RPMDrate: bimolecular chemical reaction rates from ring polymer molecular dynamics

Yu. V. Suleimanov[a,b,*], J. W. Allen[a], W. H. Green[a]

[a] *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139*
[b] *Department of Mechanical and Aerospace Engineering, Combustion Energy Frontier Research Center, Princeton University, Princeton, New Jersey 08544*

Abstract

We present RPMDrate, a computer program for the calculation of gas phase bimolecular reaction rate coefficients using the ring polymer molecular dynamics (RPMD) method. The RPMD rate coefficient is calculated using the Bennett-Chandler method as a product of a static (centroid density quantum transition state theory (QTST) rate) and a dynamic (ring polymer transmission coefficient) factor. The computational procedure is general and can be used to treat bimolecular polyatomic reactions of any complexity in their full dimensionality. The program has been tested for the H + H_2, H + CH_4, OH + CH_4 and H + C_2H_6 reactions.

Keywords: ring polymer molecular dynamics, chemical reaction rates, kinetics, reaction coordinate, quantum mechanical effects, tunneling, zero point energy

PROGRAM SUMMARY

Program Title: RPMDrate
Catalogue identifier:
Licensing provisions: MIT license (http://www.opensource.org/licenses/mit-license)
Programming language: Fortran 90/95, Python (version 2.6.x or later, including any version of Python 3, is recommended)
Computer: Not computer specific
Operating system: Any for which Python, Fortran 90/95 compiler and the required external routines are available.
Memory required to execute with typical data: 256 Mb
Number of processors used: 1, although the program can efficiently utilise 4096+ processors, depending on problem and available computer. At low temperatures, 110 processors is reasonable for a typical umbrella integration run with an analytic potential energy function and gradients on the latest x86-64 machines.
No. of bytes in distributed program:

- Fortran files: 94 508
- Python scripts: 154 893
- RPMDrate manual: 508 350
- Data and input for test runs: 3 426 141
- Total (including installation files) :: 4 190 753

Distribution format: gzipped tar file
Supplementary material:
The distribution contains example data and a detailed manual describing the use of RPMDrate.
Keywords: ring polymer molecular dynamics, chemical reaction rates, kinetics, reaction coordinate, quantum effects, tunneling, zero point energy.
Classification: 16.12 Chemical Kinetics.
External routines:

- NumPy (http://numpy.scipy.org, version 1.5.0 or later is recommended).
- FFTW3 (http://www.fftw.org, version 3.3 or later is recommended).

Nature of problem:

- The RPMDrate program calculates thermal bimolecular rate coefficients of thermally activated atom-diatom and more complex bimolecular chemical reactions in the gas phase.

Solution method:

- The RPMD rate is calculated using the Bennett-Chandler factorization as a product of a static (centroid density quantum transition state theory (QTST) rate) and a dynamic (transmission coefficient) factor. A key feature of this procedure is that it does not require that one calculate the absolute quantum mechanical partition function of the reactants or the transition state. The centroid density QTST rate is calculated from the potential of mean force along the reaction coordinate using umbrella integration. The reaction coordinate is taken to be an interpolating function that connects two dividing surfaces: one located in the asymptotic reactant valley and the other located in the transition state region. The Hessian of the collective reaction coordinate is obtained analytically. The transmission coefficient is calculated from the RPMD simulations with the hard constraint along the reaction coordinate.

Restrictions: The applicability of RPMDrate is restricted to global potential energy surfaces with gradients. In the current release, they should be provided by Python callable objects.
Unusual features: Simple and user-friendly input system provided by Python syntax.
Additional comments: Test calculations for the H + H_2 reactions were performed using the Boothroyd-Keogh-Martin-Peterson-2 (BKMP2) potential energy surface (PES) [1]. PESs for the H + CH_4, OH + CH_4 and H + C_2H_6 reactions are taken from the online POTLIB library [2]. PESs are included within the distribution package as Fortran subroutines. Implementations of the colored-noise, generalized Langevin equation (GLE) thermostats [3-5] have been included in the current release.

*Corresponding author.
E-mail address: ysuleyma@princeton.edu

Running time: Highly dependent on the input parameters. The running time of RPMDrate depends mainly on the complexity of the potential energy surface and number of ring polymer beads. For the H + H_2, H + CH_4, and OH + CH_4 test calculations given (with 128 ring polymer beads and analytic gradients), the running time is approximately 1800, 3600 and 4000 processor hours, respectively, on the Silicon Mechanics nServ A413 servers.

[1] A. I. Boothroyd, W. J. Keogh, P. G. Martin, M. R. Peterson, J. Chem. Phys. 104 (1996) 7139.

[2] R. J. Duchovic, Y. L. Volobuev, G. C. Lynch, A. W. Jasper, D. G. Truhlar, T. C. Allison, A. F. Wagner, B. C. Garrett, J. Espinosa-García, J. C. Corchado, POTLIB, http://comp.chem.umn.edu/potlib.

[3] M. Ceriotti, G. Bussi, M. Parrinello, J. Chem. Theory Comput. 6 (2010) 1170.

[4] M. Ceriotti, G. Bussi, M. Parrinello, Phys. Rev. Lett. 103 (2009) 030603.

[5] M. Ceriotti, G. Bussi, M. Parrinello, Phys. Rev. Lett. 102 (2009) 020601.

1. Introduction

Ring polymer molecular dynamics (RPMD) is a recently developed approximate quantum mechanical approach [1, 2] which provides a simple classical-like approximation to quantum mechanical Kubo-transformed dynamical (real-time) correlation functions. The method exploits the isomorphism between the statistical properties of the quantum system and those of a classical ring polymer [3] and uses the classical molecular dynamics of the ring polymer to approximate real-time quantum evolution. The resulting RPMD correlation functions are exact in several important limits, including the short-time limit ($t \to 0$) and the classical high-temperature limit ($T \to \infty$). In the special case where one or other of the correlated operators is a linear function of the coordinates or momenta, RPMD correlation functions give the exact result for a harmonic potential [1]. They also have the same detailed balance and time-reversal symmetry properties as the exact quantum mechanical correlation functions that they approximate [1]. RPMD is also exact for a parabolic barrier bilinearly coupled to a bath of harmonic oscillators at all temperatures for which a rate coefficient of this process can be defined [4].

