MQCT: User-ready program for calculations of inelastic scattering of two molecules[☆],[☆☆]

Alexander Semenov[1],[2], Bikramaditya Mandal[2], Dmitri Babikov [*]

Chemistry Department, Wehr Chemistry Building, Marquette University, Milwaukee, WI 53201-1881, USA

ARTICLE INFO

ABSTRACT

A program named MQCT is developed for calculations of rotationally and vibrationally inelastic scattering of molecules using the mixed quantum/classical theory approach. Calculations of collisions between two general asymmetric top rotors are now possible, which is a feature unavailable in other existing codes. Vibrational states of diatomic molecules can also be included in the basis set expansion, to carry out calculations of ro-vibrational excitation and quenching. Minimal input for the code assumes several defaults and is very simple, easy to set-up and run by non-experts. Multiple options, available for expert calculations, are listed in the *Supplemental Information*. The code is parallel and takes advantage of intrinsic massive parallelism of the mixed quantum/classical approach. A Monte-Carlo sampling procedure, implemented as option in the code, enables calculations for complicated systems with many internal states and large number of partial scattering waves. The coupled-states approximation is also implemented as an option. Integral and differential cross sections can be computed for the elastic channel. Rotational symmetry of each molecule, as well as permutation symmetry of two collision partners, are implemented. Potential energy surfaces for H_2O + He, H_2O + H_2, and H_2O + H_2O are included in the code. Example input files are also provided for these systems.

Program summary
Program title: MQCT
Program files doi: http://dx.doi.org/10.17632/sg36r35njz.1
Licensing provisions: GNU GPL v3.0
Programming language: FORTRAN
Supplementary material: MQCT user guide
Nature of problem: Calculations of rotationally and vibrationally inelastic scattering of two molecules, with possible applications in astrophysics and atmospheric chemistry.
Solution method: Mixed quantum/classical theory (MQCT) approach
Additional comments: This code can be used to simulate a collision of two asymmetric top rotors, such as H_2O + H_2O.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Inelastic collisions of molecules in gas phase play important roles in atmospheric chemistry [1–3], in astrophysical phenomena [4–6] and in combustion [7–9]. In these processes energy is exchanged between translational degrees of freedom (kinetic energy of collision) and internal degrees of freedom of the molecules, such as rotations and vibrations. Theoretical treatment of excitation and quenching of rotational–vibrational molecular states requires quantum mechanics, since these states are quantized, and quite often the state-specific information about collision process is needed (*e.g.*, in spectroscopy, for analysis of state populations during absorption and emission

of light [10,11]). The work-horse of theorists today is the coupled-channel approach [12], in which not only the internal states of molecules are treated quantum mechanically, but also their collision process is described using the quantum scattering formalism [13]. Such full-quantum approach is essential in the ultra-cold physics regime, due to a small number of scattering waves involved in the process and their resultant interference effects [14,15], and when the quantum tunneling is important, for example, for description of scattering resonances [16]. Two popular computer codes are in use nowadays, MOLSCAT [17] and HYBRIDON [18], that allow users to carry out the full-quantum inelastic scattering calculations for small molecules.

Numerical effort associated with such calculations becomes very significant for larger and heavier molecules, due to their dense spectra of rotational states, and more so at higher collision energies and with heavier collision partners, when many values of the orbital angular momentum states (partial waves) should be taken into account for description of the scattering process. This is because in the full-quantum approach the internal states of molecules couple through partial scattering waves, which leads to a large system of coupled differential equations. Computationally affordable cases include diatomic + atom [19,20], triatomic + atom [21–23], and diatomic + diatomic [24,25] collisions. Rotationally inelastic scattering calculations for triatomic + diatomic systems are very demanding [26,27]. For polyatomic molecules, calculations appear to be affordable at low scattering energies only, and only with the simplest and lightest collision partner, such as He atom [28–30]. Even more complexity is brought by vibrational (in addition to rotational) excitations, that become possible in larger and floppier molecules (*e.g.*, lower-frequency modes, such as torsions or internal rotations) and/or at higher collision energies, when the vibrational bending modes may also become accessible [31,32]. There is a strong need for such calculations in the field of chemistry, physics and engineering, but, using the existing methods and codes, they are still outside of our reach.

An astonishing example is a water + water rotationally inelastic scattering process that has never been studied computationally in sufficient detail, $H_2O + H_2O$ [33,34]. Each of the water molecules in this process should be treated as an asymmetric-top rotor, but neither of the codes available to the community today can treat a collision of two asymmetric top rotors [17,18]. Another representative example includes a group of polyatomic molecules, such as $HCOOCH_3$ [28], HC_3N [35], H_2CO [36], CH_3OH [37], C_6H and C_6H^- [29,30]. For these and other systems of comparable complexity the potential energy surfaces can be computed nowadays (using the methods and codes of the electronic structure theory), but the scattering calculations are nearly impossible using the standard full-quantum scattering approach.

In the last few years we developed and tested a simplified mixed quantum/classical theory (MQCT) for inelastic scattering in which the relative motion of collision partners is described approximately, classically, whereas their internal motion is still described rigorously using quantum mechanics [38–47]. A significant speed up of this approach is achieved by classical-trajectory treatment of the scattering process that uses numerically inexpensive Newtonian mechanics, instead of the Schrodinger equation. Further computational advantage is due to the intrinsic massive parallelism of the MQCT approach, in which different trajectories are independent and can be propagated simultaneously using different processors, without any message passing. The resultant computational gain is very substantial, enabling inelastic scattering calculations for larger molecules and at higher collision energies, compared to the standard full-quantum approach. For example, we could run MQCT calculations for $HCOOCH_3 +$ He at collision energies up to 1000 cm^{-1} [44], in contrast to the full-quantum calculations affordable only up to 30 cm^{-1} [28]. We also carried out the first ever calculations for $H_2O + H_2O$ rotational excitations, in a broad energy range [48]. Accuracy of MQCT has been rigorously tested in a series of recent papers [42,47], and this question will not be reexamined here. We recommend "blending" the full-quantum calculations at low collision energies (where those are indispensable and often affordable), with MQCT calculations at higher collision energies, where they are expected to be accurate, and where no other known method is practical.

In this paper, we present a user-ready code we named `MQCT`, that can be employed for efficient calculations of rotationally inelastic scattering of any two molecules, and for some rotationally–vibrationally inelastic scattering calculations. We want to stress again that, to our best knowledge, no other code can do the inelastic scattering calculations of two general asymmetric top rotors. Potential users of the code are among the members of astrophysics community, atmospheric chemists and, of course, physical chemists. Our code is written in `FORTRAN` and is parallelized using `MPI`. Efficiency of massively parallel calculations (scaling) is explored using the example of water + water rotationally inelastic scattering. Several typical input files are given for calculations on H_2O + He, $H_2O + H_2$ and $H_2O + H_2O$ systems, as examples. Readers will find that MQCT calculations are very easy to set up and run using defaults. The most useful options are discussed in the body of the paper, but many more are described in the *Supplemental Information*. Current users of the full-quantum code MOLSCAT [17] will recognize that our input files are very similar to those they are used to. This was done intentionally, to facilitate familiarization with new program. Indeed, the input files for MOLSCAT calculations can be used for MQCT calculations with minimal modifications. An efficient parser allows communications with the code in a rather efficient way. This is the first release of the program. Its development is ongoing. Several exciting avenues are discussed in Conclusion section.

2. Theoretical framework

Here we briefly outline the major components of the mixed quantum/classical theory of inelastic scattering. More detailed description can be found in the recent literature [47,48]. The collision event can be thought of classically: At the initial moment of time two collision partners are in the asymptotic range, separated by large distance R, that shortens during the time of collision and increases again as collision partners leave the interaction region. The deflection process is determined by change of the azimuthal angle Φ. Time evolution of these continuous *classical* variables and their conjugate momenta P_R and P_Φ is described by the classical-like equations of motion:

$$\dot{R} = \frac{P_R}{\mu}; \tag{1}$$

$$\dot{\Phi} = \frac{P_\Phi}{\mu R^2}; \tag{2}$$

$$\dot{P}_R = -\sum_n \sum_{n'} e^{i\varepsilon_n^{n'} t} \sum_m \frac{\partial M_n^{n'}}{\partial R} a_{mn'}^* a_{mn} + \frac{P_\Phi^2}{\mu R^3}; \tag{3}$$

$$\dot{P}_\Phi = -i \sum_n \sum_{n'} e^{i\varepsilon_n^{n'} t} \sum_m M_n^{n'} [a_{m-1,n'}^* a_{mn} \sqrt{j'(j'+1) - m(m-1)} + a_{m+1,n'}^* a_{mn} \sqrt{j'(j'+1) - m(m+1)}$$

$$- a_{mn'}^* a_{m-1,n} \sqrt{j(j+1) - m(m-1)} - a_{mn'}^* a_{m+1,n} \sqrt{j(j+1) - m(m+1)}]/2. \tag{4}$$

