

## ABCRATE On-line Manual

Program version: 10.0

Version date: September 26, 1997

Date of most recent manual update: September 26, 1997

Copyright 1997

Bruce C. Garrett  
Environmental Molecular Science Laboratory,  
Pacific Northwest National Laboratory  
Richland, WA 99352

Gillian C. Lynch, Thomas C. Allison, and Donald G. Truhlar  
Department of Chemistry and Supercomputer Institute  
University of Minnesota, Minneapolis, MN 55455

## ABSTRACT

ABCRATE is a computer program that calculates rate constants by generalized transition state theory (GTST) for atom-diatom reactions with collinear reaction paths. The program requires a user-supplied subprogram that, given the internal coordinates, can return the potential energy and its gradient at that geometry (sample subprograms for several systems are provided with ABCRATE). The reaction path for dynamics calculations is calculated as the path of steepest descent from the collinear saddle point, which is called the minimum-energy path (MEP). Vibrational modes are treated in curvilinear internal coordinates. The GTST methods include two versions of variational transition state theory (VTST), in particular canonical variational theory (CVT) and improved canonical variational theory (ICVT), and ABCRATE also includes the canonical unified statistical (CUS) model. All vibrational energy levels are quantized, and anharmonicity may be included by several options, including Morse and Wentzel-Kramers-Brillouin (WKB) anharmonicity for stretches and quartic and centrifugal oscillator anharmonicity for bends. Transmission coefficients to account for tunneling and nonclassical reflection may be included and the methods available for transmission coefficients include the minimum energy path semiclassical adiabatic ground-state method (ZCT, also

called ZCT), the centrifugal-dominant small-curvature semiclassical adiabatic ground-state method (CD-SCSAG, also called SCT), the large-curvature ground-state method version 3 (LCG3, also called LCT), the least-action ground-state method (LAG, also called LAT), and the microcanonical optimized multidimensional tunneling method (muOMT). Reverse reaction rates are always determined by detailed balance.

Support: The development of this program has been supported in part by the U. S. Department of Energy, Office of Basic Energy Sciences.

## CONTENTS

	Page
Title page	i
Contents	ii
User Agreement	iii
1. Introduction	1- 1
2. Data input	2- 1
3. Reaction path and vibrationally adiabatic barrier calculation	3- 1
3-A Reaction path calculation	3- 1
3-B Vibrationally adiabatic barrier calculation	3- 3
4. Options for treating the generalized normal modes	4- 1
4-A Options for treating the stretch	4- 1
4-B Options for treating the bends	4- 1
5. Tunneling	5- 1
6. Canonical variational transition state theory	6- 1
7. Potential energy surfaces	7- 1
8. Input data	8- 1
9. File usage	9- 1
10. Computers on which the code has been tested	10- 1
11. Test runs	11- 1
11-A Table of potential functions used for the test suite	11- 1
11-B Description of the test runs	11- 2
11-B-1 Test run DH2tr1	11- 2
11-B-2 Test run DH2tr2	11- 3
11-B-3 Test run OH2tr1	11- 4
11-B-4 Test run ClHCltr1	11- 5
11-B-5 Test run ClHCltr2	11- 6
11-B-6 Test run ClHBrtr1	11- 7
11-B-7 Test run FH2tr1	11- 8
11-B-8 Test run DFHtr1	11- 9
11-B-9 Test run HFHtr1	11-10
11-C Timing information for test suite runs	11-11
12. File organization for the distributed version	12- 1
12-A Files in the distribution package	12- 1
12-B Installing and running the test suite	12- 3

## USER AGREEMENT

ABCRATE is a licensed program. Prospective licensees may use either the standard CPC non-profit use license printed in each volume of Computer Physics Communications (for non-profit users) or the ABCRATE license available at <http://comp.chem.umn.edu/abcrate> (for non-profit or for-profit users). The ABCRATE license does not allow redistribution, but it permits sharing and copying within a research group. Publications resulting from using this program should reference the article in Computer Physics Communications that describes the program.

## 1. INTRODUCTION

The ABCRATE code calculates collinear (1D) and 3D thermal rate constants for 3-atom reactions,  $A + BC \rightarrow AB + C$  with collinear or nearly collinear transition states. The code is designed for systems in which the minimum energy path (MEP), which is the path of steepest descent from the saddle point in isoinertial coordinates, is collinear. However, this code can also be used for systems in which the MEP is only slightly bent from collinear and in which the double-well potential for the bending motion is not too deep. A list of pertinent references for the methods used in this code is given below (acronyms in the table are all explained in the abstract):

Topic	Reference
Collinear CVT	B. C. Garrett and D. G. Truhlar, J. Phys. Chem. 83, 1052, (1979); 83, 3058(E) (1979); 87, 4553(E) (1983). B. C. Garrett and D. G. Truhlar, J. Phys. Chem. 83, 1079, (1979); 84, 682(E) (1980); 87, 4553(E) (1983).
3D CVT	B. C. Garrett and D. G. Truhlar, J. Amer. Chem. Soc. 101, 4534 (1979). B. C. Garrett and D. G. Truhlar, J. Chem. Phys. 72, 3460 (1980).
ICVT	B. C. Garrett, D. G. Truhlar, R. S. Grev, and A. W. Magnuson, J. Phys. Chem. 84, 1730 (1980).
Bend partition functions	B. C. Garrett and D. G. Truhlar, J. Phys. Chem. 83, 1915 (1979) B. C. Garrett and D. G. Truhlar, J. Phys. Chem. 95, 10374 (1991)
Semiclassical Tunneling	B. C. Garrett, and D. G. Truhlar, J. Phys. Chem. 83, 2921 (1979).
CD-SCSAG	R. T. Skodje, D. G. Truhlar, and B. C. Garrett, J. Phys. Chem. 85, 3019 (1981). R. T. Skodje, D. G. Truhlar, and B. C. Garrett, J. Chem. Phys. 77, 5955 (1982).
LAG, LCG3	B. C. Garrett and D. G. Truhlar, J. Chem. Phys. 79, 4931 (1983). B. C. Garrett, N. Abusalbi, D. J. Kouri, and D. G. Truhlar, J. Chem. Phys. 83, 2252 (1985).
System with no saddle point	S. N. Rai and D. G. Truhlar, J. Chem. Phys.

WKB energy levels	79, 6046 (1983). B. C. Garrett and D. G. Truhlar, J. Chem. Phys. 81, 309 (1984).
Computational methods	D. G. Truhlar, A. D. Isaacson, and B. C. Garrett, "Generalized Transition State Theory", in The Theory of Chemical Reaction Dynamics, ed. M. Baer, CRC Press, Boca Raton (1985).

The code has four stages: data input, calculation of reaction path and vibrationally adiabatic ground-state potential curve, calculation of transmission coefficients, and calculation of generalized free-energy-of-activation curves and variational TST rate constants. The program can be stopped after completion of any of these stages. Brief descriptions of the operation of each of these stages are given in Sections 2 through 6 of this manual. The user must supply the potential energy surface, and it must be in

the form described in Section 7. The input data is described in detail in Section 8, and the file usage is outlined in Section 9. Section 10 lists the computers on which the code has been tested along with timings for the test runs that make up the test suite. The test runs are described in Section 11, and Section 12 contains a description of the installation procedure for the program.

## 2. DATA INPUT

In this section of the program all data is read and variables are initialized. A line-by-line description of the input data is provided in Section 8; only an overview of the operation of this portion of the program is given here.

After the data has been read in, the energy for the dissociation to three separated atoms is computed, asymptotic geometries and Morse parameters of the diatomic reactants and products are computed, the saddle point(s) is (are) located, and a normal mode analysis is completed for each stationary point. Next, partition functions are computed, and finally conventional transition state theory rate constants are evaluated.

The zero of energy is taken to be the bottom of the asymptotic vibrational well for the reactants, and the potential energy function must be consistent with this convention. The arrangement of the forward reaction, which is  $A + BC \rightarrow AB + C$ , is dictated by the masses  $XMA$ ,  $XMB$ , and  $XMC$  (see page 8-1). The bond lengths passed to the potential energy surface reflect this arrangement and the user-supplied potential energy surface routine, which is described in Section 7, must be consistent with this arrangement.

