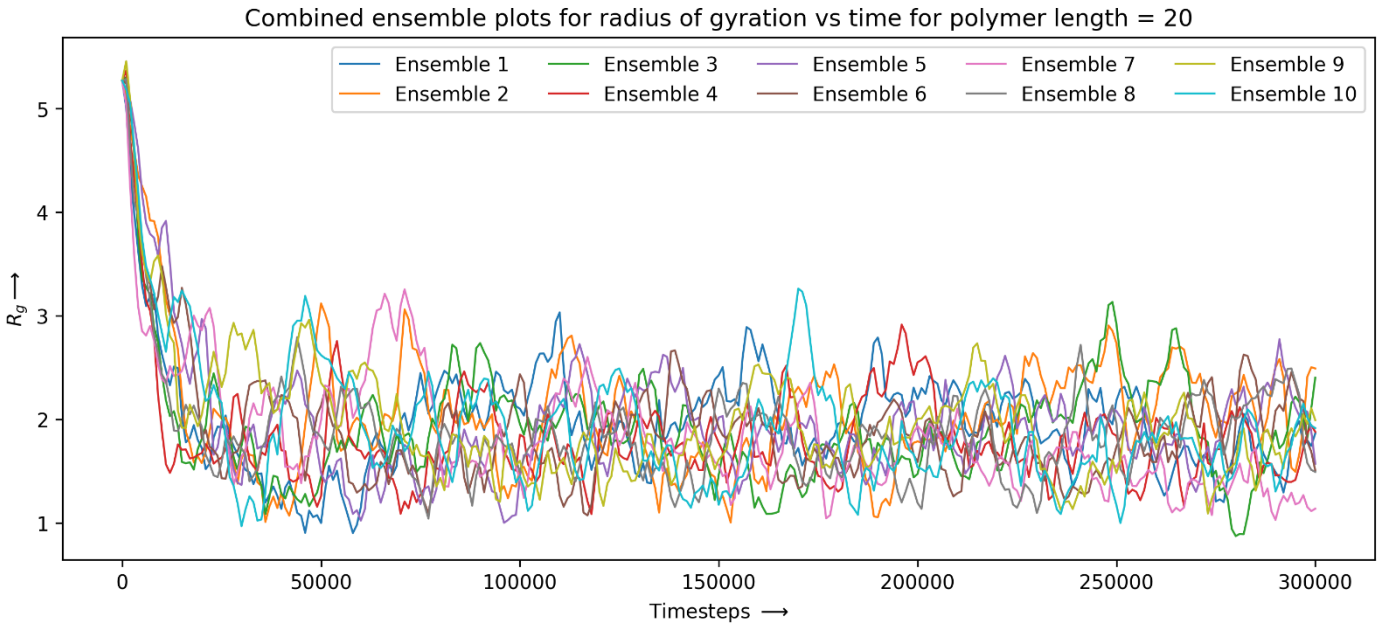
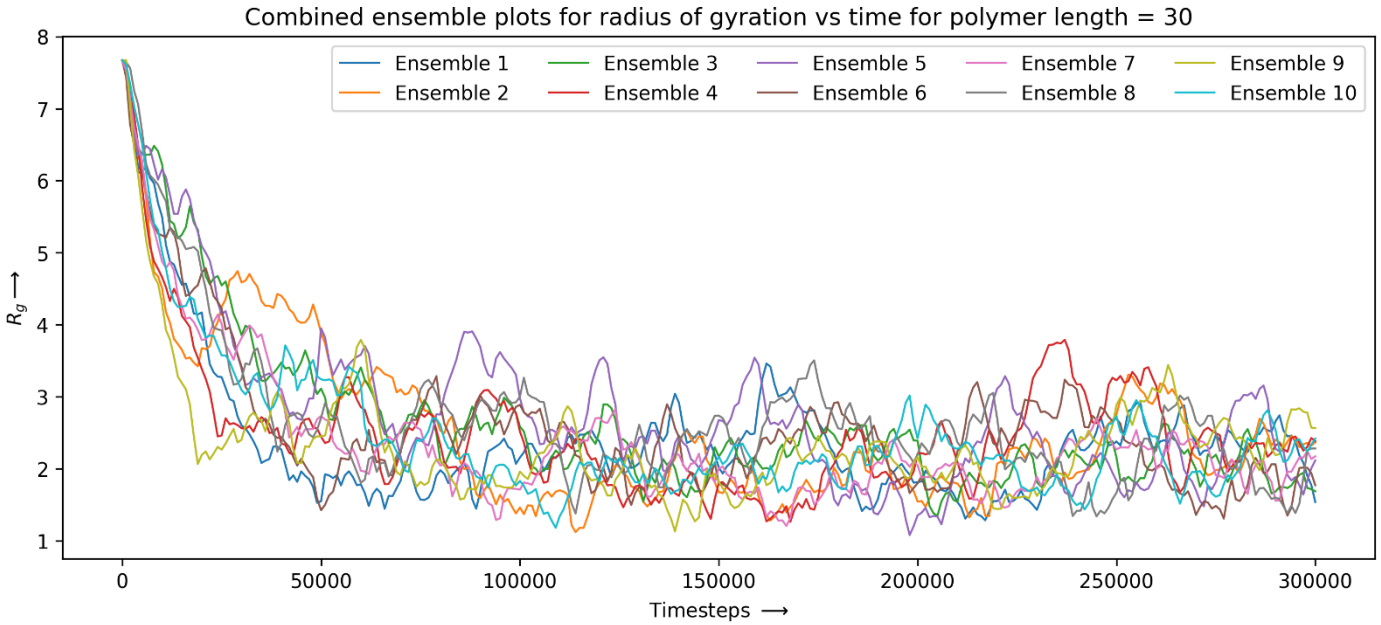