Since its introduction, RPMD has been applied to a number of different problems [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]. Application to condensed phase systems has demonstrated that RPMD method is particularly well suited to the calculation of diffusion coefficients [5, 6, 7, 8, 9, 16, 22, 29] and chemical reaction rates in complex systems [4, 8, 10, 11, 24]. In most cases, the results suggest that RPMD is able to capture the dominant quantum mechanical (zero point energy and tunneling) effects in a wide variety of different situations. Furthermore, the RPMD approximation is simple enough to allow one to perform calculations on systems with a wide range of different sizes. The RPMD simulation typically requires only n times more computational efforts than the corresponding classical simulations (n is the number of ring polymer beads) and is applicable to systems involving up to thousands of atoms [20, 24, 26].

Application of RPMD to the study of bimolecular gas-phase reactions [18, 25, 27] is one of the most recent developments. It exhibited some very desirable features:

- The RPMD rate theory has a well-defined short-time limit that provides an upper bound on the RPMD rate. When the dividing surface is defined in terms of the centroid of the ring polymer, the short-time limit of the RPMD rate coincides with a well-known (centroid density) [30, 31, 32] version of quantum transition state theory (QTST) [10].

- The RPMD rate coefficient is rigorously independent of the choice of the transition state dividing surface that is used to compute it [10]. This is a highly desirable feature of the theory for applications to multidimensional reactions for which the optimum dividing surface can be very difficult to determine.

- The RPMD rate becomes exact in the high temperature limit, where the ring polymer collapses to a single bead [4].

- The RPMD rate captures almost perfectly the zero point energy effect [27]. The RPMD rate coefficient also captures the tunneling effect and is usually within a factor of 2-3 of accurate QM results at very low temperatures in the deep quantum tunneling regime [18, 25].

The methodology proposed for bimolecular polyatomic chemical reactions in the gas phase [25] is based on the Bennett-Chandler factorization [33, 34]. A key feature of this methodology is that it does not require that one calculate the absolute value of either the reactant or the transition state partition function. This computational procedure is rather universal and it can be applied in a straightforward way to any complex reactive bimolecular system in its full dimensionality. In view of these features and given the simplicity of the method, it seems that RPMD should provide an ideal tool for studying complex polyatomic chemical reactions for which exact quantum mechanical calculations are infeasible. Construction of global potential energy surfaces for polyatomic reactive systems has made strong progress over the last decade [35, 36] suggesting a promising future of full-dimensional dynamical techniques such as RPMD.

In this paper, we present the computer program RPMDRATE which calculates the RPMD bimolecular rate coefficients for polyatomic chemical reactions in the gas phase. We begin in Sec. II by reviewing the main working expressions of the RPMD method. In Sec. III we describe the computational procedures implemented in RPMDRATE. Sec. IV presents the results of example calculations: rate coefficients for several benchmark reactive systems (H + H_2, H + CH_4, OH + CH_4 and H + C_2H_6). We present our final remarks on the current release in Sec. V.

2. General methodology of ring polymer rate theory

The bimolecular rate is calculated by combining RPMD rate theory [4, 10] with the Bennett-Chandler method [33, 34]. This approach has been discussed extensively in Refs. [18, 25, 27] so only the final working expressions are given here.

Consider a general bimolecular chemical reaction between reactants A and B in the gas phase.

$$\mathrm{A} + \mathrm{B} \rightarrow \text{Transition} \quad \text{State} \rightarrow \text{Products}. \tag{1}$$

The ring polymer Hamiltonian is written in the atomic Cartesian coordinates as

$$\begin{aligned} H_n(\mathbf{p},\mathbf{q}) &= H_n^0(\mathbf{p}_\mathrm{A},\mathbf{q}_\mathrm{A}) + H_n^0(\mathbf{p}_\mathrm{B},\mathbf{q}_\mathrm{B}) \\ &\quad + \sum_{j=1}^{n} V(\mathbf{q}_{1_\mathrm{A}}^{(j)},\ldots,\mathbf{q}_{N_\mathrm{A}}^{(j)};\mathbf{q}_{1_\mathrm{B}}^{(j)},\ldots,\mathbf{q}_{N_\mathrm{B}}^{(j)}), \end{aligned} \tag{2}$$

where

$$H_n^0(\mathbf{p}_\mathrm{X},\mathbf{q}_\mathrm{X}) = \sum_{i_\mathrm{X}=1}^{N_\mathrm{X}} \sum_{j=1}^{n} \left(\frac{\left|\mathbf{p}_{i_\mathrm{X}}^{(j)}\right|^2}{2m_{i_\mathrm{X}}} + \frac{1}{2} m_{i_\mathrm{X}} \omega_n^2 \left|\mathbf{q}_{i_\mathrm{X}}^{(j)} - \mathbf{q}_{i_\mathrm{X}}^{(j-1)}\right|^2 \right), \tag{3}$$

with $\omega_n = 1/(\beta_n\hbar)$ and $\mathbf{q}_{i_\mathrm{X}}^{(0)} \equiv \mathbf{q}_{i_\mathrm{X}}^{(n)}$. Here $\beta_n = \beta/n$ with $\beta = 1/(k_\mathrm{B}T)$ is the appropriate reciprocal temperature of the ring polymer system, n is the number of beads in the necklace for each atom, N_X is the number of atoms in the reactant X (X = A, B), $\mathbf{p}_{i_\mathrm{X}}^{(j)}$ and $\mathbf{q}_{i_\mathrm{X}}^{(j)}$ are the momentum and position vectors of the jth bead in the ring-polymer necklace of atom i_X belonging to the reactant X and m_{i_X} is its atomic mass.

The method begins by introducing two dividing surfaces defined in terms of the ring polymer centroid variables [10]. The first dividing surface is located in the asymptotic reactant valley

$$s_0(\overline{\mathbf{q}}) = R_\infty - \left|\overline{\mathbf{R}}\right|, \tag{4}$$

where $\overline{\mathbf{R}}$ is the centroid of the vector that connects the centers of mass of the reactant molecules A and B,

$$\overline{\mathbf{R}} = \frac{1}{m_\mathrm{B}} \sum_{i_\mathrm{B}=1}^{N_\mathrm{B}} m_{i_\mathrm{B}} \overline{\mathbf{q}}_{i_\mathrm{B}} - \frac{1}{m_\mathrm{A}} \sum_{i_\mathrm{A}=1}^{N_\mathrm{A}} m_{i_\mathrm{A}} \overline{\mathbf{q}}_{i_\mathrm{A}}, \tag{5}$$

with $m_\mathrm{X} = \sum_{i_\mathrm{X}=1}^{N_\mathrm{X}} m_{i_\mathrm{X}}$ and

$$\overline{\mathbf{q}}_{i_\mathrm{X}} = \sum_{j=1}^{n} \mathbf{q}_{i_\mathrm{X}}^{(j)}, \tag{6}$$

and R_∞ is an adjustable parameter that is chosen to be sufficiently large as to make the interaction between the reactants negligible.

