Homework 4: Introduction to Molecular Simulation

CHEM4050/5050 Fall 2025

Tuesday, October 14, 2025, at 11:59 PM Central

Contents

1 Numerical Computation of Work in Thermodynamic Processes

In this problem, you will numerically compute the work done on an ***ideal gas*** undergoing various thermodynamic processes. The work is defined as

$$w = -\int P(V)\,dV \tag{1}$$

where $P(V)$ is the pressure as a function of volume V, and the integral is taken over the volume change in the process. For this assignment, consider a sample of an ***ideal gas*** undergoing two processes:

1. **Isothermal Expansion**: The ***ideal gas*** expands isothermally from an initial volume V_i to a final volume V_f at a constant temperature T (in Kelvin).
2. **Adiabatic Expansion**: The ***ideal gas*** expands adiabatically from an initial volume V_i to a final volume V_f, with no heat exchange.

You will compute the work done on the ***ideal gas*** in both processes. The ***ideal gas*** follows the ***ideal gas law***

$$PV = nRT \tag{2}$$

where n is the number of moles of ***ideal gas***, and R is the ***ideal gas*** constant. Additionally, for the adiabatic process, the ***ideal gas*** follows the equation

$$PV^\gamma = \text{constant} \tag{3}$$

where γ is the adiabatic index, defined as the ratio C_P/C_V of the heat capacity at constant pressure C_P and volume C_V.

1.1 Tasks

1.1.1 Isothermal Process

- ☐ Write a Python function named `compute_work_isothermal.py` to compute the work done during an isothermal expansion from V_i to V_f.
- ☐ Use `scipy.integrate.trapezoid` to compute the work. The pressure in this case is given by the ***ideal gas law***.

$$w_{\text{iso}} = -\int_{V_i}^{V_f} \frac{nRT}{V}\,dV \tag{4}$$

  where "iso" is an abbreviation for "isothermal."

1.1.2 Adiabatic Process

- ☐ Write a Python function named `compute_work_adiabatic.py` to compute the work done during an adiabatic expansion. Use the relation $P(V) = \text{constant}/V^\gamma$ to compute the pressure as a function of volume.

$$w_{\text{adi}} = -\int_{V_i}^{V_f} P(V)\,dV \tag{5}$$

  where "adi" is an abbreviation for "adiabatic."

- ☐ Compute the constant from the initial conditions and perform numerical integration to find the work.

1.1.3 Comparison

- ☐ Plot the work done in both processes as a function of the final volume V_f for values of V_f between V_i and $3V_i$.
- ☐ Assume the following parameters for the ***ideal gas***

Parameter	**Value**	**Units**
n	1	mol
R	8.314	J/mol-K
T	300	K
V_i	0.1	m^3
γ	1.4	N/A

Table 1

1.1.4 Discussion

- ☐ Briefly discuss the results. How does the work done in an isothermal process compare with the work done in an adiabatic process for the same volume change?

1.2 Submission Guidelines

- ☐ Push Python codes named `compute_work_isothermal.py` and `compute_work_adiabatic.py` implementing the functions to compute the work for both processes to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-1`.
- ☐ When executed, these Python codes should
  - ☐ Use `scipy.integrate.trapezoid`
  - ☐ Use the parameters in Table 1
  - ☐ Write a `csv` file containing work *vs.* final volume
- ☐ Push a Python code named `plot_work.py` plotting the work done as a function of final volume for both processes to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-1`.
- ☐ Push a Markdown file named `work_discussion.md` containing your discussion.
- ☐ Include clear comments in your code, explaining each key step.
- ☐ Ensure your plots are well-labeled and properly formatted.

2 Thermodynamic Properties of Ce(3) in Different Environments

In this problem, you will construct the partition function and compute the thermodynamic properties (internal energy, free energy, and entropy) of the 4f electron in Ce^{3+} using the following three cases

1. **Isolated Ce^{3+}** (no spin-orbit coupling or crystal field splitting): The 14-fold degenerate 4f states are equally populated. In this case, the probability of occupying each state is $p_i = \frac{1}{14}$, and the energy of all states is zero.
2. **Ce^{3+} with Spin-Orbit Coupling (SOC)**: The 14-fold degenerate 4f states split into two groups, a lower-energy $^2F_{5/2}$ level with 6-fold degeneracy and a higher-energy $^2F_{7/2}$ level with 8-fold degeneracy, separated by $\Delta E = 0.28\,\text{eV}$ from $^2F_{5/2}$.
3. **Ce^{3+} with Both SOC and Crystal Field Splitting (CFS)**: The $^2F_{5/2}$ and $^2F_{7/2}$ levels further split into five distinct energy levels (as shown in the provided figure). Each level has a specific degeneracy.

