Manual
AMBERPLUS – version 2010/AT1.4.16-A11.11-M2010

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

Program version: 2010
Program version date: December 16, 2010
Manual version date: December 23, 2010
Copyright 2010 (program) and 2010 (manual)

Abstract. AMBERPLUS version 2010 is a patch that incorporates the following methods into one of the modules of AMBER, in particular *sander*, which carry outs energy minimization, molecular dynamics (MD), and NMR refinements. AMBERPLUS introduces:

- the TINKER tapering function for long-range electrostatic interactions
- the link atom method with a fixed value of the ratio of the Q1-QL bond length to the Q1-M1 bond length
- multi-configuration Shepard interpolation (MCSI) or electrostatically embedded multi-configuration Shepard interpolation (EE-MCSI) method as one of the methods for the quantum mechanical (QM) region of a combined QM/MM calculation; this is accomplished by using MCSI 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. AMBERPLUS adds another type of link atom method for QM/MM calculations in *sander*, namely the multi-configuration Shepard interpolation (MCSI) method, which was previously called multi-configuration molecular mechanics (MCMM). The MCSI method enables one to generate semiglobal potential energy surfaces of reactive systems efficiently in the gas phase. The EE-MCSI method is an extension of the MCSI method; it can generate semiglobal potential energy surfaces in the presence of an electrostatic potential. In AMBERPLUS the combined EE-MCSI and molecular mechanical (EE-MCSI/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-MCSI potential energy. To install the AMBERPLUS version 2010, one requires AMBERTOOLS version 1.4 with bugfixes 1-16, AMBER version 11 with bugfixes 1-11, and MCSI version 2010.

CONTENTS

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 2010, University of Minnesota, Minneapolis, MN, 2010.

2   D. A. Case, T. A. Darden, T. E. Cheatham, III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, R. C. Walker, W. Zhang, K. M. Merz, B. Roberts, B. Wang, S. Hayik, A. Roitberg, G. Seabra, I. Kolossváry, K. F. Wong, F. Paesani, J. Vanicek, J. Liu, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M.-J. Hsieh, G. Cui, D. R. Roe, D. H. Mathews, M. G. Seetin, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, and P. A. Kollman (2010), AMBER–version 11, University of California, San Francisco.

3   O. Tishchenko, M. Higashi, T. V. Albu, J. C. Corchado, Y. Kim, J. Villà, J. Xing, H. Lin, and D. G. Truhlar, MCSI–version 2010, University of Minnesota, Minneapolis, MN, 2010.

4   J. W. Ponder, TINKER–version 3.5, Washington University, St. Louis, MO, 1997.

For example: "The calculations were carried out using the AMBERPLUS[1] computer program, which incorporates some methods into the AMBER[2] program and interfaces the AMBER[2] and MC-TINKER[3] programs. MCSI[3] is built on single-configuration molecular mechanics subroutines from the TINKER[4] 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 outs energy minimization, molecular dynamics (MD), and NMR refinements, and that interfaces the *sander* program with the MCSI [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 add another type of link atom method to the QM/MM calculations. The AMBER version 11 program itself has two types of link atom methods. In the first type, the link atoms are located on the bonds between the QM and MM boundary atoms with the distance between the QM boundary and link atoms fixed [6]. In the second type, the link atoms are placed on the MM boundary atoms. The additional method is similar to the former one. The link atoms are located on the bonds between the QM and MM boundary atoms, but the ratio of the bond length between the QM boundary and link atoms to the one between QM and MM boundary atoms is fixed [7].

The MCSI 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 MCSI 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 Shepard interpolation (MCSI) (which was called multi-configuration molecular mechanics (MCMM) previously) [8-15], calculations based on electrostatically embedded MM or SCMM (which may be called either EE-MM or EE-SCMM) [16,17] (also see Sec. II.E. in the MCSI manual) in which MM is applied to systems in the presence of an electrostatic potential, and calculations based on electrostatically embedded MCSI

(EE-MCSI) [18-20] in which MCSI is applied to systems in the presence of an electrostatic potential. Note that EE-MCSI was previously called EE-MCMM.

Throughout this manual, we use MM and SCMM interchangeably as synonyms.

In the AMBERPLUS program, the potential energy calculated by the MCSI program can be employed as one of quantum mechanical (QM) methods in a QM calculation or a combined QM and MM (QM/MM) calculation. By replacing the electrostatically embedded QM energy in the QM/MM system with the EE-MM or EE-MCSI energy, a procedure that can be called EE-MM/MM or EE-MCSI/MM, the QM/MM potential energy surface with high-level QM calculations can be generated with low computational cost [19,20]. Energy minimization (geometry optimization) and molecular dynamics simulation can be carried out with the potential energy surface calculated by the MCSI program.

