1. **An ideal gas on a lattice.** In class we introduced a microscopic definition of entropy due to Boltzmann:

$$S_{\mathrm{B}} = k_{\mathrm{B}} \ln \Omega.$$

A more general expression for entropy due to Gibbs is:

$$S_{\mathrm{G}} = -k_{\mathrm{B}} \sum_{\nu} P(\nu) \ln P(\nu),$$

where ν is a microstate.

(i) For an isolated system (with fixed energy E, number of molecules N, and volume V) show that S_{B} and S_{G} are identical. Recall that in such a system, all allowed microstates are equally likely.

Let's start with the Gibbs form using the uniform distribution over microstates $P(\nu) = 1/\Omega$.

$$S_{\mathrm{G}} = -k_{\mathrm{B}} \sum_{\nu} \frac{1}{\Omega} \ln \frac{1}{\Omega} = k_{\mathrm{B}} \sum_{\nu} \frac{1}{\Omega} \ln \Omega = k_{\mathrm{B}} \frac{1}{\Omega} \ln \Omega \sum_{\nu} 1 = k_{\mathrm{B}} \ln \Omega = S_{\mathrm{B}}$$

(ii) For the remaining parts of this problem, consider a collection of N indistinguishable particles arranged on a lattice of M cells. Each cell can be occupied by at most one particle, and particles in different cells do not interact. Calculate the total number $\Omega(M, N)$ of possible configurations for this system.

We need to place N particles in M cells. If multiple particles could fit into a single cell we would have M^N possible ways to arrange the particles, but we said that each cell can only hold a single particle. Therefore we need to choose N of the M cells to hold a particle:

$$\Omega(M, N) = \frac{M!}{N!(M-N)!}.$$

(iii) Assuming M, N, and $M - N$ are all very large, use Stirling's approximation to write the Boltzmann entropy per cell, S_{B}/M, as a function of $f \equiv N/M$ alone.

Using Stirling's approximation for $M!$, $N!$, and $(M-N)!$, we get

$$\begin{aligned} S/k_{\mathrm{B}} &= \ln \Omega \\ &= \ln M! - \ln N! - \ln(M-N)! \\ &\approx M \ln M - M - N \ln N + N - (M-N) \ln(M-N) + (M-N) \\ &= M \ln M - N \ln N - (M-N) \ln(M-N) \end{aligned}$$

We divide through by M and use $N = fM$ to replace all instances of N yielding

$$\begin{aligned}\frac{S}{k_{\mathrm{B}}M} &= \ln M - f\ln fM - \ln M(1-f) + f\ln M(1-f)\\ &= \ln\frac{1}{1-f} - f\ln\frac{f}{1-f}\\ &= -(1-f)\ln(1-f) - f\ln f\end{aligned}$$

(iv) The occupation state of one cell in this lattice system is not affected by that of any other cell. As a result, the total entropy can be written as $S = Ms_{\mathrm{cell}}$, where s_{cell} is the entropy of a single cell. Using Gibbs' definition of entropy, calculate s_{cell} in terms of p_1 and p_0, the probabilities of finding a particular cell occupied or unoccupied, respectively.

Each cell can be occupied with probability p_1 or unoccupied with probability p_0, so the Gibbs entropy per cell is particularly simple to compute:

$$s_{\mathrm{cell}}/k_{\mathrm{B}} = -p_0\ln p_0 - p_1\ln p_1.$$

(v) By expressing p_0 and p_1 in terms of f, the fraction of occupied cells, demonstrate that the Boltzmann and Gibbs lattice gas entropies are the same.

This is simple enough once we note that $p_0 = 1 - f$ and $p_1 = f$ (each site must be either occupied or unoccupied). Inserting into (iv) gives

$$\frac{S_{\mathrm{G}}}{k_{\mathrm{B}}M} = \frac{s_{\mathrm{cell}}}{k_{\mathrm{B}}} = -f\ln f - (1-f)\ln(1-f),$$

in agreement with the Boltzmann expression in (iii). (You might ponder why the Boltzmann expression relied on the Stirling approximation while the Gibbs expression did not.)

