

CRATE:
A Package Supporting CHARMMRATE, which is a
Module of CHARMM that Interfaces it to POLYRATE

CRATE
Version 9.0.1
APRIL 18, 2006

Mireia Garcia-Viloca, Cristobal Alhambra, José Corchado,
María Luz Sánchez, J. Villà, Jiali Gao, and Donald G. Truhlar
Department of Chemistry and Supercomputer Institute, University of Minnesota,
Minneapolis, MN 55455-0431, U.S.A.

Date of this version of utility: April 18, 2006
Date of most recent manual update: April 18, 2006

Copyright 1999, 2002, 2006

CRATE-version 9.0.1 is a package supporting the use of POLYRATE with the CHARMMRATE module of CHARMM. It is available at <http://comp.chem.umn.edu>.

Note: The version number of CRATE is 9.0.1. This means that CRATE-version 9.0.1 is based on POLYRATE-version 9.0. The interface of CHARMM-versions c28b1 to c28b2 with POLYRATE-version 9.0 will result in CHARMMRATE-1.0, whereas the interface of CHARMM-version c28b3 and later with POLYRATE-version 9.0 gives CHARMMRATE-2.0.

CRATE abstract

CRATE is a package supporting the CHARMMRATE module of CHARMM. This package contains (1) a utility to prepare POLYRATE for use with the CHARMMRATE module of CHARMM, (2) test runs for enzyme kinetics simulations with CHARMMRATE, and (3) additional background information related to CRATE and CHARMMRATE.

The standard POLYRATE program requires modifications to make it compatible with CHARMM. The modifications are made to enable efficient transfer of information between CHARMM and POLYRATE and to eliminate conflicts and other problems during compilation. This package contains the modifications to be made.

The files `ef.src`, `energetics.src`, `fromblas.src`, `interface.src`, `polyag.src`, `polyrr.src`, and `polysz.src` are modified by a script called `install_cr.com` that is provided as part of the CRATE utility. In particular, subroutine names are modified to remove conflicts with the CHARMM libraries, an additional keyword is added to the input read options in subroutine `read5`, some statements are commented out because they call ACES routines that are not part of POLYRATE, some subroutines are commented because they will never be used, all tab characters are removed to allow the CHARMM preprocessor to process the code properly, and all source code files are given the extension required by CHARMM. The subroutine `dattim.f90` is copied from the `polyrate/util/` directory by the script `install_cr.com` in order to substitute the machine-dependent subroutine.

In addition, the CRATE utility contains (i) a subroutine called “`maino`,” which replaces the POLYRATE main routine, (ii) a new set of “hooks” (interface subprograms) that are required to interface CHARMM with POLYRATE, (iii) a version of the POLYRATE `param.inc` file that has dimensions large enough to run the CHARMMRATE test run, and (iv) three source files, `fdiag.f`, `pmf_projct.f`, and `charmmrate.inc`. The first file is a POLYRATE subroutine that has been modified, and the last two are new source files. They contain the additional code necessary to carry out projected instantaneous normal mode analysis. The

CRATE package contains information about the changes to be made by the user to perform this kind of calculation.

The CRATE package also contains two test runs for xylose isomerase. The first of these test runs reads in the coordinates for 4 configurations of the system. These are taken from a CHARMM trajectory file, and for each of them the test run calculates the projected instantaneous normal mode frequencies. This kind of calculation is used in step 2 of stage 1 of the EA-VTST/MT method to include quantum mechanical vibrational effects. The second test run performs a dynamics calculation for a representative configuration of the ensemble of transition state structures for the hydride transfer reaction catalyzed by xylose isomerase. This kind of calculation is used in stage 2 of the EA-VTST/MT method to determine the individual transmission coefficients for a given configuration.