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

In the original *sander* program, unless the particle meth 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,

$$V_{ab}\left(r_{ab}\right) = \begin{cases} \dfrac{Q_a Q_b}{r_{ab}} & r_{ab} < r_{cut} \\ 0 & r_{ab} \geq r_{cut} \end{cases} \tag{1}$$

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 $r_{ab} = r_{cut}$, 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

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

where $r_{tap}\left(< r_{cut}\right)$ is a tapering distance, the beginning of the tapering window, $r_c = \dfrac{1}{2}\left(r_{tap} + r_{cut}\right)$, and c_k and f_k are coefficients calculated from r_{tap} and r_{cut} and determined to connect V_{ab} at $r_{ab} = r_{tap}$ and $r_{ab} = r_{cut}$ 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 link atom method for QM/MM calculations

When a QM–MM boundary cuts a covalent bond in QM/MM methods, the link atom method is often used to cap the QM subsystem. The position of the link atom has been discussed, and several different choices for its location have been made in various implementations. Generally, to avoid adding additional degrees of freedom when a link atom is used, a constraint is imposed on the link atom.

The AMBER version 11 program (without AMBERPLUS) has two choices for the location of the link atom. The first choice is that the link atoms QL are located along the Q1-M1 bonds, where Q1 and M1 denote the QM and MM boundary atoms, respectively. Then the Q1-QL bond length is fixed [6]:

$$\mathbf{R}_{\mathrm{QL}} = \mathbf{R}_{\mathrm{Q1}} + d_{\mathrm{QL}} \frac{\mathbf{R}_{\mathrm{M1}} - \mathbf{R}_{\mathrm{Q1}}}{\left| \mathbf{R}_{\mathrm{M1}} - \mathbf{R}_{\mathrm{Q1}} \right|}, \tag{3}$$

where $\mathbf{R}_{\mathrm{X}}$ denotes the Cartesian coordinates of atom X, and d_{QL} is a the length of the Q1-QL bond. For the second choice, the link atoms are placed on the MM boundary atoms:

$$\mathbf{R}_{\mathrm{QL}} = \mathbf{R}_{\mathrm{M1}}. \tag{4}$$

This option is used with `lnk_method=2` in the `&qmmm` namelist; it is designed for use with a specifically parameterized link atom that behaves like a univalent version of the M1 atom.

In AMBERPLUS, another type of link atom method, proposed by Morokuma and coworkers [7], has been made available. When this method is chosen, the ratio of the Q1-QL bond length to the Q1-M1 bond length is fixed:

$$\mathbf{R}_{\mathrm{QL}} = \mathbf{R}_{\mathrm{Q1}} + C_{\mathrm{QL}} \left(\mathbf{R}_{\mathrm{M1}} - \mathbf{R}_{\mathrm{Q1}} \right), \tag{5}$$

where C_{QL} is constant and is specified in the input file. Note that eqs. (3) and (5) are similar, but they have different advantages and disadvantages. The vibration of the Q1-M1 bond can affect the QM energy with eq. (5), but it does not affect it with eq. (3). On the other hand, the vibration of the Q1-M1 bond is described purely by the MM force field with eq. (3), but an additional QM force can affect the Q1-M1 vibration with eq. (5).

2.3. Short description of the multi-configuration Shepard interpolation (MCSI) and electrostatically embedded multi-configuration Shepard interpolation (EE-MCSI) methods

Since details of the MCSI and EE-MCSI method are presented in Refs. [8-15,18-20] and the MCSI manual, we describe the methods only briefly here. The MCSI method can generate semiglobal potential energy surfaces (PES) of gas-phase reactions. In the MCSI method, the Born-Oppenheimer potential energy at geometry $\mathbf{q}$ is represented as the lowest eigenvalue of the 2×2 diabatic Hamiltonian matrix:

$$\mathbf{U}^{\mathrm{MCSI}}(\mathbf{q}) = \begin{pmatrix} U_{11}(\mathbf{q}) & U_{12}(\mathbf{q}) \\ U_{12}(\mathbf{q}) & U_{22}(\mathbf{q}) \end{pmatrix} \tag{6}$$

where the diagonal elements, U_{11} and U_{22}, are MM energy functions [21] that describe reactants and products, respectively. The off-diagonal element U_{12}, is the key feature of the MCSI algorithm. U_{12} and its derivatives are determined to reproduce [22] high-level electronic structure calculation of the energy, gradient, and Hessian at some reference points called Shepard points, and modified Shepard interpolation [23,24] is used to interpolate the PES between the trust regions of the resulting set of second-order Taylor series. This MCSI method has been successful in describing semiglobal PES of gas-phase reactions and calculating their reaction rates with multidimentional tunneling contributions [9-13].

The EE-MCSI 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 MCSI method. The EE-MCSI method can reproduce the electrostatically embedded QM energy $V^{\mathrm{EE\text{-}QM}}$, which is the sum of QM energy and QM/MM electrostatic interaction energy with a site-site representation,

$$V^{\mathrm{EE\text{-}QM}}(\mathbf{R}, \boldsymbol{\Phi}) = \langle \Psi | \hat{H}_0 + \hat{\mathbf{Q}}^{\mathsf{T}} \boldsymbol{\Phi} | \Psi \rangle \tag{7}$$

where $\mathbf{R}$ stands for the collection of the Cartesian coordinates $\mathbf{R}_a$ ($a = 1, 2, \cdots, N^{\mathrm{QM}}$, 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 :

$$Q_a = \langle \Psi | \hat{Q}_a | \Psi \rangle \tag{8}$$

and $\mathbf{\Phi}$ 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 MCSI method [8], the potential energy $V^{\mathrm{EE\text{-}MCMM}}$ in EE-MCSI is defined as the lowest eigenvalue of a 2×2 diabatic Hamiltonian matrix [18],

$$\mathbf{U}^{\mathrm{EE\text{-}MCSI}}\left(\mathbf{q},\mathbf{\Phi}\right) = \begin{pmatrix} U_{11}\left(\mathbf{q},\mathbf{\Phi}\right) & U_{12}\left(\mathbf{q},\mathbf{\Phi}\right) \\ U_{12}\left(\mathbf{q},\mathbf{\Phi}\right) & U_{22}\left(\mathbf{q},\mathbf{\Phi}\right) \end{pmatrix} \tag{9}$$

