Monte Carlo calculation of the free energy of a quantum system

I. Free energy in the Canonical Ensemble: Background

Consider a system coupled to a heat bath which is constrained to have an average energy U. A question arises as to how much of this energy U can be converted into work (for example lifting a mass against the force of gravity. The maximal energy available for use as work is called the "free" energy F. In order to compute free energy from first principles for a physical system described by a Hamiltonian $\hat{H}$ with eigenstates ψ_n and eigen-energies ϵ_n we need to derive a microscopic expression for it. A quantum system coupled to a macroscopic heat bath does not have a well defined quantum stationary state. Instead, the state at any instant fluctuates. The energy of the system also fluctuates in a microscopic time scale. However, the nature of the bath is such that when one averages over the fluctuations one obtains an average energy U. Suppose that our bath is characterized by this imposed U. We want to build a theory for a fluctuating system with fluctuations averaging to U. Let us think of large number N of replicas of the microscopic system. Each replica is in a well-defined quantum state. This is called an ensemble. The N replicas of the ensemble have a given total energy $E = UN$. Now we ask: under this constraint, how many systems n_n are in state ϵ_n? If we can answer this, we will be able to claim that $p_n = \frac{n_n}{N}$ is the probability of finding the physical system in state n.

First, what are the constraints on n_n? Obviously:

$$\begin{gathered}\sum_n n_n = N \\ \sum_n n_n \epsilon_n = E = UN\end{gathered} \tag{0.1}$$

Now, since we have no information on the coupling to the bath or on any detail, we can assume that the distribution n_n is just that distribution that can be realized the largest number of times. A given distribution n_n can be realized in

$$\Omega[\{n_n\}] = \frac{N!}{n_1!\,n_2!\dots} \tag{0.2}$$

ways. We search for n_n^*, the distribution that, subject to constraints (0.1) maximizes Ω. This is equivalent to maximizing $\ln\Omega$ (since $\ln(x)$ is a monotonically increasing function) which, using Stirling's formula $\ln n! \approx n\ln n - n$, can be expressed as:

$$\ln\Omega = N(\ln N - 1) - \sum_n n_n(\ln n_n - 1) \tag{0.3}$$

The constraints can be accounted for by 2 Lagrange multipliers α and β. We subtract from $\ln\Omega$ the first constraint as $\alpha(\sum_n n_n - N)$ and the second as $\beta(\sum_n n_n\,\epsilon_n - E)$. This leads to an "unconstrained" minimization of the following Lagrangian:

$$L[\{n_n\}] = -\sum_n n_n(\ln n_n - 1 + \alpha + \beta\epsilon_n) \tag{0.4}$$

The stationary point n^* is characterized by zero variation:

$$0 = \delta\ln\Omega = -\sum_n(\ln n_n^* + \alpha + \beta\epsilon_n)\delta n_n \tag{0.5}$$

Since the δn_n are now formally unconstrained, we find that each of the coefficients of δn_nmust separately be zero, so:

$$\ln n_n^* = -\alpha - \beta\epsilon_n \to n_n^* = e^{-\beta\epsilon_n-\alpha} \tag{0.6}$$

Applying the constraints themselves we can implicitly determine α and β:

$$\begin{gathered} N = \sum_n n_n^* = e^{-\alpha}\sum_n e^{-\beta\epsilon_n} \\ UN = e^{-\alpha}\sum_n \epsilon_n e^{-\beta\epsilon_n} \end{gathered} \tag{0.7}$$

We find that the optimal probability distribution is:

$$p_n = \frac{n_n^*}{N} = \frac{e^{-\beta\epsilon_n}}{Z} \tag{0.8}$$

This is the Boltzmann distribution. From normalization:

$$Z = \sum_n e^{-\beta\epsilon_n} \tag{0.9}$$

Is called the partition function. Note it is dependent only on the parameter β. Thus we can think of our system, coupled to the heat bath as being controlled by β. The Lagrange multiplier β is directly related to the temperate of the system via the identification:

$$T = \frac{1}{k_B\beta} \tag{0.10}$$

Thus our heat bath is characterized by the temperature. Note that in terms of p_n we have for the maximal number of realization:

$$\begin{aligned}\ln\Omega^* &= N(\ln N - 1) - \sum_n n_n^*(\ln n_n^* - 1) = N(\ln N - 1) - \sum_n Np_n(\ln Np_n - 1)\\ &= N(\ln N - 1) - \sum_n Np_n(\ln p_n + \ln N - 1) = -N\sum_n p_n \ln p_n\end{aligned} \tag{0.11}$$

