U_{12} and its derivatives are determined to reproduce [23] high-level electronic structure calculation of the energy, gradient, and Hessian at some reference points called Shepard points, and modified Shepard interpolation [24,25] is used to interpolate the PES between the trust regions of the resulting set of second-order Taylor series. This MCMM method has been successful in describing semiglobal PES of gas-phase reactions and calculating their reaction rates with multidimensional tunneling contributions [9-13].

The EE-MCMM method is based on the combined quantum mechanical and molecular mechanical (QM/MM) method with electronic embedding, and it extends the domain of applicability of the MCMM method. The EE-MCMM method can reproduce the electrostatically embedded QM energy $V^{\text{EE-QM}}$, which is the sum of QM energy and QM/MM electrostatic interaction energy with a site-site representation,

$$V^{\text{EE-QM}}(\mathbf{R}, \Phi) = \langle \Psi | \hat{H}_0 + \hat{\mathbf{Q}}^T \Phi | \Psi \rangle \quad (6)$$

where \mathbf{R} stands for the collection of the Cartesian coordinates \mathbf{R}_a ($a=1,2,\dots,N^{\text{QM}}$, where N^{QM} is the number of the QM atoms) in the QM region, Ψ is the electronic wave function, \hat{H}_0 is the electronic Hamiltonian (including nuclear repulsion) of the QM region, $\hat{\mathbf{Q}}$ is the population operator vector of order N^{QM} whose components \hat{Q}_a are the population operators that generate the partial charges on QM atomic sites a :

$$Q_a = \langle \Psi | \hat{Q}_a | \Psi \rangle \quad (2.13)$$

and Φ is the electrostatic potential distribution, which is a vector of order N^{QM} , each of whose components Φ_a is the electrostatic potential due to the MM region acting at QM atom a . As in the case of the original MCMM method [9], the potential energy $V^{\text{EE-MCMM}}$ in EE-MCMM is defined as the lowest eigenvalue of a 2×2 diabatic Hamiltonian matrix [16],

$$\mathbf{U}^{\text{EE-MCMM}}(\mathbf{q}, \Phi) = \begin{pmatrix} U_{11}(\mathbf{q}, \Phi) & U_{12}(\mathbf{q}, \Phi) \\ U_{12}(\mathbf{q}, \Phi) & U_{22}(\mathbf{q}, \Phi) \end{pmatrix} \quad (7)$$

where we use nonredundant or redundant internal coordinates \mathbf{q} to represent the nuclear coordinates of the QM subsystem. U_{11} and U_{22} are analytic functions that describe V^{EEQM} in the region of reactants and products. U_{12} is based on a set of Shepard points $(\mathbf{R}^{(k)}, \Phi^{(k)})$, where $k=1, 2, \dots, N$. We evaluate $[U_{12}(\mathbf{q}, \Phi; k)]^2$ by a second-order Taylor expansion around each Shepard point $(\mathbf{R}^{(k)}, \Phi^{(k)})$, where the Taylor series coefficients are determined such that $V^{\text{EE-MCMM}}$ reproduces V^{EEQM} and its first and second derivatives with respect to \mathbf{q} and Φ at Shepard point $(\mathbf{R}^{(k)}, \Phi^{(k)})$ or are set to zero. The Shepard points where these coefficients are nonzero are called electronic structure Shepard points, and the ones where they are zero are called MM Shepard points. Then, we construct $U_{12}(\mathbf{q}, \Phi)$ at any arbitrary geometry by Shepard interpolation of these expressions.

The EE-MCMM method can be applied the QM/MM method by replacing the electrostatically embedded QM energy with the EE-MCMM energy [17]; we label the resulting potential energy surface as EE-MCMM/MM. The total potential energy V is given by

$$V(\mathbf{R}, \mathbf{R}^{\text{MM}}) = V^{\text{EE-MCMM}}(\mathbf{R}, \Phi(\mathbf{R}, \mathbf{R}^{\text{MM}})) + V_{\text{vdW}}^{\text{QM/MM}}(\mathbf{R}, \mathbf{R}^{\text{MM}}) + V_{\text{val}}^{\text{QM/MM}}(\mathbf{R}, \mathbf{R}^{\text{MM}}) + V^{\text{MM}}(\mathbf{R}^{\text{MM}}) \quad (8)$$

where $V_{\text{vdW}}^{\text{QM/MM}}$ and $V_{\text{val}}^{\text{QM/MM}}$ are respectively the van der Waals and valence interaction energies between the QM and MM regions, and V^{MM} is the MM potential energy.

Geometry optimization and MD simulations require the first derivative of V with respect to \mathbf{R} and \mathbf{R}^{MM} . These derivatives are given by

$$\frac{\partial V}{\partial \mathbf{R}_a} = \frac{\partial V^{\text{EE-MCMM}}}{\partial \mathbf{R}_a} + \frac{\partial V^{\text{EE-MCMM}}}{\partial \Phi_a} \frac{\partial \Phi_a}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{vdW}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{val}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} \quad (9)$$

and

$$\frac{\partial V}{\partial \mathbf{R}_a} = \frac{\partial V^{\text{EE-MCMM}}}{\partial \mathbf{R}_a} + \frac{\partial V^{\text{EE-MCMM}}}{\partial \Phi_a} \frac{\partial \Phi_a}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{vdW}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{val}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} \quad (10)$$

$\frac{\partial V^{\text{EE-MCMM}}}{\partial \Phi_a}$ corresponds to the partial charge $Q_a^{\text{EE-MCMM}}$. Therefore, we can regard the electrostatic interaction between the QM and MM regions in the EE-MCMM/MM method as

$$V_{\text{ele}}^{\text{QM/MM}} = \sum_a^{N^{\text{QM}}} Q_a^{\text{EE-MCMM}} \Phi_a \quad (11)$$

In the AMBERPLUS program, which interfaces the *sander* program with the MC-TINKER program, the potential energy calculated by the MC-TINKER program can be employed as one of QM methods in a QM or QM/MM system; geometry optimization and MD simulation in a MCMM or EE-MCMM/MM system are available.