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 $\left(\mathbf{R}^{(k)},\mathbf{\Phi}^{(k)}\right)$, where $k=1,2,\cdots,N$. We evaluate $\left[U_{12}\left(\mathbf{q},\mathbf{\Phi};k\right)\right]^2$ by a second-order Taylor expansion around each Shepard point $\left(\mathbf{R}^{(k)},\mathbf{\Phi}^{(k)}\right)$, where the Taylor series coefficients are determined such that $V^{\mathrm{EE\text{-}MCSI}}$ reproduces V^{EEQM} and its first and second derivatives with respect to $\mathbf{q}$ and $\mathbf{\Phi}$ at Shepard point $\left(\mathbf{R}^{(k)},\mathbf{\Phi}^{(k)}\right)$ 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 $U_{12}\left(\mathbf{q},\mathbf{\Phi}\right)$ at any arbitrary geometry by Shepard interpolation of these expressions.

The EE-MCSI method can be applied the QM/MM method by replacing the electrostatically embedded QM energy with the EE-MCSI energy [19]; we label the resulting potential energy surface as EE-MCSI/MM. The total potential energy V is given by

$$\begin{aligned} V\left(\mathbf{R},\mathbf{R}^{\mathrm{MM}}\right) = {}& V^{\mathrm{EE\text{-}MCSI}}\left(\mathbf{R},\mathbf{\Phi}\left(\mathbf{R},\mathbf{R}^{\mathrm{MM}}\right)\right) + V_{\mathrm{vdW}}^{\mathrm{QM/MM}}\left(\mathbf{R},\mathbf{R}^{\mathrm{MM}}\right) \\ & + V_{\mathrm{val}}^{\mathrm{QM/MM}}\left(\mathbf{R},\mathbf{R}^{\mathrm{MM}}\right) + V^{\mathrm{MM}}\left(\mathbf{R}^{\mathrm{MM}}\right) \end{aligned} \tag{10}$$

where $V_{\mathrm{vdW}}^{\mathrm{QM/MM}}$ and $V_{\mathrm{val}}^{\mathrm{QM/MM}}$ 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 $\mathbf{R}^{\mathrm{MM}}$. These derivatives are given by

$$\frac{\partial V}{\partial \mathbf{R}_a} = \frac{\partial V^{\text{EE-MCSI}}}{\partial \mathbf{R}_a} + \frac{\partial V^{\text{EE-MCSI}}}{\partial \Phi_a}\frac{\partial \Phi_a}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{vdW}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{val}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} \tag{11}$$

and

$$\frac{\partial V}{\partial \mathbf{R}_a} = \frac{\partial V^{\text{EE-MCSI}}}{\partial \mathbf{R}_a} + \frac{\partial V^{\text{EE-MCSI}}}{\partial \Phi_a}\frac{\partial \Phi_a}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{vdW}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} + \frac{\partial V_{\text{val}}^{\text{QM/MM}}}{\partial \mathbf{R}_a} \tag{12}$$

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

$$V_{\text{ele}}^{\text{QM/MM}} = \sum_a^{N^{\text{QM}}} Q_a^{\text{EE-MCSI}}\Phi_a \tag{13}$$

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

2.4. Notes on AMBERPLUS parallelization

The current version of AMBERPLUS supports parallel calculations with MPI as does the AMBER itself, although some parts of the program are not fully parallelized yet. Parallel calculations with the TINKER tapering function for long-range electrostatic interactions and with the version of the link atom method in which the ratio of the Q1-QL bond length to the Q1-M1 bond length is fixed can be carried out with no restrictions. These options can be used just as in AMBER. On the other hand, the calculations with the MCSI program are only partially parallelized because MCSI itself does not fully support parallel calculations; the SCMM and EE-SCMM calculations with the MCSI program are not parallelized, while the MCSI and EE-MCSI calculations are parallelized. However, the current implementation is very simple: each odd-numbered CPU calculates U_{11}, while each even-numbered CPU calculates U_{22}. Then the data are transferred between processors.

Therefore, the parallel efficiency is maximized when 2 CPUs are used. No speed-up is expected with more than 2 CPUs in the MCSI and EE-MCSI calculations. However, if the number of MM atoms is much larger than the number of QM (EE-MCSI) atoms in the EE-MCSI/MM calculations, the time-determining step is the MM force calculation, which is carried out by the original AMBER program and is fully parallelized. Therefore, using more than 2 CPUs may still be useful for speeding up EE-MCSI/MM MD simulations.