In these equations $\varepsilon_n^{n'} = E_{n'} - E_n$ is used to label energy differences between the initial (lower index) and the final (upper index) internal states of the system, whereas $a_{mn}(t)$ represent time-evolving probability amplitudes for these *quantized* states:

$$\dot{a}_{mn} = -i \sum_{n'} e^{i\varepsilon_{n'}^n t} M_{n'}^n a_{mn'}$$

$$-i\dot{\Phi}[a_{m-1,n}\sqrt{j(j+1) - m(m-1)} + a_{m+1,n}\sqrt{j(j+1) - m(m+1)}]/2. \tag{5}$$

The last term of Eq. (5) describes Coriolis coupling between states with $\Delta m = \pm 1$, driven by classical angular speed $\dot{\Phi}(t)$. Neglecting this term leads to the coupled-states (CS) approximation within MQCT, while retaining this term corresponds to the fully-coupled version of MQCT (or coupled-channel MQCT). Matrix $M_{n'}^n(R)$ in Eqs. (3)–(5) is a potential coupling matrix, its R-dependent elements are real, time independent, and are different for different values of m. The range of m is between $-\min(j, j')$ and $+\min(j, j')$. The total angular momentum of two molecules $\mathbf{j} = \mathbf{j}_1 + \mathbf{j}_2$ is quantized in MQCT. The corresponding eigenfunctions can be formally expressed through states of two coupled rotors:

$$\Psi_{mn}(\Lambda_1, \Lambda_2) = \sum_{m_1 m_2} (j_1 m_1 j_2 m_2 | jm) \Psi_{m_1 n_1}(\Lambda_1) \Psi_{m_2 n_2}(\Lambda_2) \tag{6}$$

Coefficients of this expansion, $(j_1 m_1 j_2 m_2 | jm)$, the so-called Clebsch–Gordan coefficients, are non-zero only if $m = m_1 + m_2$ and $|j_1 - j_2| \le j \le j_1 + j_2$, where m is projection of $\mathbf{j}$ onto the molecule–molecule vector $\mathbf{Q} = (R, \Phi)$. A composite index n is used to label the total set of quantum numbers for the system, $n = \{j j_1 k_{a_1} k_{c_1} j_2 k_{a_2} k_{c_2}\}$. For the two scattering partners the sets of state labels are $n_1 = \{j_1 k_{a_1} k_{c_1}\}$ and $n_2 = \{j_2 k_{a_2} k_{c_2}\}$, thus, one can also write $n = \{j n_1 n_2\}$. The rotation of each scattering partner is treated quantum mechanically and is described by a set of usual Euler angles: $\Lambda_1 = (\alpha_1, \beta_1, \gamma_1)$ for molecule one and $\Lambda_2 = (\alpha_2, \beta_2, \gamma_2)$ for molecule two. According to standard notation, rotational states of an asymmetric top are labeled (in addition to j_1 and m_1 for molecule one) by quantum numbers k_{a_1} and k_{c_1} that represent projections of $\mathbf{j}_1$ onto the principal axis of inertia with smallest and largest values of rotational constants, respectively. And similar for k_{a_2} and k_{c_2} for the angular momentum $\mathbf{j}_2$ of molecule two (in addition to j_2 and m_2).

The potential coupling matrix is diagonal in m (*i.e.*, its elements for transition $nm \leftarrow n'm'$ are non-zero only if $m = m'$) but the actual values of non-zero matrix elements do depend on m. For given m the matrix element for transition $n \leftarrow n'$ is:

$$M_{n'}^n(R) = \langle \Psi_{mn}(\Lambda_1, \Lambda_2) | V(R, \Lambda_1, \Lambda_2) | \Psi_{mn'}(\Lambda_1, \Lambda_2) \rangle \tag{7}$$

Here $V(R, \alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$ is the potential energy hypersurface for the molecule–molecule interaction, expressed through the same variables.

It should be stressed that here we use the so-called *body-fixed* reference frame, where z-axis is defined to pass through the centers of mass of two molecules (*i.e.*, is tied to the classical molecule–molecule vector $\mathbf{Q}$). As collision progresses, this axis turns together with collision partners relative to the space-fixed reference frame (same as vector $\mathbf{Q}$), and this effect is incorporated into the equations of motion (1)–(5) as discussed in our earlier papers. Projection m_1 of momentum $\mathbf{j}_1$ and projection m_2 of momentum $\mathbf{j}_2$ are made onto this body-fixed z-axis, or equivalently on $\mathbf{Q}$.

3. Three components of the input

MQCT calculations of inelastic scattering include time-propagation of the Newtonian-like equations (1)–(4), and the Schrodinger-like equations for evolution of quantum state populations of the molecules, the system of coupled equations (5). The initial conditions for each of those are defined separately in two blocks of the input file, called $SYSTEM and $BASIS. The potential energy surface describes interaction between the quantum and classical parts of the system, and is defined in the third block of the input file, called $POTENTIAL. This block should be the last in the input file, while the order of the first two blocks is interchangeable, since they are independent. Example below sets up input for the default calculations of H_2O + He rotational excitation:

```
$BASIS
     SYS_TYPE=4, A=27.877, B=9.285, C=14.512
$END

$SYSTEM
     LABEL="H2O+He", MASS_RED=3.2748, RMIN=4.5, RMAX=20.0,
     NMB_ENERGS=1, U_ENERGY=200., JTOTL=0, JTOTU=20,
     TIME_STEP=15.0, TIME_LIM=3.5E+6
$END

$POTENTIAL
     E_UNITS=A.U., R_UNITS=A.U.,
     GRD_R=50, GRD_ANG1=8, GRD_ANG2=25, GRD_ANG3=25
$END
```

Each block starts with its name and is finalized by the $END. All entries inside the block are separated by coma. You can have as many spaces, lines or tabs as you want between the keywords and their values, all is taken care of by the parser.

Table 1
Types of systems handled by MQCT, with required and optional input data.

SYS_TYPE	Collision partners	Required constants	Channel labels	Optional input
1	rigid diatom + atom	B_e, D_e	j	
2	vibrating diatom + atom	B_e, D_e, ω_e, x_e	j, v	Vibrational functions, grid
3	symmetric top + atom	A, C	j, k	
4	asymmetric top + atom	A, B, C	j, k_a, k_c	Expansion over sym. top basis
5	rigid diatom + rigid diatom	B_{e_1}, D_{e_1}, B_{e_2}, D_{e_2}	j_1, j_2	
6	vibrating diatom + vibrating diatom	B_{e_1}, D_{e_1}, ω_{e1}, x_{e1}, B_{e_2}, D_{e_2}, ω_{e2}, x_{e2}	j_1, v_1, j_2, v_2	Vibrational functions, grid
7	symmetric top + rigid diatom	A, C, B_e, D_e	j_1, k_1, j_2	
8	asymmetric top + rigid diatom	A, B, C, B_e, D_e	j_1, k_{a_1}, k_{c_1}, j_2	Expansion over sym. top basis
9	asymmetric top + symmetric top	A_1, B_1, C_1, A_2, C_2	j_1, k_{a_1}, k_{c_1}, j_2, k_2	Expansion over sym. top basis
0	asymmetric top + asymmetric top	A_1, B_1, C_1, A_2, B_2, C_2	j_1, k_{a_1}, k_{c_1}, j_2, k_{a_2}, k_{c_2}	Expansion over sym. top basis

First keyword of the block $BASIS indicates type of the system (SYS_TYPE=4 corresponds to the asymmetric-top rotor molecule + atom, see below), followed by three values of the rotational constants of the molecule (in the units of wavenumber). The block $SYSTEM contains reduced mass of the collision partners (MASS_RED, in atomic mass units), the minimum and maximum values of distance between collision partners to initialize and terminate trajectories (RMIN and RMAX in the units of Bohr), the number of collision energies to compute NMB_ENERGS (here one only) and their effective values (U_ENERGY in wavenumbers), the range of total angular momentum quantum number J (from JTOTL to JTOTU), propagation time-step TIME_STEP and the time limit TIME_LIM to terminate trajectories (both in atomic units). In the block $POTENTIAL, the units of energy and distance for the potential energy surface subroutine (supplied by the user, see below) are indicated first, followed by the number of quadrature points for integration of matrix elements for state-to-state transitions.

As one can see the default input is very short. Below we discuss the three input blocks in more detail, and some of the most useful options.

