

A Guide for the Practical Work

Mario Barbatti

Aix Marseille University, CNRS, ICR, Marseille, France

Institut Universitaire de France, 75231, Paris, France

mario.barbatti@univ.amu.fr; www.barbatti.org

Abstract. This document is a short guide for the practical work in the [2023 Molecular Modeling course](#). It details the goals and procedures for TP3, TP4, and TP5.

Introduction

Practical works TP3, TP4, and TP5 aim to help the students develop basic coding skills for molecular modeling. This document discusses the theory that will be employed. We will write, test, and employ a Python program to run molecular dynamics on a model potential energy surface (PES).

These tasks will be accomplished through three parts:

- In the first part (TP3), we will implement a code to compute the adiabatic Spin-Boson PES discussed in masterclass CM2.
- In the second part (TP4), we will implement a code to run molecular dynamics based on the Velocity Verlet algorithm studied in masterclass CM1.
- Finally, in the third part (TP5), we will implement the Andersen thermostat introduced in masterclass SM1.

TP3: PES modeling

The adiabatic Spin-Boson Hamiltonian (SBH),¹ is one of the most used models in quantum mechanics. It represents a two-state system (like spin 1/2) coupled to an environment of harmonic oscillators (the *bosons*). This model allows the simulation of different phenomena, from a photon decay within a cavity to a molecule undergoing decoherence.

In the SBH model, the potential energy surfaces are given by a set of simple analytical functions. Thus, it delivers a simple approximation for the energies, energy gradients, and coupling without solving the Schrödinger equation, making it very fast.

For a two-state system coupled to N harmonic oscillators, the SBH model gives two adiabatic potential energy surfaces. The ground-state ($i = 1$) and excited-state ($i = 2$) potential energies are

$$E_i = \frac{1}{2} \sum_{j=1}^N M_j \omega_j^2 R_j^2 + (-1)^i \left[\eta^2 + v_0^2 \right]^{1/2} \quad (i=1,2) \quad (1)$$

where

$$\eta = \left(\sum_{j=1}^N g_j R_j + \varepsilon_0 \right) \quad (2)$$

In these equations, the nuclear coordinates are $\mathbf{R} = (R_1, R_2, \dots, R_N)$. The specific values of the SBH model are determined by several parameters ε_0 , M_i , ω_i , g_i , and ν_0 . Their values are discussed later.

The potential energy gradient of the two adiabatic states is

$$\frac{\partial E_i}{\partial R_k} = M_k \omega_k^2 R_k + (-1)^i g_k \left[\frac{\eta}{[\eta^2 + \nu_0^2]^{1/2}} \right] \quad (i=1,2; \quad k=1 \dots N) \quad (3)$$

Task 1. Create a notebook (Jupyter or Colab) and implement a Python code to compute the adiabatic SBH model's energy and the energy gradient for the lowest state ($i = 1$) in one dimension ($N = 1$).

- The code must contain two functions. The first function takes the geometry as input and gives the energy as output. The second function takes the geometry as input and outputs the energy gradient.
- The code must be fully documented.
- The equations that you implemented in the functions must be shown as markdown (LaTeX).

Task 2. Use your code to compute energy and gradients for the parameters given in Table 1.

- Plot the graph of energy x geometry.
- Plot the graph of energy gradient x geometry.
- Play with the parameters. How do the graphs change when you change the parameters? What is the effect of each one?

Tip: Work in [Hartree atomic units](#).

Task 3. Determine approximately (R_1, R_2) for the two minima.

Tip. You can find the approximate minima by plotting $E \times R$ in a small region around the minima. You can also plot the energy gradient and check the values of R_1 that have a gradient equal to zero.

Table 1. Parameters for the Spin-Boson model.

