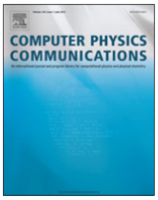
Computer Physics Communications 184 (2013) 885–890

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



A computer program to evaluate the NVM propagator for rigid asymmetric tops for use in path integral simulations of rigid bodies☆

Carl McBride [a,*], Eva G. Noya [b], Carlos Vega [a]

[a] *Departamento de Química Física, Facultad de Ciencias Químicas, Universidad Complutense de Madrid, 28040 Madrid, Spain*

[b] *Instituto de Química Física Rocasolano, Consejo Superior de Investigaciones Científicas, CSIC, Calle Serrano 119, 28006 Madrid, Spain*

ARTICLE INFO

Article history:
Received 25 July 2012
Received in revised form
15 October 2012
Accepted 22 October 2012
Available online 30 October 2012

Keywords:
Rotational propagator
Path integral
Rigid asymmetric tops

ABSTRACT

Here we provide FORTRAN source code to facilitate the calculation of the "Noya–Vega–McBride" (NVM) rotational propagator for asymmetric tops [E.G. Noya, C. Vega, C. McBride, J. Chem. Phys. 134 (2011) 054117] for a given value of PT and A, B and C, where P is the number of beads, T is the temperature, and A, B and C are the rotational constants for the system in question. The resulting NVM propagator calculated by the code provided can then be used to obtain the quantum rotational energy during a path integral Monte Carlo simulation of rigid bodies.

Program summary

Program title: NVM

Catalogue identifier: AEOA_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEOA_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: Standard CPC licence, http://cpc.cs.qub.ac.uk/licence/licence.html

No. of lines in distributed program, including test data, etc.: 624734

No. of bytes in distributed program, including test data, etc.: 9890026

Distribution format: tar.gz

Programming language: Fortran.

Computer: Any.

Operating system: Any.

RAM: <2 Mbytes

Classification: 16.13.

External routines: Lapack routine, dsyev (code included in the distribution package).

Nature of problem: Calculation of the NVM rotational propagator

Solution method: Fortran implementation of the NVM propagator equation.

Additional comments: Example and test calculations are provided.

Running time: 2–200 hours. Two examples are provided. The PT_1497 example will take approximately 11 hours to run. The quick_test should only take a few minutes.

1. Introduction

Nuclear quantum effects are important for low atomic masses and at low temperatures, and in those cases they should be explicitly included in the description of the system. The path integral formalism proposed by Feynman and Hibbs [1] allows one to incorporate such quantum effects in simulations by

☆ This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

* Corresponding author.
E-mail address: carl@ender.quim.ucm.es (C. McBride).

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.10.025

representing molecules as ring polymers consisting of P "beads". The appropriate value of P depends on the temperature and on the mass of the atoms. Thanks to this the basic scheme used for classical simulations can be relatively easily adapted to perform quantum simulations [2,3].

Since many condensed matter properties depend mainly on the intermolecular rather than intramolecular vibrations, several schemes have been proposed to perform path integral simulations treating molecules as rigid tops, either approximately or exactly (in the limit of infinite P) [4–9]. This permits a significant saving in computational time since the high frequency intramolecular vibrations require the use of ring polymers with a larger number of beads, thus considerably increasing the computational cost.

Müser and Berne derived a path integral formalism for rigid rotors [6], in which the translational degrees of freedom (coordinates of centre of mass) are treated using the usual path integral formalism, and where the orientational degrees of freedom (Euler angles) are treated using a rotational propagator, which is exact for spherical and symmetric tops. Recently we extended this formalism to the more general case of asymmetric tops [8], and used it to study many properties of water, using a rigid non-polarizable model, namely TIP4PQ/2005 [10]. This facilitated the study of isotopic effects on the melting temperature and the temperature of maximum density [11] as well as the phase diagram of water including many of its solid phases [12].

In this contribution we provide FORTRAN source code which calculates the rotational propagator for asymmetric tops [8] for a given value of PT and A, B and C, where P is the number of beads, T is the temperature, and A, B and C are the rotational constants for the system in question. The computational cost of evaluating the propagator is quite high, so it is convenient to calculate it over a grid of Euler angles (θ, ϕ, χ) and store it in a file which can be read by the Monte Carlo code. Note that this propagator depends on the temperature, on the rotational constants of the molecule, in other words the magnitude and distribution of mass, and on the number of beads to be used in the simulations. The number of beads needs to be chosen specifically for each system and temperature, the number of beads being larger the more quantum the system is. In what follows we provide the equations that are needed to evaluate the rotational propagator for asymmetric tops and also provide ways to save time calculating the propagator. More details about the derivation of the propagator as well as examples of its application to study quantum effects in water can be found in Refs. [8,11–13].