The second dividing surface is located in the transition state region and is defined in terms of the bond-breaking and bond-forming distances

$$s_1(\overline{\mathbf{q}}) = \min\left\{s_{1,1}(\overline{\mathbf{q}}),\ldots,s_{1,N_\mathrm{bonds}}(\overline{\mathbf{q}})\right\} \tag{7}$$

where N_bonds is the number relevant combinations of breaking and forming bonds and

$$s_{1,k}(\overline{\mathbf{q}}) = \max\left\{s_{1,k}^{(1)}(\overline{\mathbf{q}}),\ldots,s_{1,k}^{(N_\mathrm{channel})}(\overline{\mathbf{q}})\right\}, \tag{8}$$

where N_channel is the number of equivalent product channels and

$$s_{1,k}^{(l)}(\overline{\mathbf{q}}) = \left(\left|\overline{\mathbf{q}}_{12,k}^{(l)}\right| - q_{12,k}^{\ddagger(l)}\right) - \left(\left|\overline{\mathbf{q}}_{23,k}^{(l)}\right| - q_{23,k}^{\ddagger(l)}\right), \tag{9}$$

is the dividing surface for each individual combination. Here one bond breaks between atoms 1 and 2 and one bond forms between atoms 2 and 3. $\overline{\mathbf{q}}_{ij}$ denotes the vector that connects the centroids of atoms i and j and $q_{ij}^{\ddagger}$ is the corresponding interatomic distance at the transition state saddle point. Clearly, such generalized expression for the transition dividing surface won't be optimal in many complex polyatomic systems. However, the RPMD results are guaranteed to be the same for *any* choice of dividing surface [10], whereas the transition state theory results would change if the dividing surface were chosen differently.

The reaction coordinate ξ is taken to be an interpolating function that connects these dividing surfaces,

$$\xi(\overline{\mathbf{q}}) = \frac{s_0(\overline{\mathbf{q}})}{s_0(\overline{\mathbf{q}}) - s_1(\overline{\mathbf{q}})}, \tag{10}$$

such that $\xi \rightarrow 0$ as $s_0 \rightarrow 0$ and $\xi \rightarrow 1$ as $s_1 \rightarrow 0$.

The RPMD rate coefficient is then written in the Bennett-Chandler form [33, 34]

$$k_\mathrm{RPMD}(T) = k_\mathrm{QTST}(T;\xi^\ddagger)\kappa(t \rightarrow t_p;\xi^\ddagger). \tag{11}$$

Here the first factor, $k_\mathrm{QTST}(T;\xi^\ddagger)$, is the centroid-density quantum transition state theory (QTST) rate coefficient [30, 31, 32] evaluated at the transition state $\xi^\ddagger$ along the reaction coordinate $\xi(\overline{\mathbf{q}})$. It can be expressed in terms of the centroid potential of mean force (PMF) along the reaction coordinate, $W(\xi)$, as [18, 25]

$$k_\mathrm{QTST}(T;\xi^\ddagger) = 4\pi R_\infty^2 \left(\frac{m_\mathrm{A}+m_\mathrm{B}}{2\pi\beta m_\mathrm{A} m_\mathrm{B}}\right)^{1/2} e^{-\beta[W(\xi^\ddagger)-W(0)]}. \tag{12}$$

The $k_\mathrm{QTST}(T;\xi^\ddagger)$ in Eq. (11) is calculated by umbrella integration [37, 38, 39]. The PMF difference in Eq. (12) is calculated as

$$\begin{aligned} W(\xi^\ddagger) - W(0) &= \int_0^{\xi^\ddagger} \sum_{i=1}^{N_\mathrm{windows}} \left[\frac{N_i P_i(\xi)}{\sum_{j=1}^{N_\mathrm{windows}} N_j P_j(\xi)} \right. \\ &\quad \left. \left(\frac{1}{\beta} \frac{\xi - \bar{\xi}_i}{(\sigma_i)^2} - k_i(\xi - \xi_i) \right) \right] d\xi, \end{aligned} \tag{13}$$

with

$$P_i(\xi) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\xi - \bar{\xi}_i}{\sigma_i}\right)^2\right]. \tag{14}$$

Here N_windows is the number of biasing windows placed along the reaction coordinate with a specific value ξ_i assigned to each window, N_i is the total number of steps sampled for window i, $\bar{\xi}_i$ and σ_i^2 are the mean value and the variance calculated for ith window from the trajectory generated by the modified ring polymer Hamiltonian

$$\tilde{H}_n(\mathbf{p},\mathbf{q}) = H_n(\mathbf{p},\mathbf{q}) - \frac{1}{\beta_n} \ln f_s(\overline{\mathbf{q}}) + \frac{1}{2} k_i (\xi(\overline{\mathbf{q}}) - \xi_i)^2, \tag{15}$$

where k_i is the force constant which defines the strength of the bias in window i and

$$f_s(\overline{\mathbf{q}}) = \left\{ \sum_{i=1}^{N_\mathrm{A}+N_\mathrm{B}} \frac{1}{2\pi\beta m_i} \left| \frac{\partial\xi(\overline{\mathbf{q}})}{\partial \overline{\mathbf{q}}_i} \right|^2 \right\}^{1/2}. \tag{16}$$