The equilibrium properties of the system are determined by β and this is done via the partition function. For example, the average energy of a system in the ensemble of N systems:

$$U = \frac{E}{N} = \sum_n p_n\epsilon_n = \frac{1}{Z}\sum_n e^{-\beta\epsilon_n}\epsilon_n = \frac{1}{Z}\left(-\frac{\partial}{\partial\beta}\sum_n e^{-\beta\epsilon_n}\right) = -\frac{\partial\ln Z}{\partial\beta} \tag{0.12}$$

In thermodynamics the concept of entropy exists as a state function which is maximal under constraints. This prompted Boltzmann to postulate that the entropy of a system in the ensemble of N systems is defined as log the number of realizations of the most probable state $\frac{k_B\ln\Omega^*}{N}$:

$$\begin{aligned}S = \frac{k_B\ln\Omega^*}{N} &= -k_B\sum_n p_n\ln p_n = -k_B\frac{1}{Z}\sum_n e^{-\beta\epsilon_n}(-\beta\epsilon_n - \ln Z)\\ &= k_B\beta U + k_B\ln Z\end{aligned} \tag{0.13}$$

Indeed, multiplying (0.13) by $T=(k_B\beta)^{-1}$ we obtain:

$$U-TS=-\beta^{-1}\ln Z \tag{0.14}$$

Let us denote the left hand side by and energy function F. Suppose we want to extract work from the bath through the use of the quantum system. When we attach the system to a device for doing work, it's Hamiltonian becomes dependent on a parameter (the height of the weight). So as the weight is pulled up, the Hamiltonian changes, so it's partition function changes. The process of doing work (the change in the height) is done very slowly so that the bath has time to equilibrate the system. Such a process is called adiabatic or reversible. The temperature of the bath, which is too big to be affected, remains constant. Thus, in this process, the internal energy of the system changes by ΔU and the internal entropy by ΔS. This change in entropy of the system signifies that heat $Q=T\Delta S$ was transferred from the bath to the system. The work done is then $W=\Delta U-Q$ (by the first law) which means that $W=\Delta U-T\Delta S$. Thus we see that the work done by a system as it is connected to a heat bath is simply ΔF for this process. We see that $F=U-TS$ is the free energy for processes which preserve the temperature. Thus, the partition function, a microscopic quantity, is directly related to the free energy, a thermodynamic property:

$$F=-\beta^{-1}\ln Z \tag{0.15}$$

We see from the above that:

$$U=\frac{\partial\beta F}{\partial\beta}=-T^2\frac{\partial F/T}{\partial T}=F-T\frac{\partial F}{\partial T} \tag{0.16}$$

And the entropy is:

$$S=\frac{1}{T}(U-F)=-\frac{\partial F}{\partial T} \tag{0.17}$$

We have thus formed a microscopic theory enabling the computation of the free energy F and from it all properties are derived. For example the change in internal energy when the temperature is changes:

$$C=\frac{\partial U}{\partial T} \tag{0.18}$$

This is in fact the "heat capacity". To see this, we not that from Eq. (0.16) and Eq. (0.17):

$$C=-T\frac{\partial^2 F}{\partial T^2}=T\frac{\partial S}{\partial T} \tag{0.19}$$

The change in entropy dS between two thermodynamic states of the system is the ratio of the reversible change of heat dQ to the temperature T:

$$dQ=TdS \tag{0.20}$$

So:

$$C=\frac{\partial Q}{\partial T} \tag{0.21}$$

Hence the name "heat capacity".

II. Computing the Partition Function

We saw that the free energy of a quantum system can determine its equilibrium properties when coupled to a bath of temperature $T=(k_B\beta)^{-1}$. The free energy is given by Eq. (0.15) is computable once we have the partition function $Z(\beta)$. This can be written as:

$$Z(\beta)\ =tr\left[e^{-\beta\hat{H}}\right] \tag{0.22}$$

Where the trace is realized given any complete basis having M orthonormal ψ_m functions:

$$Z=\sum_{m=1}^{M}\langle\psi_m|e^{-\beta\hat{H}}|\psi_m\rangle \tag{0.23}$$

With:

$$\langle\psi_m|\psi_{m'}\rangle=\delta_{mm'} \tag{0.24}$$

We assume, in an actual computation that:

$$\hat{H}=-\frac{\hbar^2}{2M}\nabla^2+V(\boldsymbol{r}) \tag{0.25}$$

If we use a grid to implement the system for a computer the Laplacian can be taken as:

$$\psi \to \psi_n = \psi(\boldsymbol{r}_n) \qquad \nabla^2\psi(\boldsymbol{r}_n) \to \left(L^h\psi\right)_n \tag{0.26}$$

Where L^h is a finite difference formula and thus a very sparse matrix. For example, in 2D:

$$\left(L^h\psi\right)_{n,m} = \frac{\psi_{(n+1),m} + \psi_{(n-1),m} - 2\psi_{nm} + \psi_{n,(m+1)} + \psi_{n,(m-1)}}{h^2} \tag{0.27}$$