(vi) Show that for the low density lattice gas ($f \ll 1$),

$$S \approx -k_{\mathrm{B}}V[\rho\ln(\rho v) - \rho],$$

where v is the volume of a single lattice cell, and $\rho = N/V = N/(Mv)$ is the density.

We first want to Taylor expand our previous results for small f. Specifically, we can think of the term $(1-f)$ as 1 minus a small perturbation f, leading us to Taylor expand $(1-f)\ln(1-f)$.

Then we re-express terms using $\rho = N/(Mv)$ and $v\rho = N/M = f$.

$$\begin{aligned}
\frac{S}{k_{\mathrm{B}}M} &= -f\ln f - (1-f)\ln(1-f) \\
&= -f\ln f - (1-0)\ln(1-0) - f\,\frac{\partial}{\partial f}[(1-f)\ln(1-f)]\bigg|_{f=0} + \mathcal{O}(f^2) \\
&= -f\ln f - 0 - f\left[-\frac{1-f}{1-f} - \ln(1-f)\right]_{f=0} + \mathcal{O}(f^2) \\
&= f - f\ln f + \mathcal{O}(f^2) \\
&= -v\rho\,(\ln v\rho - 1)\,.
\end{aligned}$$

Multiplying through by M gives

$$S = -Nk_{\mathrm{B}}[\ln v\rho - 1] = -Vk_{\mathrm{B}}[\rho\ln v\rho - \rho].$$

(vii) From macroscopic thermodynamics we saw that

$$dS = \frac{1}{T}dE + \frac{p}{T}dV - \frac{\mu}{T}dN,$$

so

$$\frac{p}{T} = \left(\frac{\partial S}{\partial V}\right)_{E,N}.$$

Use the expression you just obtained for the entropy of a low density lattice gas to explicitly compute the partial derivative. Express your result as the familiar ideal gas law: $pV = Nk_{\mathrm{B}}T$. (You may be more familiar with $pV = nRT$. Make sure you think through the distinction between the two forms if it's not immediately obvious to you.)

You must be a little careful in differentiating S with respect to V. At first glance it may look like S is a constant times V, but volume terms are also hidden in the ρ terms:

$$S = -k_{\mathrm{B}}V\left[\frac{N}{V}\ln\frac{N}{M} - \frac{N}{V}\right] = k_{\mathrm{B}}\left[N\ln\frac{N}{M} - N\right] = -k_{\mathrm{B}}N\left[\ln\frac{Nv}{V} - 1\right].$$

Now we can take the partial derivative.

$$\frac{p}{T} = \left(\frac{\partial S}{\partial V}\right)_{E,N} = k_{\mathrm{B}}N\frac{V}{Nv}\frac{Nv}{V^2} = \frac{k_{\mathrm{B}}N}{V}.$$

The difference between this and $pV = nRT$ is whether you measure the number of particles in moles. That is, $R = k_{\mathrm{B}}N_{\mathrm{A}}$ with N_{A} being Avogadro's number. You may have seen a different (more mechanical) derivation of an ideal gas law that considers the force imparted by particles hitting a wall, which requires you to consider the momenta of the particles. The power of the current presentation is that it emphasizes how general the ideal gas law really is. It's not limited to gas molecules flying through vacuum until they collide with a wall. Rather, the "equation of state" $pV = nRT$ derives from noninteracting degrees of freedom which have statistically independent fluctuations. Any time you have that decorrelation, you should anticipate something with an ideal gas law structure (though we may call the pressure the osmotic pressure, for example).

(viii) An ideal gas is one with particles which do not interact. Is the lattice gas model whose entropy you found in parts (iii) through (v) an ideal gas? In other words, are there any interactions between the particles? Explain how your answer does or does not agree with your derivation of an ideal gas law in (vii).

> No, the gas in the first parts is not ideal. Even though the particles don't interact at a distance, they do affect each other because only one particle can be in every site. A truly non-interacting situation would have allowed there to be multiple particles at a site. Nevertheless, in the low-density limit, our interacting gas model acts like an ideal gas because the particles are so dilute that there is no real difference between explicitly forbidding them from being on top of each other and allowing that possibility (but knowing that it will essentially never happen when there are so few particles).