The second factor in Eq. (11), $\kappa(t \to t_p; \xi^\ddagger)$, is the long-time limit of a time-dependent ring polymer transmission coefficient. This is a dynamical correction to centroid density QTST that accounts for recrossing of the dividing surface at $t \to t_p$, where t_p is a "plateau" time [34], and ensures that the resulting RPMD rate coefficient $k_{\mathrm{RPMD}}(T)$ will be independent of this choice of the dividing surface [10]. It is given by [25]

$$\kappa(t \to t_p; \xi^\ddagger) = \lim_{t \to t_p} \frac{\left\langle f_s(\overline{\mathbf{q}}_0)^{-1} v_s(\mathbf{p}_0, \mathbf{q}_0) h[s_{\xi^\ddagger}(\overline{\mathbf{q}}_t)] \right\rangle_{s_{\xi^\ddagger}}}{\left\langle f_s(\overline{\mathbf{q}}_0)^{-1} v_s(\mathbf{p}_0, \mathbf{q}_0) h[v_s(\overline{\mathbf{p}}_0, \overline{\mathbf{q}}_0)] \right\rangle_{s_{\xi^\ddagger}}}, \quad (17)$$

where

$$v_s(\mathbf{p}_0, \mathbf{q}_0) = \sum_{i=1}^{N_\mathrm{A}+N_\mathrm{B}} \sum_{j=1}^{n} \frac{\partial s_{\xi^\ddagger}}{\partial \mathbf{q}_i^{(j)}} \frac{\mathbf{p}_i^{(j)}}{m_i} \Bigg|_{t=0} \quad (18)$$

is the initial velocity along the reaction coordinate and $h[\ldots]$ is a Heaviside step function which counts ring polymer trajectories $(\overline{\mathbf{p}}_t, \overline{\mathbf{q}}_t)$ that are on the product side at time t of the dividing surface

$$s_{\xi^\ddagger}(\overline{\mathbf{q}}) = \xi^\ddagger s_1(\overline{\mathbf{q}}) + (1 - \xi^\ddagger) s_0(\overline{\mathbf{q}}). \quad (19)$$

The angular brackets in Eq. (17) denote a canonical average and the superscript on the brackets indicates that the average is taken over the ensemble constrained at $t = 0$ ($s_{\xi^\ddagger}(\overline{\mathbf{q}}_0) = 0$). The factor of $f_s(\overline{\mathbf{q}}_0)^{-1}$ in Eq. (19) is a metric tensor correction for the effect of the hard constraint [40].

Note that the same strategy can be used to calculate purely classical reaction rate coefficients simply by replacing the ring polymer with a single bead ($n = 1$).

3. Program structure

The work carried out by the RPMDrate program consists of six subsequent steps:

1. data input;
2. generating the initial configurations for umbrella integration;
3. biased sampling along the reaction coordinate;
4. calculation of the potential of mean force;
5. calculation of the transmission coefficient;
6. calculation of the final RPMD rate coefficient.

The program can be stopped after the completion of any of the calculation steps. Steps 3 and 5 are the most cpu time consuming. During these steps, the program creates a series of checkpoints which allow RPMDrate to be safely stopped at any point in its execution and then to be restarted later.

The RPMD thermal rate coefficients are calculated separately for each temperature. To execute a RPMDrate job, invoke the command

$$\$ \textit{ python rpmdrate.py input.py } \texttt{Temp Nbeads} -p \texttt{ Nprocs}, \quad (20)$$

where `Temp` is the temperature, `Nbeads` is the number of ring polymer beads, `Nprocs` is the number of requested processors. The job will run and the results will be saved to `Temp/Nbeads` in the same directory as the input file.

An example of RPMDrate input file for the H + H_2 reaction is given in Figure 1. The format of RPMDrate input files is based on Python syntax. In fact, RPMDrate input files are valid Python source code, and this is used to facilitate reading of the file. Each section is made up of one or more function calls, where parameters are specified as text strings, numbers, or objects. Text strings must be wrapped in either single or double quotes. The format of the input file is explained in more detail in the RPMDrate manual.

Though, the RPMD results are independent on the choice of the reaction coordinate $\xi^\ddagger$, it is advantageous to start calculations by obtaining the centroid-density QTST rate coefficient and locate the barrier of PMF along the reaction coordinate so as to to reduce number of recrossings in the subsequent calculations of the transmission coefficient [25].

In umbrella integration, the reaction coordinate is divided into a series of windows $\{\xi_i\}_{i=1}^{N_\mathrm{windows}}$. At the first step of umbrella integration, initial classical (number of beads set to 1) configurations are generated for each window from short biased trajectories. For each window i, the biased system is then equilibrated for $t_\mathrm{equilibration}$ and $\bar{\xi}_i$ and σ_i are accumulated along the RPMD thermostatted trajectory run for t_sampling. The trajectory is thermostatted using simple stochastic Andersen thermostatting scheme [41] or more elaborate colored noise, generalized Langevin equation (GLE) thermostats [42, 43, 44]. Both thermostats are implemented in the current release of the RPMDrate program. The whole procedure is repeated several times ($N_\mathrm{trajectory}$) so as to accumulate enough steps of sampling. The program design is such that the RPMD simulations for each window can be executed in parallel on separate processors and the calculations can be restarted from the last checkpoint (after each trajectory). At the last step of umbrella integration, the reaction coordinate is divided into N_bins bins that evenly span the whole range of ξ and are independent of the windows [37]. The integration in Eq. (14) is then performed using the trapezoidal rule between the bins.

The next step in the calculation is the transmission coefficient. Having obtained the optimal values of $\xi^\ddagger$ from the PMF profile for each temperature, it is evaluated from Eq. (17) by performing a constrained RPMD simulation in the presence of a thermostat (parent trajectory) to obtain an average of the bracketed quantity in the denominator and generate a series of independent configurations $\mathbf{q}_0$ with centroids on the transition state dividing surface $s_{\xi^\ddagger}(\overline{\mathbf{q}}) = 0$. The RATTLE algorithm [45] is introduced into the time integration to constrain the centroid $\overline{\mathbf{q}}_0$ to the dividing surface $s_{\xi^\ddagger}(\overline{\mathbf{q}}) = 0$. For each of these constrained configurations obtained from the parent trajectory with a sampling increment of $t_\mathrm{childsampling}$, a number (N_child) of momentum vectors $\mathbf{p}_0$ is sampled at random from the Maxwell distribution contained in $e^{-\beta_n H_n(\mathbf{p}_0, \mathbf{q}_0)}$ and the resulting recrossing (child) trajectories are evolved forward in time (t_child) without the thermostat or the dividing surface constraint to obtain $\mathbf{q}_t$. The bracketed quantity in the numerator is calculated by averaging over a large number ($N_\mathrm{totalchild}$) of these trajectories.

The centroid density QTST rate coefficient is obtained from the PMF profile using Eq. (12) and the final RPMD rate coefficient is calculated using the Bennett-Chandler factorization using Eq. (11).