3. DISTRIBUTION

AMBERPLUS is a set of subroutines for interfacing AMBER and MC-TINKER. The user should obtain three items of code:

- AMBER The AMBER code is a licensed code available from the University of California. The AMBERPLUS program supports version 9.0 code with bug fixes 1-46. For further information see:
<http://amber.scripps.edu/>

- MC-TINKER The MC-TINKER code is a licensed code available from the University of Minnesota. For further information see:
<http://comp.chem.umn.edu/mc-tinker>

- AMBERPLUS The AMBERPLUS code is a licensed code available from the University of Minnesota. For further information see:
<http://comp.chem.umn.edu/amberplus>

The prospective user does not need to obtain TINKER because a complete copy of the required version of TINKER is included (with permission) in MC-TINKER. The prospective user of AMBERPLUS must obtain and install AMBER and MC-TINKER before proceeding with AMBERPLUS.

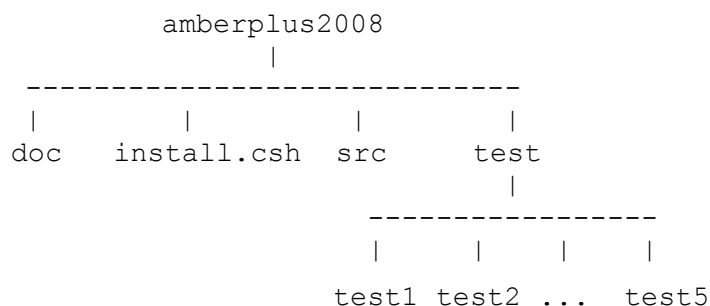
The AMBERPLUS program is distributed as a compressed tar file named `amberplus2008.tar.gz`. To uncompress, enter:

```
gunzip amberplus2008.tar.gz
```

The uncompressed file will be named `amberplus2008.tar`. After extracting the files from the tar file, which can be done with the command:

```
tar xvf amberplus2008.tar
```

a new directory, `amberplus2008`, is created. This directory contains all the files included in the distribution package, which are located in subdirectories according to the following tree structure:



doc: 1 file: `amberplus2008.pdf`. This file is the present AMBERPLUS manual.

install.csh: this file is a compilation script for AMBERPLUS.

src: 18 files comprising the patch and source code: The `def_time.patch`, `dynlib.patch`, `ew_setup.patch`, `force.patch`, `Makefile.patch`, `mdfil.patch`, `new_time.patch`, `nmr.patch`, `printe.patch`, `qmmm_module.patch`, `runmd.patch`, `sander.patch`, and `short_ene.patch` files are patch files that will be used to patch the AMBER source code; `getprm.f` file is a modified MC-TINKER file that will be used in compiling the program instead of the file distributed in the MC-TINKER source code; and `Makefile.mc-tinker`, `mcinter1.f`, `mcinter2.f`, and `taper.f` files comprise the AMBERPLUS source code.

test: 1 file: `Run.alltests`, and 5 subdirectories (`test1`, `test2`, ..., and `test5`) with the input and output files for the test calculations. Script `Run.alltests` does tests 1~5 and checks the output files.

4. INSTALLATION

First the user should obtain licensed copies of the AMBER version 9.0 and MC-TINKER version 2008 packages. Next the user should obtain a licensed copy of the AMBERPLUS package. After downloading, uncompressing, and untarring the files (as discussed in Section 3 of this manual), the AMBERPLUS package should appear as a directory, as described in Section 3.

There are a few variables that have to be set large enough to accommodate the system or systems to be studied. These variables are set in the MC-TINKER include files. The user should consult the MC-TINKER manuals for additional details; in addition, we note that these variables are explained in comment lines in the include files themselves.

The installation of the AMBERPLUS program is carried out in 4 steps. The first 3 steps are the same as in the case of the original AMBER program. The user should consult the AMBER manuals for additional details.

- (1) Set up the `AMBERHOME` environment variable to point to where the AMBER tree resides on your machine. For example,

```
Using csh, tcsh, etc:      setenv AMBERHOME /usr/amber9
Using bash, sh, zsh, etc: set AMBERHOME=/usr/amber9
                          export AMBERHOME
```

- (2) Go to the AMBER 9.0 bug fixes web site, <http://amber.scripps.edu/bugfixes90.html>, and download the `bugfix.all` file, which contains all of the bug fixes. Then patch your system running:

```
cd $AMBERHOME
patch -p0 -N -r patch_rejects < bugfix.all
```

The current version of AMBERPLUS supports AMBER 9.0 with bug fixes 1-46. If any problems are encountered with any other version of AMBER or other bug fixes, please contact the AMBERPLUS developer (see <http://comp.chem.umn.edu/amberplus> for contact information).