2. The equations

The density matrix for an individual free rotor can be evaluated using the following expression [7,13]:

$$\rho_{\text{rot}}^{t,t+1}(\beta/P) = \left\langle \omega^t \left| \exp\left(-\frac{\beta}{P}\hat{T}_{\text{rot}}\right) \right| \omega^{t+1} \right\rangle \tag{1}$$

where $\beta = 1/k_B T$, $\hat{T}_{\text{rot}}$ is the rotational kinetic energy operator and P is the number of beads. The eigenfunctions of the angular position $|\omega^{t+1}\rangle$ can be expanded in a basis set of the eigenfunctions of the top in question, in this case the asymmetric top $|JM\hat{K}\rangle$:

$$|\omega^{t+1}\rangle = \sum_{JM\hat{K}} \langle JM\hat{K}|\omega^{t+1}\rangle |JM\hat{K}\rangle. \tag{2}$$

Note that the integer $\hat{K}$ is not a true quantum number (i.e., it does not quantise any observable) it is simply a number used to label the $(2J + 1)$ possible values of the energy available for each value of J and M. This leads to [8]

$$\left\langle \omega^t \left| \exp\left(-\frac{\beta}{P}\hat{T}^{\text{rot}}\right) \right| \omega^{t+1} \right\rangle = \sum_{JM\hat{K}} \left\langle \omega^t |JM\hat{K}\right\rangle \exp\left(-\frac{\beta}{P}E_{\hat{K}}^{(JM)}\right) \left\langle JM\hat{K}|\omega^{t+1}\right\rangle. \tag{3}$$

The location of the laboratory frame used to define the Euler angles is arbitrary, and so we can choose a laboratory frame such that the Euler angles of bead t are zero (i.e., $\Omega^t = (\theta^t, \phi^t, \chi^t) = (0, 0, 0)$). This leads to:

$$|\omega^t\rangle = \delta(\Omega - \Omega^t) = \delta(\Omega). \tag{4}$$

A tilde is added to Ω (i.e., $\tilde{\Omega}$) in order to remind ourselves that the Euler angles are defined in a laboratory frame in which the Euler angles of bead t are zero. Thus $\tilde{\Omega}^{t+1}$ are the Euler angles of bead $t + 1$ in this arbitrary frame, i.e.

$$|\omega^{t+1}\rangle = \delta(\tilde{\Omega} - \tilde{\Omega}^{t+1}). \tag{5}$$

This expression can be further simplified by expanding the eigenfunctions of the asymmetric top $|JM\hat{K}\rangle$ in a basis set formed by the eigenfunctions of the symmetric top ($|JMK\rangle$):

$$|JM\hat{K}\rangle = \sum_K A_{\hat{K}K}^{JM} |JMK\rangle \tag{6}$$

where the eigenvectors $A_{\hat{K}K}^{JM}$ can be calculated solving a secular determinant for each value of J and M whose expression is given in Section 2.2.

The eigenfunctions of the symmetric top are given by [14]

$$\Psi_{JMK}(\theta, \phi, \chi) = \left(\frac{2J+1}{8\pi^2}\right)^{1/2} \exp(iM\phi) d_{MK}^J(\theta) \exp(iK\chi) \tag{7}$$

where $d_{MK}^J(\theta)$ represents Wigner's reduced d-matrix, which is given by [15–17]:

$$\begin{aligned} d_{MK}^J(\theta) &= [(J+M)!(J-M)!(J+K)!(J-K)!]^{1/2} \\ &\times \sum_\chi \frac{(-1)^\chi}{(J-M-\chi)!(J+K-\chi)!(\chi)!(\chi+M-K)!} \\ &\times [\cos(\theta/2)]^{(2J-2\chi+K-M)}[-\sin(\theta/2)]^{(2\chi+M-K)} \end{aligned} \tag{8}$$

where the sum over χ is restricted to those values that do not lead to negative factorials. The d-matrix is calculated in the subroutine `wignerCalc.f90`.