3.1. Quantum degrees of freedom, block $BASIS

In this first release of MQCT there are ten system types, summarized in Table 1, from the simplest diatom + atom, to the most general case of two asymmetric top rotor molecules. For each system, the required rotational and vibrational constants are listed in the Table. Those are used by the code to set up and diagonalize Hamiltonian matrix for rotational motion (using basis set of oblate symmetric-top functions), in order to determine rotational states of the system. An option of invoking the externally-computed user-supplied rotational–vibrational states (*e.g.*, vibrational wave functions defined on a grid of points, or vibrationally distorted rotational states, such as those of Kyro model Hamiltonian, expanded over basis set of the oblate symmetric-top functions) are indicated, where available. When symmetric tops are involved in collisions, the case of an oblate top is handled in a standard way, with rotational constants indicated such that $A > C$, while for a prolate top the input should be in the form $A < C$, opposite to the standard notation. In both cases it is assumed that $B = A$. Although the case of spherical top is not explicitly included, it can be handled also, by entering equal values for A and C. For a general asymmetric top molecule the values of A, B and C should correspond to the x, y and z-axes and be consistent with the potential energy surface routine provided for the system. If the molecule has a symmetry axis, user can take advantage of this, as usual, by aligning it with z, to enable independent calculations for para- and ortho-states of the molecule (and thus to reduce the rotational basis set size accordingly).

By default, the program will form a basis set of six lowest energy channels of the system (and will include all corresponding degenerate states, see below), and will choose the ground state as the initial channel. Alternatively, one can indicate the number of channels, list them explicitly, and choose the initial channel using optional keywords. For example, for the H_2O + He system discussed above, an equivalent input would be:

```
NMB_CHNLS=6,
CHNLS_LIST=0,0,0, 1,0,1, 1,1,0, 1,1,1, 2,0,2, 2,2,1,
INIT_CHNL=0,0,0
```

Here the rotational states of the asymmetric top rotor are labeled using the standard notation: j, k_a, k_c (where k_a and k_c are projections of j onto the principal axis with smallest and largest moments of inertia, respectively). Another useful optional keyword allows forming basis set out of all states below given energy, for example: EMAX=135.4 commands to include in the basis set all states at energies below 135.4 cm^{-1} (which, again, would be the same six channels, see Table C1 in Ref. [48]). Rotational quenching calculations can be initiated by choosing the excited state as initial, for example (in the H_2O + He system): INIT_CHNL=2,0,2.

By default, the most general fully-coupled version of MQCT calculations is carried out, in which the transitions due to the Coriolis coupling are included. Such "coupled-channel" calculations are referred to as CC-MQCT. One important option, initiated by the keyword CS_APPROX=YES, is to run the so-called *coupled-states* calculations, CS-MQCT [42,44,47], where the Coriolis-driven transitions are neglected. Speed up is a factor of ~20, due to a much simpler form of the mean-field potential in this case, and a much smaller number of states accessible from a given initial state [42,44,47]. Recall that (within each channel) the Coriolis force couples $2j + 1$ degenerate states labeled by m, which is projection of j on the molecule–molecule axis, and j is the total angular momentum quantum number of two molecules, also quantized in MQCT [45–48]. (If one of the collision partners is an atom, then j is simply equal to the angular momentum of the second partner, the molecule.) So, in CS-MQCT these Coriolis couplings and transitions are neglected, and the calculations are done *independently* for various fixed values of m.

3.2. Classical degrees of freedom, block $SYSTEM

Initial conditions for MQCT trajectories are generated by the code using `RMAX` as the initial distance between the colliding partners at time equal zero. Initial velocity is defined by reduced mass `MASS_RED` and the effective collision energy `U_ENERGY`. If several values of energy are indicated, the program will run `NMB_ENERGS` independent calculations, for example:

```
NMB_ENERGS=4, U_ENERGY=200.5, 345.3, 1203.7, 45.637
```

Equally spaced collision energies can be defined using the smallest value and an increment:

```
NMB_ENERGS=6, UMIN=175.0, DU=25.0
```

Or, alternatively, the largest value and the decrement:

```
NMB_ENERGS=6, UMAX=300.0, DU=25.0
```

The last two inputs are equivalent.

Collision impact parameters are generated automatically by the code, using the indicated range of values of the total angular momentum J of the system, and the values of individual angular momenta j_1 and j_2 of collision partners in the initial rotational state (defined by `INIT_CHNL` or zero by default, see above). Namely, for given j_1 and j_2 the range of possible values of j which is internal angular momentum quantum number of two molecules is $|j_1 - j_2| \le j \le j_1 + j_2$ (these degenerate states are also quantized in MQCT [45–48]). Moreover, for each j there are $2j + 1$ space-degenerate states labeled by m (which is projection of j on the molecule–molecule axis). Independent calculations with different initial values of j and $|m|$ are required, and all are done automatically by the code. For each of these initial degenerate states the allowed values of orbital angular momentum ℓ (integer, quantized in MQCT) are varied in the range $|J - j| \le \ell \le J + j$. These are used to initiate MQCT trajectories with different impact parameters, since ℓ is closely related to the collision impact parameter b.

There is an option of setting the maximum value of impact parameter, in the units of Bohr, for example: `B_IMPCT=9.50`. If this is specified, then the upper and lower limits of J are not required (`JTOTL` and `JTOTU` are ignored, even if indicated). This option is convenient for calculations in a broad energy range, since the maximum impact parameter is less sensitive to collision energy. At each collision energy, the upper limit of J is determined individually, based on the indicated value of maximum impact parameter.

By default, fourth-order Runge–Kutta method is employed to propagate MQCT trajectories (together with quantum equations for state populations) using a constant step-size `TIME_STEP`. Trajectories in MQCT are integrated through the interaction region and are terminated when the molecule–molecule distance exceeds `RMAX`. If optional time-limit is indicated (by `TIME_LIM`, as in the example above) the trajectory is terminated as soon as either condition is fulfilled. This is recommended at low collision energies, when orbiting trajectories are possible (analogous to quantum scattering resonances, see below).

3.3. Potential energy surface block $POTENTIAL

In this block, the units for user-supplied potential energy subroutine must be specified. For the units of distance (keyword `R_UNITS`) Bohr and Angstrom are available, as defined by `A.U.` and `ANGS`, respectively. For energy units (keyword `E_UNITS`) possible values are `CM−1`, `KLVN` and `KCAL` that correspond to wavenumbers, Kelvin and kilocalories per mole, respectively. Note that angles are always assumed to be in radians.

By default, state-to-state transition matrix elements are computed by direct numerical integration over all internal degrees of freedom. The number of Gauss–Legendre quadrature points should be indicated for each angular coordinate, as in the example above (three Euler angles for an asymmetric top rotor, such as H_2O). Integration over angles is carried out at each point of the molecule–molecule distance grid (`GRD_R` equally-spaced points in the range between `RMIN` and `RMAX`). In the collision dynamics calculations, each matrix element is splined over the distance between the grid points, using 3rd-order one-dimensional spline [49]. Optionally, computed matrix elements can be saved to a file, using keyword `SAVE_MTRX=YES`. By default, this would be in the binary form (unformatted). Formatted matrix output can be requested by the keyword `UNFORMAT=NO`, *e.g.*, for visual inspection by the user.

Alternatively, user can expand the potential energy surface over the basis set of spherical harmonics, as it is usually done in the full quantum inelastic scattering calculations, and is implemented in the existing popular codes MOLSCAT [17] and HYBRIDON [18]. We implemented this approach in MQCT as well, as an option, to facilitate comparison with full-quantum calculations. But, users should be aware that although this procedure is routinely done for small molecules, it is known to produce unphysical results for larger molecules due to truncation issues [28], and is also computationally inefficient. For this representation of the PES, the following keywords are needed (*e.g.*, for H_2O + He [43]):

```
EXPANSION=YES, NMB_TERMS=12, TERMS=
0,0, 1,0, 2,0, 2,2, 3,0, 3,2, 4,0, 4,2, 4,4, 6,0, 6,2, 6,4
```

where the expansion terms taken into consideration are explicitly listed (labeled by λ and μ, see Eqs. (11–12) in Ref. [43]). The externally generated file `PES_EXPAN_TERMS.DAT` is also needed in this case, which should contain radial dependencies of the expansion coefficients on the grid of points in the range between `RMIN` and `RMAX`. Examples of such files are distributed with the code. These expansion coefficients will be used to compute analytically elements of the state-to-state transition matrix. In the course of trajectory, those will be splined between the grid points by the code. The code can also generate a suitable expansion, using keyword `CALC_EXPANSION=YES`, for all system types except 2 and 6 where the vibrational motion is included. These more involved topics are discussed in detail in Appendixes C, D and E in the *Supplemental Information*.