2. **A polymer off a lattice.** There are many ways in which you might argue that the polymer model from last week's problem set is lacking. For one thing, you might complain that real polymers don't live on a grid; we should have let each step move in a random direction, not just along a Cartesian axis. You might also complain that polymers can stretch. Perhaps we should have allowed the size of each link between monomers to vary. In this problem we will explore a model of a Gaussian polymer (in three dimensions) which addresses both of these complaints. It's a model we'll refer back to in lectures as we discuss heat, work, and free energy.

   We denote the position of the $n+1$ monomers (still n segments as in last week's lattice polymer) by $\boldsymbol{R}_0, \boldsymbol{R}_1, \ldots \boldsymbol{R}_n$ and define our coordinate system by putting the origin at the location of the initial monomer, so $\boldsymbol{R}_0 = 0$. The next monomer's position is randomly selected from the Gaussian distribution

$$P(\boldsymbol{R}_1) = \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\left(\boldsymbol{R}_1\right)^2}{2\ell^2}\right). \tag{1}$$

   Observe that the bond between monomers 0 and 1 can be of variable length and can point in any direction in three-dimensional space, in contrast to Problem 1. Monomer 2's position is similarly selected from a Gaussian, but now a Gaussian centered at the location of the second monomer:

$$P(\boldsymbol{R}_2|\boldsymbol{R}_1) = \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\left(\boldsymbol{R}_2-\boldsymbol{R}_1\right)^2}{2\ell^2}\right).$$

   The left-hand-side is read as "the probability of monomer 2 position $\boldsymbol{R}_2$ given that monomer 1 is at $\boldsymbol{R}_1$." The pattern continues, so the position of monomer $i+1$ is randomly selected from

$$P(\boldsymbol{R}_{i+1}|\boldsymbol{R}_i) = \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\left(\boldsymbol{R}_{i+1}-\boldsymbol{R}_i\right)^2}{2\ell^2}\right).$$

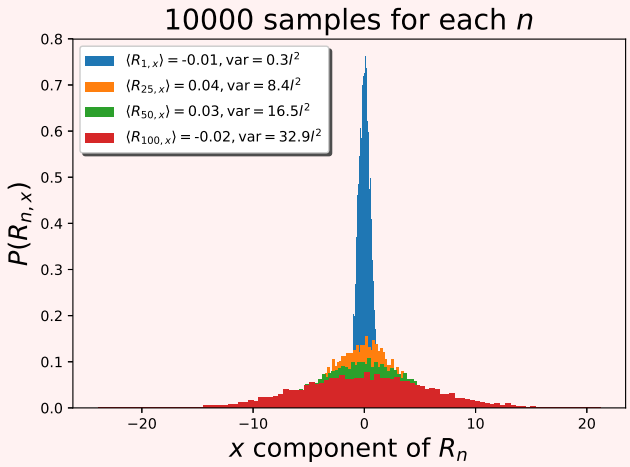
   The goal of this problem is to see that this model has the exact same root mean squared end-to-end distance as the simpler polymer in Problem 1. (The secondary goal is to become familiar with this model since it will come up in lecture.)

   (i) First, let's see the result by sampling polymers with a computer. Adapt your polymer sampling code from last week to generate random Gaussian polymers with n segments. If you're using Python, you'll probably find the function np.random.normal to be useful. By generating a collection of many

random polymers with the same length, plot for yourself a histogram for the x-component of $\boldsymbol{R}_n$. What is the mean of this histogram? In terms of l, what is the variance? Repeat these calculations for various n from 1 to 100 and plot the variance of the x-component of $\boldsymbol{R}_n$ as a function of n. As in experimental plots, it is best to include error bars. Your error bars should indicate how much uncertainty there is due to the fact that you only sampled some of the possible polymer configurations.