Referencing CRATE

CRATE is a package that contains (1) a utility for preparing POLYRATE for use with CHARMMRATE and (2) test runs facilitating the use of CHARMMRATE. Since CHARMMRATE and CRATE were developed as a package, it is not necessary to reference CRATE if CHARMMRATE is referenced. The recommended referencing for CHARMMRATE is as follows:

A) J. Chem. Phys. format:

The rate constant (or reaction path or geometry optimization, etc.) calculations were carried out using the CHARMMRATE program.¹⁻³

1. M. Garcia-Viloca, C. Alhambra, J. C. Corchado, M. L. Sánchez, J. Villà, J. Gao, and D. G. Truhlar, CHARMMRATE–version 2.0, University of Minnesota, Minneapolis, 2002, a module of CHARMM (Ref. 2) for interfacing it with POLYRATE (Ref. 3).
2. *Chemistry at HARvard Macromolecular Mechanics* (CHARMM) computer program, as described in B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *J. Compt. Chem.* **4**, 187 (1983).
3. J. C. Corchado, Y.-Y. Chuang, P. L. Fast, J. Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitiño, A. Fernandez-Ramos, J. Pu, and T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE–version 9.0, University of Minnesota, Minneapolis, 2002.

B) American Chemical Society format:

The rate constant (or reaction path or geometry optimization, etc.) calculations were carried out using the CHARMMRATE program.¹⁻³

1. M. Garcia-Viloca, C. Alhambra, J. C. Corchado, M. L. Sánchez, J. Villà, J. Gao, and D. G. Truhlar, CHARMMRATE–version 2.0, University of Minnesota,

Minneapolis, 2002, a module of CHARMM computer program (Ref. 2) for interfacing it with POLYRATE program (Ref. 3).

2. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. Statyes, S. Swaminathan, M. Karpuls, *J. Comput. Chem.* **1983**, 4, 187.
3. J. C. Corchado, Y.-Y. Chuang, P. L. Fast, J. Villà, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, V. S. Melissas, B. J. Lynch, I. Rossi, E. L. Coitiño, A. Fernandez-Ramos, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, IA. D. Isaacson, D. G. Truhlar, POLYRATEversion 9.0, University of Minnesota, Minneapolis, 2002.

In cases where the older version of CHARMMRATE is used, one may substitute that reference (CHARMMRATE-1.0) for the one given above.

Contents

Title and URL.....	1
CRATE abstract.....	2
Referencing CRATE.....	4
Contents.....	6
Description of CRATE.....	7
1. Utility for modifying POLYRATE.....	7
2. Test runs for xylose isomerase.....	13
Additional background for CRATE and CHARMMRATE.....	14
1. Selection of configurations for the transition state ensemble.....	14
2. Summary of CRATE and CHARMMRATE versions.....	18
Revision history of CRATE versions.....	20

Utility for modifying POLYRATE

The CRATE utility will automatically make changes to the source code of POLYRATE to allow the interface between CHARMM and POLYRATE. When both CHARMM (with the CHARMMRATE module) and POLYRATE have been obtained, integration of the code into a single executable called CHARMMRATE is possible with the utility, CRATE, which is available from the University of Minnesota (<http://comp.chem.umn.edu>).

The CHARMMRATE interface for CHARMM and POLYRATE takes advantage of the modular nature of both programs, and, consequently, minimal modifications of CHARMM and POLYRATE were required. The CHARMM program is the main driver of the integrated program, which makes FORTRAN call to the interface subprogram, CHARMMRATE, to initiate VTST calculations by POLYRATE. The energy and energy gradients for the primary-zone atoms required by POLYRATE are determined by CHARMM through the interface subprogram and are supplied to POLYRATE through a set of subroutines called the POLYRATE hooks.

In general, all the standard CHARMM directives were followed to make CHARMMRATE possible. Specific modifications of the original POLYRATE program have been made for efficient transfer of information between CHARMM and POLYRATE. The CRATE utility will automatically make these modifications, which are accomplished in two ways.

- (1) A subroutine called “maino,” which replaces the POLYRATE main routine, and a new set of POLYRATE hooks are provided as part of the CRATE utility. For further details of these modifications, see sections **a** and **b** below.
- (2) Several other files, namely ef.src, energetics.src, fromblas.src, interface.src, polyag.src, polyrr.src and polysz are modified by the script file called install_cr.com that is provided as part of the CRATE utility. In particular, subroutine names are modified to remove conflicts with the CHARMM libraries, an additional keyword is added to the input read options in

subroutine read5, some statements are commented out because they call ACES routines that are not part of POLYRATE, some subroutines are commented because they will never be used, all tab characters found in the source code and all characters beyond column 72 are removed to allow the CHARMM preprocessor to process the code properly, and all source code files are given the extension required by CHARMM. Further details of these changes are provided in parts *c* through *h* below.