- (3) Go to the `$AMBERHOME/src` directory, and create a configuration file for a serial version. The command

```
./configure -help
```

will show you the options available. Choose a machine/compiler name, for example:

```
./configure ifort_x86_64
```

This will create a `config.h` file for a `x86_64` processor using `ifort` compiler. Do not choose any parallel options for the AMBERPLUS installation because the current version of the AMBERPLUS patch does not support parallel calculation. If you want to install the original AMBER program, you should issue the command

```
make serial
```

before step 4.

- (4) Go to the AMBERPLUS directory, and run a C shell script, `install.csh`. To run the script, the user should also give the paths of the `mc-tinker2008` and the `amber9` (AMBERHOME) directories. For example, if the `mc-tinker2008` path is `/usr/mc-tinker2008`, and the `amber9` path is `/usr/amber9`, then the command should be

```
./install.csh /usr/mc-tinker2008 /usr/amber9
```

These scripts will compile the source code and will generate an executable `sanderplus` file in the `$AMBERHOME/exe` directory.

In Section 8 we list the computers and operating systems on which the code has been tested.

5. USAGE

Usage of `sanderplus` is almost the same as usage of the original `sander`. Therefore, users of `sanderplus` should consult the *AMBER 9 Users' Manual* (particularly Sec. 5 “Sander basics” and Sec. 6 “Using Sander”) that is distributed with the AMBER package.

5.1 File usage

```
sanderplus [-help] [-O] [-A] -i mdin -o mdout -p prmtop
           -c inpcrd -r restrt -ref refc -x mdcrd -y inptraj
           -v mdvel -e mden -inf mdinfo -radii radii -cpin cpin
           -cpout cpout -cprestrt cprestrt -evbin evbin
           -mc81 esp.fu81 -mc82 esp.fu82 -mc83 esp.fu83
           -mc85 esp.fu85 -mcprm param.prm -mcene mcene
           -mcchg mcchg -mcelp mcelp
```

- O Overwrite output files if they exist.
- A Append output files if they exist, (used mainly for replica exchange).

The table below is a brief description of the files referred to above; the first five files are used for every run, whereas the remainder are only used when certain options are chosen. The last eight files are added in the AMBERPLUS patch and are employed for the MC-TINKER input and output files. Note that the default file names are the same as the file names in the table below.

<i>file</i>	<i>in/out</i>	<i>purpose</i>
mdin	input	control data for the min/md run
mdout	output	user-readable state info and diagnostics -o stdout will send output to stdout (to the terminal) instead of to a file.
mdinfo	output	latest mdout-format energy info
prmtop	input	molecular topology, force field, periodic box type, atom and residue names
inpcrd	input	initial coordinates and (optionally) velocities and periodic box size
refc	input	(optional) reference coordinates for position restraints; also used for targeted MD
mdcrd	output	coordinate sets saved over trajectory
inptraj	input	input coordinate sets in trajectory format, when imin=5
mdvel	output	velocity sets saved over trajectory
mden	output	extensive energy data over trajectory
restrt	output	final coordinates, velocity, and box dimensions if any - for restarting run
inpdip	input	polarizable dipole file, when indmeth=3
rstdip	output	polarizable dipole file, when indmeth=3
cpin	input	protonation state definitions
cprestrt		protonation state definitions, final protonation states for restart (same format as cpin)
cpout	output	protonation state data saved over trajectory
evbin	input	input for EVB potentials (not used for MCMM or EE-MCMM)
esp.fu81	input	MC-TINKER esp.fu81 input file
esp.fu82	input	MC-TINKER esp.fu82 input file
esp.fu83	input	MC-TINKER esp.fu83 input file
esp.fu85	input	MC-TINKER esp.fu85 input file
param.prm	input	MC-TINKER param.prm input file
mcene	output	matrix elements of (EE-)MCMM diabatic Hamiltonian and the QM/MM electrostatic interaction energy saved over a trajectory
mcchg	output	EE-MCMM partial charges saved over a trajectory
mcelp	output	electrostatic potential distribution from the MM region saved over a trajectory in the EE-MCMM/MM calculation

5.2 How to use the TINKER tapering function for long-range electrostatic interactions

In `sanderplus`, two additional parameters are added in the `&ewald` namelist to treat the TINKER tapering function for long-range electrostatic interactions. The current version of the AMBERPLUS does not support a polarizable force field; `IPOL` in the `&ctrl` namelist must be equal to 0.

ITAPER	Specifies whether the TINKER tapering function is used or not:
ITAPER=0	Not used. The electrostatic interaction is sharply truncated at CUT specified in the <code>&ctrl</code> namelist in the (default)
ITAPER=1	The TINKER tapering function is employed for long-range electrostatic interactions. In the case of periodic boundary condition (<code>NTB≠0</code> in the <code>&ctrl</code> namelist), you must specify not to use the PME or Ewald summation for long-range electrostatic interactions; you must set <code>EEDMETH=4</code> and <code>USE_PME=0</code> .
CTAPER	Defines the fraction of r_{tap} with respect to r_{cut} , which is specified by CUT in the <code>&ctrl</code> namelist (See Sec. 2.1); $r_{\text{tap}} = \text{CTAPER} \times r_{\text{cut}}$. The default value is 0.65, which is the same value as that in TINKER.