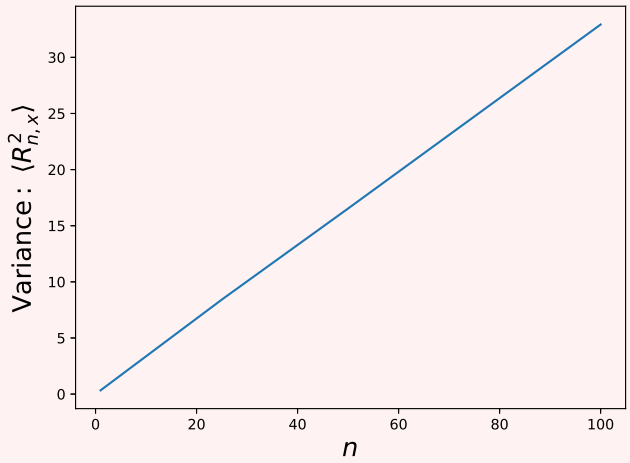
I generated histograms for $n = 1, 25, 50, 100$ using the following code:

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
plt.style.use('default')

step_mean = 0
step_variance = 1/np.sqrt(3)
dim = 3

Num_Polymer_Samples = 10000;

Num_Steps_List = [1, 25, 50, 100];

x_distances = np.zeros([Num_Polymer_Samples, len(Num_Steps_List)])

for k in range(len(Num_Steps_List)):
  Num_Steps = Num_Steps_List[k]
  for j in range(Num_Polymer_Samples):
    # Initialize the 3d polymer at the origin
    polymer3D = np.zeros([Num_Steps + 1, 3])
    # Draw a random Gaussian for each step. Note that the variance for the
    # x, y, and z components is not 1 but rather 1 / sqrt(dim)
    for i in range(Num_Steps):
      polymer3D[i+1] = polymer3D[i] + np.random.normal(step_mean, \
                                                       step_variance, dim)
    (x,y,z) = np.transpose(polymer3D)
    x_distances[j, k] = x[[-1]] # record the x position of the final monomer

for k in range(len(Num_Steps_List)):
   plt.hist(np.transpose(x_distances)[k],100,density=True, \
            label=r'$\langle R_{'+str(Num_Steps_List[k])+\
            r',x} \rangle = $'+\
            str(np.round(np.mean(np.transpose(x_distances)[k]),2))+\
           r'$, \mathrm{var} = $'+\
           str(np.round(np.var(np.transpose(x_distances)[k]),1))+'$l^2$');
plt.xlabel(r'$x \mathrm{\ component\ of\ }R_n$', fontsize=18);
plt.ylabel(r'$P(R_{n,x})$', fontsize=18);
plt.title(str(Num_Polymer_Samples)+r' samples for each $n$', fontsize=20);
plt.tight_layout()
plt.legend(loc=''upper left'', fancybox=True, shadow=True);
plt.savefig('PS3P2p1Hist.eps',format='eps',transparent=True);

from google.colab import files
files.download('PS3P2p1Hist.eps')
```



The plot of the variance versus n is pretty simple to make without error bars. I note that these variances were already computed and shown on the legend of my histogram, so I just pull those variances out and plot them with the following code:

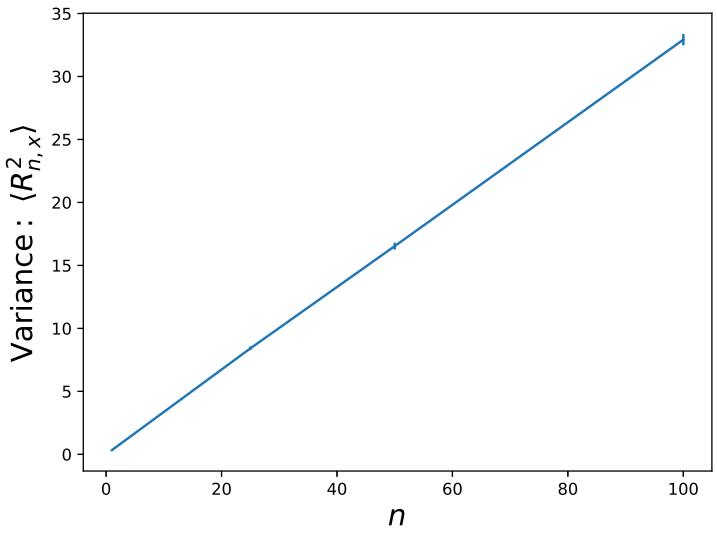
```
Variance_List = np.zeros(len(Num_Steps_List));
for k in range(len(Num_Steps_List)):
  Variance_List[k] = np.var(np.transpose(x_distances)[k])