In order to compute the operation of and exponent of an operator on a function suppose we use the following scheme:

$$e^{-\beta\hat{H}}\psi = \lim_{\mathrm{N}\to\infty}\left(1 - \frac{\beta\hat{H}}{N}\right)^N \psi \tag{0.28}$$

Thus, for a very large N, once we select N we can apply the operator $\left(1 - \frac{\beta\hat{H}}{N}\right)$ to ψ N times. In a typical problem the number of grid-points can reach $M = 10^6$ while $N \approx 10^4$. The operation of $e^{-\beta\hat{H}}$ on a ψ costs $MN \approx 10^{10}$ operations, quite doable on a computer. But for the trace one needs M functions, so the evaluation of the trace incurs $M^2N \approx N^{16}$ operations, which is very expensive. (Note: below we describe a stable method and accurate method for the calculation of $e^{-\beta\hat{H}}\psi$. However, it will scale similarly to the above estimate).

Can this large effort be reduced?

We can evaluate the trace using a Monte Carlo procedure. We give up (hopefully) a small amount of accuracy while reducing the computational load significantly. Let us define a “random” wave function:

$$\chi = \sum_{m=1}^{M} O_m\psi_m \tag{0.29}$$

The numbers O_m are either 1 or -1 taken at random. For each random determination of χ we compute $\left\langle\chi\middle|e^{-\beta\hat{H}}\middle|\chi\right\rangle$. We the average over the O’s. This means we generate I random

$\chi's$, call each χ_i and calculate the expectation value for each then sum them all up and divided by I:

$$\left\langle\langle\chi|e^{-\beta\hat{H}}|\chi\rangle\right\rangle_O \equiv \lim_{I\to\infty}\frac{1}{I}\sum_{i=1}^{I}\langle\chi_i|e^{-\beta\hat{H}}|\chi_i\rangle \approx \frac{1}{I}\sum_{i=1}^{I}\langle\chi_i|e^{-\beta\hat{H}}|\chi_i\rangle \tag{0.30}$$

For this type of averaging:

$$\begin{aligned}\left\langle\langle\chi|e^{-\beta\hat{H}}|\chi\rangle\right\rangle_O &= \left\langle\sum_{m,m'=1}^{M} O_m O_{m'}\langle\psi_m|e^{-\beta\hat{H}}|\psi_{m'}\rangle\right\rangle_O \\ &= \sum_{m,m'=1}^{M}\langle O_m O_{m'}\rangle_O\langle\psi_m|e^{-\beta\hat{H}}|\psi_{m'}\rangle\end{aligned} \tag{0.31}$$

But since O_m are random and are not correlated with $O_{m'}$ we have, for $m \neq m'$:

$$\langle O_m O_{m'}\rangle_O = \langle O_m\rangle_O\langle O_{m'}\rangle_O = 0 \tag{0.32}$$

On the other hand $\langle O_m O_m\rangle_O = \langle 1\rangle_O = 1$. Thus:

$$\langle O_m O_{m'}\rangle_O = \delta_{mm'} \tag{0.33}$$

From this we find:

$$\left\langle\langle\chi|e^{-\beta\hat{H}}|\chi\rangle\right\rangle_O = \sum_{m=1}^{M}\langle\psi_m|e^{-\beta\hat{H}}|\psi_m\rangle = tr\left[e^{-\beta\hat{H}}\right] \tag{0.34}$$

We approximate $\left\langle\langle\chi|e^{-\beta\hat{H}}|\chi\rangle\right\rangle_O \approx \frac{1}{I}\sum_{i=1}^{I}\langle\chi_i|e^{-\beta\hat{H}}|\chi_i\rangle$ (i.e. do not take the limit $I\to\infty$) and hope that $I \ll M$ gives an estimate of the trace to good accuracy. If we are willing to give up some accuracy, we may gain a large factor in the trace calculation.

How many iterations must we make in order to improve an estimate? Here we can use the the central limit theorem of statistics which shows that process of averaging I samples from a distribution of mean μ and variance σ^2 gives a random number which is distributed as a normal distribution with mean μ and variance $\frac{\sigma^2}{I}$. Thus, in order to reduce the statistical error by a factor 10, one must increase I by a factor 100.

III. Newton interpolation

The computation of a function of a matrix can be done in several ways. We will learn a neat way, called the Newton method.