Example of mdin input file:

```
test1 input file
&cntrl
  ntx=5,  irest=1,  ntp=1,
  nstlim=10, dt=0.001, ntt=0,
  ntc=2,  ntf=7,  ntb=1,  cut=13.0,
/
&ewald
  eedmeth=4, use_pme=0, vdwmeth=0, itaper=1,
/
```

5.3 How to carry out umbrella sampling along a reaction coordinate that is a difference between two bond distances

In the original `sander` program, one can add a restraining harmonic potential on a distance, angle, or torsion by specifying `NMROPT=1` in the `&cntrl` namelist and defining the `&rst` namelist in a file specified by `DISANG` keyword (See Sec 6.7 “Umbrella sampling” and Sec 6.12.1 “Distance, angle torsional restraints” in the *AMBER 9 Users’ Manual*). For example, a `mdin` input file for umbrella sampling is as follows,

```
test of umbrella sampling
&cntrl
  nmropt=1
  ntx=5,  irest=1,  ntp=1,
  nstlim=10, dt=0.0005,
  ntt=0,  ntc=2,  ntf=7,
/
&wt type='DUMPFREQ' istep1=10 /
&wt type='END' /
DISANG=test.us
DUMPAVE=test.dist
END
```

and the `&rst` namelist is given in the `test.us` file,

```
&rst
  iat=1,6, r1=0.0, r2=3.0, r3=3.0, r4=10.0,
  rk2=30.0, rk3=30.0
/
```

In this case, one can add a harmonic potential with a minimum at 3.0 Å and a force constant 30.0 kcal mol⁻¹ Å⁻² the distance between atom number 1 and 6. Note that one must set `r2=r3` and `rk2=rk3` for a harmonic restraint because if `R` is the value of the restraint in question:

$R < r_1$	Linear, with the slope of the "left-hand" parabola at the point $R=r_1$.
$r_1 \leq R < r_2$	Parabolic, with restraint energy $rk_2 \times (R - r_2)^2$.
$r_2 \leq R < r_3$	Restraint energy is zero
$r_3 \leq R < r_4$	Parabolic, with restraint energy $rk_3 \times (R - r_3)^2$.
$r_4 \leq R$	Linear, with the slope of the "right-hand" parabola at the point $R=r_4$.

The target value (the distance between atom number 1 and 6 in this case) is stored in a file specified by `DUMPAVE` keyword every `istep1` step.

Which distance, angle, or torsion is restrained is defined by the IAT parameter in the `&rst` namelist. IAT is an array which has 4 component, IAT(1)-IAT(4). If IAT(3) \leq 0, this is a distance restraint between IAT(1) and IAT(2). If IAT(4) \leq 0, this is an IAT(1)-IAT(2)-IAT(3) angle restraint. Otherwise, this is an IAT(1)-IAT(2)-IAT(3)-IAT(4) torsional restraint. All the default values of IAT are zero.

In the `sanderplus` program, not only distance, angle, or torsional restraints, but also restraints on a difference between two distances can be used. If IAT(1) $>$ 0, IAT(2) $>$ 0, IAT(3) $<$ 0, and IAT(4) $>$ 0, then one can add a restraint on the difference between the distance between IAT(1) and IAT(2) and the distance between IAT(3) and IAT(4). For example, if `iat=3,4,-5,6`, then the difference between the distance between atom numbers 3 and 4 and the distance between numbers 5 and 6 will be restrained and stored in a file specified by DUMPAVE keyword every `istep1` step.

Example of `&rst` namelist input file:

```
&rst
  iat=1,6,-1,5 r1=-10.0, r2=0.0, r3=0.0, r4=10.0,
  rk2=30.0, rk3=30.0
/
&rst
  iat=1,6, r1=0.0, r2=0.0, r3=0.0, r4=0.0,
  rk2=0.0, rk3=0.0
/
&rst
  iat=1,5 r1=0.0, r2=0.0, r3=0.0, r4=0.0,
  rk2=0.0, rk3=0.0
/
```

Note that if you want to store the distances themselves as well as the difference between the distances at the same time, you set more restraints with `rk2=rk3=0.0` (No additional restraint potentials will be imposed). Then the output file specified by DUMPAVE keyword is like

0	0.234	2.317	2.083
10	0.214	2.290	2.076
20	0.152	2.243	2.090
30	0.071	2.187	2.115
40	-0.004	2.135	2.139
50	-0.041	2.104	2.145
60	-0.025	2.103	2.128
70	0.031	2.126	2.094
80	0.104	2.160	2.057
90	0.171	2.195	2.024

where the first column is the number of steps, the second is the difference between the two distances, and the third and fourth are the distances.

5.4 How to use the MCMM and EE-MCMM/MM methods

To employ the potential energy calculated by the MC-TINKER program, one sets QMTHOERY=8 in the `&qmmm` namelist. In the current version of `sanderplus`, PME or Ewald sum to calculate the EE-MCMM/MM long-range electrostatic interactions is unavailable; you must set EEDMETH=4 and USE_PME=0 in the `&ewald` namelist. ITAPER option (See Sec. 5.2) is available for the EE-MCMM/MM calculation.

QMTHOERY	Specifies level of theory to use for the QM region of the simulation.
QMTHOERY=1	PM3 (default)
QMTHOERY=2	AM1
QMTHOERY=3	MNDO
QMTHOERY=4	PDDG/PM3
QMTHOERY=5	PDDG/MNDO
QMTHOERY=6	PM3CARB1
QMTHOERY=7	DFTB/SCC-DFTB
QMTHOERY=8	(EE-)MCMM

Example of mdin input file:

```
test input file of MCMM or EE-MCMM/MM calculation
&cntrl
  nmropt=1
  ntx=5,  irest=1,  ntp=1,
  nstlim=10,  dt=0.0005,
  ntt=0,  temp0=300,  tempi=300,
  ntf=1,  ntb=0,  cut=100.0,  ifqnt=1
/
&qmmm
  qmmask=':1-2',  qmtheory=8,
  qmcharge=-1,  qmshake=0,
/
END
```

In this case, residues or molecules 1 and 2 are treated as the QM region, and their energy is calculated by the MC-TINKER program. Details of the `&qmmm` namelist are shown in Sec 6.4.7 “General QM/MM `&qmmm` Namelist Variables” in the *AMBER 9 Users’ Manual*.