Using Eq. (6), together with Eq. (2) and the properties of the Wigner function (Eqs. (7) and (8)), the exact rotational propagator for rigid asymmetric tops, as a function of the number of beads (P), and the temperature (T), can be written:

$$\begin{aligned} \rho_{\text{rot}}^{t,t+1}(\beta/P, \tilde{\theta}, \tilde{\phi}, \tilde{\chi}) &= \sum_{J=0}^{\infty} \sum_{M=-J}^{J} \sum_{\hat{K}=-J}^{J} \left(\frac{2J+1}{8\pi^2}\right) A_{\hat{K}M}^{(JM)} \\ &\times \exp\left(-\frac{\beta}{P}E_{\hat{K}}^{(JM)}\right) \sum_{K=-J}^{J} A_{\hat{K}K}^{(JM)} d_{MK}^J(\tilde{\theta}) \\ &\times \cos(M\tilde{\phi} + K\tilde{\chi}) \end{aligned} \tag{9}$$

where $\beta = 1/k_B T$. $\rho_{\text{rot}}^{t,t+1}$ is calculated in the main section of the code provided; `NVM_propagator.f90`.

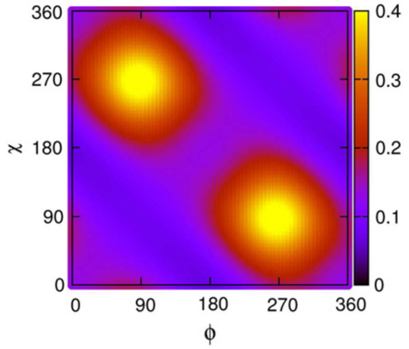


Fig. 1. Plot of the (ϕ, χ) plane for the normalised $PT = 120$ K propagator for $A = 100\ \text{cm}^{-1}$, $B = 10\ \text{cm}^{-1}$, $C = 1\ \text{cm}^{-1}$ at $\theta = 30°$. One can clearly see two mirror planes $(\phi = \chi)$ and $(\phi + \chi = 2\pi)$.

2.1. The energy

The contribution to the rotational energy of the interactions between beads t and $t+1$ is given by:

$$e_{\text{rot}}^{t,t+1}(\beta/P, \tilde{\theta}, \tilde{\phi}, \tilde{\chi}) = \frac{1}{\rho_{\text{rot}}^{t,t+1}} \sum_{JM\hat{K}} \left(\frac{2J+1}{8\pi^2}\right) A_{\hat{K}M}^{(JM)} E_{\hat{K}}^{(JM)} \times \exp\left(-\frac{\beta}{P} E_{\hat{K}}^{(JM)}\right) \sum_K A_{\hat{K}K}^{(JM)} d_{MK}^{J}(\tilde{\theta}) \times \cos(M\tilde{\phi} + K\tilde{\chi}) \qquad (10)$$

where $(\tilde{\theta}, \tilde{\phi}, \tilde{\chi})$ are the Euler angles of bead $t+1$ expressed in the body frame of bead t (see Appendix 3 of [13] for a complete description). $e_{\text{rot}}^{t,t+1}$ is also calculated in the main section of the code provided (`NVM_propagator.f90`).

2.2. The secular determinant

In the subroutine `secular_determinant.f90` we calculate the secular determinant. Using the so-called *bca* convention, i.e. the x axis is associated with the b axis of inertia, the y axis is associated with the c and the z axis is associated with the a axis of inertia, the energies $E_{\hat{K}}^{(JM)}$ of the asymmetric top and the coefficients $A_{\hat{K}K}^{(JM)}$ (i.e., the eigenvectors $A_{\hat{K}}^{(JM)}$) can be obtained solving the following secular determinant (see Refs. [18,19]) for each value of J and M:

$$H_{KK} = \frac{1}{2}(B+C)[J(J+1) - K^2] + AK^2 \qquad (11)$$

$$H_{KK\pm 2} = \frac{1}{4}(B-C)[J(J+1) - K(K\pm 1)]^{1/2} \times [J(J+1) - (K\pm 1)(K\pm 2)]^{1/2} \qquad (12)$$

where K ranges from $-J$ to $+J$. The remaining elements of the determinant are zero. A, B and C are the rotational constants (in cm^{-1}), where $A = \frac{\hbar}{4\pi c I_a}$, $B = \frac{\hbar}{4\pi c I_b}$ and $C = \frac{\hbar}{4\pi c I_c}$. Note that since $I_a \le I_b \le I_c$, it follows that $A \ge B \ge C$. This determinant has dimensions of $(2 \times J + 1)$. Therefore, $(2 \times J + 1)$ eigenvalues are obtained for each value of J and M, which are, in general, all different. These $(2 \times J + 1)$ energy levels are labelled with the $\hat{K}$ index. However, as M does not appear in the determinant, there is a $(2 \times J + 1)$ degeneracy in the energy associated with M.