The RPMDrate code performs all calculations using atomic Cartesian coordinates, thus imposing no restriction on the overall rotational or translational motion of the system [25]. The equations of motion are integrated using a symplectic integrator based on alternating free harmonic ring-polymer and external force steps with a time step dt [46]. The program uses a global representation of the Born-Oppenheimer potential energy surface of the system. This potential energy surface with gradients must be supplied in a special format described in the RPMDrate manual. The manual also provides additional information on the program structure, and options, as well as a detailed description of the input data file.

3.1. Distribution

The distribution of the current version of the RPMDrate program contains 967 files, consisting of 4 major parts: the program Fortran source code, Python scripts for compiling, linking, running, the RPMDrate manual, and the suite of 4 example calculations for the H + H_2, H + CH_4, OH + CH_4 and H + C_2H_6 reactions.

The files for the example runs consists of source code for the potential energy surfaces used in the test suite, input data files, and the corresponding output files. In general, the user supplies the source code for the potential energy surface in the form described in the RPMDrate manual.

4. Example calculations

In the test calculations reported below we used the Boothroyd-Keogh-Martin-Peterson (BKMP2) potential energy surface for H + H_2 [47], the modified and recalibrated versions of the Jordan-Gilbert [48] potential energy surface developed by Espinosa-García *et al.* for H + CH_4 [49] and OH + CH_4 [50], and the combined valence bond-molecular mechanics (CVBMM) potential energy surface for H + C_2H_6 [51]. (The last three potential energy surfaces were taken from the online POTLIB library [52].) The gradients for use in trajectory simulations are given analytically (H + H_2, H + CH_4, OH + CH_4) or approximated by finite differences (H + C_2H_6). More accurate potential energy surfaces now undoubtedly exist for some of these reactions, but since our goal here is simply to demonstrate the performance of the RPMDrate code any reasonable global potential energy surfaces should suffice.

The input parameters for these calculations are summarized in Table I. The simulations are performed at 300 K. We also performed classical simulations in which the ring polymer was replaced with a single bead, using entirely analogous procedures to calculate the classical TST rate coefficient and the classical transmission coefficient. The parameters used in the RPMD and classical procedures were checked to be sufficient to converge the rate coefficients to within a statistical error of about 5%.

The general behavior of RPMD and classical molecular dynamics simulations is summarized in Figs. 1 and 2. The first of these figures shows the potentials of mean force along the reaction coordinate ξ and the second shows the time-dependent transmission coefficients $\kappa(t)$. The results of the above RPMD and classical simulations are summarized in Table II.

5. Final remarks

In this work, we present the RPMDrate code which calculates the rate coefficients for thermally activated bimolecular chemical reactions in the gas phase using the ring polymer molecular dynamics method. The methodology is based on the Bennett-Chandler factorization which represents the rate as a product of a static (centroid density quantum transition state theory rate) and a dynamic (ring polymer transmission coefficient) factor. A key feature of this methodology is that it does not require that one calculate the absolute value of either the reactant or the transition state partition function. The computational procedure is general and can be used to treat bimolecular polyatomic reactions of any complexity in their full dimensionality. The RPMDrate code can also be used to calculate purely classical rate coefficient by simply replacing the ring polymer with a single bead ($n = 1$). We have tested the RPMDrate code for four hydrogen abstraction reactions, namely, H + H_2, H + CH_4, OH + CH_4, and H + C_2H_6, and calculated the classical and ring polymer rate coefficients at 300 K.

The RPMDrate code uses a global representation of the Born-Oppenheimer potential energy surface of the system. In the current release, this potential energy surface together with gradients must be supplied by the user. In a future release of the RPMDrate code we plan to interface the ring polymer trajectories with fitting algorithms for generating automated fits of *ab initio* PESs, such as automated interpolating moving least squares fitting method [53] or least-squares fitting of permutationally symmetrized multinomials of bond-order-like functions of internuclear distances [54, 55]. This work is currently under way.

Acknowledgement

This work is supported by the US Department of Energy, Office of Basic Energy Sciences under the Energy Frontier Research Center for Combustion Science (Grant No. DE-SC0001198). Y.V.S. acknowledges the support of a Combustion Energy Research Fellowship through the Energy Frontier Research Center for Combustion Science. J.W.A acknowledges Award No. KUS-I1-010-01 made by King Abdullah University of Science and Technology (KAUST).

[1] I. R. Craig, D. E. Manolopoulos, J. Chem. Phys. 121 (2004) 3368.
[2] B. J. Braams, D. E. Manolopoulos, J. Chem. Phys. 125 (2006) 124105.
[3] D. Chandler, P. G. Wolynes, J. Chem. Phys. 74 (1981) 4078.
[4] I. R. Craig, D. E. Manolopoulos, J. Chem. Phys. 122 (2005) 084106.
[5] T. F. Miller, D. E. Manolopoulos, J. Chem. Phys. 122 (2005) 184503.
[6] I. R. Craig, D. E. Manolopoulos, Chem. Phys. 322 (2006) 236.
[7] T. F. Miller, D. E. Manolopoulos, J. Chem. Phys. 123 (2005) 154504.
[8] T. E. Markland, S. Habershon, D. E. Manolopoulos, J. Chem. Phys. 128 (2008) 194506.
[9] T. E. Markland, J. A. Morrone, B. J. Berne, K. Miyazaki, E. Rabani, D. R. Reichman, Nature Physics, 7 (2011) 134.