Input and output files used for the MC-TINKER program can be specified by option when the `sanderplus` is executed (See Sec. 5.1). See the MC-TINKER manual for description of those input files.

Example of sanderplus running script:

```
sanderplus -i test.inp -p test.top -c test.crd \  
          -inf test.inf -r test.rst -o test.out \  
          -mc81 test.81 -mc82 test.82 -mc83 test.83 \  
          -mc85 test.85 -mcprm test.prm
```

In addition to the original MC-TINKER sections and options, a SIMULATION section is added in the `esp.fu85` file to output matrix elements of the (EE-)MCMM diabatic Hamiltonian, (EE-)MCMM potential energy, QM/MM electrostatic interaction energy (See Sec. 2.3) partial charges, and electrostatic potential distribution from the MM region during a simulation.

Glossary of SIMULATION keywords

- NSTPENE** Every NSTPENE step the matrix elements of the (EE-)MCMM diabatic Hamiltonian, the (EE-)MCMM potential energy and QM/MM electrostatic interaction energy will be written to a file specified by `-mcene` option. The default value is zero, which means no output.
- NSTPCHG** Every NSTPCHG step the partial charges in the QM region calculated by the EE-MCMM method will be written to a file specified by `-mcchg` option. The default value is zero, which means no output.
- NSTPELP** Every NSTPELP step the electrostatic potentials from the MM region will be written to a file specified by `-mcelp` option. The default value is zero, which means no output.

Example of esp.fu85 file:

```
*MCGENERAL

MCTITLE
  Cl- + CH3Cl -> ClCH3 + Cl-
  Electronic structure theory: MPW1K/6-31(+)G**
  Molecular mechanics: MM3
END

MCATOMS
1 C
2 H
3 H
4 H
5 CL
6 CL
END

UNITINPUT angstrom
ORDERCALC mcmm
TYPECALC gradient

*EEGENERAL

EECALC true

*MCENERGETICS

ZERO1  0.000507
ZERO2  0.000507
EDIFF  0.000000

*RESONANCE

ICSHEPARD
1-5 1-6 5-6 1-2 1-3 1-4
2-1-3 3-1-4 4-1-2
2-1-5 3-1-5 4-1-5
2-1-6 3-1-6 4-1-6
END

ICDISTANCE
1-5 1-6 5-6
END

ISHMM  false

*SIMULATION

NSTPENE 10
NSTPCHG 10
NSTPELP 10
```

Example of mcene file:

-53.71180	-62.08102	50.06119	-108.13219	-113.22813
-38.84318	-64.28695	49.82608	-102.98962	-110.36186
-35.93807	-64.91736	49.32426	-101.83620	-107.63343
-42.99261	-60.64741	47.69047	-100.32056	-106.09449
-48.51289	-49.16756	51.02331	-99.86459	-107.16055
-59.39881	-53.29123	48.22638	-104.66799	-110.63633
-60.07918	-55.51584	50.07179	-107.92125	-114.79133
-64.56524	-60.24303	48.59133	-111.04350	-117.64564
-65.00547	-52.04319	50.76324	-109.69963	-117.10705
-76.23199	-46.02469	46.54969	-110.06702	-114.78106

where the first, second, and third columns are U_{11} , U_{22} , and U_{12} , respectively. The fourth column gives the (EE-)MCMM potential energy, and the fifth column gives the QM/MM electrostatic interaction energy.

Example of mcchg and mcelp files:

0.0137295	0.1328831	0.1398529	0.1191530	-0.7176742	-0.6879444
0.0123489	0.1344027	0.1383553	0.1164482	-0.7356300	-0.6659251
0.0085104	0.1346136	0.1387383	0.1119046	-0.7370989	-0.6566681
0.0037560	0.1353874	0.1306323	0.1270156	-0.7286033	-0.6681879
0.0094405	0.1331812	0.1340876	0.1196719	-0.7089574	-0.6874239
0.0055715	0.1362960	0.1355875	0.1200311	-0.6968589	-0.7006271
0.0054269	0.1373900	0.1350747	0.1038647	-0.6797068	-0.7020495
-0.0060488	0.1359985	0.1314302	0.1267406	-0.6707805	-0.7173400
-0.0101357	0.1250390	0.1241392	0.1268864	-0.6431707	-0.7227582
-0.0277731	0.1195461	0.1243475	0.1412180	-0.6166210	-0.7407175

where the format is

FORMAT(6F12.7) (X(I), I=1, NQM)

X: array of partial charges or electrostatic potentials

NQM: the number of QM atoms

6. SAMPLE TEST RUNS

The test suite includes five test runs. Each of these test runs is described below. The `Run.testx` script in the `test/testx` directory (where `x` means the number of test program) runs and checks the test suite. To execute `Run.testx`, `$AMBERHOME` environment must be set to point where the AMBER tree resides on your machine. To run the complete test suite, one can run the `Run.alltests` script located in the `test` directory.