In addition, the CRATE utility contains a version of the POLYRATE param.inc file that has dimensions large enough to run the CHARMMRATE test runs. The dimensions of this parameter file are also large enough to run the CRATE test runs.

For CHARMMRATE-2.0, three source files are provided in the CRATE package. These source files contain the additional code necessary to carry out projected instantaneous normal mode analysis. Further details of these changes are provided in parts *i* and *j* below.

Description of the changes that have been made in POLYRATE main program and in POLYRATE hooks:

a. differences of maino.src from main.f

- a.1. At the beginning of the file, the POLYRATE main program is converted into a subroutine by adding the line: subroutine ChaRate.
- a.2. When the logical variable ChRate is true the calls to open the files fu5 and fu6 are skipped since these units have already been opened by CHARMM.
 - call openfi(fu5...
 - call openfi(fu6...
- a.3. For similar reasons the call to fiopen is skipped.
 - call fiopen

- a.4. Any calls to fcrate and the subsequent stop statement are deactivated. This allows control to return to CHARMM once the execution of POLYRATE is over.
- a.5. call readic is replaced by call cr_readic.
- a.6. A return statement is placed at the end of the routine.

b. differences of the file hooks.src distributed in CRATE from the generic hooks.f file of POLYRATE.

In this file, the ACES potential option has been deactivated to avoid problems during compilation. In addition,

- b.1. in subroutine ehook, the following lines are commented:

```
else if (ipot.eq.2)then
  call acalc(1,format)
```

- b.2. in subroutine ghook, the following line is commented:

```
call acalc(2,format)
```

- b.3. in subroutine ohook, the following lines are commented:

```
if(opt.eq.5) call gseta
call acalc(4,format)
```

- b.4. in subroutine prepj, the following line is commented:

```
call acaset(jtype)
```

- b.5. subroutine hhok, the following line is commented:

```
call acalc(3,format)
```

- b.6. the following line is added in subroutine surf to initiate the energy/gradient evaluation:

```
call cr_ener
```

- b.7. subroutine setup is deactivated.

- b.8. subroutine prep is deactivated.

Description of the changes made by the script file install_cr.com:***c. interface.src***

c.1. subroutine read5

The option *finish is added to terminate the command stream for POLYRATE from CHARMM.

The following 3 lines are added:

```
C      Stop input steam from CHARMM file
      else if (string(j:j+5).eq.'finish') then
      go to 1201
```

The next “call intab” was replaced by:

```
1201  call intab
```

c.2. subroutine renerg

The SPECBASIS options have been commented out since they are not available.

c.3. subroutine rline

The function case was changed to casito to avoid problems during the compilation.

c.4. subroutine readic was replaced by subroutine cr_readic.

d. energetics

d.1. subroutine deriv2 was deactivated to avoid problems during the compilation.

d.2. subroutine derv24 was deactivated to avoid problems during the complication.

e. polyag

e.1. The function erf has been replaced by cr_erf.

e.2. call readic is replaced by call cr_readic.

f. polysz

f.1. The function erf has been replaced by cr_erf.

g. ef, fromblas, polyrr

g.1. All tq1rat were replaced by cr_tq1rat.

g.2. All tq12 were replaced by cr_tq12.

h. All source files must have the file extension .src and mustn't have more than 72 columns to be processed by CHARMM preprocessor during compilation.

i. The file charmmrate.inc, which contains the definition of the CHARMMRATE variables required for new capabilities in CHARMMRATE-2.0, is copied to the directory chmroot/source/prate. The variable *lpmf* is defined here and activates the call to the subroutine pmf_project after the calculation of the primary-system Hessian matrix. The variables *itag_a*, *itag_b*, and *itag_c*, which define the three atoms involved in the reaction, are also defined in this file.