plt.plot(Num_Steps_List, Variance_List)
plt.xlabel(r'$n$', fontsize=18);
plt.ylabel(r'$\mathrm{Variance: \ }\langle R_{n,x} ^2 \rangle$', fontsize
    =18);
plt.tight_layout()
plt.savefig('PS3P2p1MSD.eps',format='eps',transparent=True);
from google.colab import files
files.download('PS3P2p1MSD.eps')
```



It is a little more painful to include error bars. The simplest way to get error bars is to do N independent experiments, compute the quantity of interest from each experiment, then find the standard deviation of the values from those N experiments. If the experiments truly are independent of each

other, then the standard deviation will drop like $1/\sqrt{N}$. Here, I generate 10,000 polymer samples then split those into 10 independent subsamples (independent experiments). The standard deviation I would expect from 10,000 samples is $1/\sqrt{10}$ times as much as the standard deviation I would expect from 1,000 samples. I am able to estimate this 1,000 sample standard deviation from my 10 subsamples, then rescale it to plot 10,000 sample error bars. Yes, it's tedious to do this. Yes, the error bars are very small in this case. But this type of manipulation will probably be something you'll want to use throughout graduate school when responsibly analyzing data.

```
ErrorBar_List = np.zeros(len(Num_Steps_List));

Num_Subsamples = 10;

for k in range(len(Num_Steps_List)):
  subsamples = np.split(np.transpose(x_distances)[k], Num_Subsamples);
  Subsample_Variances = np.zeros(Num_Subsamples)
  for m in range(Num_Subsamples):
    Subsample_Variances[m] = np.var(subsamples[m])
  ErrorBar_List[k] = np.sqrt(np.var(Subsample_Variances) \
                             / float(Num_Subsamples))