The `Run.testx` scripts compare the results with distributed outputs using `dacdif` script in the `$AMBERHOME/test/` directory. Note that sometimes “possible FAILURE” messages are found because of machine-dependent round-off errors. If such messages are found, you should compare those results manually.

6.1. Test run 1

MM MD simulation with the TINKER tapering function for long-range electrostatic interactions of the Cl⁻ anion in aqueous solution

This test run is an MM MD simulation with the TINKER tapering function for long-range electrostatic interactions. The system contains 1 Cl⁻ anion and 1023 water molecules. A cubic unit cell is used with a box length of 31.2 Å and with periodic boundary condition. The electrostatic interaction smoothly becomes zero at 13 Å. A cutoff of 13 Å is also employed for the Lennard-Jones interactions. The equation of motion is integrated by the velocity Verlet method with a time step of 1 fs at a temperature of 300 K. The SHAKE algorithm is used to fix the intramolecular distances of the water solvent molecules.

INPUT FILES:

test1.inp	mdin	input file for control data
test1.crd	inpcrd	input file for initial coordinates and velocities
test1.top	prmtop	input file for molecular topology and force field
Run.test1		Script file to run sanderplus

OUTPUT FILE:

test1.out	mdout	general output file
test1.inf	mdinfo	output file of latest energy information
test1.rst	restrt	output file for restarting run

6.2. Test run 2

QM/MM MD simulation with a harmonic restraint on the difference between two reactive-bond distances of the aqueous $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system

This test run is a QM/MM MD simulation for the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system in aqueous solution. The system contains 1 Cl^- anion, 1 $\text{CH}_3\text{Cl}'$ molecule, and 1021 water molecules. The $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system is treated by the AM1 Hamiltonian, while TIP3P model is employed for water molecules. A harmonic potential with a minimum at 0.0 Å and a force constant 60.0 kcal mol⁻¹ Å⁻² is added on the difference between C-Cl and C-Cl' distances. A cubic unit cell is used with a box length of 31.2 Å and with periodic boundary conditions. The PME method is used for long-range electrostatic interactions. The equations of motion are integrated by the velocity Verlet method with a time step of 0.5 fs at a temperature of 300 K. The SHAKE algorithm is used to fix all three intramolecular distances of the water solvent molecules.

INPUT FILES:

test2.inp	mdin	input file for control data
test2.crd	inpcrd	input file for initial coordinates and velocities
test2.top	prmtop	input file for molecular topology and force field
test2.us	DISANG	input file for restraints
Run.test2		Script file to run sanderplus

OUTPUT FILE:

test2.out	mdout	general output file
test2.inf	mdinfo	output file of latest energy information
test2.rst	restrt	output file for restarting run
test2.dist	DUMPAVE	output file of the target restrained values.

6.3. Test run 3

MCMM MD simulation of the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system

This test run is an MCMM MD simulation for the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system. The entire system is treated by the MCMM method. This system is treated in Ref. [13]. Configuration 1 corresponds to the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system while configuration 2 corresponds to the $\text{ClCH}_3 + \text{Cl}'^-$ system. The electronic structure theory used is MPW1K, and the basis set is 6-31G(d,p) for C and H atoms and 6-31+G(d,p) for Cl. U_{12} is determined using Shepard interpolation with three stationary electronic structure theory points (the precursor ion-dipole complex, the saddle point, and the successor ion-dipole complex) and nine nonstationary electronic structure theory points. There are no Shepard interpolation points in this calculation. The Shepard interpolation is carried out by using fifteen internal coordinates, which is a redundant set. Three of these coordinates (C-Cl, C-Cl', and Cl-Cl') are used to calculate the generalized distance used to weight the various terms in the interpolation of $[U_{12}]^2$. A harmonic potential with a minimum at 0.0 Å and a force constant $30.0 \text{ kcal mol}^{-1} \text{ \AA}^{-2}$ is added on the difference between C-Cl and C-Cl' distances. The equations of motion are integrated by the velocity Verlet method with a time step of 0.5 fs.

INPUT FILES:

test3.inp	mdin	input file for control data
test3.crd	inpcrd	input file for initial coordinates and velocities
test3.top	prmtop	input file for molecular topology and force field
test3.us	DISANG	input file for restraints
test3.81	MC-TINKER esp.fu81	input file for configuration 1
test3.82	MC-TINKER esp.fu82	input file for configuration 2
test3.83	MC-TINKER esp.fu83	input file for Shepard points
test3.85	MC-TINKER esp.fu85	general input file
test3.prm	MC-TINKER param.prm	input file for TINKER parameter
Run.test3		Script file to run sanderplus

OUTPUT FILE:

test3.out	mdout	general output file
test3.inf	mdinfo	output file of latest energy information
test3.rst	restrt	output file for restarting run
test3.dist	DUMPAVE	output file of the target restrained values.

6.4. Test run 4

MCMM geometry optimization of the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system

This test run is an MCMM MD geometry optimization for the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system. The entire system is treated by the MCMM method. The details of the MCMM calculation are the same as those in test run 3. Geometry optimization starts at the geometry near the ion-dipole complex ($\text{Cl}^- \cdots \text{CH}_3\text{Cl}'$), and it proceeds to the ion-dipole complex by the conjugate gradient method.