3. DISTRIBUTION

AMBERPLUS is a set of subroutines for interfacing AMBER and MCSI. The user should obtain four items of code:

- AMBERTOOLS   The AMBERTOOLS code is a licensed code available from the University of California. The AMBERPLUS program supports version 1.4 code with bug fixes 1-16. For further information see:
  `http://amber.scripps.edu/`

- AMBER   The AMBER code is a licensed code available from the University of California. The AMBERPLUS program supports version 11 code with bug fixes 1-11. For further information see:
  `http://amber.scripps.edu/`

- MCSI   The MCSI code is a licensed code available from the University of Minnesota. The AMBERPLUS program supports version 2010 code For further information see:
  `http://comp.chem.umn.edu/mcsi`

- 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 MCSI. The prospective user of AMBERPLUS must obtain and install AMBER and MCSI before proceeding with AMBERPLUS.
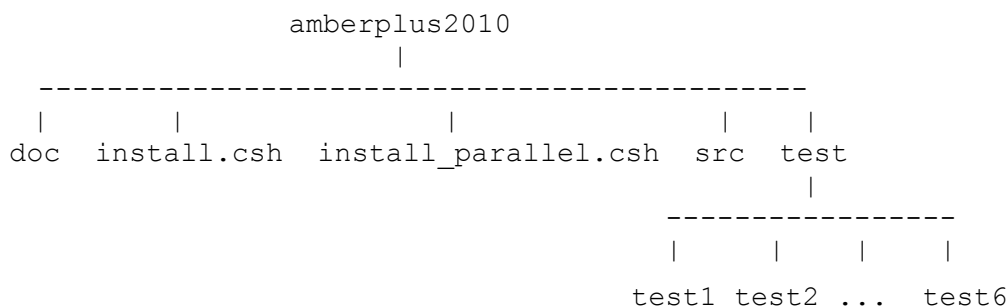
The AMBERPLUS program is distributed as a compressed tar file named `amberplus2010.tar.gz`. To uncompress, enter:

```
gunzip amberplus2010.tar.gz
```

The uncompressed file will be named `amberplus2010.tar`. After extracting the files from the tar file, which can be done with the command:

```
tar xvf amberplus2010.tar
```

a new directory, `amberplus2010`, is created. This directory contains all the files included in the distribution package, which are located in subdirectories according to the following tree structure:

```
                    amberplus2010
                         |
       ---------------------------------------------
       |      |               |                |     |
      doc  install.csh  install_parallel.csh  src   test
                                                     |
                                             -----------------
                                             |    |    |    |
                                           test1 test2 ... test6
```

doc: 1 file: `amberplus2010.pdf`. This file is the present AMBERPLUS manual.

install.csh: this file is a compilation script for the serial version of AMBERPLUS.

install_parallel.csh: this file is a compilation script for the parallel version of AMBERPLUS.

src: 27 files comprising the patch and source code: The `Makefile.sander.patch`, `Makefile.sqm.patch`, `def_time.patch`, `dynlib.patch`, `ew_setup.patch`, `force.patch`, `makedepend.sander.patch`, `makedepend.sqm.patch`, `mdfil.patch`, `mdread.patch`, `new_time.patch`, `parallel.patch`, `printe.patch`, `qm2_read_nm_and_alloc.patch`, `qm_link_atoms.patch`, `qmmm_module.patch`, `qmmm_qmtheorymodule.patch`, `runmd.patch`, `sander.patch`, and `short_ene.patch` files are patch files that will be used to patch the AMBER source code; The `getprm.f` file is a modified MCSI file that will be used in compiling the program instead of the file distributed in the

MC-TINKER source code; and `extra.h`, `Makefile.mcsi`, `mcinter1.f`, `mcinter2.f`, `parallel.h`, and `taper.f` files comprise the AMBERPLUS source code.

`test:` 2 files: `Run.alltests` and `Run.alltests.parallel`, and 6 subdirectories (`test1`, `test2`, ..., and `test6`) with the input and output files for the test calculations. Script `Run.alltests` (`Run.alltests.parallel`) does tests 1-6 and checks the output files by the serial (parallel) version of AMBERPLUS.

4. INSTALLATION

First the user should obtain licensed copies of the AMBERTOOLS version 1.4, AMBER version 11, and MCSI version 2010 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 MCSI include files. The user should consult the MCSI 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 for the serial version, and additional 2 steps are required for the parallel version. 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) First, extract the AMBERTOOLS and AMBER compressed files in a user-selected location (we use /usr/local as an example here):

```
cd /usr/local
tar xvfj AmberTools-1.4.tar.bz2
tar xvfj Amber11.tar.bz2
```

Then go to the AMBER bug fixes web site, http://ambermd.org/bugfixes.html, and follow the instructions there to install needed updates. The current version of AMBERPLUS supports AMBERTOOLS version 1.4 with bug fixes 1-16 and AMBER version 11 with bug fixes 1-11. If any problems are encountered with any other version of AMBER or other bug fixes, please contact an AMBERPLUS developer (see http://comp.chem.umn.edu/amberplus for contact information).