We evaluate the secular determinant using the LAPACK routine `dsyev` (along with its corresponding dependencies) which computes all eigenvalues, and optionally, eigenvectors of a real, symmetric matrix [20]. The reliability of the LAPACK (Linear Algebra PACKage) is well established and it is freely-available software based upon work supported by the National Science Foundation.

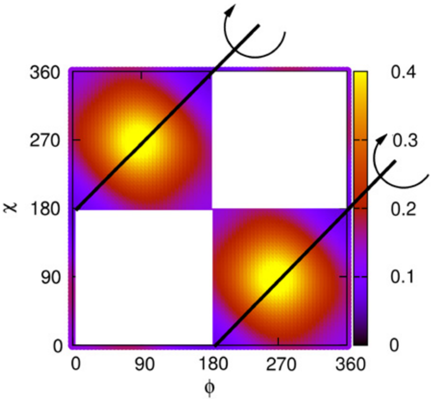


Fig. 2. Same plot as Fig. 1. Here one can see the mirror plane $(\chi = \pi + \phi)$.

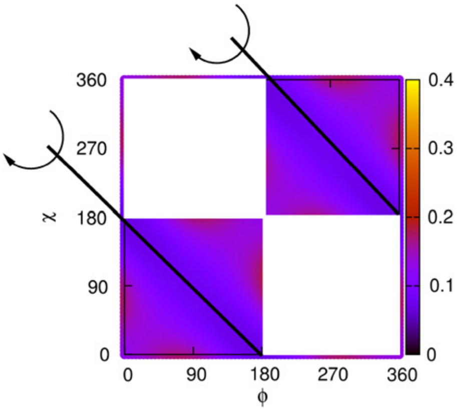


Fig. 3. Same plot as Fig. 1. Here one can see the mirror plane $(\chi = \pi - \phi)$.

2.3. The mass factor λ

The "mass factor" (λ) is a numerical value that scales the masses of the atoms of the molecule by a factor λ, thus scaling A, B and C by A/λ, B/λ and C/λ. This has been implemented to facilitate the mass integration which is used if one wishes to calculate the free energy (see Section 2.2 of [12] for a detailed explanation).

3. Time saving tricks

Calculating the whole propagator $(\theta(0 < \pi), \phi(0 < 2\pi)$ and $\chi(0 < 2\pi))$ is time consuming. Calculation over a 1° grid results in over 23 million angles that would each need to be evaluated. To reduce this we make use of the following "time-saving tricks":

3.1. Symmetry

Visual inspection of the (ϕ, χ) plane the propagator (Fig. 1) indicates the following four mirror planes: (1) $(\phi = \chi)$ (2) $(\phi + \chi = 2\pi)$ along with (3) $(\chi = \pi + \phi)$ (see Fig. 2) and finally (4) $(\chi = \pi - \phi)$ (see Fig. 3).

The use of symmetry reduces the number of points to be initially calculated to 1/8 of the original number. One only needs to calculate one "triangle" from Fig. 2 along with one from Fig. 3. Of the 16 possible combinations we have chosen the two triangles shown in Fig. 4 due to the simplicity of the DO LOOPS required to cover them.

We only write the reduced area to disk to save space. One can then "unfold" the propagator using the sample code `QuantumMatrixRead.f`, following the scheme drawn in Fig. 5, into an array for use within a path integral Monte Carlo code.

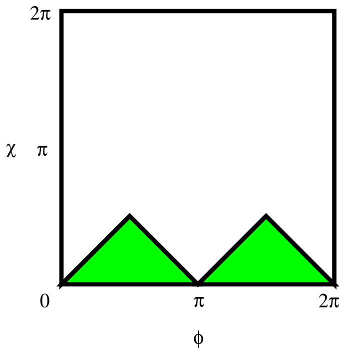


Fig. 4. The irreducible area the (ϕ, χ) plane to be calculated for each value of θ, covering 1/8 th of the total area.