INPUT FILES:

test4.inp	mdin	input file for control data
test4.crd	inpcrd	input file for initial coordinates and velocities
test4.top	prmtop	input file for molecular topology and force field
test4.81	MC-TINKER esp.fu81	input file for configuration 1
test4.82	MC-TINKER esp.fu82	input file for configuration 2
test4.83	MC-TINKER esp.fu83	input file for Shepard points
test4.85	MC-TINKER esp.fu85	general input file
test4.prm	MC-TINKER param.prm	input file for TINKER parameter
Run.test4		Script file to run sanderplus

OUTPUT FILE:

test4.out	mdout	general output file
test4.inf	mdinfo	output file of latest energy information
test4.rst	restrt	output file for restarting run

6.5. Test run 5

EE-MCMM/MM MD simulation of the aqueous $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system

This test run is a EE-MCMM/MM MD simulation for the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system in aqueous solution. The system contains 1 Cl^- anion, 1 $\text{CH}_3\text{Cl}'$ molecule and 1021 water molecules. The $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system is treated by the EE-MCMM method, while the TIP3P model is employed for water molecules. This system is treated in Refs. [13] and [14]. Configuration 1 corresponds to the $\text{Cl}^- + \text{CH}_3\text{Cl}'$ system, while configuration 2 corresponds to the $\text{ClCH}_3 + \text{Cl}^-$ system. The electronic structure theory used is MPW1K, and the basis set is 6-31G(d,p) for C and H atoms and 6-31+G(d,p) for Cl. U_{12} is determined using Shepard interpolation with three stationary electronic structure theory points (the precursor ion-dipole complex, the saddle point, and the successor ion-dipole complex) and eleven nonstationary electronic structure theory points. There are no MM Shepard points in this run. The Shepard interpolation is carried out by using fifteen internal coordinates, which is a redundant set. Three of these coordinates (C-Cl, C-Cl', and Cl-Cl') are used to calculate the generalized distance that is used to weight the various Shepard-point contributions to the interpolation of $[U_{12}]^2$. A harmonic potential with a minimum at 0.0 Å and a force constant $60.0 \text{ kcal mol}^{-1} \text{ Å}^{-2}$ is added on the difference between C-Cl and C-Cl' distances. A cubic unit cell is used with a box length of 31.2 Å and with periodic boundary conditions. The TINKER tapering function for long-range electrostatic interactions is employed, and the electrostatic interaction becomes zero smoothly at 15 Å. A cutoff of 15 Å is also employed for the Lennard-Jones interactions; the Lennard-Jones interaction becomes zero sharply at 15 Å. The equations of motion are integrated by the velocity Verlet method with a time step of 0.5 fs at a temperature of 300 K. The SHAKE algorithm is used to fix all three intramolecular distances of the water solvent molecules.

INPUT FILES:

test5.inp	mdin	input file for control data
test5.crd	inpcrd	input file for initial coordinates and velocities
test5.top	prmtop	input file for molecular topology and force field
test5.us	DISANG	input file for restraints
test5.81	MC-TINKER esp.fu81	input file for configuration 1
test5.82	MC-TINKER esp.fu82	input file for configuration 2
test5.83	MC-TINKER esp.fu83	input file for Shepard points
test5.85	MC-TINKER esp.fu85	general input file
test5.prm	MC-TINKER param.prm	input file for TINKER parameter
Run.test5		Script file to run sanderplus

OUTPUT FILE:

test5.out	mdout	general output file
test5.inf	mdinfo	output file of latest energy information
test5.rst	restrt	output file for restarting run
test5.dist	DUMPAVE	output file of the target restrained values
test5.ene	mcene	output file of the matrix elements, EE-MCMM energy, and the QM/MM interaction energy during the MD simulation
test5.chg	mchg	output file of the partial charges during the MD simulation
test5.elp	mcelp	output file of the electrostatic potentials at the nuclei of the QM subsystem during a QM/MM MD simulation

7. BIBLIOGRAPHY

- [1] Case, D. A.; Darden, T. A.; Cheatham, I., T. E.; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M.; Pearlman, D. A.; Crowley, M.; Walker, R. C.; Zhang, W.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Wong, K. F.; Paesani, F.; Wu, X.; Brozell, S.; Tsui, V.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.; Cui, G.; Beroza, P.; Mathews, D. H.; Schafmeister, C.; Ross, W. S.; Kollman, P. A.; AMBER 9; University of California: San Francisco, 2006.
- [2] Tishchenko, O.; Higashi, M.; Albu, T. V.; Corchado, J. C.; Kim, Y.; Villà, J.; Xing, J.; Lin, H.; Truhlar, D. G.; MC-TINKER–version 2008; University of Minnesota: Minneapolis, MN, 2008.
- [3] Ponder, J. W.; TINKER–version 3.5; Washington University: St. Louis, MO, 1997.
- [4] Darden, T.; York, D.; Pedersen, L. G.; *J. Chem. Phys.* **1993**, *98*, 10089.
- [5] Steinbach, P. J.; Brooks, B. R.; *J. Comput. Chem.* **1993**, *15*, 667.
- [6] Chandrasekhar, J.; Smith, S. F.; Jorgensen, W. L.; *J. Am. Chem. Soc.* **1984**, *106*, 3049.
- [7] Alhambra, C.; Corchado, J. C.; Sánchez, M. L.; Gao, J.; Truhlar, D. G. *J. Am. Chem. Soc.* **2000**, *122*, 8197.
- [8] Alhambra, C.; Corchado, J. C.; Sánchez, M. L.; Gao, J.; Truhlar, D. G. *J. Am. Chem. Soc.* **2001**, *105*, 11326.
- [9] Kim, Y.; Corchado, J. C.; Villa, J.; Xing, J.; Truhlar, D. G. *J. Chem. Phys.* **2000**, *112*, 2718.
- [10] Albu, T. V.; Corchado, J. C.; Truhlar, D. G. *J. Phys. Chem. A* **2001**, *105*, 8465.
- [11] Lin, H.; Zhao, Y.; Tishchenko, O.; Truhlar, D. G. *J. Chem. Theory Comput.* **2006**, *2*, 1237.
- [12] Tishchenko, O.; Truhlar, D. G. *J. Phys. Chem. A* **2006**, *110*, 13530.
- [13] Tishchenko, O.; Truhlar, D. G. *J. Chem. Theory Comput.* **2007**, *3*, 938.
- [14] Morita, A.; Kato, S. *J. Am. Chem. Soc.* **1997**, *119*, 4021.
- [15] Iuchi, S.; Morita, A.; Kato, S. *J. Phys. Chem. B* **2002**, *106*, 3466.
- [16] Higashi, M.; Truhlar, D. G. *J. Chem. Theory Comput.* **2008**, *4*, 790.
- [17] Higashi, M.; Truhlar, D. G. *J. Chem. Theory Comput.* **2008**, *4*, 1032.
- [18] Valleau, J. P.; Torrie, G. M. In *Modern Theoretical Chemistry, Vol. 5: Statistical Mechanics, Part A, Equilibrium Techniques*; Berne, B. J., Ed.; Plenum Press: New York, 1977.
- [19] Kumar, S.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; Rosenberg, J. M. *J. Comput. Chem.* **1992**, *13*, 1011.