4. Efficient use of the code

4.1. Compiling and running the code

User supplied subroutine for the PES should be compiled first, to create an object file, for example `PES_H2O+He.o`. It should be copied into (or linked to) the main program directory `/MQCT_v1.01`. The MQCT code itself is compiled independently to create the object file `head.o`, and then is linked with the desired PES to create an executable file. Examples of this procedure are given in the files `./comp_MQCT` and `./link_ALL`. These can be executed as commands, after changing access:

```
> chmod +x ./comp_PES ./comp_MQCT ./link_ALL
> ./comp_PES
> ./comp_MQCT
> ./link_ALL
```

Input file for MQCT should have the extension `*.inp`, and its name should be placed in the file `INPUT_NAME.inp`. This permits user to store multiple input files (*e.g.*, for different molecules) in the program directory, but run actual calculations with one specific input file.

There are two general ways of running the code. In the straightforward approach, which is also the default, the program computes elements of the state-to-state transition matrix and then propagates trajectories for collisions, all in a single run. In the optional two-step approach, which we recommend following, the program is run first with small number of processors to compute transition matrix, save it into the file and stop (without doing the calculations of collision). This is done by indicating the following optional keywords:

```
SAVE_MTRX=YES, PROG_RUN=NO
```

Then the program is run again to read the transition matrix (computed previously) and perform massively parallel trajectory calculations using large number of processors. Keywords required for this are:

```
READ_MTRX=YES, PROG_RUN=YES
```

This approach is also convenient when multiple calculations are needed with different input parameters (such as collision energy, initial state, number of trajectories, time step, *etc.*) but with the same basis set, which determines the matrix size. Clearly, the matrix must be computed only once, can be saved in the file and then reused later as many times as needed. The file name is `MTRX_UF.dat` for the binary form (unformatted) and is `MTRX.dat` for the formatted option of the matrix. Note that all intermediate data files created or used by in the code have extension `*.dat`.

There are two levels of parallelization in the code. At the first level, propagation of each trajectory can be done by multiple processors used as a group (*e.g.*, all processors of a node) to compute right-hand sides of the classical and quantum equations of motion. This requires some minimal message passing. At the second level, propagation of different trajectories can be assigned to different groups, which requires virtually no message passing. The program attempts to evenly split all requested trajectories between the groups. For example, if the code is submitted for execution using 60 nodes of the machine with 32 cores per node (1920 processors total) the following option, indicated in the `$SYSTEM` block:

```
MPI_PERTRAJ=32, NMB_TRAJ=300
```

will result in formation of 60 groups, and assignment of 5 trajectories per group. Typically, trajectories with larger impact parameters are shorter and faster to propagate. Thus, for an equalized load, and optimal use of resources, it is recommended to assign several trajectories per group, not just one. Note that `NMB_TRAJ` is an optional keyword used only in conjunction with Monte-Carlo sampling of the initial states (see below). If it is not used, the code automatically determines the required number of MQCT trajectories and attempts to split them evenly between the groups.

The equations of motion (1)–(5) represent a system of ordinary differential equations, and their right-hand sides contain a matrix-vector product (for quantum degrees of freedom) and a vector-matrix-vector product (for classical degrees of freedom). In the code, however, this procedure is optimized by excluding transitions with zero matrix elements and computing each right-hand side using one vector-vector dot product. Calculations of these right-hand sides are parallelized very efficiently in our code using inter-communication protocol within MPI. In Fig. 1 we present the wall-clock data collected from `Cray XC40` machine at NERSC (the Cori, http://www.nersc.gov/systems/cori/). They represent CC-MQCT calculations for the $H_2O + H_2O$ system [48] with 91 channels (3483 states total, including the degenerate states), using one head-on collision trajectory with kinetic energy 8000 cm^{-1}. The figure shows a nearly perfect scaling of MQCT code up to 256 processors. With 1024 processors some small overhead becomes visible. These data correspond to a well parallelized code.

Overall, the two levels of parallelization discussed above (first distributing trajectories over the groups of processors, and then distributing the load of trajectory propagation between processors of a group) enable massive parallelization of MQCT calculations. If the number of trajectories varies from say 100 to 1000 (depending on the system, degeneracy of the initial state and the collision energy), then one can employ rather efficiently, without any significant overhead, on the order of ten thousand processors using the present version of MQCT code.

4.2. Understanding the output

All output files have extension `*.out`. System setup is written into the file `USER_INPUT_CHECK.out` and should be checked by user for correctness. The file `STATES.out` (written if the option `PRNT_STATES=YES` is chosen) contains the list of all quantum states
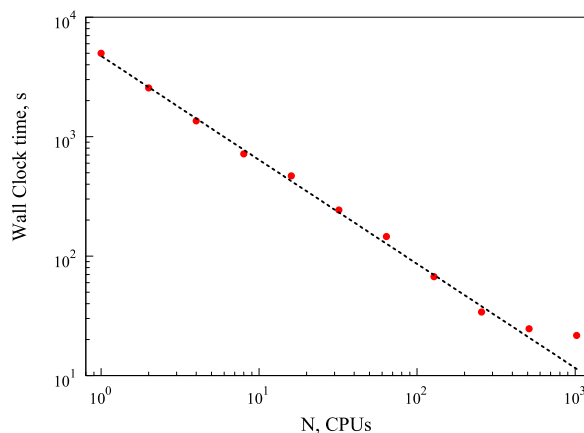
Fig. 1. Efficient scaling of MQCT calculations over a large number of parallel processors. Performance of the code remains nearly constant up to 256 processors. With 1024 processors some small MPI overhead becomes visible. See text for details of this test.

involved in calculations, including the channel number, the values of j and m, and the assigned quantum numbers. Major results are found in `CROSS_SECTIONS.out`. Other, problem-specific output files, are discussed in the next section. Here we cover only the major parts of the output.

Examples of output files for the H_2O + He system are also distributed with the code. Cross sections for transitions from the initial state to the final states (all states of the basis, including the elastic channel) are listed in the output file `CROSS_SECTIONS.out`, for each effective collision energy specified in the input. For each transition, the actual collision energy `E_COLL` is also given in the output file, which depends on the effective collision energy `U_ENERGY` and the state-to-state energy difference. These are different for different transitions (particularly at lower collision energies), which is a property of the mixed quantum/classical approach, discussed in detail in several recent papers [40,47]. Next in the output file goes an important information about the largest values of energy conservation error, and the probability conservation error (both given as % of the initial value) encountered during the propagation. User should check these numbers to ensure that they are reasonable (say below `1.0E−03`). Excessive values indicate that modification of the propagation parameters in the input file is needed. The last in the output file is CPU-time statistics of the code execution.

More detailed information about propagation accuracy can be found in the file `INTEGRATOR_ERRORS.out`, where the data collected for the less accurate (worst) trajectory in the batch are printed, for each collision energy. Other generally useful information about behavior of MQCT trajectories (classically-treated translational degrees of freedom) is found in the file `DEFLECTION_FUNCTION.out`. Final probability amplitudes of the internal states (quantum degrees of freedom) can be found in the file `OPACITY_FUNCTIONS.out`. These files are generated by default. We recommend that users inspect all these dependencies after each program execution, to make sure that the overall behavior of the system is reasonable.

4.3. Random sampling of the initial conditions

For larger and heavier molecules (with large J and ℓ) and/or highly excited rotational states (with large j_1 and/or j_2), the number of initial states that should be taken into account may become prohibitively large, especially at high energies. Running MQCT trajectories for all possible initial states, which is a default in our code, may become computationally expensive and, in fact, unnecessary. In such cases, it is more efficient to sample the values of J, ℓ, j and m randomly and simultaneously, using an efficient multi-dimensional Monte-Carlo procedure [42,45,47]. This option is initiated by the following keywords:

```
MONTE_CARLO=YES, NMB_TRAJ=300
```

where `NMB_TRAJ` is the user-defined number of trajectories to compute. Note that in the case of Monte-Carlo sampling the elastic cross section and the differential cross section is not computed (due to the lack of accurate phase information, see below).

Evolution of the Monte-Carlo error during this sampling process is written into the file `MONTE_CARLO_ERROR.out` for each state of the system individually, and can be monitored by user. The final standard errors of transition cross sections are written into the main output file `CROSS_SECTIONS.out`, for each transition individually. Also, the projection is made for the number of trajectories needed to reach the desired accuracy, as indicated by the keyword `EPS_MONCAR`. After one Monte-Carlo run the user can increase the value of `NMB_TRAJ` accordingly, set `RESTART=YES` and run the code again to finalize calculations, starting from the checkpoint file `CHECK_POINT.DAT`.