Although by far the most common case encountered is a single saddle point between a given set of reactants and products, the program will handle from 0 to 4 saddle points, and the user must supply approximate geometries for each of them. If more than one saddle point is listed the order of the approximate geometries must be the same as the order that the saddle points will be encountered in moving along the MEP from reactants to products. In the case where there is no saddle point the user supplies the geometry of an approximate location along the MEP in the asymptotic reactant region; the reaction coordinate calculation begins on the MEP near this location and follows the steepest descent path into the product region. For this reason, systems with no saddle point must be oriented in the exoergic direction.

### 3. REACTION PATH AND VIBRATIONALLY ADIABATIC BARRIER CALCULATION

#### 3-A. Reaction path calculation

The MEP is found by following the negative of the mass-scaled gradient of the potential in a mass-scaled coordinate system. The distance along the MEP is called  $s$ ;  $s$  is zero at the saddle point (if there is more than one saddle point,  $s = 0$  at the one for which the ground-state adiabatic energy is greatest),  $s$  is negative in the reactant region, and  $s$  is positive in the product region. In the case of a system with no saddle point, the calculation begins in the asymptotic reactant region with  $s=0$ , and follows the negative mass-scaled gradient into the product region. Therefore, for a system without a saddle point the potential must be arranged so that the forward reaction is exoergic. If the reverse reaction is the one of interest, the rates for this reaction are also computed using detailed balance. For systems with a single saddle point, the calculation begins at the saddle point and the negative mass-scaled gradient is followed into both the reactant and product regions. The initial step is taken along the normal coordinate for the unbound motion at the saddle point. If a well in the potential energy surface along the MEP is encountered, the reaction-coordinate calculation stops at the bottom of the well. The calculation is restarted at the asymptotic geometry for the direction being followed and proceeds back towards the well. If the calculation makes it back into the well the two segments are joined into one. However, if it is unable to find its way back into the well an error message is issued and execution of the program is terminated. Also, if the energy at the bottom of the well is greater than that at the asymptotic value of the MEP, another saddle point exists between the well and the asymptote, and the program quits. For systems in which more than one saddle point is listed, a single continuous reaction coordinate connecting reactants to products is generated by sequentially starting the calculation at each of the saddle points and computing segments in both the forward and reverse directions. Checks are made that the wells between the saddle points connect the adjacent saddle points. If the well between two adjacent saddle points does not connect them an error message is issued and the program terminates.

The negative of the mass-scaled gradient is followed using a fixed step size DEL. At regular intervals DELSV along the MEP, the local potential in directions transverse to the MEP is fitted to a Morse potential for the stretching motion and a harmonic-quartic potential for the bending motion. The values of the coordinates, potential energy, Morse parameters, bending parameters, mass-scaled gradient vector, moment of inertia, and curvature of the reaction path are stored in arrays in the common block BLKPCM. This grid

of reaction-coordinate information will be used to compute the vibrationally adiabatic curve, as well as in the tunneling and VTST calculations. The Morse parameters are evaluated using the Morse approximation I [J. Phys. Chem. 83, 1079 (1979)]. The bending parameters are fitted in one of four ways depending on the option chosen (IBOPT). If IBOPT<3, the harmonic force constant FB is first fitted using the second derivative with respect to the angle at the equilibrium geometry. If FB is positive, the quartic force constant AB is fitted using the fourth derivative with respect to the angle. If FB is negative, the collinear reaction coordinate is not the minimum energy path, and the bending potential for geometries near collinear will be a double well. If the depth of the well is not too large, the methods used in this program are still valid and one of the following two methods is used to refit FB and to fit

AB. If IBOPT=1, FB and AB are fitted to reproduce the well depth and the outer turning point for the bend for the energy at the collinear geometry. If IBOPT=2, FB and AB are fitted to reproduce the location and depth of the well. The fourth method of fitting FB and AB can be used for  $FB < 0$  or  $FB > 0$ . For IBOPT=3, FB is not checked to be positive, FB and AB are fitted to reproduce the correct bending potential at two angles, THETA(1) and THETA(2).

Two criteria are checked for ending the reaction coordinate calculation in an asymptotic region. The first criterion is : in the reactant region, the calculation is ended if  $s < SLM$ ; in the product region, the calculation is ended if  $s > SLP$ . The second criterion is: if the values of the Morse parameters and the potential along the reaction coordinate have a relative difference from the asymptotic values of less than EPSASY, then the calculation is ended. Once the reaction coordinate calculation is sufficiently far into the asymptotic region, the gradient is no longer followed. Instead, steps are taken along the asymptotic gradient and the MEP is located by finding the minimum in the potential well normal to the this vector. The switch is made to this mode when the cosine of the angle between the gradient vector and the asymptotic gradient vector is greater than ACASY for three consecutive locations of grid points along the reaction coordinate.

The curvature of the reaction coordinate is calculated using a two-point central difference formula for the derivative of the gradient. This numerical derivative is computed with a step size DELCUR except near the saddle point where it is computed with a step size DELCSP. The curvature is calculated from  $s = SM$  to  $s = SP$ .

After completion of the reaction-path calculation the grid of stored information may be written, in formatted form, to the file path.1 which is linked to FORTRAN unit 1. Subsequent runs on the same reaction can reuse this information without having to recalculate the reaction coordinate information.

When the collinear saddle point has an imaginary bend frequency, it is a hilltop in three dimensions (third-order saddle point) rather than a first-order saddle point. In this case we still compute a reaction path by following the mass-scaled path of steepest descent from the collinear saddle point. This ridge path is taken as a reference path, and it is treated just like the case of a first-order saddle point. Thus, in the rest of this manual this reference path is called the reaction path or MEP, just as when the bend frequency is real. A ridge path is also followed in the case where the bend frequency is real at the saddle point but becomes imaginary as one proceeds along the reaction path. In such a case the code follows the collinear MEP

even after the bend frequency has become imaginary.

### 3-B. Vibrationally adiabatic barrier calculation

The adiabatic barriers are calculated at the grid points along the MEP at which the reaction-coordinate information has been stored. The global maximum on this grid is located, and the maximum of the adiabatic barrier is determined by the following method. A quadratic fit of the adiabatic barrier near the maximum is made using the maximum on the grid and one point on either side of the maximum. New reaction coordinate information is generated at the maximum of the quadratic fit and for one point on either side of the maximum. These three new points are equally spaced; the spacing is one-half of the spacing in the old grid. These three new points are then used to obtain a new quadratic fit; the maximum of this quadratic fit is used as the new maximum and the procedure is repeated. This procedure is repeated up to 10 times or until the location of the maximum of the quadratic fit has a relative difference from the previous value of less than 1.E-6. A cubic spline fit is made to the adiabatic energies including this new point.

#### 4. OPTIONS FOR TREATING THE GENERALIZED NORMAL MODES

The generalized normal modes, i. e., the normal modes orthogonal to the reaction coordinate at nonstationary points as well as stationary ones, are treated as independent. For a 3-body system in 3D with a collinear reference path there are three normal modes: a symmetric stretch and a doubly degenerate bend. All the generalized normal modes can be treated by a harmonic approximation, and several anharmonicity options are also available, but these options depend on the type of mode being treated.

Before discussing anharmonicity we note that at nonstationary points even the harmonic frequencies are not unique; in particular they depend on the coordinate system. ABCRATE uses the following coordinates for the vibrations orthogonal to the MEP. First, the stretch coordinate is taken as the linear combination of the two mass-scaled Jacobian coordinates for collinear ABC that are orthogonal to the mass-scaled gradient. Then the bend coordinates are defined as the deviations of the ABC bond angle from  $\pi$  in each of the two directions perpendicular to the collinear axis.

##### 4-A. Options for treating the stretch

The vibrational stretch can be treated by a harmonic approximation, by the Morse I approximation, or by the WKB method. If the harmonic [IGS(5)=1] or Morse I [IGS(5)=2] method is chosen, the local potential at each save grid point is fit to a harmonic or a Morse curve and all eigenvalues (ground-state and excited state) are determined from the parameters for the fitted curves. If the WKB [IGS(5)-3-5] method is chosen all the WKB calculations are performed using the true potential energy curve, i. e., any energy or derivative of the energy with respect to the coordinates are obtained from calls to the user supplied potential not from a fit of the potential. When the WKB option is chosen for a thermal reaction the ground-state eigenvalues are determined by the WKB approximation using the true potential; all excited state eigenvalues are determined by the Morse I approximation. For the case in which WKB anharmonicity is chosen for a state-selected calculation, the eigenvalue for the stretch vibrational state equal to NSTATE is computed by the WKB approximation using the true potential. If IGS(5)=4, the WKB energy levels are written to the file wkb.14 which is linked to FORTRAN unit 14. The WKB energy level calculations can be avoided in subsequent runs by reading in these WKB energy levels, this is controlled by IGS(5)=5.