- [20] Kumar, S.; Rosenberg, J. M.; Bouzida, D.; Swendsen, R. H.; Kollman, P. A.; *J. Comput. Chem.* **1995**, *16*, 1339.
- [21] Roux, B. *Comput. Phys. Comm.* **1995**, *91*, 275.
- [22] Warshel, A.; Weiss, R. M. *J. Am. Chem. Soc.* **1980**, *102*, 6218.
- [23] Chang, Y. T.; Miller, W. H. *J. Phys. Chem.* **1990**, *94*, 5884.
- [24] Ischtwan, J.; Collins, M. A. *J. Chem. Phys.* **1994**, *100*, 8080.
- [25] Nguyen, K.; Rossi, I.; Truhlar, D. G. *J. Chem. Phys.* **1995**, *103*, 5522.

8. COMPUTERS AND OPERATING SYSTEMS ON WHICH THE CODE HAS BEEN DEVELOPED AND TESTED

The computers, operating systems, compiler/architecture option which is specified when a configuration file for AMBER is created, and compiler versions on which various versions of AMBERPLUS have been tested are listed in Tables 8.1, 8.2, and 8.3, respectively.

Table 8.1. Operating systems on the various machines on which various versions of the code have been tested

<u>Version</u>	<u>Machine</u>	<u>Operating system</u>
2008	IBM Regatta Power4	AIX 5.3
	IBM Netfinity Linux cluster	Red Hat Enterprise Linux WS release 3 (Taroon Update 9) – 2.4.21 kernel
	SGI Altix 3000	SuSE Linux Enterprise Server 9 (ia64) – 2.6.5 kernel
	IBM BladeCenter Linux Cluster	SuSE Linux Enterprise Server 9 (x86_64) – 2.6.5 kernel
	SGI Altix XE 1300 Linux cluster	SuSE Linux Enterprise Server 10 (x86_64) – 2.6.16 kernel

Table 8.2. Specified compiler/architecture options used for various machines in creating an AMBER configuration file

<u>Version</u>	<u>Machine</u>	<u>Operating system</u>
2008	IBM Regatta Power4	xlf90_aix
	IBM Netfinity Linux cluster	ifort_ia32
	SGI Altix 3000	sgi_altix
	IBM BladeCenter Linux Cluster	ifort_x86_64
	IBM BladeCenter Linux Cluster	pgf90
	SGI Altix XE 1300 Linux cluster	ifort_x86_64
	SGI Altix XE 1300 Linux cluster	gfortran

Table 8.3. Compiler versions used for various machines

<u>Version</u>	<u>Machine</u>	<u>Compiler version</u>
2008	IBM Regatta Power4	XL Fortran for AIX version 10.1
	IBM Netfinity Linux cluster	Intel Fortran complier version 8.1
	SGI Altix 3000	Intel Fortran complier version 8.1
	IBM BladeCenter Linux Cluster	Intel Fortran complier version 9.1
	IBM BladeCenter Linux Cluster	Portland Group Fortran 90 compiler version 6.2-5
	SGI Altix XE 1300 Linux cluster	Intel Fortran complier version 10.1
	SGI Altix XE 1300 Linux cluster	gfortran version 4.1.2

9. REVISION HISTORY

AMBERPLUS version numbers have the general form 200x/Ax'.y'-M200x" where 200x is the overall version number, x' is the number of the version of AMBER on which it is based, y' is the number of AMBER bug fix patches which are applied to AMBER, and 200x" is the number of the version of MC-TINKER on which it is based.

Note: we sometimes update one or more of the manuals without updating the version number. The manual version is determined by the date of its most recent change and is given on its first page. Any changes other than the manual(s) always involve a change in version number of the code.

9.1. AMBERPLUS–version 2008/A9.46–M2008 (July 2008)

Authors: M. Higashi and D. G. Truhlar

AMBER version: 9 with 46 bug fix patches

MC-TINKER version: 2008

This is the first version of AMBERPLUS.