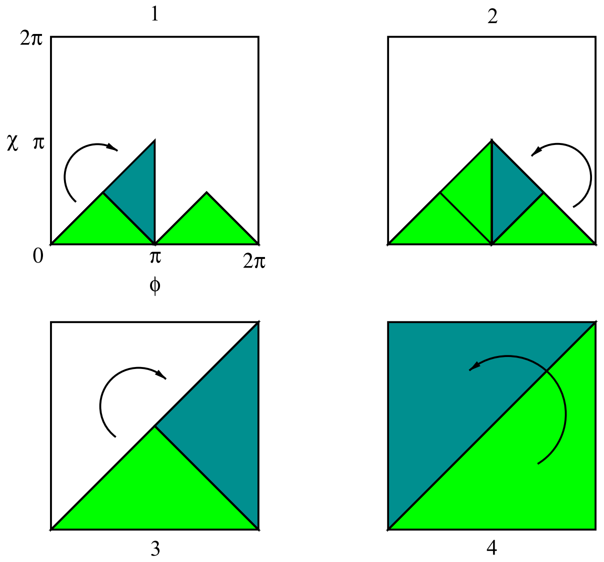


Fig. 5. The four operations that we used to "unfold" the irreducible area to produce the whole propagator, implemented in the subroutine `QuantumMatrixRead.f`.

3.2. Sparse

We have observed that for a typical propagator for water, for $PT > 1000$ K, around 95% of the propagator is almost zero (less than 1 part in 10,000 when compared to the maximum at $\rho(0, 0, 0)$), i.e. the propagator is sparse. Also, we assume without proof that:

- The propagator for a given PT and mass has a global maximum at $\rho(0, 0, 0)$.
- One can see from Eq. (9) that as the value of PT increases, the propagator can only be $\leq$ to a lower PT. Thus the peaks in the propagator shrink and become narrower. i.e. as the temperature increases, the propagator becomes less quantum and more classical; in the classical limit the propagator is zero for all angles, with a Dirac delta function at $\rho(0, 0, 0)$. For example, in Fig. 6 we see that on going from $PT = 1000$ K to $PT = 2000$ K the peaks become narrower, the significant data-points are in the same area, and their number becomes reduced. This allows us to use a lower PT angle set to "seed" a higher PT calculation.
- As the mass of the molecule increases, the peaks in the propagator shrink and become narrower, i.e. as the mass increases, the propagator becomes less quantum and more classical, again in the classical limit the propagator is zero for all angles, with a Dirac delta function at $\rho(0, 0, 0)$. For example, in Fig. 7 we see that on going from $\lambda = 1$–1.2 to $\lambda = 2$ at $PT = 1500$ K the peaks become narrower, the significant data-points are in the same area, and their number becomes reduced. This allows us to use the set of angles from a lower mass to "seed" a higher mass calculation.
- The energy, e_{rot}, mimics these trends found for ρ.

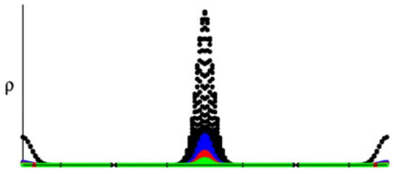


Fig. 6. Comparison of $PT = 1000$ K (black), 1500 K (blue), 1750 K (red) and 2000 K (green) for TIP4PQ/2005 water (for $\rho(0, 0, 0)$ normalised to 1) at $\theta = 30°$ and viewed along the $(\phi+\chi = 2\pi)$ mirror plane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

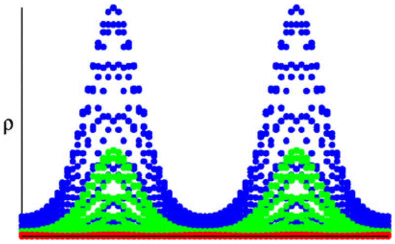


Fig. 7. Comparison of $\lambda = 1$ (blue), 1.2 (green) and $\lambda = 2$ (red) for $PT = 1500$ K for TIP4PQ/2005 water (for $\rho(0, 0, 0)$ normalised to 1) at $\theta = 30°$ and viewed along the $(\phi = \chi)$ mirror plane. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

To make use of this we first, say, calculate the propagator for all of the data points at a modest PT and mass (say $PT = 1400$ K and having a mass "factor" of $\lambda = 1$). This is time consuming, but necessary to provide a starting point (for example, the $PT = 1400$ K $\lambda = 1$ propagator for TIP4PQ/2005 water takes about 207 h to calculate[1]). Once calculated we only write out the angles for which the values of the propagator are significant. If one now wishes to calculate a new propagator at, say, $PT = 1497$ K, we read in the set of significant angles found for $PT = 1400$ K and only re-calculate this reduced set of angles, reducing the calculation time to 2–3 h.