##### 4-B. Options for treating the bends

The vibrational bends can be treated by a harmonic [IGS(6)=1] or a harmonic-quartic approximation [IGS(6)=2-3], and as described in Section 3 the local potential at each save point, DELSV, is fit in one of four ways depending on the value of IBOPT. The harmonic-quartic energy levels can be computed by a semiclassical method [IGS(6)=2], a perturbation-variation method [IGS(6)=3], or the centrifugal oscillator formulation [IGS(6)=4]. With the first two methods the bending modes are treated as a doubly degenerate pair of independent modes. For state-selected calculations [LSBEND], the total quanta in the bending modes can be divided between the two bends [NBND1, NBND2], when the bends are treated as uncoupled, i.e., when IGS(6) does not equal 4. For example, if a state-selected calculation is chosen, i.e., LSBEND does not equal 0, and the calculation specifies 4 quanta in the bends this can be distributed in any of the following ways: (NBND1=0, NBND2=4), (NBND1=2, NBND2=2), or (NBND1=1, NBND2=3).

The centrifugal oscillator formulation includes coupling between the two bending degrees of freedom, and the quantum numbers for this coupled motion are no longer NBEND1 and NBEND2, but they are  $v_2$ , the bending quantum number, and K, the vibrational angular momentum quantum number which depends on the total angular momentum, J, of the system. In state-selected centrifugal oscillator calculations the input for the bend quantum numbers [NBND1, NBND2] has a different meaning from the calculations in which the bending motions are treated as uncoupled. Table 1 demonstrates the correspondence between NBND1 and NBND2 and  $v_2$  and K for state-selected calculations for the centrifugal oscillator method.

TABLE 1. Correspondence between NBND1, NBND2 and  $v_2$ , K.

total quanta	$v_2$	K	NBND1	NBND2
0	0	0	0	0
1	1	1	0	1
2	2	0	1	0
4	4	0	2	0
6	6	0	3	0
	6	2	2	2
	6	4	1	4

## 5. TUNNELING

All tunneling is done semiclassically with parabolic uniformization. Several vibrationally adiabatic and two nonadiabatic methods are used; in all the methods the vibrationally adiabatic ground-state (or selected stretch-excited) potential energy curve is used.

For the vibrationally adiabatic methods the barrier penetration integrals are evaluated using the spline fit to the adiabatic barrier. The barrier penetration integrals are evaluated using repeated Kronrod quadrature in NSEG segments with quadratures of size  $NQ2$  and  $2*NQ2+1$ . Reaction-path curvature is accounted for approximately in most of these methods using a reaction-coordinate-dependent reduced mass which becomes smaller in regions of high curvature. For values of  $s$  off the grid of stored parameters the reduced mass is computed using AITKEN interpolations of the stored parameters.

In the nonadiabatic methods the tunneling paths are parametrized paths from reactants to products. The end points for the paths are determined by the turning points along the adiabatic potential. The effective potential along the tunneling path is either the adiabatic potential if the point lies within the adiabatic region or the actual potential energy surface for points outside the adiabatic region. The barrier penetration integrals are also evaluated using repeated Kronrod quadrature.

The tunneling probabilities are computed on a grid of energies, again using Kronrod's quadrature scheme with quadratures of size  $NQ$  and  $2*NQ+1$ . It is this grid of energies that is used in the evaluation of the integrals to obtain the transmission coefficients.

The code include three nonoptimized tunneling approximations (ZCT, SCT, and LCT) and two optimized ones (LAT and muOMT).

The ZCT method is zero-curvature tunneling. It is the same as MEPSAG.

SCT denotes small-curvature tunneling. For atom-diatom reactions with collinear reference paths it is the same as SCSAG and CD-SCSAG.

LCT denotes large-curvature tunneling. It is the same as LCG3.

LAT denotes least-action tunneling. It is the same as LAG.

The G in the old abbreviations denotes ground-state. We still use the

ground-state transmission coefficient when we compute thermal rate coefficients, but we no longer include the G in the method name. Previously we would shorten the acronyms to MEPSA, SCSA, CD-SCSA, LC3, and LA when we used these methods for excited-state reactions. The new acronyms, ending in T, may be used more invariantly.

In this version of the program the microcanonical optimized multidimensional tunneling transmission (muOMT) method is also available. The muOMT coefficient is the ratio of the Boltzmann average of the quantum muOMT

transmission probability to the Boltzmann average of the classical transmission probability. The muOMT quantum transmission probability is taken as the larger of the SCT and the LCT probabilities at each energy on the grid of energies at which the tunneling probabilities are evaluated.

The reaction rate calculations do not have to be carried out in the exothermic direction, unless the large-curvature tunneling methods with tunneling into product excited states are being used; then the reaction must be set up in the exothermic direction. This directional specificity is necessary because the code does not perform any checks to determine the exothermic direction, and the tunneling calculations are carried out in the direction specified. Therefore, to include product excited states in the large-curvature tunneling calculations requires that the reaction be carried out in the exothermic direction. If the reaction in the endothermic direction is the one of interest, this does not prevent its study as the reverse rates are always computed by detailed balance.

## 6. CANONICAL VARIATIONAL TRANSITION STATE THEORY

The location of the CVT transition state for each temperature is the maximum in the generalized free-energy-of-activation curve. The generalized free energy of activation is calculated on the grid of stored reaction-coordinate information. At each of the two highest local maxima, a quadratic fit is obtained, and three new points of reaction-coordinate information are generated: at the maximum of the quadratic fit and at equally spaced points on either side of the maximum. The generalized free energy is computed on this 3-point grid, and a new quadratic fit is obtained. This procedure is iterated to convergence within  $1.E-6$  or for 10 iterations. This procedure is then repeated with the threshold correction added to the generalized free-energy-of-activation curves. Finding the absolute maximum of the generalized free-energy-of-activation curve including the threshold correction gives the improved CVT (ICVT) results.

After the maxima are located, the reaction-coordinate information for the CVT and ICVT transition state locations are summarized, and the rate constants are computed. The rate constants for the Goren model are also calculated.

This entire procedure is repeated for each temperature specified in the input data file. Next, the Arrhenius activation parameter and energy are computed for each pair of temperatures listed in the input data file. Finally, a summary of the transmission coefficients, the rate constants, and the Arrhenius parameters are provided for both the forward and the reverse reactions at the end of the output file.

## 7. POTENTIAL ENERGY SURFACES

The form of the subroutine for evaluating the potential energy is as follows. The user must supply a subroutine PREPEF which optionally reads in potential parameters and sets up any necessary data. This routine is called only once in each run of the program. If no initialization is required by the potential energy function, PREPEF may simply return without doing anything. The user must also supply a subroutine PEF. The subroutine PEF evaluates the potential and the derivatives of the potential with respect to the internal coordinates (in particular, with respect to the three internuclear distances) for a given configuration. The subprogram PREPEF is called with no argument list, but the subprogram PEF is called with the following argument list

```
CALL PEF (R1, R2, R3, ENERGY, DEDR1, DEDR2, DEDR3, NDER)
```

where R1, R2, and R3 are the internuclear bond distances (A to B, B to C, and A to C, respectively). ENERGY is the classical Born-Oppenheimer energy, and DEDR1, DEDR2, and DEDR3 are the derivatives of the energy with respect to the internal coordinates R1, R2, and R3, respectively. The units are hartrees for ENERGY, bohr for R1, R2, and R3, and hartrees per bohr for DEDR1, DEDR2, and DEDR3. NDER is an integer flag which controls the calculation of the derivatives of the energy with respect to the coordinates. If NDER is equal to 0 then the derivatives are not needed and only the energy is required, but if NDER is equal to 1 then both the energy and the derivatives are needed. This means that the derivatives are not requested if they are not needed and this may lead to a savings in CPU time if the potential energy routine can respond efficiently to both type of call.