[10] I. R. Craig, D. E. Manolopoulos, J. Chem. Phys. 123 (2005) 034102.
[11] R. Collepardo, I. R. Craig, D. E. Manolopoulos, J. Chem. Phys. 128 (2008) 144502.
[12] M. Shiga,A. Nakayama, Chem. Phys. Lett. 451 (2008) 175.
[13] S. Habershon, G. S. Fanourgakis, D. E. Manolopoulos, J. Chem. Phys. 129 (2008) 074501.
[14] X. Huanga, S. Habershon, J. M. Bowman, Chem. Phys. Lett. 450 (2008) 253.
[15] T. D. Hone, J. A. Poulsen, P. J. Rossky, D. E. Manolopoulos, J. Phys. Chem. B 112 (2008) 294.
[16] S. Habershon, D. E. Manolopoulos, J. Chem. Phys. 131 (2009) 244518.
[17] A. Kaczmarek, M. Shiga, D. Marx, J. Phys. Chem. A 113 (2009) 1985.
[18] R. Collepardo-Guevara, Yu. V. Suleimanov, D. E. Manolopoulos, J. Chem. Phys. 130 (2009) 174713; 133 (2010) 049902.
[19] A. Witt, S. D. Ivanov, M. Shiga, H. Forbert, D. Marx, J. Chem. Phys. 130 (2009) 194510.
[20] A. R. Menzeleev, T. F. Miller, J. Chem. Phys. 132 (2010) 034106.
[21] F. Calvo, P. Parneix, N.-T. Van-Oanh, J. Chem. Phys. 132 (2010) 124308.
[22] F. Calvo, D. Costa, J. Chem. Theory Comput. 6 (2010) 508.
[23] F. Calvo, Physica D 240 (2011) 1001.
[24] A. R. Menzeleev, N. Ananth, T. F. Miller, J. Chem. Phys. 135 (2011) 074106.
[25] Yu. V. Suleimanov, R. Collepardo-Guevara, D. E. Manolopoulos, J. Chem. Phys. 134 (2011) 044131.
[26] N. Boekelheide, R. Salomón-Ferrer, T. F. Miller, Proc. Natl. Acad. Sci. 108 (2011) 16159.
[27] R. Perez de Tudela, F. J. Aoiz, Yu. V. Suleimanov, D. E. Manolopoulos, J. Phys. Chem. Lett. 3 (2012) 493.
[28] N.-T. Van-Oanh, C. Falvo, F. Calvo, D. Lauvergnat, M. Basire, M.-P. Gaigeot and P. Parneix, Phys. Chem. Chem. Phys. 14 (2012) 2381.
[29] Yu. V. Suleimanov, J. Phys. Chem. C 116 (2012) 11141.
[30] M. J. Gillan, Phys. Rev. Lett. 58 (1987) 563.
[31] M. J. Gillan, J. Phys. C 20 (1987) 3621.
[32] G. A. Voth, D. Chandler, W. H. Miller, J. Chem. Phys. 91 (1989) 7749.
[33] C. H. Bennett, in *Algorithms for Chemical Computations*, ACS Symposium Series No. 46, edited by R. E. Christofferson (American Chemical Society, 1977), p. 63.
[34] D. Chandler, J. Chem. Phys. 68 (1978) 2959.
[35] J. M. Bowman, G. Czakó, B. Fu, Phys. Chem. Chem. Phys. 13 (2011) 8094.
[36] J. Espinosa-Garcia, M. Monge-Palacios, J. C. Corchado, Adv. Phys. Chem. 2012 (2012) 164752.
[37] J. Kästner, W. Thiel, J. Chem. Phys. 123 (2005) 144104.
[38] J. Kästner, W. Thiel, J. Chem. Phys. 124 (2006) 234106.
[39] J. Kästner, J. Chem. Phys. 131 (2009) 034109.
[40] D. Frenkel, B. Smit, Understanding Molecular Simulation, Academic, New York, 2002, Chaps. 15 and 16.
[41] H. C. Andersen, J. Chem. Phys. 72 (1980) 2384.
[42] M. Ceriotti, G. Bussi, M. Parrinello, J. Chem. Theory Comput. 6 (2010) 1170.
[43] M. Ceriotti, G. Bussi, M. Parrinello, Phys. Rev. Lett. 103 (2009) 030603.
[44] M. Ceriotti, G. Bussi, M. Parrinello, Phys. Rev. Lett. 102 (2009) 020601.
[45] H. C. Andersen, J. Comput. Phys. 52 (1983) 24.
[46] M. Ceriotti, M. Parrinello, T. E. Markland, D. E. Manolopoulos, J. Chem. Phys. 133 (2010) 124104.
[47] A. I. Boothroyd, W. J. Keogh, P. G. Martin, M. R. Peterson, J. Chem. Phys. 104 (1996) 7139.
[48] M. J. T. Jordan, R. G. Gilbert, J. Chem. Phys. 102 (1995) 5669.
[49] J. C. Corchado, J. L. Bravo, J. Espinosa-García, J. Chem. Phys. 130 (2009)184314.
[50] J. Espinosa-García, J. C. Corchado, J. Chem. Phys. 112 (2000) 5731.
[51] A. Chakraborty, Y. Zhao, H. Lin, D. G. Truhlar, J. Chem. Phys. 124 (2006) 044315.
[52] R. J. Duchovic, Y. L. Volobuev, G. C. Lynch, A. W. Jasper, D. G. Truhlar, T. C. Allison, A. F. Wagner, B. C. Garrett, J. Espinosa-García, J. C. Corchado, POTLIB, http://comp.chem.umn.edu/potlib.
[53] R. Dawes, A. Passalacqua, A. F. Wagner, T. D. Sewell, M. Minkoff, D. L. Thompson, J. Chem. Phys. 130 (2009) 144107.
[54] Z. Jin, B. J. Braams, J. M. Bowman, J. Phys. Chem. A 110 (2006) 1569.
[55] G. Czakó, B. C. Shepler, B. J. Braams, and J. M. Bowman, J. Chem. Phys. 130 (2009) 084301.

Table 1: Input parameters for the test calculations on the H + H_2, H + CH_4, OH + CH_4 and H + C_2H_6 reactions. The format of the input file is explained in detail in the RPMDRATE manual.

Parameter	Reaction				Explanation
	H + H_2	H + CH_4	OH + CH_4	H + C_2H_6	
Command line parameters					
`Temp`	300	300	300	300	Temperature (K)
`Nbeads`	1(128)	1(128)	1(128)	1(32)	Number of beads in the classical (RPMD) calculations
`Nprocs`	48	48	48	48	Number of processors
Dividing surface parameters					
R_∞	30	30	15	15	Dividing surface s_1 parameter (a_0)
N_{bonds}	1	1	1	1	Number of forming and breaking bonds
N_{channel}	2	4	4	3[a]	Number of equivalent product channels
Thermostat					
`thermostat`	'GLE'	'Andersen'	'Andersen'	'Andersen'	Thermostat option
Biased sampling parameters					
N_{windows}	111	111	111	106	Number of windows
ξ_1	-0.05	-0.05	-0.05	-0.05	Center of the first window
$d\xi$	0.01	0.01	0.01	0.01	Window spacing step
ξ_N	1.05	1.05	1.05	1.00	Center of the last window
dt	0.0001	0.0001	0.0001	0.0001	Time step (ps)
k_i	30	30	30	30	Umbrella force constant (hartree)
$N_{\text{trajectory}}$	200	200	200	150	Number of trajectories
$t_{\text{equilibration}}$	20	20	20	20	Equilibration period (ps)
t_{sampling}	100	100	100	100	Sampling period in each trajectory (ps)
N_i	2×10^8	2×10^8	2×10^8	2×10^8	Total number of sampling points
Potential of mean force calculation					
ξ_0	-0.02	-0.02	-0.02	-0.05	Start of umbrella integration
$\xi^\ddagger$	1.02	1.02	1.02	1.00	End of umbrella integration
N_{bins}	5000	5000	5000	5000	Number of bins
Recrossing factor calculation					
dt	0.0001	0.0001	0.0001	0.0001	Time step (ps)
$t_{\text{equilibration}}$	20	20	20	20	Equilibration period (ps) in the constrained (parent) trajectory
$N_{\text{totalchild}}$	100000	100000	100000	100000	Total number of unconstrained (child) trajectories
$t_{\text{childsampling}}$	2	2	2	2	Sampling increment along the parent trajectory (ps)
N_{child}	100	100	100	100	Number of child trajectories per one initially constrained configuration
t_{child}	0.05	0.05	0.05	0.05	Length of child trajectories (ps)