Description of the changes to be made by the user to perform projected instantaneous normal mode calculations with CHARMMRATE-2.0:

j. Two subroutines with file extension .f are provided in the CRATE package. These subroutines contain the additional code necessary to project the distinguished reaction coordinate used in the computation of the classical PMF out of the Hessian matrix of the primary-zone subsystem during a projected instantaneous normal mode analysis carried out with CHARMMRATE-version 2.0. The README file in the crate9.0 directory indicates the changes to be made by the user to perform this kind of calculation. Once the user has made the indicated changes, there will be a new subroutine fdiag in polyag.src and a new subroutine pmf_project in polymq.src. Then,

when the logical variable *lpmf* (defined in `charmmrate.inc`) is true, the subroutine `pmf_project` is called from the subroutine `fdiag` contained in the file `polyag.src`.

For details about CRATE installation, see the CHARMMRATE manual.

Test runs

CRATE-9.0 contains two test runs illustrating the use of CHARMMRATE for enzyme simulations. Each test job includes a full input file, initial coordinates, parameter files, and output files. It is recommended to read charmmrate.doc before trying to understand these test jobs.

Test run 1 performs projected instantaneous normal mode calculations for 4 configurations in the reactant window of the classical mechanical potential of mean force obtained for the hydride transfer reaction catalyzed by xylose isomerase. This job reads the 4 configurations from a CHARMM trajectory file, and for each of them it calculates the normal mode frequencies and stores them by writing to unit 62. Although this test run involves only 4 configurations, a real run for step 2 of stage 1 of the EA-VTST/MT method would typically involve many more configurations, for example, a hundred. Details about the input and output files for this job are provided in the README file in the crate9.0/test directory. After using CHARMMRATE to calculate projected instantaneous normal mode frequencies, the four files created by writing to unit 62 contain the frequencies for each configuration in cm^{-1} . These files may be used to calculate the quantum vibrational free energy correction using the utility programs in the crate9.0/util directory.

Test run 2 performs a dynamics calculation for one representative configuration of the ensemble of transition state structures obtained for the hydride transfer catalyzed by xylose isomerase. This kind of calculation is used in stage 2 of the EA-VTST/MT method to determine the individual transmission coefficients T_i and κ_i for a given configuration i . Details of the input and output files for this job are provided in the README file in the crate9.0/test directory.

Selection of configurations for the transition state ensemble

In order to illustrate the procedures that have been used in the EA-VTST/MT method, we here provide extra details of the actual selection of points in the transition state ensembles for the first several applications (LADH, MADH, XyI, and DHFR) of the EA-VTST/MT method.

In the first article on liver alcohol dehydrogenase¹ (LADH), we selected 20 configurations for the calculation of the transmission coefficient. In the second article² on this enzyme, where we employed the EA-VTST/MT method for the first time, we used 18 of these (two of the data sets were no longer available in the form required for EA-VTST/MT). These configurations were approximately evenly spaced in time, ~2.5 ps apart, from a 50 ps MD simulation. In this study, the transition state sampling involved a separate simulation with harmonic restraints on the breaking and forming bond distances. This is not the final version of our method, but the LADH calculations were nearly complete before we developed the final version of EA-VTST/MT. In the final version of EA-VTST/MT, the members of the transition state ensemble (TSE), i.e., the configurations to be used for calculating the average transmission coefficient γ , are selected directly from saved configurations in the umbrella sampling step, as described below.

The final value of the SSZ average transmission coefficient γ for LADH is 2.42 with a standard deviation of 0.20. The ESZ calculations were based on a smaller ensemble, as discussed in the article: *J. Phys. Chem. B* 105 (2001) 11326.

The methylamine dehydrogenase (MADH) calculations³ were also carried out before we developed the final version of the EA-VTST/MT method, and, as stated on page 391 of the paper [the reference for the corrected version of the paper is C. Alhambra et al., *Chem. Phys. Lett.* 355 (2002) 388], the TSE was selected from a separate MD run in which we used harmonic restraints (similar to those used for LADH) on the making and breaking bond distances. The paper notes that these restraints are not required by the

theory but are imposed in order to decrease the likelihood of an outlier in a small ensemble. In particular, the paper states that we selected 6 regularly spaced configurations from a 48 ps MD run. The actual spacing between points is 4.8 ps. The average transmission coefficient γ for MADH is stated on page 392 of the paper to be 57; the standard deviation (unpublished) is 34.