Because the program is designed to compute 3D reaction rates, a full 3D potential energy surface is required. A potential energy surface routine which is limited to collinear geometries can be generalized to a full 3D PES using the BEBO-bend method of J. Chem. Phys. 76, 2321 (1982).

The code is designed such that the energy at the bottom of the BC well, i.e., the energy computed for A far from BC and BC at its equilibrium diatomic distance, must be defined as zero in the user-supplied potential energy surface. If the energy at this configuration is not zero in the user-supplied potential, the code will print an error message and stop. The calculations do not have to be carried out in the exothermic direction in all cases; however, if the large-curvature tunneling methods are being used with tunneling into product excited states, then the reaction must

be set up in the exothermic direction as discussed in the last paragraph of Section 5.

All common block names in ABCRATE end with "COM" or "CM." Therefore the user should be careful that any common block names in the potential energy subprograms that end in these strings do not pose conflicts.

## 8. INPUT DATA

A record-by-record description of the input to ABCRATE follows. The meaning of \* and \*\* is explained at the end of this section on page 8-8. LFN denotes logical file number, i.e., the FORTRAN unit number which is used for the data input or output.

Note that for logical variables, a blank is the same as false.

## Record Format Description

- |     |                   |   |
|-----|-------------------|---|
| 1,2 | 120A1             | Two title lines, limited to 120 characters per line.  |
| 3   | 5X,<br>3(3X,A2)   | Atomic labels.  |
| 4   | 3F20.10           | XMA**, XMB**, XMC** - masses of A, B, and C, respectively, in atomic units. These masses dictate the direction of the forward reaction. |
| 5   | 10I2,<br>2X, 20I2 | IGS(1-10), IOPT(1-20) - Calculation options   |

IGS( 1) - Allows the program to be stopped after certain stages.

- 1 - The program exits after computing reactant, product, and saddle point properties, and TST rates.
- 2 - The program exits after the reaction coordinate and the vibrationally adiabatic potential energy curve are calculated.
- 3 - The program exits after the transmission coefficients are calculated.
- 4 - Full calculation.

IGS( 2) - Controls the type of calculation to be performed.

- 1 - Collinear calculation (1D) only.
- 2 - Three dimensional (3D) calculation only.
- 3 - Both 1D and 3D calculations are performed.
- 4 - Both 1D and 3D calculations are performed, but electronic degeneracies and electronic excited states are omitted in the 1D calculation.

IGS( 3) - Selects the symmetry of the reaction path.

- 1 - The reaction path is symmetric about the saddle point.
- 2 - The reaction path is asymmetric about the saddle point.

IGS( 4) - Controls the reaction path calculation.

- 1 - The reaction path is calculated during program execution.
- 2 - The reaction path is read from LFN 1.
- 3 - The reaction path is calculated and written to LFN 1.

IGS( 5) - Controls the treatment of the vibrational stretches.

- 1 - Harmonic method.
- 2 - Morse I method.
- 3 - WKB method without special read/write options.
- 4 - WKB method. Results are written to LFN 14.
- 5 - WKB method. Results are read from LFN 14.

- IGS( 6) - Controls treatment of the vibrational bends.
- 1 - Harmonic method.
  - 2 - Harmonic-quartic method with energy levels computed using the semiclassical method.
  - 3 - Harmonic-quartic method with energy levels computed using the semiclassical method if the harmonic force constant is negative and using the perturbation-variation method if the harmonic force constant is positive.
  - 4 - Centrifugal oscillator method.
  - 5 - Centrifugal oscillator method w/ Pitzer-Gwinn for functions. Parameters of harm-quart. potential fit to give WKB ground state energy level on real potential.
- IGS( 7) - Controls the tunneling methods used.
- 1 - Use ZCT and SCT methods.
  - 2 - Use ZCT and SCT methods, and read LCT and LAT transmission probabilities as needed from LFN 11 for 1D reactions, and/or from LFN 12 for 3D reactions.
  - 3 - Use ZCT, SCT, LCT, and three-point LAT methods. In the three-point LAT method, the code does not perform a root search for the optimal alpha in the LAT calculation; rather the minimum in theta is taken from the three points, alpha = 0.0, 0.5, and 1.0.
  - 4 - Use ZCT, SCT, LCT, and LAT methods. The optimal alpha is found in the LAT calculation.
  - 5 - Use ZCT, SCT, LCT, and three-point LAT methods. The code does not optimize alpha in the LAT calculation; the minimum in theta is taken from three points, alpha = 0.0, 0.5, and 1.0. Write the transmission probabilities from the methods including large-curvature tunneling (LCT and LAT) to LFN 11 for 1D reactions, and to LFN 12 for 3D reactions.
  - 6 - Use ZCT, SCT, LCT, and LAT methods, including the optimization of alpha in the LAT calculation. Write the transmission probabilities from the large-curvature tunneling methods (LCT and LAT) to LFN 11 for 1D reactions, and to LFN 12 for 3D reactions.
- IGS( 8) - Controls printing of summary tables.
- 1 - Do not print any summary tables.
  - 2 - Print a table of TST properties to LFN 21.
  - 3 - Print a table of selected rate constants for LFN 22.
  - 4 - Print both tables.

IGS( 9) - Not used.

IGS(10) - Not used.

IOPT(1) - if IOPT(1) is not zero, a state specific LAT calculation is performed where the initial state is NSTATE (see item 15 below) and a sum is performed over final states from IOPT(2) to IOPT(3).

IOPT(2) - if IOPT(1) is not zero, specifies the smallest value in range of final states.

IOPT(3) - if IOPT(1) is not zero, specifies the largest value in range of final states.

IOPT(4) - if IOPT(4) is not zero, then the reactant asymptotic value is not used in finding the maxima of the vibrationally adiabatic ground-state curve (VaG).

IOPT(5) - if IOPT(5) is not zero, then the product asymptotic value is not used in finding the maxima of the vibrationally adiabatic ground-state curve (VaG).

IOPT(6) - controls the printing of the generalized free energy of activation curves.

= 0, the free energy data is printed at all the temperatures.

= 1, the free energy data is not printed.

= 2, the free energy data is printed at the temperatures specified in records 19 and 20.

IOPT(7-20) - not used

6        7I5, F10. 5,    NSEG, NQ, NSEG2, NQ2, NSEG3, NQ3, IOPTAU, SINT, NSS,  
           2I5, E10. 2    NINTS, TOLLAG - parameters for Kappas.

NSEG, NQ, NSEG2, NQ2 - number of segments (NSEG) and Legendre quadrature points (NQ) in each segment for energy integration (Boltzmann average) involved in computing the transmission coefficient and the number of segments (NSEG2) and Legendre quadrature points (NQ2) for the phase integral quadratures for the SAG methods. The number of Kronrod quadrature points is twice the number of Legendre quadrature points plus one.

NSEG3, NQ3 - are the number of segments and quadrature points used for the integration of the LCT phase integrals.

IOPTAU - =0, the period of the vibrational period of the stretch approximated by the zero-point energy method.

=1, the period is calculated from the derivative of the phase integral.

SINT, NS, NINTS - the approximate segment length, number of quadrature points in each segment, and order of interpolation used in the integration over s in the LAT methods.

TOLLAG - controls the accuracy to which the minimization of the LAT probabilities are carried out with respect to alpha. Recommended value is 1.E-7.

- 7      415            NINT, NQWK, NQCO, NLEVEL
- NINT, NQWK - order of Lagrange interpolation used to obtain the reaction coordinate information from the stored grid, and number of quadrature points used in integration of phase integrals for WKB eigenvalues.
- NQCO, NLEVEL - number of quadrature points used in the centrifugal oscillator phase integrals, and the maximum primary quantum number in the centrifugal oscillator partition functions.
- 8      F20.10          RIASY, R2ASY - approximate equilibrium geometries for product and reactant diatomics, respectively, in bohr.
- 9      I5                NSAD\*\* - number of saddle points on along the minimum energy path. (This version of the code allows a maximum of 4.) The next lines are the approximate geometries of the saddle points. There is one line per saddle point. If NSAD=0, one line is read: an approximate geometry along the reaction coordinate in the asymptotic reactant region. If the system is being treated as symmetric (i.e., LSYM = .T.), then NSAD should be the number of saddle points encountered along the part of the reaction coordinate that is actually computed. Thus for a symmetric system with twin saddle points, NSAD should be set to 1.
- 10     2F20.10          (R1SP(I), R2SP(I), I=1, NSAD) - approximate geometries of the saddle points (A to B, and B to C internuclear distances, respectively, in bohr.)
- 11     2F20.10,          SM2, SP2, STRUNL, STRUNR - reaction-coordinate parameters.  
2F15.8
- SM2, SP2 - values of s in the reactant and product valleys at which the curvature is fitted to an exponential form to be one-half the value at the values SM, SP, respectively.