(2) Set up the AMBERHOME environmental variable to point to where the AMBER tree resides on your machine. For example,

```
Using csh, tcsh, etc: setenv AMBERHOME /usr/local/amber11
Using bash, zsh, etc: export AMBERHOME=/usr/local/amber11
```

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

```
./configure -help
```

will show you the options available. Choose a complier name, for example:

```
./configure intel
```

This will create a `config.h` file for the Intel complier. Do not choose any parallel options at this point. If you want to install the original AMBERTOOLS and AMBER programs, you should issue the command

```
make
cd $AMBERHOME/src
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 `mcsi2010-1` and the `amber11` (`AMBERHOME`) directories. For example, if the `mcsi2010-1` path is `/usr/local/mcsi2010-1`, and the `amber11` path is `/usr/local/amber11`, then the command should be

```
./install.csh /usr/local/mcsi2010-1 /usr/local/amber11
```

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

(5) If you want to install the parallel version of AMBERPLUS, go to the `$AMBERHOME/AmberTools/src` directory again, and create a configuration file for the parallel version. For example,

```
   ./configure -mpi intel
```

will create a `config.h` file for the Intel complier with MPI. If you want to install the parallel version of the original AMBER programs, you should issue the command

```
   cd $AMBERHOME/src
   make parallel
```

before step 6.

(6) Go to the `amberplus` directory again, and run a C shell script, `install_parallel.csh`. The usage is the same as for `install.csh` (see step 4). For example, if the mcsi2010-1 path is `/usr/local/mcsi2010-1`, and the amber11 path is `/usr/local/amber11`, then the command should be

```
   ./install_parallel.csh /usr/local/mcsi2010-1 /usr/local/amber11
```

These scripts will compile the source code and will generate an executable `sanderplus.MPI` 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 11 Users' Manualt6t6t6t5* (particularly Sec. 2 "Sander basics") 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.

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 MCSI input and output files. Note that the default file names are the same as the file names in the table below.

file	in/out	purpose
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 MCSI or EE-MCSI)
esp.fu81	input	MCSI esp.fu81 input file
esp.fu82	input	MCSI esp.fu82 input file
esp.fu83	input	MCSI esp.fu83 input file
esp.fu85	input	MCSI esp.fu85 input file
param.prm	input	MCSI param.prm input file
mcene	output	matrix elements of (EE-)MCSI diabatic Hamiltonian and the QM/MM electrostatic interaction energy saved over a trajectory
mcchg	output	EE-MCSI partial charges saved over a trajectory
mcelp	output	electrostatic potential distribution from the MM region saved over a trajectory in the EE-MCSI/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 polaraizable 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 `&ctrl` namelist (default).

ITAPER=1   The TINKER tapering function is employed for long-range electrostatic interactions. In the case of periodic boundary condition (NTB≠0 in the `&ctrl` namelist), you must specify not to use the PME or Ewald summation for long-range electrostatic interactions; you must set EEDMETH=4 and USE_PME=0.

CTAPER   Defines the fraction of r_{tap} with respect to r_{cut}, which is specified by CUT in the `&ctrl` namelist (See Sec. 2.1); r_{tap}=CTAPER×r_{cut}. The default value is 0.65, which is the same value as that in TINKER.

Example of `mdin` *input file:*

```
test input file for the tapering function
 &cntrl
   ntx=5, irest=1, ntpr=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, ctaper=0.80
 /
```

5.3 How to carry out link atom calculations with the ratio of the Q1-QL bond length to the Q1-M1 bond length fixed

In `sanderplus`, two additional parameters are added in the `&qmmm` namelist to treat the link atom method with the ratio of the Q1-QL bond length to the Q1-M1 bond length fixed.

LNKTYP	Specifies which link atom method is used:

LNKTYP=0   The link atom method in the original sander is used (default).

LNKTYP=1   The link atom method with the ratio of the Q1-QL bond length to the Q1-M1 bond length fixed is used.

CLINK   Defines the ratio of the Q1-QL bond length to the Q1-M1 bond length. The default value is 0.7143, which is the ratio of the equilibrium bond length for the CT-HC bond (1.090 Å) to that for the CT-CT bond (1.526 Å) in the AMBER parm99 and parm03 force fields.

Example of `mdin` *input file:*

```
test input file for the added link method
 &cntrl
   ntx=5, irest=1, ntpr=1,
   nstlim=10, dt=0.0005, ntt=0,
   ntc=2, ntf=1, ntb=0, cut=1000.0,
   ifqnt=1,
 /
 &qmmm
  iqmatoms=1974,1975,1976,1977,1978,1979,
          4859,4860,4861,4862,4863,4864,4865,4866,
  lnktyp=1, clink=0.70
 /
```

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

To employ the potential energy calculated by the MCSI program, one sets QM_THEORY='MCSI' or 'MCMM' in the `&qmmm` namelist. In the current version of `sanderplus`, PME or Ewald sum to calculate the EE-MCSI/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-MCSI/MM calculation.

QM_THOERY Specifies level of theory to use for the QM region of the simulation. In the original `sander`, available options are AM1, RM1, MNDO, PM3-PDDG, MNDO-PDDG, PM3-CARB1, PM6, and DFTB. In the `sanderplus`, MCSI and MCMM options, which both call the MCSI program, are added.

Example of `mdin` *input file:*

```
test input file of MCSI or EE-MCSI/MM calculation
 &cntrl
   nmropt=1
   ntx=5, irest=1, ntpr=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', qm_theory='MCSI',
   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 MCSI program. Details of the `&qmmm` namelist are shown in Sec 3.6.7 "General QM/MM &qmmm Namlist Variables" in the *AMBER 11 Users' Manual*.