4.4. Convergence studies

Convergence studies with respect to several input parameters should be carried out by user. For the quantum part of the system sensitivity of results (cross section values) should be checked with respect to the basis set size `NMB_CHNLS`, the range of molecule–molecule distances used to compute transition matrix elements `RMIN` and `RMAX`, the corresponding number of points `GRD_R`, and the number of integration points for each internal degree of freedom (*e.g.*, parameters `GRD_ANG1`, `GRD_ANG2` and `GRD_ANG3` in the case of H_2O + He system). If the PES is represented by expansion over spherical harmonics, then the convergence parameter is the number of terms `NMB_TERMS` (and, of course, what terms are included). For trajectory propagation, sensitivity of energy and probability

conservation errors should be checked with respect to step size TIME_STEP, and, if the adaptive step-size is used, the value of tolerance EPS_ODEINT and the minimum time step MIN_TMSTP (see below). For sampling of the initial conditions, convergence parameter is the upper limit of total angular momentum JTOTU (or, optionally, the maximum value of collision impact parameter B_IMPCT). If optional Monte-Carlo sampling of the initial states is chosen (MONTE_CARLO=YES) then the number of trajectories NMB_TRAJ is also a convergence parameter. The value of RMAX may also affect MQCT trajectories, since it is used to set up the initial molecule–molecule separation. Note that RMAX can be specified larger than the range of the PES. In this case, the code will automatically extrapolate matrix elements using an attractive $-1/R^6$ function in the asymptotic range of distances (a linear function is used for extrapolation in the repulsive short-distance range).

Often, user will want to increase the number of channels relative to the previously run calculations, for example, in order to check convergence with respect to the basis set size, or to do calculations at higher collision energy, where the number of channels is typically larger. In this case our code allows to add new needed elements to the existing transition matrix, without re-computing the entire matrix. This is achieved, simply, by rerunning the code with new increased number of channels, using any method of channel specification available, for example, increasing the value of keyword NMB_CHNLS and adding new channels to the list CHNLS_LIST (or increasing the value of keyword EMAX, or using other appropriate keywords indicated in Table A1 of *Supplemental Information*). The code will read the existing matrix from file, check whether all needed matrix elements are present, automatically compute the missing elements, update the matrix file and/or proceed with trajectory calculations, depending on what options are indicated by the user. One requirement is that *R*-grid remains identical to the one used in the previous calculations. However, the way of computing elements of the state-to-state transition matrix can be changed, for example, by increasing the number of integration points for the internal coordinates, or increasing the number of terms in the PES expansion, etc. (see Options 1–4 in Appendix C of *Supplemental Information*). The only requirement is that new channels are added at the end of the list, not at the beginning or stuffed in the middle (automatically taken care of if EMAX is used).

Also, user can always run calculations of dynamics with the number of channels smaller than that in the saved matrix file MTRX_UF.dat. The code will read the file, and chose only those matrix elements that are necessary for this run. Again, the requirement is that the active channels are listed continuously from the beginning of the list, and only the channels at the end of the list can be omitted. If the user wants to exclude some states from the beginning or from the middle of the list, the keyword EXCLUDE_STATES can be used as explained in Appendix B (the corresponding data file should be provided, see *Supplemental Information*).

4.5. Propagation options

Besides the default RK4 propagator, one can choose an adaptive step-size method by indicating:

```
PROPAGATOR=ODEINT, TIME_STEP=500.0, MIN_TMSTP=10.0,
EPS_ODEINT=1.0E-3
```

This method adjusts time-steps along each trajectory trying to keep accuracy below EPS_ODEINT, and is a slightly modified version of the code from Numerical Recipes [49]. Our version of this propagator enforces MIN_TMSTP to avoid an excessively long integration near the turning point. We recommend that users run their first calculations using the default RK4 propagator to determine a suitable value of TIME_STEP for their system and collision conditions, (by monitoring energy and norm conservation in the output file). Then, one can try to switch to the ODEINT by simultaneously setting MIN_TMSTP equal to this value, and increasing the value of TIME_STEP by an order of magnitude or even more (which in ODEINT plays the role of the maximum time step allowed). We found that for the systems with deep molecule-molecule attraction potentials, such as dipole-dipole, this approach gives a considerable computational advantage. Also, the ODEINT propagator has an option SINGLE_STEP that commands to integrate the entire trajectory in one step. In this case the code automatically estimates the termination time for each trajectory (based on the impact parameter, the initial separation of collision partners, and the collision energy) and uses this number as maximum time step. This is the most efficient propagation option recommended for the production runs. Note that if SINGLE_STEP is chosen, then no propagation information is printed along the trajectory, no phases are computed, and thus no elastic or differential cross sections are calculated.

5. Special cases of MQCT

5.1. Vibrational states and transitions

For the vibrating-diatom cases, SYS_TYPE=2 and 6, the code will automatically generate a specified set of vibrational states using Morse parameters indicated in the input file. Vibrational energies are computed using a standard Dunham's formula. Vibrational wave functions are computed using recurrent relations [50], and then are used to compute matrix elements for the corresponding state-to-state transitions. Integration is between the points RMIN_VIBGRID and RMAX_VIBGRID (indicated in the block $BASIS) using a constant-step quadrature of GRD_VIB points (indicated in the block $POTENTIAL). These are convergence parameters, and should be checked by the user.

Note that this way of describing ro-vibrational transitions is expected to be reasonably accurate only for low to moderate rotational and vibrational excitations, since it neglects ro-vibrational interaction. More reliable approach, accurate up to dissociation limit, is to compute numerically accurate ro-vibrational states using an external code (not provided) and feed them as input for the MQCT code. This is achieved by the keyword LEVELS_FILE=YES. If specified, the file named USER_DEFINED_BASIS.DAT is also required (copied or linked to the code directory), that should contain energies and wave functions of pre-computed states. The number of states in the files should be equal to NMB_CHNLS (indicated in the block $BASIS). The values and weights of grid points for numerical integration (can be non-equidistant) should also be specified. The number of points should be equal to GRD_VIB (indicated in the block $POTENTIAL). Examples of such files are distributed with the code.

5.2. Differential cross sections and elastic scattering

Calculations of differential cross sections involve phase information and computation of a coherent sum over all partial scattering waves [42], and thus requires propagation of MQCT trajectories for *all* allowed integer values of J and ℓ [51], which is default in the code. The keyword DIFF_CROSS=YES can be used to request construction of the differential cross section. Angular resolution of the differential cross section is defined by ANG_RES. Angular dependence of differential cross section is printed into the file DIFF_CROSS.out.

However, when the Monte-Carlo sampling of the initial conditions is requested (for numerical efficiency, see above) the values of J and ℓ will be chosen randomly and only a few of them may be available for each initial degenerate state (j, m). In this case the meaningful determination of the deflection function, scattering phase and the differential cross section is technically challenging, and is not implemented in the code. So, MONTE_CARLO=YES should not be used simultaneously with calculations of the differential cross section. Same considerations apply to the integral cross section for the elastic scattering channel, since it also requires the scattering phase. If the MONTE_CARLO=YES option is used, the value of elastic cross section should normally be ignored (zero is printed in the output file). In many applications, the elastic and/or differential cross section are not needed. Then MONTE_CARLO=YES is the best option.

If the elastic and/or differential cross sections are needed for complex systems, the most efficient and robust approach is to use the optional keyword DL, in order to skip some values of ℓ and thus make the overall calculations more affordable. For calculation of the differential cross section the code still computes a coherent sum over all ℓ, but within the "boxes" of size DL the same values of scattering phase and probability amplitudes are used. The magnitude of DL becomes a convergence parameter in this case, and should be carefully checked by convergence studies. Note that the full-quantum calculations often follow a similar approach, by skipping some values of the total angular momentum J of the system.

5.3. Trajectory analysis and resonances

There are several options in the code that allow users to visualize MQCT trajectories, or their most important properties. Opacity functions and deflection functions are printed by default, which gives dependencies of transition probabilities and scattering angle on ℓ and b (but only for MONTE_CARLO=NO, which is default). Another useful option is to use keyword PRN_TRJCT to print out all information for a trajectory with indicated value of ℓ. By default, the information from trajectories with the initial $j = \max(j_1, j_2)$ and $m = 0$ is printed. Other values of the initial j and $|m|$ can be specified using optional keyword, for example: PRN_JM=3,3. Plotting and inspecting the deflection function, opacity functions and trajectories is recommended, particularly at low scattering energies, when trajectories may be trapped at certain values of ℓ.

There are options in the code to deal automatically with trapped trajectories, if those occur. For such trajectories the number of loops (due to mutual rotation of collision partners around the origin), and the number of periods (due to mutual oscillations of collision partners along the intermolecular distance R) is determined. The option NMB_LOOPS=2 commands to stop propagation of orbiting trajectories after two full loops (the default value is 1). The option NMB_OSCIL=5 commands to stop propagation at the fifth outer turning point (the default value is 1). When such trapped trajectory is forced to stop, it is still analyzed in a standard way. However, the result of such analysis is somewhat arbitrary, since the termination point is also arbitrary. The option NO_RESONANCE=YES tells the code to remove looping and oscillating trajectories from analysis, which can be used for calculation of non-resonant contribution to the integral cross sections.