To arrive at a particular value of PT and mass factor one could first calculate the new value of PT then subsequently calculate the new value of the mass factor, or *vice versa*. In our experience the PT-mass route takes about the same time as the mass-PT route, leading to the same propagator.

3.3. Convergence

The main summation loop in Eqs. (9) and (10) range from $J = 0$ to $J = \infty$. However, the sum converges for a finite number of J, so the calculation is only performed up to a certain $J (J_{\max})$. Each successive J the following equation is evaluated

$$\texttt{cebolla} = \frac{1}{N_{\text{angles}}} \sum_{\text{angles}} \left(\frac{\rho_J(\theta, \phi, \chi)}{\rho_J(0, 0, 0)} - \frac{\rho_{J-1}(\theta, \phi, \chi)}{\rho_{J-1}(0, 0, 0)} \right)^2 \qquad (13)$$

when `cebolla` is less than a given cut-off (here we have used 1×10^{-12}) the loop over J is stopped. In Eq. (13) ρ_J represents the propagator evaluated when adding all terms up to J. Given that the value `cebolla` is a function of the grid used, we have plotted an example of the convergence of $\rho(0, 0, 0)$ with respect to J in Fig. 8. Negative values of the propagator are unphysical, however,

[1] Calculations were performed on an Intel® Xeon® CPU X5680 running at 3.33 GHz.

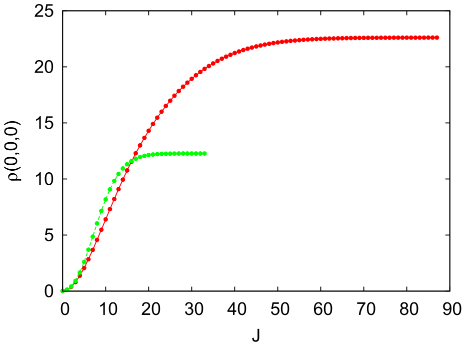


Fig. 8. Convergence plot of $\rho(0, 0, 0)$ with respect to J. Red line-points: $A = 100$ cm^{-1}, $B = 10$ cm^{-1} and $C = 1$ cm^{-1}, green line-points TIP4PQ/2005 water ($PT = 1500$ K). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

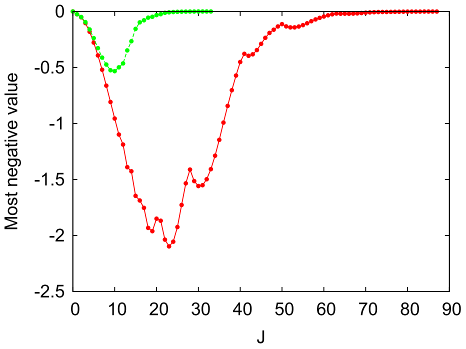


Fig. 9. Convergence plot of the most negative value with respect to J. Red line-points: $A = 100$ cm^{-1}, $B = 10$ cm^{-1} and $C = 1$ cm^{-1}, green line-points TIP4PQ/2005 water ($PT = 1500$ K). Points calculated on a 3° grid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

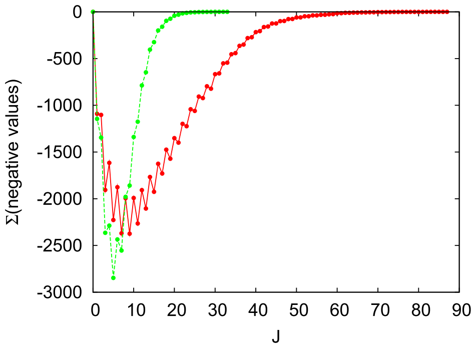


Fig. 10. Convergence plot of the sum of all of the negative values with respect to J. Red line-points: $A = 100$ cm^{-1}, $B = 10$ cm^{-1} and $C = 1$ cm^{-1}, green line-points TIP4PQ/2005 water ($PT = 1500$ K). Points calculated on a 3° grid. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

until the propagator has converged negative values do arise. For example, in Figs. 8–10 we plot the convergence of $\rho(0, 0, 0)$ with respect to J, the most negative value, and the global sum of negative values with respect to J for two systems. Convergence information is written to the file `convergence`. It is interesting to note an *odd–even* effect with respect to J in the plot of the sum of the negative values (Fig. 10).