Input and output files used for the MCSI program can be specified by option when the `sanderplus` is executed (See Sec. 5.1). See the MCSI 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 MCSI sections and options, a SIMULATION section is added in the `esp.fu85` file to set some variables.

Glossary of SIMULATION keywords

NSTPENE   Every NSTPENE step the matrix elements of the (EE-)MCSI diabatic Hamiltonian, the (EE-)MCSI potential energy and QM/MM electrostatic interaction energy (See Sec. 2.3) will be written to a file specified by the `-mcene` option. The default value is zero, which means no output.

    Example:
     NSTPENE 10

NSTPCHG   Every NSTPCHG step the partial charges in the QM region calculated by the EE-MCSI method will be written to a file specified by `-mcchg` option. The default value is zero, which means no output.

    Example:
     NSTPCHG 10

NSTPELP   Every NSTPELP step the electrostatic potentials from the MM region will be written to a file specified by the `-mcelp` option. The default value is zero, which means no output.

    Example:
     NSTPELP 10

IMCSIATOM IMCSIATOM is a list keyword that specifies which atoms in the `prmtop` parameter and topology file correspond to MCSI atoms in the MCSI input file. For example,

```
IMCSIATOM
1974
1972
1975
1976
1977
1978
1979
END
```

means that the first, second, third, … MCSI atoms given in the MCSI input correspond to 1974th, 1972nd, 1976th, … atoms in the `sander` parameter file. If an MCSI atom corresponds to a link atom, the number of the M1 atom should be specified. The default is the ordering determined by the original `sander` according to IQMATOMS or QMMASK in the `&qmmm` namelist. Note that the original `sander` sets the order of link atoms after QM atoms.

BOUNDARY BOUNDARY is a variable keyword that specifies how the QM−MM electrostatic interaction near the QM−MM boundary is treated. The default is *amber*.

Option	Description
amber	The method specified in the `&qmmm` namelist is used.
rc	The redistributed charge scheme [25] is used.
rcd	The redistributed charge scheme [25] is used.
brc	The balanced redistributed charge scheme [26] is used.
brcd	The balanced redistributed charge and dipole scheme [26] is used.

Example:
```
BOUNDARY    brc
```

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-)MCSI 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 six test runs. Each of these test runs is described below. The `Run.testx` script in the `test/testx` directory (where x denotes the number of the test) runs and checks the test suite by the serial version of `sanderplus`. To execute `Run.testx`, the `$AMBERHOME` environment must be set to point to where the AMBER tree resides on your machine. To run the complete test suite by the serial version of `sanderplus`, one can run the `Run.alltests` script located in the `test` directory. The `Run.testx.parallel` and `Run.alltests.parallel` scripts are the parallel version of `Run.testx` and `Run.alltests`.

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 one 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 of the S_N2 reaction in haloalkane dehalogenase with the link atom method with the ratio of the Q1-QL bond length to the Q1-M1 bond length fixed.

This test run is a QM/MM MD simulation for the reaction $RCH_2C(O)O^- + CH_2ClCH_2Cl \rightarrow RCH_2C(O)OCH_2CH_2Cl + Cl^-$ in haloalkane dehalogenase. The system contains the enzyme-substrate complex constructed from the X-ray crystal structure (Protein Data Bank code 2DHC) and 317 water molecules. The $CH_3C(O)O^- + CH_2ClCH_2Cl$ system is treated by the PM3 Hamiltonian, while AMBER ff03 force field and TIP3P model are employed for protein and water molecules, respectively. The link atom method with the ratio of the Q1−QL bond length to the Q1−M1 bond length fixed is used. A harmonic potential with a minimum at 0.0 Å and a force constant 200.0 kcal mol^{-1} Å^{-2} is added as a function of the difference between breaking C−Cl and forming C−O distances. MM atoms beyond 20 Å from the reaction center are fixed. A half-hamonic potential is added at 20 Å from the reaction center to prevent water molecules from drifting away. All the non-bonding interactions are considered. 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

MCSI MD simulation of the $Cl^- + CH_3Cl'$ reaction

This test run is an MCSI MD simulation for the $Cl^- + CH_3Cl'$ system. The entire system is treated by the MCSI method. This system is treated in Ref. [18]. Configuration 1 corresponds to the $Cl^- + CH_3Cl'$ system while configuration 2 corresponds to the $ClCH_3 + Cl'^-$ system. The electronic structure theory used is density functional theory with the MPW1K density functional, 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 kcal mol^{-1} Å^{-2} is added as a function of 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`	MCSI `esp.fu81`	input file for configuration 1
`test3.82`	MCSI `esp.fu82`	input file for configuration 2
`test3.83`	MCSI `esp.fu83`	input file for Shepard points
`test3.85`	MCSI `esp.fu85`	general input file
`test3.prm`	MCSI `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