Note that differential cross sections and the elastic channel integral cross sections cannot be computed rigorously at collision energies when at least one resonant trajectory is present, since in such cases the deflection function is undefined in the range of small values of ℓ [42,51]. Good recipe for extracting resonance information from trapped trajectories is yet to be found [51]. In any case the code automatically detects trajectories that exhibit resonant behavior and prints some basic information about these trajectories in the file RESONANCE_TRAJECT.out, including the relevant values of ℓ. Then, if desired, the user can rerun the code with the option PRN_TRJCT employed, to obtain more detailed information for each resonant trajectory.

5.4. Molecular symmetry

Rotational states of asymmetric-top molecules are split onto two groups, called para- and ortho-states. We define them based on what values of the quantum number k participate in expansion of wave function over the basis of symmetric-top eigenstates. Namely, for each j even values of k produce para-states, while odd values of k produce ortho-states. Even and odd values of k never mix [52]. For symmetric molecules, such as H_2O, transitions are allowed within each group only, and are exactly forbidden between the para- and ortho-states, due to the symmetry of potential of interaction of the molecule with a quencher (any quencher). Including all states would not cause a problem, but would be meaningless since the efficiency of calculations would be reduced. If the states are specified explicitly as a list, the user should take care of this issue manually (for assignments of rotational states of water as ortho/para see Table C1 in Ref. [48]). But, if the basis set is generated automatically (*e.g.*, using the keyword EMAX), user has an option to indicate SYMMETRY=YES, for reducing the basis set size to one symmetry only, depending on symmetry of the initial state.

Note that in asymmetric molecules, such as methyl formate [44], all transitions are allowed, and all states should be included in the basis. Thus, indicating the keyword SYMMETRY=YES, would lead to unphysical results in this and other cases with no symmetry, and normally should not be done. In the molecule + molecule case, symmetry consideration will be applied to the states of each molecule individually, for example, one can compute collisions of para-water with ortho-hydrogen, etc.

5.5. Collisions of identical molecules

When identical molecules are collided (*e.g.*, $H_2O + H_2O$) one should choose to treat them as indistinguishable, by indicating the keyword IDENTICAL=YES in the input file. In this case only one set of rotational quantum numbers is needed on input, and the unique channels only should be specified. For example, if the state (1,1,1, 0,0,0) of $H_2O + H_2O$ is already specified, one should not include the state (0,0,0, 1,1,1) since two molecules are treated as indistinguishable.

Symmetry considerations discussed in the previous subsection apply to each molecule individually, which results in three manifolds of uncoupled rotational states for the system of two indistinguishable molecules: para + para, ortho + ortho, and para + ortho collisions. If the basis set is specified explicitly as a list of states, the user should make efforts to exclude the uncoupled states. If the basis set is generated automatically (employing the keyword EMAX, or keywords EMAX1 and EMAX2), user has option to minimize the number of channels by choosing SYMMETRY=YES. Again, only the states coupled to the initial state would be included, based on symmetry considerations.

For scattering of two identical collision partners there is one more effect of symmetry, related to swapping the two molecules, which can be done in two ways, by adding or by subtracting their wave functions. The resultant states are energetically degenerate, of course, but transitions between them are forbidden [48]. To distinguish between the two manifolds of uncoupled states one should use the *exchange parity* ($\pm$). This sign, together with the orbital angular momentum quantum number ℓ of collision, affect the values of state-to-state matrix elements through a factor of $\pm(-1)^{\ell}$, as it was demonstrated by Eq. (25) in Ref. [48]. It appears that in general, *two* separate MQCT calculations are needed in the case of identical particle scattering: first run for $+(-1)^{\text{evn}}$ and $-(-1)^{\text{odd}}$, and second run for $+(-1)^{\text{odd}}$ and $-(-1)^{\text{evn}}$. Note that each of these two calculations includes all values of ℓ, both even and odd. The results are converted then into four cross sections: $\sigma^{(+)\text{evn}}$ and $\sigma^{(-)\text{odd}}$ from the first run, and, $\sigma^{(+)\text{odd}}$ and $\sigma^{(-)\text{evn}}$ from the second run, respectively. If IDENTICAL=YES is chosen in the input file, user can also employ the optional keyword WGHT_POSPAR. The default value is one, which leads to the overall cross section computed as $\sigma^{(+)} = \sigma^{(+)\text{evn}} + \sigma^{(+)\text{odd}}$ from results of the two runs (done independently and consequently by the code). The value of zero for WGHT_POSPAR, in contrast, commands to do produce $\sigma^{(-)} = \sigma^{(-)\text{evn}} + \sigma^{(-)\text{odd}}$ based on results of the two runs. For any value of this keyword in the range between zero and one, both $\sigma^{(+)}$ and $\sigma^{(-)}$ will be computed and included into the final cross section with appropriate weights (such that their sum is unity).

Trivial cases occur when both molecules are in the same initial state, such as (0,0,0, 0,0,0) state of $H_2O + H_2O$ system. In this case, negative values of $\pm(-1)^{\ell}$ turn wave function of the system into null (since the total internal parity p is positive,[3] see Eq. (21) and Table C1 of Ref. [48]). Thus, only one MQCT run is needed, which produces $\sigma^{(+)} = \sigma^{(+)\text{evn}}$ and $\sigma^{(-)} = \sigma^{(-)\text{evn}}$. Moreover, if the spin weight of the positive parity is one, as indicated by the keyword WGHT_POSPAR, then the overall cross section is equal to just $\sigma^{(+)\text{evn}}$, which requires calculations with even values of ℓ only. The case of negative total internal parity of the initial state would be handled similarly and would require only odd values of ℓ. This would be true for $j = 1$ component of the (1,1,1, 1,1,1) state of $H_2O + H_2O$ system. Our MQCT code recognizes such special cases and caries out only the necessary calculations.

6. Example calculations

As example, here we present new data obtained with MQCT code for $H_2O + H_2$. Note that in the past the mixed quantum classical approach has not been applied to any asymmetric top rotor + diatom system, so, these are the first calculations of this sort.

In Fig. 2 we show results for quenching of the excited rotational state 2_{11} of H_2O into three lower energy states: 0_{00}, 1_{11} and 2_{02}. Collision energy varies through four orders of magnitude range of values, from 10 to 10,000 cm^{-1}. Each frame of Fig. 2 has four cross section dependencies that correspond to various initial/final states of H_2. Namely, black and blue symbols describe elastic H_2 channels $j_2 = 0$ and $j_2 = 2$, red symbols describe rotational excitation of H_2 from $j_2 = 0$ to $j_2 = 2$, whereas green symbols describe rotational quenching of H_2 from $j_2 = 2$ to $j_2 = 0$ (all happening simultaneously with rotational quenching of H_2O, as explained above). From Fig. 2 one can see that the overall quenching processes (black, green, blue) have no energy thresholds and in the low energy range the values of cross sections slowly grow as collision energy is reduced. In contrast, the overall excitation processes (red in three frames of Fig. 2) all have a well-defined threshold energy, where the value of cross section drops sharply as energy is reduced. This happens because the excitation of H_2 from $j_2 = 0$ to $j_2 = 2$ requires more energy than can be released by the quenching of H_2O (2_{11}), so, even when combined these transitions require some minimal amount of energy, which should come from the relative motion of the colliding partners.

In order to benchmark the accuracy of these MQCT predictions we carried out the full-quantum calculations using MOLSCAT package [17], for the same collision processes in the $H_2O + H_2$ system. The size of rotational basis set was also the same: for water the states up to $j = 4$ and for hydrogen the states up to $j = 2$ were considered, but only the states with combined rotational energy below 600 cm^{-1} were retained. These full-quantum data are presented in Fig. 2 as solid lines (of matching colors), and thus can be compared with MQCT results (symbols). One can see that, overall, the two sets of data are very similar. The behavior of cross section near threshold energy for the excitation process is predicted correctly by MQCT (red symbols vs lines), as well as the overall dependencies of cross sections on collision energy. At higher energies the results of MQCT become nearly identical to the full-quantum results of MOLSCAT, as expected. At lower collision energies the differences become somewhat larger, as one could also expect, but the relative magnitudes of all state-to-state transition cross sections in the system still remain correct. For simplicity, we removed resonances from Fig. 2, but it should be stressed that the mixed quantum/classical approach can offer some useful insight into the resonant phenomena too (through analysis of orbiting trajectories [51]).

It can also be noted that MQCT data presented above are in very good agreement with results of the most rigorous and complete study of water quenching, carried out by Dubernet and coworkers [26]. Three frames of Fig. 2 here can be compared with Fig. 1a-c of Ref. [26], which indicates good agreement, although it should be noted that our convergence parameters were not identical to those of Ref. [26], where the basis sets size was progressively increased as a function of collision energy.