STRUNL,STRUNR - values of  $s$  in the reactant and product valleys at which the reaction path is truncated. If both are zero no truncation is performed.

If IGS(4)=2 then skip to record 15.

- 12      2F20. 10      SLM, SLP - values of the reaction coordinate at which the reaction coordinate calculation is ended. SLM is in the reactant region and SLP is in the product region.
- 13      4F20. 10      DEL, DELSV, ACASY, EPSASY - reaction-coordinate parameters
- DEL -      step size between gradient evaluations when calculating the MEP.
- DELSV -    interval at which the reaction-coordinate information is stored.
- ACASY -    determines when the calculation of the reaction coordinate switches from following the gradient to taking steps along the asymptotic value of the gradient.
- EPSASY -   if the relative difference between the potential parameters at a point along the reaction coordinate is less than EPSASY, the calculation is ended.
- 14      4F20. 10      DELCUR, DELCSP, SM, SP - curvature parameters
- DELCUR -   step size used in the central difference numerical first derivative.
- DELCSP -   special step size used for the numerical derivatives near the saddle point.
- SM, SP -   bounds on the values of the reaction coordinate for which the curvature is calculated.

- 15 I5, 2F10. 4, IBOPT\*, (THETA(I)\*, I=1, 2), LSTATE, NSTATE, SAD1, SAD2, LSBEND,  
 1X, I1, I5, NBND1, NBND2 -bend-potential parameters  
 2F20. 10,  
 1X, I1, 2I3
- IBOPT\* - =1, the harmonic force constant FB is fitted using the second derivative with respect to the angle at the equilibrium geometry (collinear). If FB is positive, the quartic force constant AB is fitted using the fourth derivative with respect to the angle. If FB is negative, FB and AB are fitted to reproduce the well depth and the turning point for an energy equal to the local maximum of the double well.  
 =2, same as IBOPT=1 except if FB is negative. If FB is negative, FB and AB are fitted to reproduce the location and depth of the well.  
 =3, FB and AB are fitted using two values of the actual potential at the two angles THETA.
- THETA\* - the two angles used if IBOPT=3.
- LSTATE - =0, if thermal average over stretch vibrational states.  
 =1, if state-selected option is chosen for stretch.
- NSTATE - the stretch vibrational state.
- SAD1, SAD2 - limits on a region in s in which a maximum is searched for in the adiabatic potential.
- LSBEND - =0, if thermal average over bend vibrational states.  
 =1, if state-selected option is chosen for bend.
- NBND1, NBND2 - bend vibrational states  
 if IGS(6)=4, then these two variables represent v2 and K, respectively.
- 16 615 NTMAX, NPGTST, NPRP, NPAD, NPSAG, NPLAG - number of temperatures and print options.
- NTMAX - number of temperatures to be read in (max. of 100).
- NPGTST - the number of grid values skipped between printing in free energy calculations.
- NPRP - the number of grid values skipped between

printing the reaction-path summary.  
NPAD - the number of grid values skipped between  
printing the adiabatic potential curves.  
NPSAG - 0 if SAG-type probabilities are printed,  
otherwise printing is suppressed.  
NPLAG - 0 if LCT-type probabilities are printed,  
otherwise printing is suppressed.

17      10(1X, F7. 1)      TEMP(I=1, NTMAX) - temperatures (in Kelvin)

Note: the code will look for one line of record 17  
data if NTMAX is less than or equal to 10, it will look  
for two lines if NTMAX is between 1 and 20, and so  
forth, up to 10 lines if NTMAX is in the range 91-100.

18        1X, L1, 5X, L1        LEQ(1), LEQ(2)

LEQ(I) - if true, additional rates are read in for calculating ratios of the computed rates to those read in. I=1 for 1D, I=2 for 3D.

Loop over lines 19-21 for I = 1, 2 (1D, 3D)

If LEQ(I) = .FALSE. then skip over lines 19-21

19        4G20. 10            RKQ(J=1, NTMAX, I) - list of accurate quantal rates for (I=1) the collinear reaction, and (I=2) the 3D reaction.

20        4G20. 10            RMVT(J=1, NTMAX, I) - list of microcanonical VTST thermal rates for the collinear reaction if I=1, and for the 3D reaction if I=2.

21        4G20. 10            RUS(J=1, NTMAX, I) - list of unified statistical rates for the collinear reaction if I=1, and for the 3D reaction if I=2.

If IOPT(6) does not equal 2, skip to line 24

22        2X, I3                NIDX     - the number of temperatures at which free energy data will be printed.

23        10I5                 ITEMP     - indices for the temperatures at which the free energy data is printed. There are NIDX indices.

24        I5                    IACT -     number of activation energies to be calculated (max. of 10).

25        10I5                 (NT1(I), NT2(I), I=1, IACT) - pair of indices for the temperatures over which the activation energies are calculated. There are IACT pairs of these indices.

26      5I5            NEL(I=1,5) - number of electronic states for each of  
                         the species, A, BC, ABC, AB, and C  
                         (max. of 10).  
                         Setting any of these to zero gives the  
                         default value of 1 state of degeneracy 1.  
                         The ground state is always zero energy. If  
                         NEL(I) = 0 for all I, then skip to line 29.

Loop over lines 27 and 28 for every nonzero element of NEL.

27      10I5            NDGEN(J=1,NEL(I)) - degeneracies of the electronic  
                         states of species I.

28      8F10.4           EL(J=1,NEL(I)) - electronic energies of the states of  
                         species I in kcal/mol.

29      I5               NESPEC - number of special energies at which tunneling  
                         calculations will be performed (max. of 40).  
                         If NESPEC is zero, the tunneling calculations  
                         are performed on the Kronrod grid.

If NESPEC = 0, then skip line 30.

30      4F20.10           ESPEC(I=1,NESPEC) - special energies (total energies  
                         in kcal/mol) at which the tunneling  
                         calculations are performed.

\* If a reaction path is being read in from a restart file, these variables must have identical values to those used in the run which created the reaction path; otherwise an error message will be issued and the program will be stopped.

\*\* If a reaction path is being read in from a restart file, these variables should have identical values to those used in the run which created the reaction path, but no check is made for consistency.

## 9. FILE USAGE

LFN denotes the logical file number, and all files are referred to as abc.LFN, e.g., abc.5 is the input file opened by the program. All input files must be assigned to the logical file name externally. For example, in Unix use a statement like

```
ln clhbrtr1.1 abc.1
```

LFN	Usage
1	Reaction path information.
5	Input data.
6	Output.
8	Potential data. This file is not always needed; it depends upon the user-supplied potential subroutine. However, because some PREPEF routines will need to read data, we have reserved LFN 8 for this purpose.
11	Restart information for 1D LAT calculation. This file needs to be assigned only if IGS(1)=3 or 4, IGS(2)=1 or 3, and IGS(7)=2.
12	Restart information for 3D LAT calculation. This file needs to be assigned only if IGS(1)=3 or 4, IGS(2)=2 or 3, and IGS(7)=2.
14	WKB energy levels for the stretching vibration. This file needs to be assigned only if IGS(1)=3 or 4 and IGS(5)= 4 or 5.
20	Output a special table of ratios of approximate to accurate rate constants. This file needs to be assigned only if LEQ is true and IGS(1)=4.
21	Output a special table of bottleneck properties. This table is created only if IGS(8)=2 or 4.
22	Output a special table of selected rate constants. This table is created only if IGS(8)=3 or 4.

10. COMPUTERS ON WHICH THE CODE HAS BEEN TESTED

ABCRATE is a portable program which has been tested on both supercomputers and workstations. The computers and the operating systems on which ABCRATE has been tested are listed in Table 1. The code is in FORTRAN and executes in double precision on all the workstations on which the program was tested. On the Cray supercomputer single precision corresponds to 64-bit words. Therefore, the REAL floating point variables in the program are interpreted as REAL\*8 on the Cray computers in single precision and so on this machine the double precision typing is disabled by the `-dp` qualifier with the `cft77` command. The compiler and loader commands used for testing the program are listed in Tables 2 and 3, respectively.