MCSI geometry optimization of the Cl^- + CH_3Cl' system

This test run is an MCSI MD geometry optimization for the Cl^- + CH_3Cl' system. The entire system is treated by the MCSI method. The details of the MCSI calculation are the same as those in test run 3. Geometry optimization starts at the geometry near the ion-dipole complex (Cl^-…CH_3Cl'), 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	MCSI esp.fu81	input file for configuration 1
test4.82	MCSI esp.fu82	input file for configuration 2
test4.83	MCSI esp.fu83	input file for Shepard points
test4.85	MCSI esp.fu85	general input file
test4.prm	MCSI 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-MCSI/MM MD simulation of the aqueous $Cl^- + CH_3Cl'$ system

This test run is a EE-MCSI/MM MD simulation for the $Cl^- + CH_3Cl'$ system in aqueous solution. The system contains one Cl^- anion, one CH_3Cl' molecule and 1021 water molecules. The $Cl^- + CH_3Cl'$ system is treated by the EE-MCSI method, while the TIP3P model is employed for water molecules. This system is treated in Refs. [18] and [19]. Configuration 1 corresponds to the $Cl^- + CH_3Cl'$ system, while configuration 2 corresponds to the $ClCH_3 + Cl'^-$ system. The electronic structure theory used is density functional theory with the MPW1K density functional, 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 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 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`	MCSI `esp.fu81`	input file for configuration 1
`test5.82`	MCSI `esp.fu82`	input file for configuration 2
`test5.83`	MCSI `esp.fu83`	input file for Shepard points
`test5.85`	MCSI `esp.fu85`	general input file
`test5.prm`	MCSI `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-MCSI energy, and the QM/MM interaction energy during the MD simulation
`test5.chg`	`mcchg`	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

6.6. Test run 6

EE-MCSI/MM MD simulation of the S_N2 reaction in haloalkane dehalogenase

This test run is an EE-MCSI/MM MD simulation for the reaction $RCH_2C(O)O^- + CH_2ClCH_2Cl \rightarrow RCH_2C(O)OCH_2CH_2Cl + Cl^-$ in haloalkane dehalogenase. The system contains the enzyme-substrate complex constructed from the X-ray crystal structure (Protein Data Bank code 2DHC) and 317 water molecules. The $CH_3C(O)O^- + CH_2ClCH_2Cl$ system is treated by the EE-MCSI method, while the AMBER ff03 force field and TIP3P model are employed for protein and water molecules respectively. This system is treated in Refs. [20]. Configuration 1 corresponds to the $RCH_2C(O)O^- + CH_2ClCH_2Cl$ system, while configuration 2 corresponds to the $RCH_2C(O)OCH_2CH_2Cl + Cl^-$ system. The electronic structure theory used is density functional theory with the MPW1K density functional, and the basis set is 6-31G(d,p) for C and H atoms and 6-31+G(d,p) for O and Cl. The diabatic coupling element U_{12} is determined using Shepard interpolation with three stationary electronic structure theory points (the reactant, the saddle point, and the product complex) and five nonstationary electronic structure theory points. The Shepard interpolation is carried out by using 66 internal coordinates, which constitute a redundant set. Three of these coordinates (the breaking C–Cl bond, the forming C–O bond, and C–O–C–C torsional angle) are used to calculate the generalized distance that is used to weight the various Shepard-point contributions to the interpolation of $[U_{12}]^2$. The link atom method with the ratio of the Q1–QL bond length to the Q1–M1 bond length fixed and the redistributed charge scheme is used. A harmonic potential with a minimum at 0.0 Å and a force constant 200.0 kcal mol^{-1} Å^{-2} is added on the difference between breaking C–Cl and forming C–O distances. MM atoms beyond 20 Å from the reaction center are fixed. A half-hamonic potential is added at 20 Å from the reaction center to prevent water molecules from drifting away. The TINKER tapering function for long-range electrostatic interactions is employed, and the electrostatic interaction smoothly becomes zero 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:

`test6.inp`	`mdin`	input file for control data
`test6.crd`	`inpcrd`	input file for initial coordinates and velocities
`test6.top`	`prmtop`	input file for molecular topology and force field
`test6.us`	`DISANG`	input file for restraints
`test6.81`	MCSI `esp.fu81`	input file for configuration 1
`test6.82`	MCSI `esp.fu82`	input file for configuration 2
`test6.83`	MCSI `esp.fu83`	input file for Shepard points
`test6.85`	MCSI `esp.fu85`	general input file
`test6.prm`	MCSI `param.prm`	input file for TINKER parameter
`Run.test6`		Script file to run `sanderplus`

OUTPUT FILE:

`test6.out`	`mdout`	general output file
`test6.inf`	`mdinfo`	output file of latest energy information
`test6.rst`	`restrt`	output file for restarting run
`test6.dist`	`DUMPAVE`	output file of the target restrained values
`test6.ene`	`mcene`	output file of the matrix elements, EE-MCSI energy, and the QM/MM interaction energy during the MD simulation
`test6.chg`	`mcchg`	output file of the partial charges during the MD simulation
`test6.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 III., T. E.; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Walker, R. C.; Zhang, W.; Merz, K. M.; Roberts, B.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Kolossváry, I.; Wong, K. F.; Paesani, F.; Vanicek, J.; Liu, J.; Wu, X.; Brozell, S. R.; Steinbrecher, T.; Gohlke, H.; Cai, Q.; Ye, X.; Wang, J.; Hsieh, M.-J.; Cui, G.; Roe, D. R.; Mathews, D. H.; Seetin, M. G.; Sagui, C; Babin, V.; Luchko, T.; Gusarov, S.; Kovalenko, A.; Kollman, P. A.; AMBER 11; University of California: San Francisco, 2010.