4. Source code and compilation

The `.tar` file contains the main program, `NVM_propagator.f90` which uses the MODULEs `wigner` and `secular`. It also makes use of the LAPACK driver routine `DSYEV`, which computes all eigenvalues and, optionally, eigenvectors of a real symmetric matrix. `DSYEV` and all of its dependencies are included in the file `dsyev_source.f`. A sample `Makefile` is included, which uses the `ifort` compiler in conjunction with Intel's Math Kernel Library. In general the program is compiled with the $-r8$ flag, however, the MODULE wigner is compiled with $-r16$. The function `factorial` within the MODULE wigner is a `REAL` function rather than `INTEGER` because as an integer it would be limited to a maximum factorial of 20! on a 64-bit machine. `wignerDmatrix` should be compiled in quadruple precision ($-r16$) to allow factorials greater than 170! (which is the limit imposed if one compiles with $-r8$) and so permit us to go beyond $J = 85$ if necessary. If one compiles this subroutine with $-r16$ one can arrive at 1754! which corresponds to a $J_{\max}$ of 876 which, in our experience, is more than enough for any propagator of practical interest. A version of the source code suitable for compilation using the GNU gfortran compiler is also included.

5. Input

The following is an example `input_file`:

```
#PT
2205.
# mass factor
1.
# A (cm-1)
15.39
# B (cm-1)
7.36
# C (cm-1)
4.98
```

The second line is the value of $P \times T$. The fourth line is the mass factor (λ), followed by the values of A, B and C in units of inverse centimetres. If no `NVM_propagator_old` file is present, the propagator will be calculated from scratch. If a `NVM_propagator_old` is present, then the propagator will be evaluated only for the angles supplied in the old propagator file.

6. Output

A finished run produces the following files: of principal interest is the propagator, consisting of $\rho_{\text{rot}}^{t,t+1}$ and $e_{\text{rot}}^{t,t+1}$ (in units of cm^{-1}), which is written to the file `NVM_propagator_new`. During a run the progress of the calculation is written to the file `convergence`, and a dump of the current J value of the propagator is written to `NVM_propagator_Jdump` (which is not normalized).

The `NVM_propagator_new` file only contains the 1/8th of the propagator, i.e. the region shown in Fig. 4. A sample subroutine, `QuantumMatrixRead.f` is provided to "unfold" the propagator for use in subsequent simulations. Note, `QuantumMatrixRead.f` has been extracted from our own in-house simulation code, and will need to be adapted for the individual user, and is only included as a guide. The value of the propagator for any particular $\hat{\theta}$, $\hat{\phi}$ and $\hat{\chi}$ can then estimated using an interpolation algorithm.

A sample file of the $PT = 1497$ K propagator for $\lambda = 1$ for water is provided, which was calculated from the $PT = 1400$ K propagator, also provided.

7. Monte Carlo simulations

Moves in a typical path integral Monte Carlo simulation algorithm consists of the following steps: (1) a bead s of a molecule i is chosen randomly (the old configuration is represented by o), (2) this bead s of molecule i is either translated or rotated randomly (the new configuration is represented by n), (3) the following quantity is evaluated

$$W = W_{\text{rot},i} \times W_{\text{tras},i} \times W_{\text{pot}} \qquad (14)$$

where the subindex *i* indicates that the propagator is evaluated for molecule *i*, and

$$W_{\mathrm{rot},i} = \frac{\prod_{t=1}^{P} \rho_{\mathrm{rot},i}^{t,t+1}(n)}{\prod_{t=1}^{P} \rho_{\mathrm{rot},i}^{t,t+1}(o)}, \tag{15}$$

$$W_{\mathrm{tras},i} \; = \; \exp\left(-\beta \frac{MP}{2\beta^2\hbar^2}\left(\sum_{t=1}^{P}(\mathbf{r}_i^t(n) - \mathbf{r}_i^{t+1}(n))^2 - \sum_{t=1}^{P}(\mathbf{r}_i^t(o) - \mathbf{r}_i^{t+1}(o))^2\right)\right) \tag{16}$$