Note: It has been our experience that the compiler on some SGI systems does not always produce correct code. On the SGI system used in the test runs for this version of the code, the subroutine `geom.f` does not compile correctly under optimization. The result of this compiler bug generally causes the program to stop execution early rather than producing erroneous results.

Table 1. Operating systems on the various machines on which ABCRATE has been tested.

Version	Machine	Operating system
10.0	Cray C90	UNICOS 8.0.3
	IBM RS/6000 550	AIX 4.1
	IRIS Power Challenge L (R8000)	IRIX 6.2

Table 2. Compiler commands used on the various machines on which ABCRATE has been tested.

Machine	Compiler commands
Cray C90	<code>cft77 -dp -i64 -astatic</code>
IBM RS/6000 550	<code>xlf -c -O3 -qstrict -qhot -qsave</code>
IRIS Power Challenge L	<code>f77 -c -64 -mips4 -O3 -static</code>

Table 3. Loader commands used on the various machines on which ABCRATE has been tested.

---

Machine	Loader commands
Cray C90	segldr -o
IBM RS/6000 550	xlfr -o
IRIS Power Challenge L	f77 -64 -o

---

11. TEST RUNS

This section contains a description of the test runs in the distribution package. Section 11-A contains a table of the potential energy functions, with corresponding references, that are used for the test runs. Section 11-B contains a description of each test run, and it lists the input and output files and the name of the C Shell script needed to run each test run.

11-A. Table of the potential energy function routines for the test runs and corresponding references for the potential energy functions.

```

*****
Test Run(s)      Potential Energy      Reference
                  Surface Source Code
*****

```

dh2tr1 dh2tr2	h3dmbe.f	A. J. C. Varandas, F. B. Brown, C. A. Mead, D. G. Truhlar, and N. C. Blais J. Chem. Phys. 86, 6258 (1987)
oh2tr1	oh2jws.f	G. C. Schatz J. Chem. Phys. 83, 5677 (1985)
clhcltr1 clhcltr2	clhclbcmr.f	D. K. Bondi, J. N. L. Connor, J. Manz, and J. Romelt J. Mol. Phys. 50, 467 (1983)
clhbrtr1	clhbrblm.f	V. K. Babamov, V. Lopez, and R. A. Marcus J. Chem. Phys. 78, 5621 (1983)
fh2tr1	fh2m5.f	J. T. Muckerman Theor. Chem. Advan. Perspectives 67, 1 (1981)
dfhtr1 hfhtr1	fh2sk.f	G. C. Schatz and A. Kuppermann J. Chem. Phys. 72, 2737 (1980)

```

*****

```

## 11-B. Description of the test runs

This section contains a description of the test runs in the ABCRATE test suite. Several options were used for each test run to illustrate the capabilities of the program.

## 11-B-1. Test run DH2tr1



For this test run all the normal modes and generalized normal modes are treated anharmonically. The symmetric stretch is treated by the WKB approximation [IGS(5)=4] and the vibrational bends are treated by the perturbation-variation method [IGS(6)=3]. For the bending potential the harmonic and quartic force constants are fit to the potential at the angles 5 and 20 degrees. The minimum energy path is followed with a step size of 1.D-4 bohr and a save grid step of 1.D-3. Both 1D and 3D rate constants are calculated. The electronic degeneracies and energies of low-lying electronic states are also included. Transmission coefficients are determined by both the small- and the large-curvature methods; these include the SCT, LCT, and LAT methods. ZCT and muOMT values are also determined. The free energy data is not printed at any temperature. The WKB eigenvalues and the 1D and 3D large-curvature tunneling probabilities are written to the files abc.14, abc.11, and abc.12, respectively.

Disk filename	ABCRATE filename	Description
Input file		
dh2tr1.dat	abc.5	input data file
Output files		
dh2tr1.out	abc.6	output file
dh2tr1.21	abc.21	bottleneck properties
dh2tr1.22	abc.22	selected forward reaction rates
dh2tr1.11	abc.11	1D large-curvature tunneling probabilities
dh2tr1.12	abc.12	3D large-curvature tunneling probabilities
dh2tr1.14	abc.14	WKB eigenvalues

## 11-B-2. Test run DH2tr2

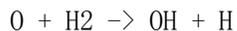
D + HH  $\rightarrow$  DH + H

DMBE potential energy surface

This test run uses the WKB eigenvalues and the large-curvature tunneling probabilities from the dh2tr1 test run. The anharmonicity options and the method used to fit the bending potential are also the same as the dh2tr1 test run. The rate constants are computed at the same temperatures as in the dh2tr1 test run plus at five additional temperatures, and the free energy data is printed at 4 temperatures. Both 1D and 3D rate constants are calculated, and electronic degeneracies and energies are included.

Disk filename	ABCRATE filename	Description
Input files		
dh2tr2.dat	abc.5	input data file
dh2tr1.11	abc.11	1D large-curvature tunneling probabilities
dh2tr1.12	abc.12	3D large-curvature tunneling probabilities
dh2tr1.14	abc.14	WKB eigenvalues
Output files		
dh2tr2.out	abc.6	output file
dh2tr2.21	abc.21	bottleneck properties
dh2tr2.22	abc.22	selected forward reaction rates

## 11-B-3. Test run OH2tr1

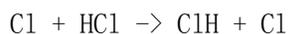


JWS potential energy surface

For this test run the symmetric stretch is treated by the WKB approximation and the vibrational bends are treated harmonically. For the bending potential the harmonic force constant is determined using the second derivative of the potential with respect to the bond angle [IBOPT=2]. The minimum energy path is followed with a step size of 1.D-4 bohr and a save grid step of 1.D-3. The state-selected option [LSTATE] is chosen for the first vibrational state; i.e., [NSTATE=1]. ZCT and SCT transmission coefficients are computed for both 1D and 3D rate constants. Electronic degeneracies and electronic energies are included in the calculation. Thus the O atom and the OHH transition state are treated as triplets.

Disk filename	ABCRATE filename	Description
Input file		
oh2tr1.dat	abc.5	input data file
Output files		
oh2tr1.out	abc.6	output file
oh2tr1.21	abc.21	bottleneck properties
oh2tr1.22	abc.22	selected forward reaction rates

## 11-B-4. Test run ClHCltr1



BCMR potential energy surface

For this test run all the normal modes and generalized normal modes are treated anharmonically. The symmetric stretch is treated by the WKB approximation [IGS(5)=3] and the vibrational bends are treated by the centrifugal oscillator method [IGS(6)=4]. For the bending potential the harmonic force constant is determined from the second derivative of the potential with respect to the bond angle and the quartic force constant is determined from the fourth derivative [IBOPT=1]. LSYM is set so that the symmetry of the reaction is used; the minimum energy path is followed in one direction only and the other direction is determined by reflection. The minimum energy path is followed with a step size of 1.D-4 bohr and a save grid step of 5.D-4. The reaction path information is written to the disk file abc.1 [IGS(4)=3] for use in test run clhcltr2. Both 1D and 3D rate constants are computed. The ZCT and SCT transmission coefficients are also computed. Factors for the electronic degeneracy and values of the electronic energy levels are included.

Disk filename	ABCRATE filename	Description
Input file		
clhcltr1.dat	abc.5	input data file
Output files		
clhcltr1.out	abc.6	output file
clhcltr1.21	abc.21	bottleneck properties
clhcltr1.22	abc.22	selected forward reaction rates
clhcltr1.1	abc.1	reaction path restart file

11-B-5. Test run ClHCltr2

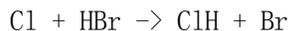


BCMR potential energy surface

This test run uses the reaction path information from clhcltr1. The mode anharmonicity and the method used to fit the bending the potential are the same as those used for the clhcltr1 test run. Both 1D and 3D rate constants are computed. The large curvature tunneling calculation [IGS(7)=4] is turned on for this test run so the ZCT, SCT, LCT, LAT, and muOMT transmission coefficients are determined. The rate constants in both the forward and the reverse direction are computed at the same temperatures as in clhcltr1 plus at five additional temperatures. As in test run clhcltr1, the electronic degeneracies and energies are included.