[2]   Tishchenko, O.; Higashi, M.; Albu, T. V.; Corchado, J. C.; Kim, Y.; Villà, J.; Xing, J.; Lin, H.; Truhlar, D. G.; MCSI–version 2010; University of Minnesota: Minneapolis, MN, 2010.

[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]   Walker, R. C.; Crowley, M. F.; Case, D. A. *J. Comput. Chem.* **2007**, *29*, 1019.

[7]   Dapprich, S.; Komáromi, I.; Byun, K.; Morokuma, K; Frisch, M. *THEOCHEM* **1999**, *461-462*, 1.

[8]   Kim, Y.; Corchado, J. C.; Villa, J.; Xing, J.; Truhlar, D. G. *J. Chem. Phys.* **2000**, *112*, 2718.

[9]   Albu, T. V.; Corchado, J. C.; Truhlar, D. G. *J. Phys. Chem. A* **2001**, *105*, 8465.

[10]   Lin, H.; Zhao, Y.; Tishchenko, O.; Truhlar D. G. *J. Chem. Theory Comput.* **2006**, *2*, 1237.

[11]   Tishchenko, O.; Truhlar, D. G. *J. Phys. Chem. A* **2006**, *110*, 13530.

[12]   Tishchenko, O.; Truhlar, D. G. *J. Chem. Theory Comput.* **2007**, *3*, 938.

[13]   Tishchenko, O.; Truhlar, D. G. *J. Chem. Phys.* **2009**, *130*, 024105.

[14]   Tishchenko, O.; Truhlar, D. G. *J. Chem. Theory Comput.* **2009**, *5*, 1454.

[15]   Tishchenko, O.; Truhlar, D. G. *J. Chem. Phys.* **2010**, *132*, 084109.

[16]   Morita, A.; Kato, S. *J. Am. Chem. Soc.* **1997**, *119*, 4021.

[17]   Iuchi, S.; Morita, A; Kato, S. *J. Phys. Chem. B* **2002**, *106*, 3466.

[18]   Higashi, M.; Truhlar, D. G. *J. Chem. Theory Comput.* **2008**, *4*, 790.

[19]   Higashi, M.; Truhlar, D. G. *J. Chem. Theory Comput.* **2008**, *4*, 1032.

[20]   Higashi, M.; Truhlar, D. G. *J. Chem. Theory Comput.* **2009**, *5*, 2925.

[21]   Warshel, A.; Weiss, R. M. *J. Am. Chem. Soc.* **1980**, *102*, 6218.

[22]   Chang, Y. T.; Miller, W. H. *J. Phys. Chem.* **1990**, *94*, 5884.

[23]   Ischtwan, J.; Collins, M. A. *J. Chem. Phys.* **1994**, *100*, 8080.

[24] Nguyen, K.; Rossi, I.; Truhlar, D. G. *J. Chem. Phys.* **1995**, *103*, 5522.

[25] Lin, H.; Truhlar, D. G. *J. Phys. Chem. A* **2005**, *109*, 3991.

[26] Wang, B.; Truhlar, D. G. *J. Chem. Theory Comput.* **2010**, *6*, 359.

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

Version	Machine	Operating system
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
2010	Linux Cluster	CentOS 4.4 (x86_64) – 2.6.9 kernel

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

Version	Machine	Operating system
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`
2010	Linux Cluster	`intel (=icc/ifort)`
	Linux Cluster	`gnu (=gcc/gfortran)`
	Linux Cluster	`pgi (=pgcc/pgf90)`

Table 8.3.   Compiler versions used for various machines

Version	Machine	Compiler version
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
2010	Linux Cluster	Intel Fortran complier version 11.0
	Linux Cluster	gfortran version 4.1.0
	Linux Cluster	Portland Group Fortran 90 compiler version 7.1-2

9. REVISION HISTORY

AMBERPLUS version numbers have the general form 20xx/ATx′.x′.y′–Ax″.y″–M20x‴x‴ where 20xx is the overall version number, x′.x′ is the number of the version of AMBERTOOLS on which it is based, y′ is the number of AMBER bug fix patches which are applied to AMBERTOOLS, 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 20x‴x‴ is the number of the version of MCSI 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.

9.2. AMBERPLUS–version 2010/AT1.4.16-A11.11-M2010 (December 2010)

Authors: M. Higashi and D. G. Truhlar
AMBERTOOLS version: 1.4 with 16 bug fix patches
AMBER version: 11 with 11 bug fix patches
MC-TINKER version: 2010

- The capability to carry out the link atom method with the ratio of the Q1-QL bond length to the Q1-M1 bond length fixed in the QM/MM method has been added.
- The capability to apply the EE-MCSI method to QM/MM systems with covalent bonds between the QM and MM subsystems has been added.
- The capability to add a restraint on the difference between two bond lengths has been removed because the current version of AMBER itself has that capability.

- Parallelization is supported partially (See Sec. 2.4).
- A bug in calculating electrostatic interactions with the TINKER tapering function has been fixed.