---

[3] Note, this is different from the total "internal" parity of the state defined in Ref. [48] as $p = (-1)^j(-1)^{\kappa_1+\kappa_2}p_1p_2$, where j is the value of total angular momentum of two molecules, κ_1 and κ_2 are their ortho/para assignments, p_1 and p_2 are their inversion parities. The value of p remains unaffected by the swap.
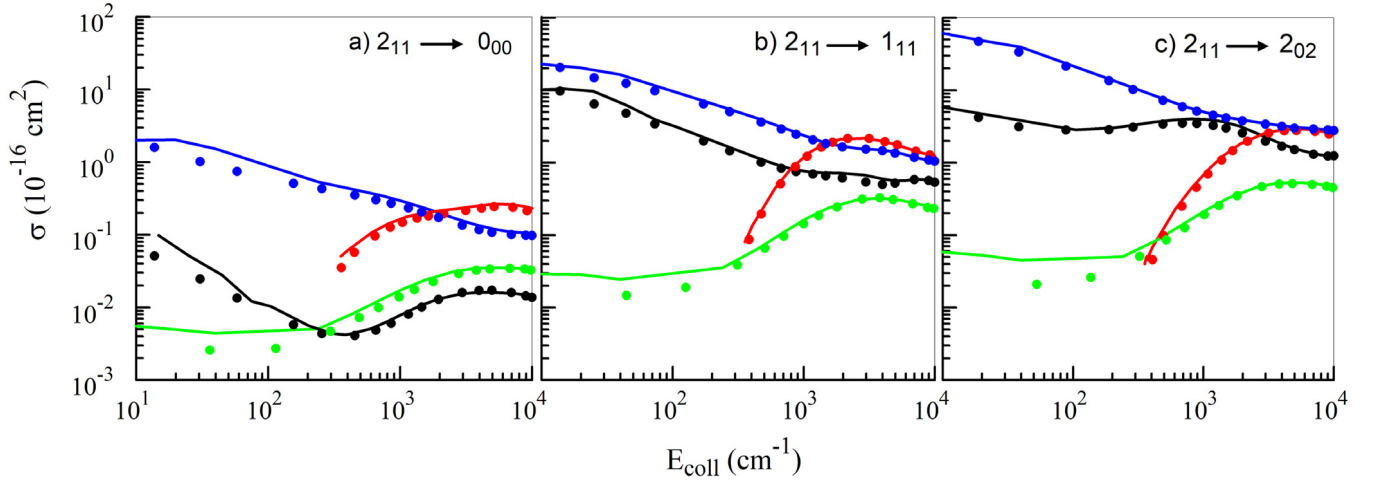
Fig. 2. Cross sections for quenching of the excited rotational state 2_{11} of H_2O onto the lower levels: 0_{00}, 1_{11}, and 2_{02}. Black and blue lines represent the elastic H_2 channels with $j_2 = 0$ and $j_2 = 2$, respectively. Red line represents excitation of H_2 from $j_2 = 0$ to $j_2 = 2$ while green line describes quenching of H_2 from $j_2 = 2$ to $j_2 = 0$. Results of MQCT are indicated by symbols; full-quantum results of MOLSCAT are shown by lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
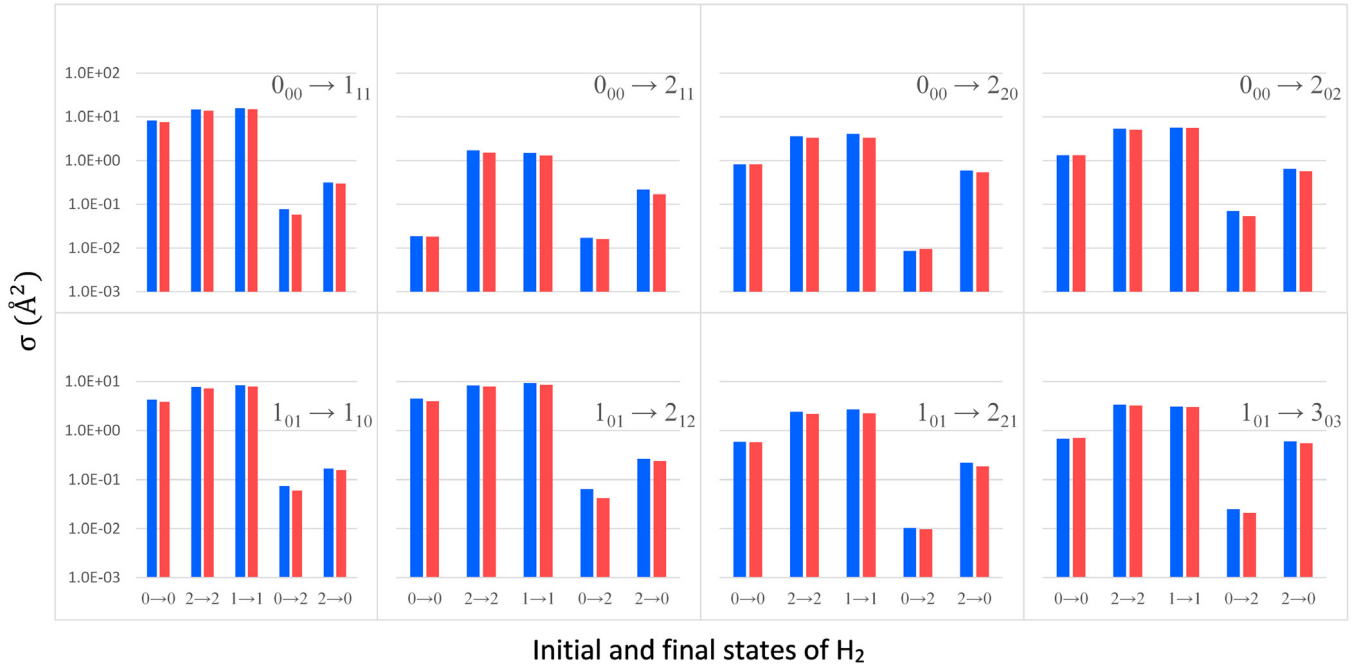
Fig. 3. Cross sections for excitation of the ground state para-H_2O (0_{00}) and ortho H_2O (1_{01}) by para-H_2 ($j_2 = 0$, 2) and ortho-H_2 ($j_2 = 1$). Upper row is for para-H_2O, lower row is for ortho-H_2O. The final state of water is indicated in each frame, while the initial and final states of H_2 are listed along the horizontal axis. Results of MQCT are in red; full-quantum results of MOLSCAT are in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We also carried out calculations for the excitation of para- and ortho-water in their ground states, H_2O (0_{00}) and H_2O (1_{01}) respectively, by para- and ortho-hydrogen ($j = 0, 1, 2$) for one value of collision energy, 574 cm^{-1}. Again, for the purpose of benchmarking, two sets of calculations were conducted, using our MQCT code and using the full-quantum code MOLSCAT [17], with the same rotational basis set as above. The resultant cross sections are presented in Fig. 3, in the upper row for the states of para-water, and in the lower row for the states of ortho-water. Their values vary through three orders of magnitude range and demonstrate a systematic very good agreement between MQCT (red) and MOLSCAT (blue) for all kinds of transitions. This comprehensive survey serves to show that symmetry properties of rotational states are captured correctly by our method and code, describing allowed and forbidden transitions between the ortho- and para-states of molecules. In particular, here we carried out separate calculations for four combinations of allowed transitions between the para and ortho states in water and in hydrogen molecules. But also, we carried out additional calculations where all these states were included to check that, for the transitions forbidden by symmetry (such as between ortho- and para-states of the same molecule), the values of cross sections would come out close to numerical zero. In practice it is also useful to run such test in order to determine the level of convergence of the code.

It can finally be noted that the data presented in Fig. 3 are in very good agreement with results of Wiesenfeld and coworkers [27] computed using MOLSCAT, but with somewhat larger basis set (see Fig. 6 and Table 3 in Ref. [27]). Those data, in turn, were shown to be in excellent agreement with experimental results, and thus we can claim that cross sections obtained by MQCT compare favorably against the available experiments. Therefore, the code MQCT can be used for reliable computational prediction of state-to-state transition cross sections in the inelastic molecular collision processes.

7. Conclusions

In this paper we presented the code MQCT for calculations of rotationally inelastic scattering of molecules using mixed quantum/classical theory. MQCT calculations are now possible for collision of two general asymmetric top rotors, which is a feature unique to this code, unavailable in other existing codes, to our best knowledge. Vibrational states of diatomic molecules can be included in the basis set expansion, to carry out ro-vibrational calculations of excitation and quenching. The simplest input for our code uses defaults and thus is very short, easy to set-up and run by non-experts. The options available to expert users are listed in the *Supplemental Information*. The code was tested and debugged under Unix environment using two different compilers, Intel and GNU (Cori machine at NERSC), under the Linux on Cray XC30 (Edison at NERSC), but also on a PC under Windows 7 and 10.