Before we refer to Newton, let us study the concept of interpolation. Suppose we have a function $f(x)$ which is very difficult to compute. Thus we spend time on computing it for a finite, say $N+1$ number of points $x_0 < x_1 \ldots < X_N$. We find $y_i = f(x_i)$. Now we want an approximation for any $x_0 < x < x_N$. One way of doing this is to use a polynomial. There is one and only one polynomial of degree at most N $p_N(x) = a_0 + a_1 x + \cdots + a_N x^N$ which obeys $p_N(x_i) = y_i$.

Lagrange wrote this polynomial as follows:

$$p_N(x) = \sum_{n=0}^{N} l_n(x) y_n$$

Where $l_n(x)$ are polynomials having the property:

$$l_n(x_m) = \delta_{nm}$$

These polynomials are easily guessed:

$$l_n(x) = \prod_{m \neq n}^{N} \frac{x - x_m}{x_n - x_m}$$

Example: the parabola going through $(0,2), (1,-1), (2,1)$ is:

$$p_2(x) = 2 \times \frac{(x-1)(x-2)}{(0-1)(0-2)} - 1 \times \frac{(x-0)(x-2)}{(1-0)(1-2)} + 1 \times \frac{(x-0)(x-1)}{(2-0)(2-1)}$$
$$= (x-1)(x-2) + x(x-2) + \frac{1}{2} x(x-1)$$

The interpolation polynomial goes through the interpolating points. A more convenient formula is due to Newton.

Suppose $p_N(x)$ is the interpolation polynomial for $x_0, \ldots, x_N$ and $p_{N-1}(x)$ that for $x_0, \ldots, x_{N-1}$. Then,

$$p_N(x) - p_{N-1}(x) = C(x)$$

And $C(x)$ is a polynomial of degree N. We also have

$$C(x_i) = 0, \qquad i = 0, \dots, N-1$$

Thus:

$$C(x) = a_N(x - x_0) \times \dots \times (x - x_{N-1})$$

For the last point (x_N, y_N) we have:

$$y_N = p_N(x_N) = p_{N-1}(x_N) + a_N(x_N - x_0) \times \dots \times (x_N - x_{N-1})$$

So that:

$$a_N = \frac{y_N - p_{N-1}(x_N)}{(x_N - x_0) \times \dots \times (x_N - x_{N-1})}$$

We thus see a way to build the next polynomial from the previous one:

$$p_N(x) = p_{N-1}(x) + \frac{y_N - p_{N-1}(x_N)}{(x_N - x_0) \times \dots \times (x_N - x_{N-1})}(x - x_0) \times \dots \times (x - x_{N-1})$$

This is a nice approach because it seems we can add points and “update” the polynomial as we go along.

The formula is:

$$f(x) \approx p_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots a_n(x - x_0) \cdots (x - x_{n-1})$$

While Newton’s method looks beautiful, it turned out that to be very unstable for a large sequence of points. However, one can overcome this if one chooses the points in a certain order. This order makes it stable.

The stable ordering was investigated by Leja and others (see references below). If the variable x is in the interval $[a, b]$ then one can redfine

$$f(x) = F(s)$$

Where $s=4\frac{\left(x-\frac{a+b}{2}\right)}{b-a}$ is in $[-2,\dots,2]$ and expand $F(s)$ in the interval. The Leja points in this $[-2,\dots,2]$ can be obtained from a Mathtype program given in the appendix of the paper of Reichel (3).

The material on Newton interpolation was based on:

1) R. Kosloff, "Propagation Methods for Quantum Molecular-Dynamics", Ann. Rev. Phys. Chem. 45, 145-178 (1994).
2) L. Reichel, "Newton interpolation at Leja points", BIT 30, 332-346 (1990).
3) J. Baglama, D. Calvetti and L. Reichel, “Fast Leja Points”, Electronic Transactions of Numerical Analysis, Vol. 7, p. 124-140 (1998) (download from: http://etna.mcs.kent.edu/volumes/1993-2000/vol7/abstract.php?vol=7&pages=124-140. A java applet for the leja points is also given (up to 500 points in [-2,2]).

IV. Computing the Function of an operator of a sparse matrix

Suppose we want to compute the operation of a function of an operator $\hat{H}$ on a wave function:

$$\psi=f(\hat{H})\phi$$

We define a shifted –scaled Hamiltonian, using the maximal and minimal eigenvalues.

$$\hat{H}_S=4\frac{\left(\hat{H}-\frac{E_{min}+E_{max}}{2}\right)}{E_{max}-E_{min}}$$

We use the Newton expansion of the function $F(s)$ in the interval $[-2,2]$. Then

$$\psi=\sum_{m=0}^{N}a_m\psi_m$$

Where

$$\psi_0=\phi_0$$

$$\psi_m=\left(\hat{H}_S-x_{m-1}\right)\psi_{m-1}$$

(note that we do not need to store any of the ψ_m, except that which is currently being used).