plt.errorbar(Num_Steps_List, Variance_List,yerr=ErrorBar_List);
plt.xlabel(r'$n$', fontsize=18);
plt.ylabel(r'$\mathrm{Variance: \ }\langle R_{n,x} ^2 \rangle$', fontsize=18);
plt.tight_layout()
plt.savefig('PS3P2p1MSDErrorBars.eps',format='eps',transparent=True);
from google.colab import files
files.download('PS3P2p1MSDErrorBars.eps')
```



(ii) In the last problem set, you found that $\langle R_n^2 \rangle = nl^2$. Compare this to your result from (i). Explain any notable similarities and differences between the two results.

In (i) you hopefully found that $\langle R_{n,x}^2 \rangle = \frac{n}{3} l^2$. Because the y and z components are independent of the x component, $\langle R_n^2 \rangle = \langle R_{n,x}^2 \rangle + \langle R_{n,y}^2 \rangle + \langle R_{n,z}^2 \rangle = 3\langle R_{n,x}^2 \rangle = nl^2$, which agrees with the way the polymers spread out in last week's problem set.

(iii) To estimate an average value, we have used the average of a set of sampled polymers, but the true average involves an integration over all possible polymers. Typically such an integral cannot be performed analytically, but in the special case of Gaussian distributions, the integrals are solvable (as you explored on the previous problem set). Using those Gaussian integrals (you don't need to rederive these!), compute the root mean squared distance of the connection between monomer 0 and 1 (it will be the same between any monomers i and $i+1$). You will probably want to write integrals in Cartesian coordinates, i.e.,

$$\begin{aligned}\langle R_1^2\rangle &= \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \int d\boldsymbol{R}_1\ (\boldsymbol{R}_1)^2\, e^{-\frac{3(\boldsymbol{R}_1)^2}{2\ell^2}} \\ &= \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \int_{-\infty}^{\infty} dx_1 \int_{-\infty}^{\infty} dy_1 \int_{-\infty}^{\infty} dz_1\ (x_1^2+y_1^2+z_1^2) \exp\left(-\frac{3(x_1^2+y_1^2+z_1^2)}{2\ell^2}\right),\end{aligned}$$

with $\boldsymbol{R}_1 = (x_1, y_1, z_1)$. If you want, you can do integrals like this in Mathematica as well.

Notice that $\langle x_1^2\rangle = \langle y_1^2\rangle = \langle z_1^2\rangle = \frac{\ell^2}{3}$, so the sum of all three just gives ℓ^2. Hey, it's almost like I constructed the Gaussian distributions just so the mean squared distance of a link between monomers would be exactly ℓ to match the (fixed) distance between monomers in last week's lattice polymer.

(iv) We now consider $n = 2$. Again, the first monomer is set to be at the origin, the second monomer is at $\boldsymbol{R}_1$ and the final monomer is at $\boldsymbol{R}_2$. In terms of $\boldsymbol{R}_1$, $\boldsymbol{R}_2$, and l, what is the (normalized) probability distribution for a microstate of the polymer $P(\boldsymbol{R}_1, \boldsymbol{R}_2)$? We said all microstates are equally likely, but you probably found that these are not. Why not?

$$\begin{aligned}P(\boldsymbol{R}_1, \boldsymbol{R}_2) &= P(\boldsymbol{R}_1 \text{ and } \boldsymbol{R}_2) = P(\boldsymbol{R}_2|\boldsymbol{R}_1)P(\boldsymbol{R}_1) \\ &= \left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\,(\boldsymbol{R}_2-\boldsymbol{R}_1)^2}{2\ell^2}\right)\left(\frac{3}{2\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\,(\boldsymbol{R}_1)^2}{2\ell^2}\right) \\ &= \left(\frac{3}{2\pi\ell^2}\right)^{3} \exp\left(-\frac{3}{2\ell^2}[(\boldsymbol{R}_2-\boldsymbol{R}_1)^2 + \boldsymbol{R}_1^2]\right)\end{aligned}$$

Because the original distributions were themselves already normalized, they remain normalized after our manipulation. Answers may vary for the next part of the problem. Trivially, we see that the initial distribution of $P(\boldsymbol{R}_1)$ wasn't uniform to begin with, so there's no reason to expect that $P(\boldsymbol{R}_1, \boldsymbol{R}_2)$ will be, either. The idea behind this is that all microstates are only equally likely in the *microcanonical ensemble*, and the probability distributions here are incompatible with this ensemble. In fact, these distributions are associated with a canonical distribution, which means that we can define an effective energy in the system associated with the coordinate $\boldsymbol{R}_1$.

(v) Suppose you only intend to measure the location of $\boldsymbol{R}_0$ (the origin) and $\boldsymbol{R}_2$ but will never explicitly observe $\boldsymbol{R}_1$. Consequently, you would like to know the *marginalized* distribution that averages over the possible values of $\boldsymbol{R}_1$:

$$P(\boldsymbol{R}_2) = \int d\boldsymbol{R}_1\, P(\boldsymbol{R}_1, \boldsymbol{R}_2).$$

Perform that integration (Mathematica or citing last week's results are fine) to determine the marginal distribution $P(\boldsymbol{R}_2)$. Compare this distribution to the distribution in Eq. (1). What is different? What is the same?

From 2iv,

$$P(\boldsymbol{R}_1, \boldsymbol{R}_2) = \left(\frac{3}{2\pi\ell^2}\right)^3 \exp\left(-\frac{3}{2\ell^2}[(\boldsymbol{R}_2 - \boldsymbol{R}_1)^2 + \boldsymbol{R}_1^2]\right).$$

Integrating over $\boldsymbol{R}_1$ involves a standard Gaussian integration. We will begin by completing the square in $\boldsymbol{R}_1$ in the argument of the exponential, with

$$(\boldsymbol{R}_2 - \boldsymbol{R}_1)^2 + \boldsymbol{R}_1^2 = 2\boldsymbol{R}_1^2 - 2\boldsymbol{R}_1 \cdot \boldsymbol{R}_2 + \boldsymbol{R}_2^2 = 2\left(\boldsymbol{R}_1 - \frac{\boldsymbol{R}_2}{2}\right)^2 + \frac{\boldsymbol{R}_2^2}{2}.$$

Performing the resulting standard Gaussian integral thus yields

$$\begin{aligned} P(\boldsymbol{R}_2) &= \int \mathrm{d}\boldsymbol{R}_1 \, P(\boldsymbol{R}_1, \boldsymbol{R}_2) \\ &= \left(\frac{3}{2\pi\ell^2}\right)^3 \int \mathrm{d}\boldsymbol{R}_1 \, \exp\left(-\frac{3}{2\ell^2}\left[2\left(\boldsymbol{R}_1 - \frac{\boldsymbol{R}_2}{2}\right)^2 + \frac{\boldsymbol{R}_2^2}{2}\right]\right) \\ &= \left(\frac{3}{4\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\boldsymbol{R}_2^2}{4\ell^2}\right). \end{aligned}$$

(vi) A similar strategy could work for larger n. You could write down a Gaussian distribution for $P(\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots \boldsymbol{R}_n)$, the probability of the microstate then get the marginal distribution for the final monomer position as:

$$P(\boldsymbol{R}_n) = \int d\boldsymbol{R}_1 \int d\boldsymbol{R}_2 \ldots \int d\boldsymbol{R}_{n-1} P(\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots \boldsymbol{R}_n).$$

Doing so many Gaussian integrals could get painful, but the result must still be a Gaussian. Based on what you learned in part (i) and (ii) about the mean and variance of $\boldsymbol{R}_n$, determine this (normalized) Gaussian distribution $P(\boldsymbol{R}_n)$.

We know that $\langle \boldsymbol{R}_n \rangle = 0$ and that $\langle \boldsymbol{R}_n^2 \rangle = n\ell^2$, so, by plugging in the mean and variance of $\boldsymbol{R}_n$ into the standard Gaussian distribution

$$P(\boldsymbol{r}) = \left(\frac{3}{2\pi\sigma^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\boldsymbol{r}^2}{2\sigma^2}\right),$$

we find

$$P(\boldsymbol{R}_n) = \left(\frac{3}{2n\pi\ell^2}\right)^{\frac{3}{2}} \exp\left(-\frac{3\boldsymbol{R}_n^2}{2n\ell^2}\right).$$

Notice all the 3s that have popped up—this is a result of the distribution being three-dimensional, rather than one-dimensional, as before. You can show that this three-dimensional Gaussian distribution can be separated into three independent one-dimensional Gaussian distributions in each coordinate. In addition, as a consistency check, you should check that the $n = 2$ result agrees with your answer in 2v.

(vii) [Optional] Prove your answer to (vi) without asserting a priori that the distribution will be Gaussian.

There are various strategies. A simple proof would involve assuming that the distribution is Gaussian and then inducting on n. More elegantly, but requiring knowledge of more advanced techniques, one could change variables from $\{\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n\}$ to $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$, $\boldsymbol{r}_i := \boldsymbol{R}_i - \boldsymbol{R}_{i-1}$, and then use the fact that

$$P(\boldsymbol{R}_n) = \left\langle \delta\left(\boldsymbol{R}_n - \sum_i r_i\right)\right\rangle,$$

where the expectation value is over the $\{\boldsymbol{r}_i\}$. To evaluate this probability density, you will likely want to make use of the Fourier representation of the δ function,

$$\delta(\boldsymbol{x}) = \int \mathrm{d}\boldsymbol{\xi}\, e^{-2\pi i \boldsymbol{x}\cdot\boldsymbol{\xi}}.$$

Another approach is to use moment-generating functions: the idea here is to work with the $\mathbf{r}_i$, noting that the moment-generating function of a sum of independent random variables is the product of those of the individual random variables. Finally, we invert the resulting moment-generating function to produce the desired distribution of $\mathbf{R}_n = \sum_i \mathbf{r}_i$.