Disk filename	ABCRATE filename	Description
Input files		
clhcltr2.dat	abc.5	input data file
clhcltr1.1	abc.1	reaction path restart file
Output files		
clhcltr2.out	abc.6	output file
clhcltr2.21	abc.21	bottleneck properties
clhcltr2.22	abc.22	selected forward reaction rates

## 11-B-6. Test run ClHBrtr1



BLM potential energy surface

For this test run only the collinear (1D) reaction is studied and electronic degeneracies and energies are not included. The symmetric stretch is treated by the Morse I approximation [IGS(6)=2]. The minimum energy path is followed with a step size of 2.D-5 bohr and a save grid step of 1.D-2. The large-curvature tunneling calculation [IGS(7)=4] is turned on and inclusion of the excited product vibrational states [IOPT(1)] is also carried out. Accurate quantal rates [D. G. Truhlar and A. Kuppermann, Chem. Phys. Lett. 92, 574 (1982)] are read in [LEQ(1)] and a table of the ratios of the accurate and the VTST rates is produced.

Disk filename	ABCRATE filename	Description
Input file		
clhbrtr1.dat	abc.5	input data file
Output files		
clhbrtr1.out	abc.6	output file
clhbrtr1.20	abc.20	table of ratios of accurate and VTST rates
clhbrtr1.21	abc.21	bottleneck properties
clhbrtr1.22	abc.22	selected forward reaction rates

When this test run is executed on the Cray platform mentioned in section 10, all ICVT results at 200 K are incorrect. This gives rise to an error of several percent for the ICVT methods when compared to the results from other platforms. The results at all other temperatures are correct.

## 11-B-7. Test run FH2tr1

F + H2 -&gt; FH + H

M5 potential energy surface

For this test run only the conventional transition state theory rate constants are computed. The vibrational stretch is treated by the WKB approximation [IGS(5)=3] and the vibrational bends are treated semiclassically [IGS(6)=2]. For the bending potential the harmonic force constant is determined using the second derivative of the potential with respect to the bond angle [IBOPT=2]. Neither minimum energy path nor the vibrationally adiabatic potential curve is computed. The electronic degeneracies and energies are included for this calculation.

Disk filename	ABCRATE filename	Description
Input file		
fh2tr1.dat	abc.5	input data file
Output files		
fh2tr1.out	abc.6	output file
fh2tr1.21	abc.21	bottleneck properties

## 11-B-8. Test run DFHtr1

D + FH -&gt; DF + H

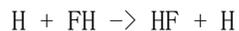
SK potential energy surface

For this test run all the normal modes and generalized normal modes are treated harmonically. The harmonic force constant for the bending potential is determined from the second derivative of the potential with respect to the bond angle [IBOPT=1]. This system has two saddle points [NSAD=2]. The minimum energy path is followed with a step size of 2.D-6 bohr and a save grid step of 2.D-3 bohr. Both 1D and 3D rate constants are computed. The ZCT and SCT transmission coefficients are computed. The electronic degeneracies and energies are included in this test run.

Disk filename	ABCRATE filename	Description
Input file		
dfhtr1.dat	abc.5	input data file
Output files		
dfhtr1.out	abc.6	output file
dfhtr1.21	abc.21	bottleneck properties
dfhtr1.22	abc.22	selected forward reaction rates

This test run is very sensitive to the convergence of the derivatives in the vicinity of the saddle point. RMS relative errors of a few percent are typical between different platforms. RMS relative errors are also generally larger for the 1-D reaction than for the 3-D reaction. This largest errors are usually given by the CUS methods.

## 11-B-9. Test run HFHtr1



SK potential energy surface

The test run uses WKB anharmonicity for the vibrational stretches and centrifugal oscillator energy levels for the vibrational bends. The stepsize used to follow the minimum energy path is 5.D-5 bohr and points on the MEP are saved every 1.D-3 bohr. The 1D rate constants are computed at 14 temperatures. Transmission coefficients are computed using the ZCT, SCT, LCT, LAT, and muOMT methods. No electronic factors are included in this calculation.

Disk filename	ABCRATE filename	Description
Input file		
dfhtr2.dat	abc.5	input data file
Output files		
dfhtr2.out	abc.6	output file
dfhtr2.21	abc.21	bottleneck properties
dfhtr2.22	abc.22	selected forward reaction rates

11-C. TIMING INFORMATION FOR TEST SUITE RUNS

The following table presents the timing information for all of the test runs on the three platforms on which the code has been tested. All times are given in seconds. Timings reflect the number of CPU seconds required to complete the test run.

test run	Platforms		
	IBM	SGI	Cray
ClHBr tr1	208.11	99.39	133.23
ClHC1 tr1	11542.40	3773.55	1634.65
ClHC1 tr2	24058.65	9315.71	4375.50
DFH tr1	95.44	48.37	36.69
DFH tr2	40110.68	13061.70	
DH2 tr1	7612.51	4674.48	3316.63
DH2 tr2	366.35	218.52	131.41
FH2 tr1	0.11	0.09	0.05
OH2 tr1	239.91	146.03	100.18
TOTAL	44123.48	18276.14	9728.34

Notes:

- 1) The IBM test platform was an IBM RS/6000-550.
- 2) The SGI test platform was an SGI Iris Power Challenge L with a single 90 MHz R8000 CPU.
- 3) The Cray test platform was a Cray YMP C90. All test runs were run on a single processor.

## 12. FILE ORGANIZATION FOR DISTRIBUTED VERSION

ABCRATE requires two subroutines, `dattim.f` which returns the current date and time and `porcpu.f` which returns the process CPU time in seconds. These files are supplied for Cray, SGI, and IBM platforms. It may be necessary to create a new version of these subroutines for other platforms. The user must, in general, also supply a potential energy surface routine which is described in detail in Section 7.

### 12-A. Files in the distribution package

ABCRATE-version 10.0 is made up of the following files:

<code>makeabc</code>	Makefile for the source code. Note: this makefile requires certain environment variables to be set before it can be properly used. The script <code>abc.c</code> sets these variables and executes the makefile.
<code>*.f</code>	FORTTRAN source files used by the code. There are 148 source files in the <code>src</code> directory, 8 source files in the <code>poten</code> directory, and 1 source file in the <code>script</code> directory. These files are listed individually in the next section.
<code>abc.inc</code>	Include file which contains a single parameter statement controlling the maximum number of points saved along the minimum energy path.
<code>dattim.&lt;mach&gt;</code>	This is the date/time subprogram for the Cray ( <code>mach=cray</code> ), IBM ( <code>mach=ibm</code> ), or SGI ( <code>mach=sgi</code> ) computers. It returns the current date and time in an 80 character string.
<code>porcpu.&lt;mach&gt;</code>	This subprogram determines the CPU time for the Cray ( <code>mach=cray</code> ), IBM ( <code>mach=ibm</code> ), or SGI ( <code>mach=sgi</code> ) computers. It returns the CPU time used by the process in seconds.
<code>abc.f</code>	Potential interface file. This file is puts the information calculated by the potential codes found in the <code>poten</code> directory into the form described in Section 7.

cab.f            Potential interface file, like abc.f, but this file also changes the coordinates and output so that a potential energy code written assuming an ABC atomic arrangement will return the potential energy and derivatives as though the arrangement were CAB.

abcrate.doc    The documentation for ABCRATE. This is the ASCII file you are currently reading. It serves as hardcopy and online documentation.

abc.c           This is a C shell script which sets compiler and machine type environment variables and calls the makefile "makeabc" to compile the source code in the src directory.

- comp.c This is a C shell script which attempts to determine the architecture on which it is being executed and, if it is able to do so, returns compiler commands, loader commands, or machine types. This script is used by most other scripts in the ABCRATE distribution. It will need to be modified for unsupported machines.
- compare.c This is a C shell script which compiles the comparison code compare.f and compares all available testrun output (<testrun>.22) to the output found in the testo directory. The comparison files are placed in the cmp directory in the testrun directory.
- compare.f This is a FORTRAN program designed to read the rate constant summary files (<testrun>.22) and report on their differences by returning a RMS relative deviation. This file is compiled and executed by the script compare.c.
- runtest.c This is a C shell script which may be used to execute all test runs or a single specific test run. Executing the script with no arguments gives a list of options. Executing the script with the argument "all" runs all test runs. Executing the script with a number from 1 to 8 runs a specific test run.
- <testrun>.20 This file is used to compare the rate constants computed by ABCRATE to accurate rate constants which are given in the input file for a given <testrun>. Currently only the ClHBr test run uses this option.
- <testrun>.21 This file gives a summary of the bottleneck properties for a given <testrun>.
- <testrun>.22 This file gives a summary of the rate constants for a given <testrun>.
- <testrun>.c This file is a job control file for a given <testrun>. It is a C shell script designed to copy the necessary files for each test run to their appropriate names for use by ABCRATE and to rename the output to the form

<testrun>.<ext>, where <ext> denotes a specific type of output file.