where *M* is the mass of the molecule and

$$W_{\mathrm{pot}} = \exp\left(-\beta\left(U(n) - U(o)\right)/P\right) \tag{17}$$

and finally (4) the movement is accepted with probability:

$$\mathrm{accept}(o \to n) = \min(1, W). \tag{18}$$

In Eq. (16) $\mathbf{r}_i^t(o)$ and $\mathbf{r}_i^t(n)$ represent the Cartesian coordinates of the centre of mass of bead *t* of molecule *i* in the configuration before and after the movement respectively. It is also possible to introduce additional movement attempts, such as the translation or the rotation of a whole ring, so that the configurational space is sampled more quickly (see Refs. [10,13]). When simulations are performed in the *NpT* ensemble, trial moves that attempt to change the volume are also incorporated (for more details see Ref. [10]). The FORTRAN code presented here provides the values of the rotational propagator $\rho_{\mathrm{rot},i}^{t,t+1}$ for a grid of values of (θ, ϕ, χ) which can be read in at the beginning of the Monte Carlo code and used to rapidly evaluate Eq. (15) when needed in the Monte Carlo simulation by way of an interpolation routine. The rotational energy of the system can also be evaluated in a Monte Carlo simulation by averaging the estimator of the energy, given by Eq. (10), and also obtained in tabulated form as a function of the angles as an output in the code provided here. Note that for each temperature *T* and number of beads *P* a new propagator has to be generated with the executable `NVM.x`. For a more detailed description about the path integral Monte Carlo simulations using the rotational propagator see Refs. [10,13].

Acknowledgments

This work was funded by grants FIS2010-16159 and FIS2010-15502 of the Dirección General de Investigación and S2009/ESP-1691 (MODELICO) of the Comunidad Autónoma de Madrid.

References

[1] R.P. Feynman, A.R. Hibbs, Path-Integrals and Quantum Mechanics, McGraw-Hill, 1965.
[2] J.A. Barker, J. Chem. Phys. 70 (1979) 2914.
[3] D. Chandler, P.G. Wolynes, J. Chem. Phys. 74 (1981) 4078.
[4] R.A. Kuharski, P.J. Rossky, Chem. Phys. Lett. 103 (1984) 357.
[5] L.H. de la Peña, P.G. Kusalik, J. Chem. Phys. 121 (2004) 5992.
[6] M.H. Müser, B.J. Berne, Phys. Rev. Lett. 77 (1996) 2638.
[7] D. Marx, M.H. Müser, J. Phys.: Condens. Matter. 11 (1999) R117.
[8] E.G. Noya, C. Vega, C. McBride, J. Chem. Phys. 134 (2011) 054117.
[9] R. Kumar, F.-F. Wang, G.R. Jenness, K.D. Jordan, J. Chem. Phys. 132 (2010) 014309.
[10] C. McBride, C. Vega, E.G. Noya, R. Ramírez, L.M. Sesé, J. Chem. Phys. 131 (2009) 024506.
[11] C. McBride, J.L. Aragones, E.G. Noya, C. Vega, Phys. Chem. Chem. Phys. 14 (2012) 15199.
[12] C. McBride, E.G. Noya, J.L. Aragones, M.M. Conde, C. Vega, Phys. Chem. Chem. Phys. 14 (2012) 10140.
[13] E.G. Noya, L.M. Sesé, R. Ramírez, C. McBride, M.M. Conde, C. Vega, Mol. Phys. 109 (2011) 149.
[14] I.N. Levine, Molecular Spectroscopy, John Wiley & Sons Inc., 1975.
[15] E.P. Wigner, Gruppentheorie Und Ihre Anwendungen Auf Die Quantenmechanik Der Atomspektren, Vieweg Verlag, Braunschweig, 1931.
[16] C.G. Gray, K.E. Gubbins, Theory of Molecular Fluids, Clarendon Press, Oxford, 1984.
[17] M.E. Rose, Elementary Theory of Angular Momentum, John Wiley & Sons, 1967.
[18] R.N. Zare, Angular Momentum: Understanding Spatial Aspects in Chemistry and Physics, John Wiley & Sons Inc., 1988.
[19] P. Jensen, P.R. Bunker, Molecular Symmetry and Spectroscopy, second ed., NRC Research Press, 1998.
[20] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.