The code takes advantage of intrinsic parallelism of the mixed quantum/classical approach and is suitable for massively parallel calculations. A Monte-Carlo sampling procedure, implemented as option in the code, enables calculations for rather complicated systems. The coupled-states approximation can also be employed, at high collision energies. Integral and differential cross sections can be computed for the elastic channel, through reconstruction of the deflection function and calculation of the scattering phase. Rotational symmetry of each molecule, and the permutation symmetry of two collision partners are both implemented. Orbiting trajectories can be identified and analyzed to gain some insight into resonant behavior at low collision energies.

In the main text of the article examples of the program input were given for $H_2O + He$. Example calculations were presented for $H_2O + H_2$ in a broad range of collision energies. Performance of the code (scaling) was studied in the most demanding calculations on $H_2O + H_2O$. The version of the code distributed with this publication includes potential energy surfaces for $H_2O + He$, $H_2O + H_2$, and $H_2O + H_2O$. Example input files are also provided for these systems, to demonstrate several useful options of the code.

In the future, third level of parallelization can be implemented in the propagation subroutine of the code. Moreover, calculations for different collision energies, that are sequential in this version of the code, could in principle be parallelized, as well as calculations for different initial states of the system (current version starts at one chosen initial state, and should be launched for each initial state). More advanced techniques for propagation of the equations of motion, such as preconditioned Lanczos method [53], could also be implemented in MQCT. Another significant development would be to expand the code for calculations of ro-vibrational transitions beyond diatomic molecules, for example, for the bending mode in triatomic molecules, which typically has the lowest frequency and thus is the most important (e.g., in H_2O).

Acknowledgments

Appendix A. Supplementary data

Supplementary material related to this article includes MQCT User Manual that can be found online at https://doi.org/10.1016/j.cpc.2020.107155.

References

[1] M. Ivanov, D. Babikov, J. Chem. Phys. 134 (2011) 174308.
[2] M. Ivanov, D. Babikov, J. Chem. Phys. 136 (2012) 184304.
[3] M. Ivanov, D. Babikov, Proc. Natl. Acad. Sci. 110 (2013) 17708.
[4] F. Daniel, M.-L. Dubernet, A. Grosjean, Astron. Astrophys. A 76 (2011) 536–546.
[5] E. Roueff, F. Lique, Chem. Rev. 113 (2013) 8906–8938.
[6] A. Faure, L. Wiesenfeld, Y. Scribano, C. Ceccarelli, Mon. Not. R. Astron. Soc. 420 (2012) 699–704.
[7] Q. Ma, P.J. Dagdigian, M. Alexander, J. Chem. Phys. 138 (2013) 104317.
[8] N.J. Brown, L.A.J. Bastien, P.N. Price, Prog. Energy Combust. Sci. 37 (5) (2011) 565–582.
[9] A. Schiffman, D. Chandler, Intern. Rev. Phys. Chem. 14 (2) (1995) 371–420.
[10] J.M. Hartmann, C. Boulet, D. Robert, Collisional Effects on Molecular Spectra, Elsevier, 2008.
[11] L. Wiesenfeld, A. Faure, Phys. Rev. A 82 (2010) 040702.
[12] P. Mcguire, D. Micha, Intern. J. Quant. Chem. 6 (1972) 111–132.
[13] A.M. Arthurs, A. A. Dalgarno, 256 1960 540.
[14] B.K. Kendrick, J. Hazra, N. Balakrishnan, Nature Commun. 6 (2015) 7918.
[15] B.K. Kendrick, J. Hazra, N. Balakrishnan, Phys. Rev. Lett. 115 (2015) 153201.
[16] A. Mack, T.K. Clark, R.C. Forrey, N. Balakrishnan, T.-G. Lee, P.C. Stancil, Phys. Rev. A 74 (2006) 052718.
[17] J. Hutson, S. Green, MOLSCAT Computer Code, version 14; Distributed by Collaborative Computational Project (6) of the Engineering and Physical Sciences Research Council, Swindon, U.K., 1994.
[18] M.H. Alexander, D.E. Manolopoulos, H.-J. Werner, B. Follmeg, HIBRIDON, available at http://www2.chem.umd.edu/groups/alexander/hibridon/hib43/hibhelp.html.
[19] N. Balakrishnan, A. Dalgarno, R.C. Forrey, J. Chem. Phys. 113 (2000) 621.
[20] J. Loreau, P. Zhang, A. Dalgarno, J. Chem. Phys. 135 (2011) 174301.
[21] S. Green, J. Chem. Phys. 64 (1976) 3463–3473.
[22] B. Yang, M. Nagao, W. Satomi, M. Kimura, P.C. Stancil, Astrophys. J. 765 (9) (2013) 77.
[23] P.J. Dagdigian, M.H. Alexander, J. Chem. Phys. 138 (2013) 164305.

[24] B. Yang, P. Zhang, X. Wang, P.C. Stancil, J.M. Bowman, N. Balakrishnan, R.C. Forrey, Nature Commun. 6 (1–10) (2015) 6629.
[25] F.S. dos Santos, N. Balakrishnan, R.C. Forrey, P.C. Stancil, J. Chem. Phys. 138 (10) (2013) 104302.
[26] F. Daniel, M.-L. Dubernet, A. Grosjean, Astron. Astrophys. A 76 (2011) 536–546.
[27] C.H. Yang, G. Sarma, D.H. Parker, J.J. ter Meulen, L. Wiesenfeld, J. Chem. Phys. 134 (12) (2011) 204308.
[28] A. Faure, K. Szalewicz, L. Wiesenfeld, J. Chem. Phys. 135 (10) (2011) 024301.
[29] K.M. Walker, F. Lique, F. Dumouchel, R. Dawes, Mon. Not. R. Astron. Soc. 466 (2016) 831.
[30] K.M. Walker, F. Dumouchel, F. Lique, R. Dawes, J. Chem. Phys. 145 (2016) 024314.
[31] A. Faure, L. Wiesenfeld, M. Wernli, P. Valiron, J. Chem. Phys. 123 (2005) 104309.
[32] A. Faure, E. Josselin, Astron. Astrophys. 492 (2008) 257.
[33] P.J. Agg, D.C. Clary, J. Chem. Phys. 95 (1991) 1037–1048.
[34] G. Buffa O. Tarrini, F. Scappini, C. Cecchi-Pestellini, Astrophys. J. Suppl. 128 (2000) 597.
[35] M. Wernli, L. Wiesenfeld, A. Faure, P. Valiron, Astron. Astrophys. 464 (2007) 1147–1154.
[36] N. Troscompt, A. Faure, L. Wiesenfeld, C. Ceccarelli, P. Valiron, Astron. Astrophys. 493 (2009) 687–696.
[37] D. Rabli, D.R. Flower, Mon. Not. R. Astron. Soc. 406 (2010) 95.
[38] A. Semenov, D. Babikov, J. Chem. Phys. 138 (2013) 164110.
[39] A. Semenov, M. Ivanov, D. Babikov, J. Chem. Phys. 139 (2013) 074306.
[40] A. Semenov, D. Babikov, J. Chem. Phys. 139 (2013) 174108.
[41] A. Semenov, D. Babikov, J. Phys. Chem. Lett. 5 (2014) 278.
[42] A. Semenov, D. Babikov, J. Chem. Phys. 140 (2014) 044306.
[43] A. Semenov, M.-L. Dubernet, D. Babikov, J. Chem. Phys. 141 (2014) 114304.
[44] A. Semenov, D. Babikov, J. Phys. Chem. Lett. 6 (2015) 1854–1858.
[45] A. Semenov, D. Babikov, J. Phys. Chem. A 119 (2015) 12329.
[46] A. Semenov, D. Babikov, J. Phys. Chem. A 120 (2016) 3861.
[47] D. Babikov, A. Semenov, J. Phys. Chem. A 120 (2016) 319.
[48] A. Semenov, D. Babikov, J. Phys. Chem. A 121 (2017) 4855.
[49] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge University Press, 1989.
[50] H. Kobeissi, J. Comput. Phys. 61 (1985) 351.
[51] B. Mandal, A. Semenov, D. Babikov, J. Phys. Chem. A 122 (2018) 6157.
[52] P.R. Bunker, P. Jensen, Fundamentals of Molecular Symmetry, IOP Publishing, 2005.
[53] P.C. Stancil1, H. You, A. Cook, A.T. Sornborger, M.R. Geller, Towards quantum simulation of chemical dynamics with prethreshold superconducting qubits, 2016, arXiv:1602.00063v1 [quant-ph].