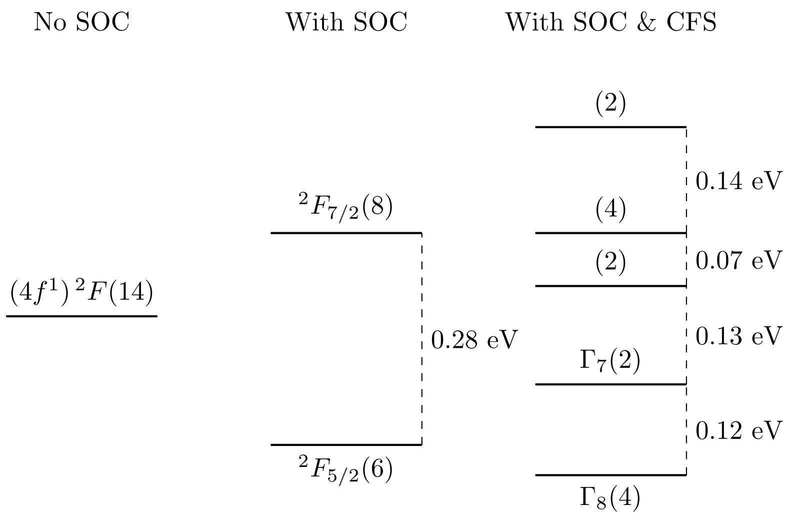

Figure 1: Energy level splitting of Ce^{3+} 4f states in three different environments: No SOC (left), With SOC (middle), and With SOC & CFS (right).

The thermodynamic properties will be computed using the partition function Z, which is given by

$$Z = \sum_i g_i e^{-E_i/k_\text{B}T} \tag{6}$$

where g_i is the degeneracy of state i, E_i is the energy of state i (relative to the ground state), and k_B is the Boltzmann constant. The thermodynamic properties can be computed as follows

Name	**Symbol**	**Expression**
Internal Energy	U	$-\frac{\partial \ln Z}{\partial \beta}$
Free Energy	F	$-k_\text{B}T\ln Z$
Entropy	S	$-\frac{\partial F}{\partial T}$

Table 2

2.1 Tasks

2.1.1 Isolated Ce(3)

- ☐ Write a Python function to compute the partition function for an isolated Ce^{3+} ion with 14-fold degeneracy, where all states have zero energy.
- ☐ Compute the thermodynamic properties (internal energy, free energy, and entropy) for this case.

2.1.2 Ce(3) with SOC

- ☐ Write a Python function to compute the partition function for Ce^{3+} with SOC. The 14 states split into $^2F_{5/2}$ (6 states) and $^2F_{7/2}$ (8 states), with a 0.28 eV energy difference between the two levels. ***Set the reference energy $\boldsymbol{E = 0}$ to the $\boldsymbol{^2F_{5/2}}$ level.***
- ☐ Use the partition function to compute the thermodynamic properties for this case.

2.1.3 Ce(3) with SOC and CFS

- ☐ Based on the figure, split the $^2F_{5/2}$ and $^2F_{7/2}$ levels into five individual energy levels. ***Set the reference energy $\boldsymbol{E = 0}$ to the $\boldsymbol{\Gamma_8}$ level.***
- ☐ Write a Python function to compute the partition function for Ce^{3+} in the presence of both SOC and CFS.
- ☐ Use the partition function to compute the thermodynamic properties for this case.

2.1.4 Comparison

- ☐ Compute and compare the partition functions and the thermodynamic properties (internal energy, free energy, and entropy) for all three cases.
- ☐ Provide plots of the thermodynamic properties as a function of temperature (from 300 K to 2000 K).

2.2 Submission Guidelines

- ☐ Push a Python code named `ce_thermo.py` that constructs the partition function and computes the thermodynamic properties for all three cases to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-2`.
- ☐ When executed, this Python code should
  - ☐ Use `scipy.constants.k`
  - ☐ Use the degeneracies and energies in Figure 1
  - ☐ Write a `csv` file containing internal energy, free energy, and entropy *vs.* temperature for all three cases
- ☐ Write plots of internal energy, free energy, and entropy as a function of temperature for the three cases to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-2`.
- ☐ Include clear comments in your code, explaining each key step.
- ☐ Ensure your plots are well-labeled and properly formatted.

3 Graduate Supplement

In this problem, you will calculate the temperature at which a dimer of Lennard-Jones (LJ) particles in a ***cubic*** box dissociates by computing the classical partition function, C_V, and determining the temperature at which C_V reaches its maximum value. To do this, you will

1. Compute the configurational partition function for two LJ particles using the trapezoidal rule for numerical integration.
2. Use the partition function to compute the thermodynamic quantities, including C_V.
3. Determine the temperature at which C_V reaches its maximum, which corresponds to the dissociation temperature.