For xylose isomerase⁴ (XyI) and dihydrofolate reductase⁵ (DHFR), all aspects of the simulations were carried out with the final version of EA-VTST/MT. In particular, we do not use an extra MD run with harmonic restraints on bond distances. In order to explain what we do, it is useful to recall some aspects of the umbrella sampling method employed by CHARMM: The umbrella sampling involves MD runs in a series of windows, which are typically (but not always) evenly spaced in the umbrella coordinate z . For umbrella sampling analysis, the generated configurations are later sorted into bins, which are narrower than the windows; the typical width of bin is 0.01 Å. The bin with the highest free energy of activation is called z_* , and this identifies the variational transition state. In addition to analyzing all of the configurations for the umbrella sampling step, in each of the windows, CHARMM has an option to save selected configurations at regular intervals, and these saved configurations may be used for more detailed analysis. In order to carry out EA-VTST/MT calculations, it is necessary to use this option. The TSE should be selected as the set of saved configurations that have z values closest to z_* .

For XyI, each window is sampled for 25 ps, excluding the equilibration period. The step size is 1 fs, and a configuration is saved every 250 steps, which corresponds to an interval of 0.25 ps. Thus we saved 100 configurations per window. The variational transition state was found to be located at $z_*=0.045$ Å, which is approximately midway between two neighboring windows, centered at $z = 0$ and $z = 0.10$ Å. From the window closest to the variational transition state, we arbitrarily selected 4 configurations at 5, 10, 20, and 25 ps and with $z = 0.029, 0.017, 0.001, \text{ and } 0.033$ Å, respectively). These 4

configurations are on the reactant side of the variational state. In order to obtain a more representative TSE, we next selected a configuration from the window centered at 0.1 Å; in particular we selected the first saved configuration (the one at 0.25 ps) from that window; this configuration has $z = 0.103$ Å. The TSE is formed by these 5 configurations that fall within 0.058 Å of z_* . The average transmission coefficient γ for Xyl is 6.6 with a standard deviation of 0.3.

For DHFR, we used the most systematically consistent selection scheme for the TSE, and we recommend this approach for further use. In particular, we do not use approximately evenly spaced samples. Each window is sampled for 40 ps, excluding the equilibration period. The step size is 1 fs, and a configuration is saved every 100 steps, which corresponds to an interval of 0.1 ps. Thus we saved 400 configurations per window, and the TSE consists of the saved configurations with z closest to z_* . In particular, for the window that contains the variational transition state, 13 of the saved configurations fall in a bin of width 0.01 Å that contains z_* ; these are selected as the TSE. This bin extends from $z = -0.140$ Å to $z = -0.150$ Å. Thus the average distance in time between members of the TSE is $40 \text{ ps} \div 13 = 3 \text{ ps}$, and because the selection is made from saved configurations, no two selected configurations can possibly be closer than 0.1 ps. It turns out that the separations in time of consecutive configurations in the TSE are between 0.2 and 6.3 ps. The average transmission coefficient γ for DHFR is 2.5 with a standard deviation of 1.6.

Finally we consider short-chain acyl-CoA dehydrogenase (SCACD). In that study⁶ an ensemble of 15 secondary-zone configurations that are closest to z_* was chosen. The 15 configurations are within the interval $z_* \pm 0.015$, and they are at least 0.6 ps apart. The 15 transition state configurations were obtained from an umbrella sampling simulation, spanning 100 ps, which yields a total of 15 secondary-zone configurations within the 0.03 Å wide interval that we specified to define the transition state ensemble. The average static-secondary-zone transmission coefficient γ for SCACD is 1.9 with a

standard deviation of 2; the standard deviation is large because the distribution is strongly bimodal.