Parameter	Value (atomic units)
ε_0	0.03674933
ν_0	0.03674933
M_1	1836
ω_1	0.01136364
g_1	0.22

TP4: Molecular dynamics (TP4)

One of the most popular algorithms to integrate Newton's second law is the Velocity Verlet²

$$\begin{aligned}\mathbf{R}_\alpha(t + \Delta t) &= \mathbf{R}_\alpha(t) + \mathbf{v}_\alpha(t)\Delta t + \frac{1}{2}\mathbf{a}_\alpha(t)\Delta t^2 \\ \mathbf{v}_\alpha(t + \Delta t) &= \mathbf{v}_\alpha(t) + \frac{1}{2}(\mathbf{a}_\alpha(t) + \mathbf{a}_\alpha(t + \Delta t))\Delta t\end{aligned}\quad (4)$$

where

$$\mathbf{a}_\alpha(t) = \frac{1}{M_\alpha} \left(-\frac{\partial E(\mathbf{R}(t))}{\partial \mathbf{R}_\alpha} + \mathbf{F}^{(e)} \right) \quad (5)$$

When these equations are applied to molecular dynamics, α represents each nuclei of mass M_α at Cartesian position $\mathbf{R}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$, Cartesian velocity $\mathbf{v}_\alpha = (v_{x,\alpha}, v_{y,\alpha}, v_{z,\alpha})$, and Cartesian acceleration $\mathbf{a}_\alpha = (a_{x,\alpha}, a_{y,\alpha}, a_{z,\alpha})$. Δt is the integration step. In Eq. (5), the first term in the parenthesis is the potential energy gradient, and $\mathbf{F}^{(e)}$ are external forces. The algorithm, including a total energy conservation check, is illustrated in Figure 1.

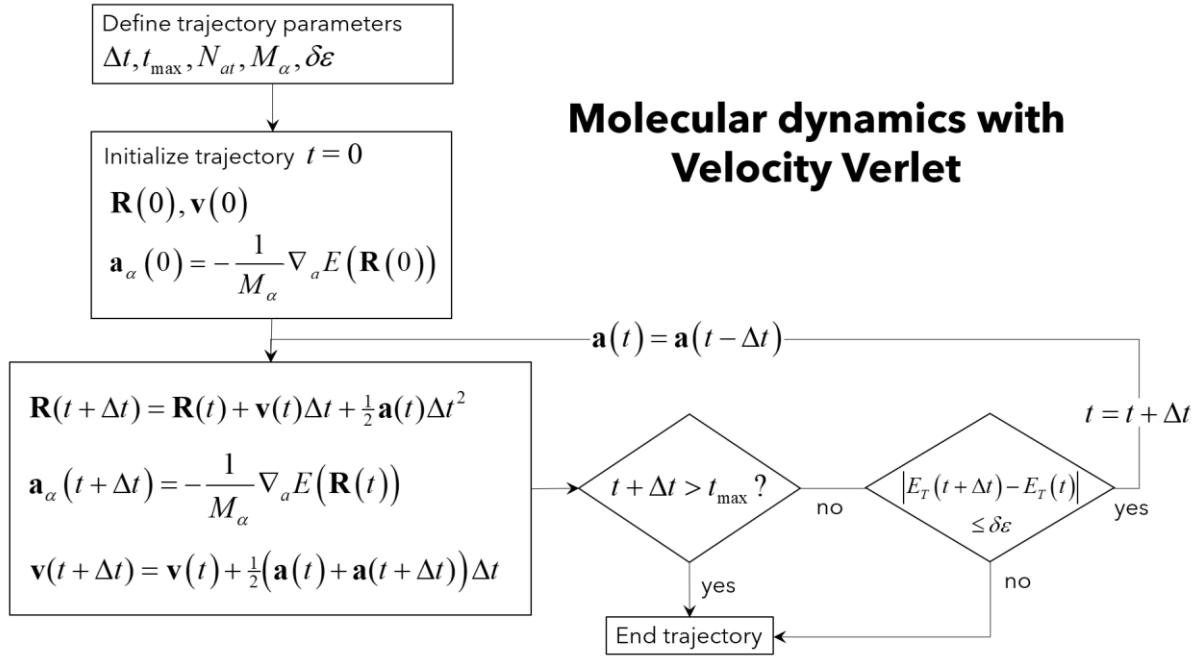


Figure 1. Schematic illustration of the Velocity Verlet algorithm.

