Luke Barbano

Metropolis Monte Carlo Algorithm Instructions:

The Metropolis Monte Carlo algorithm is a method by which to numerically simulate the canonical ensemble. To understand how it works, consider a general expression for the average energy of an Einstein solid $\bar{E}(T) = \frac{\sum_{s=1}^{\infty} E_s e^{-\beta E_s}}{\sum_{s=1}^{\infty} e^{-\beta E_s}}$ where E_s and $e^{-\beta E_s}$ are, respectively, the energy and relative probability of a microstate s. A computer is unable to generate the infinite microstates required to exactly determine the above sum, so a possible recourse is to sample the space of possible microstates by randomly generating M possible microstates and evaluating the above expression for these M microstates. In this case, the equilibrium energy is given by $\bar{E}(T) = \frac{\sum_{s=1}^{M} E_s e^{-\beta E_s}}{\sum_{s=1}^{M} e^{-\beta E_s}}$, an increasingly good approximation as M increases. However, since higher energy microstates are exponentially less probable than those with relatively lower energy, this approach is wasteful in that it effectively over-considers microstates that barely contribute to the sum. To improve the efficiency of this process, it is possible to implement a form of *importance sampling* in which only the most probable microstates are considered. This is accomplished by generating microstates that are proportional to their relative probability $e^{-\beta E_s}$.

The Metropolis algorithm itself works by comparing a microstate i with energy E_i to another microstate j with energy E_j; for simplicity, we can restrict the choice of j such that it must be adjacent to i, although this is not required and the resulting distribution will converge to the same result as if the choice of j is arbitrary. Since the probabilities of the system being in state i or j are $p_i = e^{-\beta E_i}$ and $p_j = e^{-\beta E_j}$ respectively, the probability that the system transitions from state i to state j is ${p_j}/{p_i} = e^{-\beta(E_j - E_i)} = e^{-\beta \Delta E}$. If $\Delta E < 0$, the probability of a transition is greater than 1 and the transition will decidedly occur. This is to say that the system will always transition to a state with lower energy. Otherwise, if $\Delta E > 0$, there is a probability of $p = e^{-\beta \Delta E}$ that the microstate transition will occur. By generating a random number on the unit interval and comparing it to this probability, we can decide whether to accept or reject the microstate transition.

The procedure to implement this algorithm for an Einstein solid is outlined on page 227 of G&T, but is also reproduced below. Each step can be mapped to an appropriately labeled section of the accompanying codes that execute and visualize the simulation.

1. **Initial Conditions:**
   Choose an initial microstate for a system by assigning random initial energies (or zero energy) to each particle of an Einstein solid. Also specify the temperature of the bath.

2. **Trial Move:**
   Choose a particle at random and change its energy by ± 1. Compute the change in energy of the entire system ΔE, and determine the probability that the system will transition to this new microstate. If $\Delta E < 0$, automatically accept the change. Otherwise, accept the change with probability $p = e^{-\beta \Delta E}$ by generating a random

number on the unit interval and comparing it to p. If the change is rejected, do not modify the microstate.

3. **Repeat Trial Move:**
   Repeat the trial move for a sufficiently large number of trial moves such that the system reaches equilibrium and the distribution of particle energies resembles the probability distribution.

4. **Determine relevant quantities:**
   Compute the averages of quantities of interest (i.e. $\bar{E}$, etc.)