References

1. “Quantum Dynamics of Hydride Transfer in Enzyme Catalysis,” C. Alhambra, J. C. Corchado, M. L. Sánchez, J. Gao, and D. G. Truhlar, *Journal of the American Chemical Society* **122**, 8197-8203 (2000).
2. “Canonical Variational Theory for Enzyme Kinetics with the Protein Mean Force and Multidimensional Quantum Mechanical Tunneling Dynamics. Theory and Application to Liver Alcohol Dehydrogenase,” C. Alhambra, J. C. Corchado, M. L. Sánchez, M. Garcia-Viloca, J. Gao, and D. G. Truhlar, *Journal of Physical Chemistry B* **105**, 11326-11340 (2001).
3. “Quantum Mechanical Tunneling in Methylamine Dehydrogenase,” C. Alhambra, M. L. Sánchez, J. C. Corchado, J. Gao, and D. G. Truhlar, *Chemical Physics Letters* **347**, 512-518 (2001); reprinted correctly: **355**, 388-394 (2002).
4. “Hydride Transfer Catalyzed by Xylose Isomerase: Mechanism and Quantum Effects,” M. Garcia-Viloca, C. Alhambra, D. G. Truhlar, and J. Gao, *Journal of Computational Chemistry* **24**, 177-190 (2003)
5. to be published.
6. Free Energy Surface, Reaction Paths, and Kinetic Isotope Effects of Short-Chain Acyl-CoA Dehydrogenase,” T. D. Poulsen, M. Garcia-Viloca, J. Gao, and D.G. Truhlar, to be published.

Summary of CRATE and CHARMMRATE versions

There are two versions of CRATE available from the University of Minnesota

(<http://comp.chem.umn.edu>).

1. CRATE-8.1.1 is for historical interest only. It contains a utility to prepare versions of POLYRATE previous to POLYRATE-9.0 for use with the CHARMMRATE-version 1.0 module of CHARMM.
2. CRATE-9.0.1 is the most recent updated version of the CRATE package. It contains: (1) a utility to prepare POLYRATE-version 9.0 for use with the CHARMMRATE-version 1.0 or the CHARMMRATE-version 2.0 module of CHARMM, (2) test runs for enzyme kinetics simulations with CHARMMRATE, and (3) additional background information related to CRATE and CHARMMRATE.

There are two versions of CHARMMRATE available from the University of Minnesota

(<http://comp.chem.umn.edu>).

1. CHARMMRATE-1.0: the capabilities of this module of CHARMM are explained in the manual of this version, which is also available from the same Web page. In summary, CHARMMRATE-1.0 allows the user to calculate rate constants for enzymatic reactions with the method described in the paper: J.Am.Chem.Soc. 122 (2000) 8197. In that study, the final rate constant was calculated as the average of the individual CVT/OMT rate constants calculated by POLYRATE for the 20 selected configurations of the TSE. The method evolved and the final version, the EA-VTST/MT method, formulates the rate constant as the product of the free energy of activation determined from the quasi-classical potential of mean force obtained after following the procedure described in: J.Phys.Chem.B 105 (2001) 11326, and the average transmission coefficient calculated with the procedure also described in the cited paper.

CHARMMRATE-1.0 does not have the capability to perform projected instantaneous normal mode calculations and therefore, it does not allow the user to use the EA-VTST/MT method.

2. CHARMMRATE-2.0 is the most recent updated version of CHARMMRATE module of CHARMM. The capabilities of CHARMMRATE-2.0 are explained in the manual of this

version, which is also available from the same Web page. In summary, it allows the user to calculate reaction rates of enzymatic reactions with the EA-VTST/MT method.

Revision history of CRATE versions

CRATE-9.0.1: Modifications introduced in CRATE-9.0 to make CRATE-9.0.1

- a) The script `install_cr.com` has been modified to change the names of some subroutines in POLYRATE source files to the same names with a “x” added at the beginning. The reason for doing this change is to avoid conflicts of duplicate subroutine names between CHARMM and POLYRATE source files. These conflicts result in compilation errors in LINUX machines, whereas in other machines these conflicts do not prevent compilation. The modified names are: `xgfdiag`, `xvdcmp`, `xswap`, `xeffbath`, `xintcor`, `xmnbrak`, `xsaddle`, `xdaxpy`, `xddot`, `xrestor`.
- b) A list of possible machine-dependent problems has been added to the README file in the main directory of the CRATE-9.0.1 package.