The classical partition function for a system of N particles can be written as

$$\begin{aligned}
Z &= \frac{1}{h^{3N}} \int dx_1 \cdots \int dz_N \int dp_1^{(x)} \cdots \int dp_N^{(z)}\, e^{-\beta H(x_1,\ldots,z_N,p_1^{(x)},\ldots,p_N^{(z)})} && (7)\\
&= \frac{1}{h^{3N}} \left[\int dx_1 \cdots \int dz_N\, e^{-\beta V(x_1,\ldots,z_N)}\right] \left[\int dp_1^{(x)} \cdots \int dp_N^{(z)}\, e^{-\beta T(p_1^{(x)},\ldots,p_N^{(z)})}\right] && (8)\\
&= \frac{1}{h^{3N}} \frac{1}{\lambda^{3N}} \left[\int dx_1 \cdots \int dz_N\, e^{-\beta V(x_1,\ldots,z_N)}\right] && (9)
\end{aligned}$$

where H is the classical Hamiltonian, $(x_1, \ldots, z_2, p_{x,1}, \ldots, p_{z,2})$ represents the Cartesian components of the positions and momenta of particles 1 and 2, h is the Planck constant, and $\lambda = \sqrt{\beta h^2/2\pi m}$ is often referred to as the thermal wavelength of a particle. Since H is separable in the each of the N coordinates and momenta, the partition function can be simplified from Equation 7 to 8. The momentum integral is an example of a Gaussian integral, for which the general formula is known. Applying this formula simplifies Equation 8 to 9.

The partition for an ensemble of systems of two LJ particles or Ar atoms is

$$Z = \frac{1}{h^6} \frac{1}{\lambda_{\text{Ar}}^6} \int dx_1 \int dy_1 \int dz_1 \int dx_2 \int dy_2 \int dz_2\, e^{-\beta V_{\text{LJ}}(x_1,y_1,z_1,x_2,y_2,z_2)} \quad (10)$$

For this problem, you will numerically integrate the partition function for the spatial degrees of freedom (positions) using the trapezoidal rule over the range of x_1, y_1, z_1, x_2, y_2, z_2. Once the partition function is computed, the internal energy U can be calculated using the expression in Table 2 and C_V can be computed as

$$C_V = \frac{\partial U}{\partial T} \quad (11)$$

The dissociation temperature is defined as the temperature at which C_V reaches its maximum. This corresponds to the temperature at which the cluster of three LJ particles transitions from a bound state to an unbound or "dissociated" state. You will compute the temperature at which C_V reaches its maximum by finding the peak in the C_V *vs.* T plot.

3.1 Tasks

3.1.1 Partition Function Calculation

- ☐ Write a Python function that numerically computes the classical partition function of two LJ particles in a ***cubic*** box using the trapezoidal rule for spatial integration.
- ☐ Assume the ***cubic*** box has a fixed volume, and the integration should be performed over the relative distances between the particles.

3.1.2 Thermodynamic Properties

- ☐ Write Python functions to compute U and C_V from the partition function.

3.1.3 Dissociation Temperature

- ☐ Plot C_V as a function of temperature, and determine the temperature at which C_V reaches its maximum.
- ☐ This maximum corresponds to the atomization temperature of the LJ dimer.

3.2 Submission Guidelines

- ☐ Push a Python code named `comp_part_func.py` that numerically computes the classical partition function using the trapezoidal rule to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-grad`. When executed, this Python code should
  - ☐ Use `scipy.integrate.trapezoid`
  - ☐ Use the parameters in Table 3

Parameter	**Value**	**Units**
ϵ	0.0103	eV
σ	3.4	Å
V	10^3	$Å^3$
T_{min}	10	K
T_{max}	1000	K

Table 3

  - ☐ Write a `csv` file containing the partition function *vs.* temperature
- ☐ Push a Python code named `comp_ther_prop.py` that computes U and C_V as a function of temperature. When executed, this Python code should
  - ☐ Use `numpy.gradient`
  - ☐ Use the parameters in Table 3
  - ☐ Write a `csv` file containing internal energy and heat capacity *vs.* temperature
- ☐ Push a Python code named `plot_heat_capa.py` plotting the heat capacity as a function of temperature to your GitHub repository for this course named `chem-4050-5050` in a directory named `homework-4-grad`.
- ☐ Include clear comments in your code, explaining each key step.
- ☐ Ensure your plots are well-labeled and properly formatted.