Task 4. Implement a 1D Velocity Verlet algorithm for a single coordinate and without external forces. In this case, Eqs (4) and (5) simplify to

$$\begin{aligned}x(t + \Delta t) &= x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 \\ v(t + \Delta t) &= v(t) + \frac{1}{2}(a(t) + a(t + \Delta t))\Delta t\end{aligned}\quad (6)$$

and

$$a(t) = -\frac{1}{M} \frac{dE(x(t))}{dx} \quad (7)$$

where x is the position, v is the velocity, a is the acceleration, and M is the mass.

Tip. Make a copy of the Notebook you created in TP3 and work on it.

Tip. Keep working in atomic units.

The total energy at time t is

$$E_T(t) = E_K(t) + E(t) \quad (8)$$

where the kinetic energy is

$$E_K(t) = \frac{1}{2} M v(t)^2 \quad (9)$$

The total energy is conserved if

$$|E_T(t) - E_T(t - \Delta t)| \leq \delta\epsilon \quad (10)$$

where $\delta\epsilon$ is a parameter of the dynamics. Use $\delta\epsilon = 0.004$ au.

Tip. Your code will be better if you create separate functions to compute x , v , a , E_K , and E_T .

Task 5. Run 1-ps (How much is that in atomic units of time?) dynamics on the E_1 adiabatic PES given by the 1D model of Table 1. Start at a minimum of the left well (you found it in Task 3). Use a total energy of approximately 0.1 Hartree. (If the system is at the minimum, the total energy equals the kinetic energy. Then, what's the initial velocity?) Is the total energy conserved during your trajectory? If not, your program has either a bug or you used a too-large Δt . (A reasonable value is $\Delta t = 20$ au.)

Tip. To compute the force (the negative of the derivative of the potential energy) and the potential energy (needed to check the total energy conservation), use the functions that you implemented in TP3.

Task 6. Run the 1D model with different values of total energy, starting with all kinetic energy in R_2 . What minimum energy is needed for the system to jump to the right well?

Tip. You can easily see the jump if you plot R_1 as a function of time.

TP5: Thermostat

In the previous TP, you ran microcanonical dynamics (constant energy). Now, let us run canonical dynamics (constant temperature). To do so, you should implement the Andersen thermostat.³ The algorithm is the following:

- Define a collision frequency Γ (suggestion: $\Gamma = 0.002$ au)
- Integrate dynamics in one step Δt with Velocity Verlet
- For each atom, sample a random number r_0

If $r_0 > \Gamma \Delta t$, do not change the velocity

If $r_0 \leq \Gamma \Delta t$, change the velocity as

$$v^{(new)} = r_x \sqrt{\frac{k_B T}{M_\alpha}} \quad (11)$$

where r_x is another random number and k_B is the Boltzmann constant.

Both r_0 and r_x are random numbers sampled from a Gaussian distribution of unit variance. They are not uniform random numbers!

Task 7. Implement the Andersen thermostat.

Task 8. Run dynamics of the 1D Spin Boson model at 300 K on the E_1 adiabatic PES given in Table 1. Start at the left minimum with zero kinetic energy. How long does the trajectory take to reach 300 K? Have you seen jumps to the right well?

Tip. The temperature can be estimated as⁴

$$T \approx \frac{2 \langle E_{kin} \rangle}{N k_B}$$

where $\langle E_{kin} \rangle$ is the mean kinetic energy and N is the number of dimensions. Thus, if you compute the moving average kinetic energy over the last, say, 200 integration steps, you can get T as a function of time.

Task 9. Estimate which temperature corresponds to the minimum total energy to have jumps you got in Task 6. (It should be in tens of thousands of Kelvin!) Run dynamics at this temperature. Do you see jumps?

References

(1) Chen, H.-T.; Reichman, D. R. On the Accuracy of Surface Hopping Dynamics in Condensed Phase Non-Adiabatic Problems. *J. Chem. Phys.* **2016**, *144*, 094104.

<https://doi.org/10.1063/1.4942867>

(2) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **1982**, *76*, 637-649.

<https://doi.org/10.1063/1.442716>

(3) Lowe, C. P. An alternative approach to dissipative particle dynamics. *Europhys. Lett.* **1999**, *47*, 145.
<https://doi.org/10.1209/epl/i1999-00365-x>

(4) Barbatti, M. Defining the temperature of an isolated molecule. *J. Chem. Phys.* **2022**, *156*, 204304.
<https://doi.org/10.1063/5.0090205>