[a] Hydrogen abstraction from one methyl group is only considered.

Table 2: Various quantities obtained in the present calculations. $\xi^\ddagger_{\text{cl}}$ and $\xi^\ddagger_{\text{RPMD}}$ are the positions of the maxima along the reaction coordinate ξ in the classical and centroid potentials of mean force, respectively. $k_{\text{TST}}(s_\ddagger)$ and $k_{\text{QTST}}(s_\ddagger)$ are the corresponding classical and centroid density quantum transition state theory rate coefficients, and $\kappa_{\text{cl}}(s_\ddagger)$ and $\kappa_{\text{RPMD}}(s_\ddagger)$ are the classical and RPMD transmission coefficients. $k_{\text{classical}}$ is the total classical rate coefficient and k_{RPMD} is the RPMD rate coefficient. The RPMD, TST and QTST rates are given in $\text{cm}^3\text{molecule}^{-1}\text{s}^{-1}$ and the numbers in parentheses denote powers of ten. Temperature is 300 K.

Reaction	Classical				RPMD			
	$\xi^\ddagger_{\text{cl}}$	$\kappa_{\text{cl}}(s_\ddagger)$	$k_{\text{TST}}(s_\ddagger)$	$k_{\text{classical}}$	$\xi^\ddagger_{\text{RPMD}}$	$\kappa_{\text{RPMD}}(s_\ddagger)$	$k_{\text{QTST}}(s_\ddagger)$	k_{RPMD}
H + H_2	1.000	0.994	1.56(-18)	1.55(-18)	1.000	0.776	2.51(-16)	1.95(-16)
H + CH_4	0.999	0.762	6.53(-22)	4.98(-22)	1.007	0.659	1.71(-19)	1.13(-19)
OH + CH_4	0.999	0.893	7.33(-17)	6.55(-17)	0.967	0.567	4.74(-14)	2.69(-14)
H + C_2H_6	0.988	0.699	1.53(-19)	1.07(-19)	0.996	0.707	6.04(-17)	4.28(-17)

```
#!/usr/bin/env python
# encoding: utf-8
##### Step 1. Define the potential energy surface ######################################
from PES import get_potential
##### Step 2. Define the bimolecular reactants #########################################
label = 'H + H2 -> HH + H'
reactants(
    atoms = ['H', 'H', 'H'],
    reactant1Atoms = [1,2],
    reactant2Atoms = [3],
    Rinf = (30 * 0.52918,"angstrom"),)
transitionState(
    geometry = (
        [[  0.000000,  0.000000, -1.7570],
         [  0.000000,  0.000000,  0.000000],
         [  0.000000,  0.000000,  1.7570]],
        "bohr",),
    formingBonds = [(2,3)],
    breakingBonds = [(1,2)],)
equivalentTransitionState(
    formingBonds=[(1,3)],
    breakingBonds=[(2,1)],)
##### Step 3. Define the thermostat ####################################################
thermostat('Andersen')
##### Step 4. Initial umbrella configurations ##########################################
xi_list = numpy.arange(-0.05, 1.05, 0.01)
generateUmbrellaConfigurations(
    dt = (0.0001,"ps"),
    evolutionTime = (5,"ps"),
    xi_list = xi_list,
    kforce = 0.1 * T,)
##### Step 5. Biased sampling ##########################################################
xi_list = numpy.arange(-0.05, 1.05, 0.01)
windows = []
for xi in xi_list:
    window = Window(xi=xi, kforce=0.1*T, trajectories=200, equilibrationTime=(20,"ps"),
    evolutionTime=(100,"ps"))
    windows.append(window)
conductUmbrellaSampling(
    dt = (0.0001,"ps"),
    windows = windows,)
##### Step 6. Calculate the potential of mean force ####################################
computePotentialOfMeanForce(windows=windows, xi_min=-0.02, xi_max=1.02, bins=100000)
##### Step 7. Calculate the transmission coefficient ###################################
computeRecrossingFactor(
    dt = (0.0001,"ps"),
    equilibrationTime = (20,"ps"),
    childTrajectories = 100000,
    childSamplingTime = (2,"ps"),
    childrenPerSampling = 100,
    childEvolutionTime = (0.05,"ps"),)
##### Step 8. Calculate the ring polymer rate coefficient ##############################
computeRateCoefficient()
```

Figure 1: Example input file for the H + H_2 reaction.