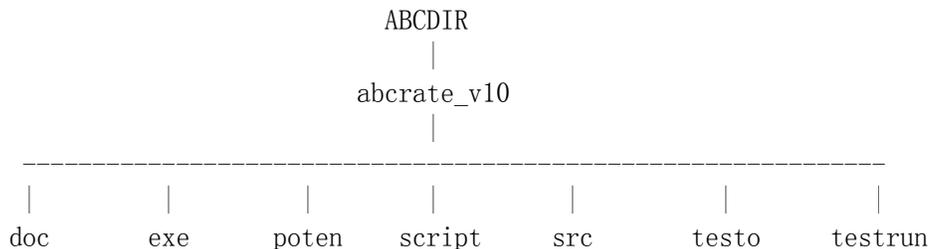
<testrun>.dat This file contains the input data for a given <testrun> as described in Section 8 of this manual. It is copied to abc.5 for use by ABCRATE.

12-B. Installing and running the test suite

The procedure for installing, testing, and using ABCRATE in a Unix environment is described in this section. If the program has been received on a tar tape, the recommended directory structure will be created automatically where the tape is untarred and the files will be placed in the correct places as described below. If the program has not been received on tar tape, the directory structure and appropriate file placement described below is highly recommended. The C shell scripts provided in this distribution package assumes the recommended structure.

Step 1: Before starting ensure that all file names are in lower case. For illustrative purposes we use upper case for example directory names, e.g., ABCDIR, where ABCDIR is the directory where ABCRATE will be installed.

Step 2: Create the following directory structure. If the program was received on a tar tape make sure that the following directory structure was created when the tape was untarred.



Step 3: Place all the files as follows or verify that they were placed correctly.

In the doc subdirectory  
 abcrate.doc

In the poten subdirectory

abc.f	cab.f	clhbrblm.f	clhclbcmr.f
fh2m5.f	fh2sk.f	h3dmbe.f	oh2jws.f

In the script subdirectory

abc.c	comp.c	compare.c	compare.f
runtest.c			

In the src subdirectory

abc.inc	abcmake	adiab.f	aitken.f
aitkf2.f	aitkn2.f	aitknf.f	alfct.f
bend.f	benmin.f	benpot.f	boltz.f
brnuli.f	chkrst.f	class.f	cobend.f
cobint.f	cobprt.f	cobtp.f	cobvex.f
cubic.f	curve.f	curvex.f	d2dx2.f
d2vdu2.f	datain.f	dattim.cray	dattim.ibm
dattim.sgi	dbwl.f	dbwlrt.f	deriv.f
ders.f	dgama.f	eact.f	ebend.f
ellip.f	elpart.f	estr.f	extras.f
extrem.f	frmt1.f	frmt2.f	fumcp.f
gaussq.f	gbslve.f	gbtql2.f	geom.f
grad.f	gtst.f	headr.f	hqsc.f

init.f	interp.f	kaptyp.f	kapva.f
kg1.f	lag.f	lagth.f	lalfmn.f
lalfor.f	lalfst.f	lbetas.f	lbuang.f
lcsa.f	lcsa2.f	lcsa3.f	liadib.f
lin2.f	locs.f	lpvag.f	param.f
pfcn.f	pfcnb.f	pfcnr.f	pfcnst.f
phid.f	phsint.f	plag.f	pmuomt.f
pnorm.f	porecpu.cray	porecpu.ibm	porecpu.sgi
pread.f	pqrt.f	psag.f	pwrite.f
qmpart.f	quadft.f	reflec.f	root2d.f
rpath.f	rphsum.f	rpr2.f	rpread.f
rpseg.f	rpseg2.f	rpstor.f	rpupdt.f
rpw2.f	rpwell.f	rpwrit.f	rsp2.f
saddle.f	sagarg.f	sagcmp.f	shift.f
spl1b1.f	spl1b2.f	spl1d1.f	st2drv.f
step1.f	step2.f	store.f	summry.f
surfit.f	tabl20.f	tabl21.f	tabl22.f
theta2.f	theta3.f	thrcor.f	timing.f
titles.f	tout.f	tp.f	tp2.f
tpsrch.f	trans.f	utp.f	vbend.f
vextr.f	vibtap.f	vibtau.f	vmin.f
vphi.f	vsplin.f	vspln2.f	vtstm.f
well.f	wkb.f	wkbpot.f	wkbset.f
wkbt.p.f	wkbt.p2.f	writef.f	writeg.f

In the testo subdirectory

clhbrtr1.20	clhbrtr1.21	clhbrtr1.22	clhbrtr1.out
clheltr1.21	clheltr1.22	clheltr1.out	clheltr2.21
clheltr2.22	clheltr2.out	dfhtr1.21	dfhtr1.22
dfhtr1.out	dh2tr1.21	dh2tr1.22	dh2tr1.out
dh2tr2.21	dh2tr2.22	dh2tr2.out	fh2tr1.out
hfhtr1.21	hfhtr1.22	hfhtr1.out	oh2tr1.21
oh2tr1.22	oh2tr1.out		

In the testrun subdirectory

clhbrtr1.c	clhbrtr1.dat	clheltr1.c	clheltr1.dat
clheltr2.c	clheltr2.dat	dfhtr1.c	dfhtr1.dat
dh2tr1.c	dh2tr1.dat	dh2tr2.c	dh2tr2.dat
fh2tr1.c	fh2tr1.dat	hfhtr1.c	hfhtr1.dat
oh2tr1.c	oh2tr1.dat		

Step 4: Change directories to ABCDIR/script. Execute the C shell script

abc.c. This script will obtain information about the machine architecture and execute the makefile script to compile the FORTRAN files in ABCDIR/src. If your machine type is not one of the three architectures currently supported, you will have to add appropriate compiler and loader commands to the comp.c script in the ABCDIR/script directory. Note: the script abc.c MUST be executed from the ABCDIR/script directory.

- Step 5: Run the test runs in the test suite. This may be done in one of several ways. Change directories to the script directory and execute the script `runtest.c`. If the script is executed with no arguments, it will print a list of options. If the script is executed with the argument "all", then all test runs will be executed. If the script is executed with a number from 1 to 8, a specific test run will be executed. Execute the script with no arguments to get a list of input numbers which correspond to a specific test run. Alternately, the user may execute the test runs from the `testrun` directory by changing directories to the `testrun` directory and executing a specific test run script `<testrun>.c`. Note that the `clhcltr1` and `dh2tr1` test runs must be run prior to running the `clhcltr2` and `dh2tr2` test runs, respectively. As in Step 4, the scripts must be executed from the appropriate directory, and the script `comp.c` must be edited on unsupported systems.
- Step 6: Compare the `testrun` output to the distributed output in the `testo` directory. To do this, change directories to the script directory and execute the script `compare.c`. As in Steps 4 and 5, you must edit the script `comp.c` if you are using an unsupported architecture.
- Step 7: Verify that the `testruns` are giving correct results by checking the output files (`<testrun>.cmp`) found in the `testrun/cmp` directory. RMS relative deviations should be small ( $< 1.e-4$ ) or zero, with the exception of test run `dfhtr1` (see notes in Section 11-B-8). (If you are running on a Cray platform see notes in Section 11-B-6 as well.) The test run `fh2tr1` must be checked by hand (just check the TST rates found at the end of the main output, `fh2tr1.out`). Significant deviations from the output found in the `testo` directory are an indication of a likely error.

Note: The C-shell script `abc.c` will accept the command line argument `clean` which will delete all object files in the `src` and `poten` directories and delete the symbolic links in the `src` directory. This feature is useful to return the source code to its distributed form.