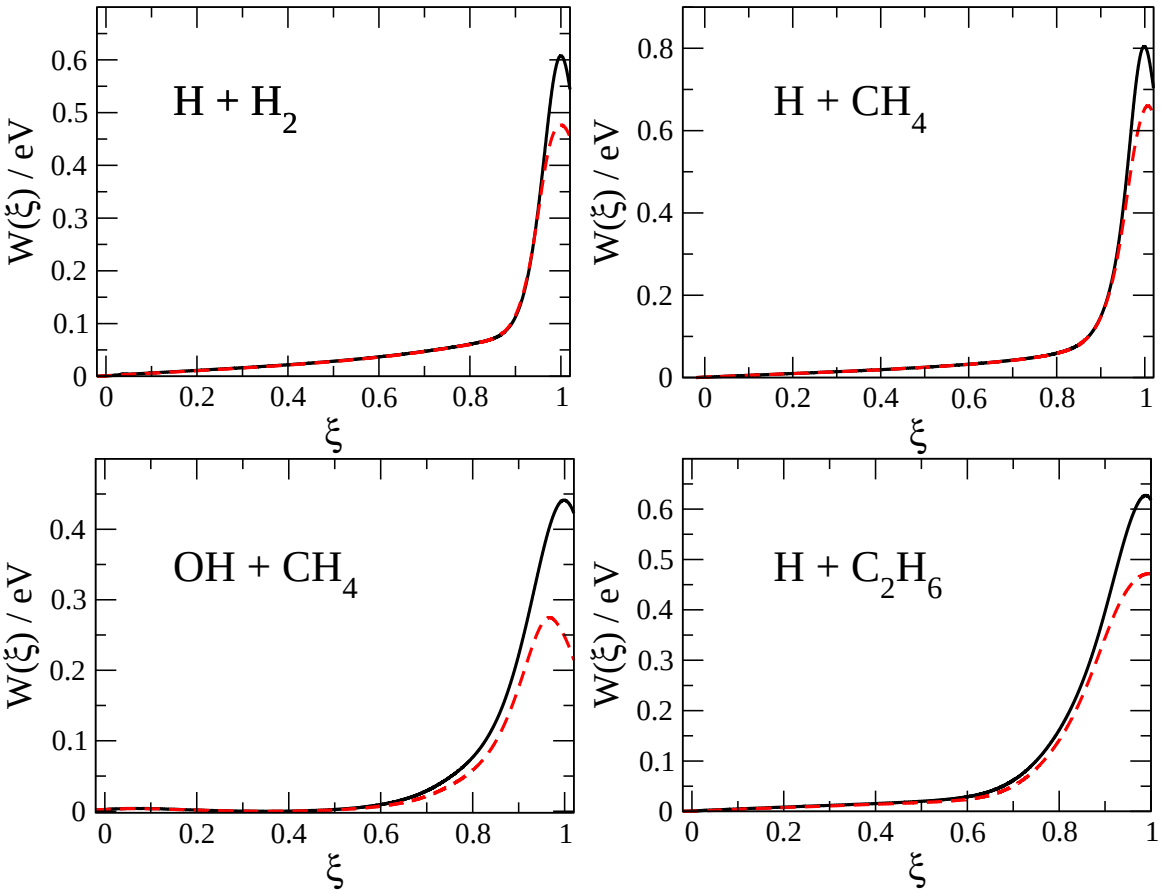

Figure 2: Classical (solid) and centroid (dashed) potentials of mean force for the H + H_2, H + CH_4, OH + CH_4, H + C_2H_6 reactions at 300 K.

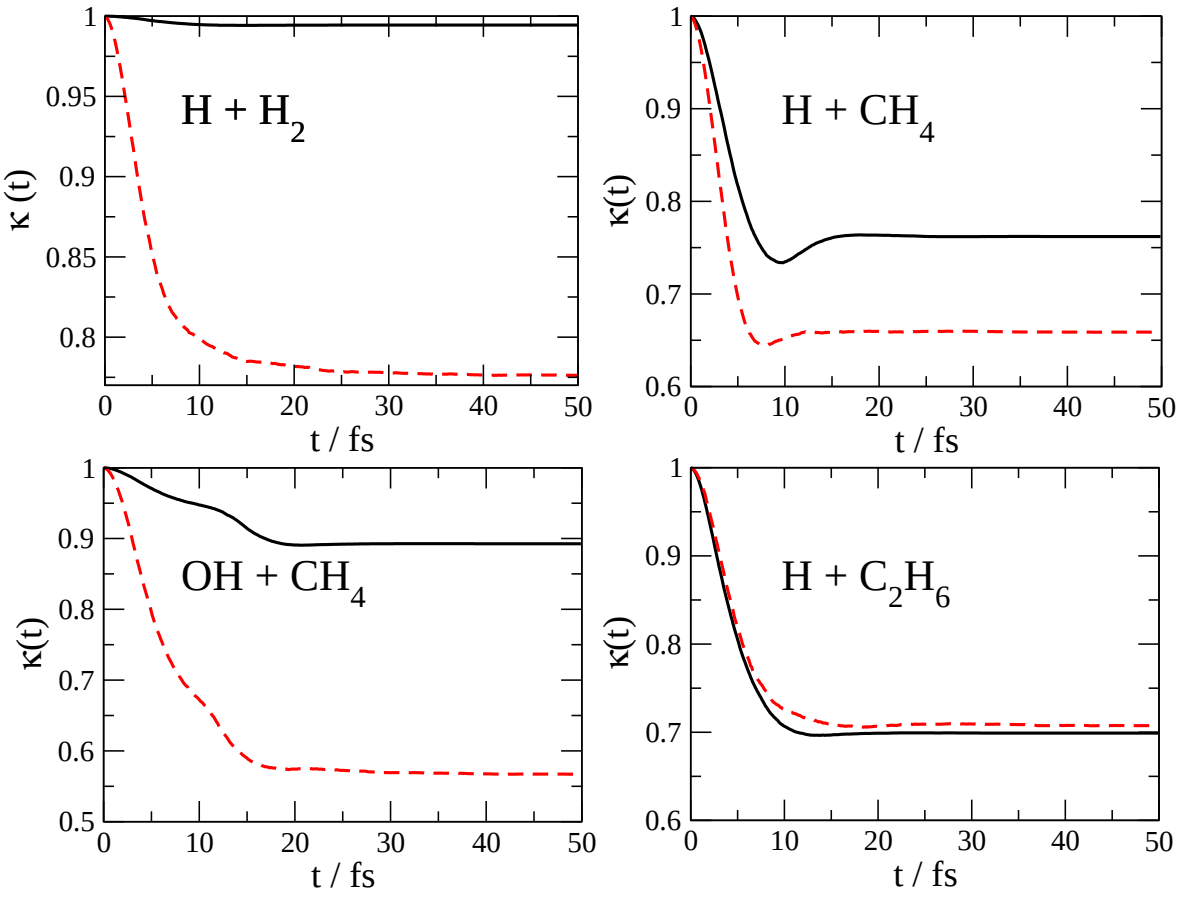

Figure 3: Classical (solid) and RPMD (dashed) time-dependent transmission coefficients for the H + H_2, H + CH_4, OH + CH_4, H + C_2H_6 reactions at 300 K.