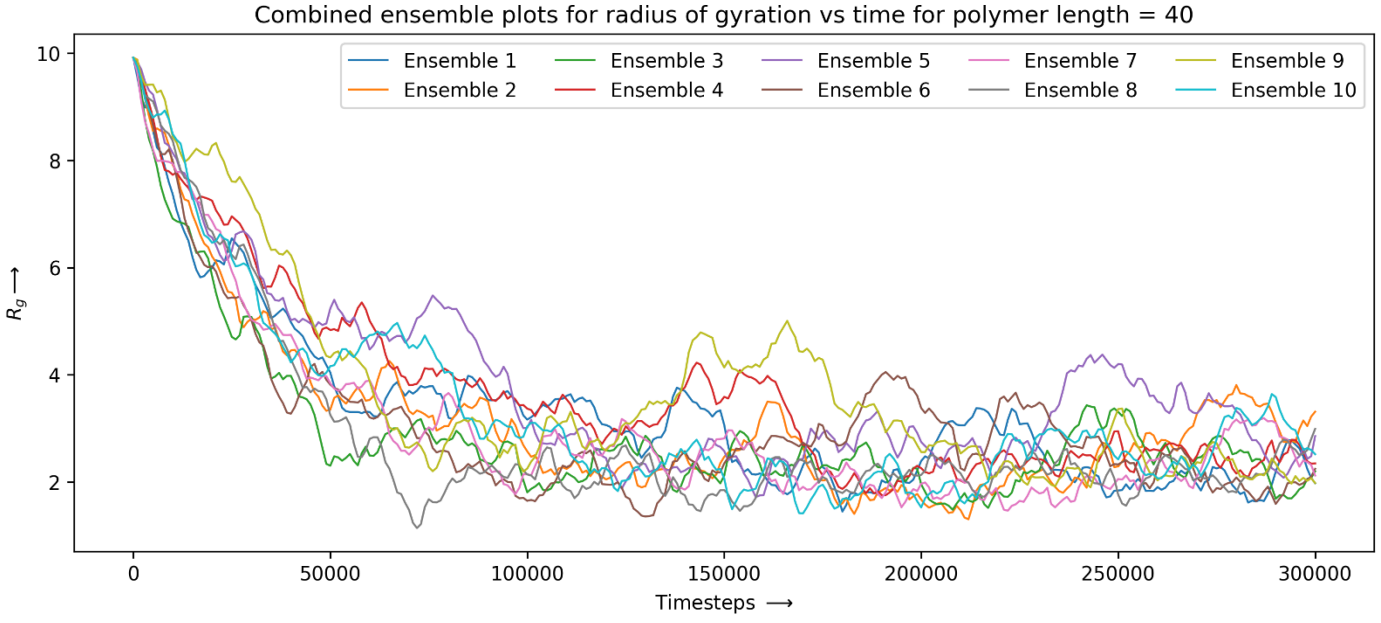
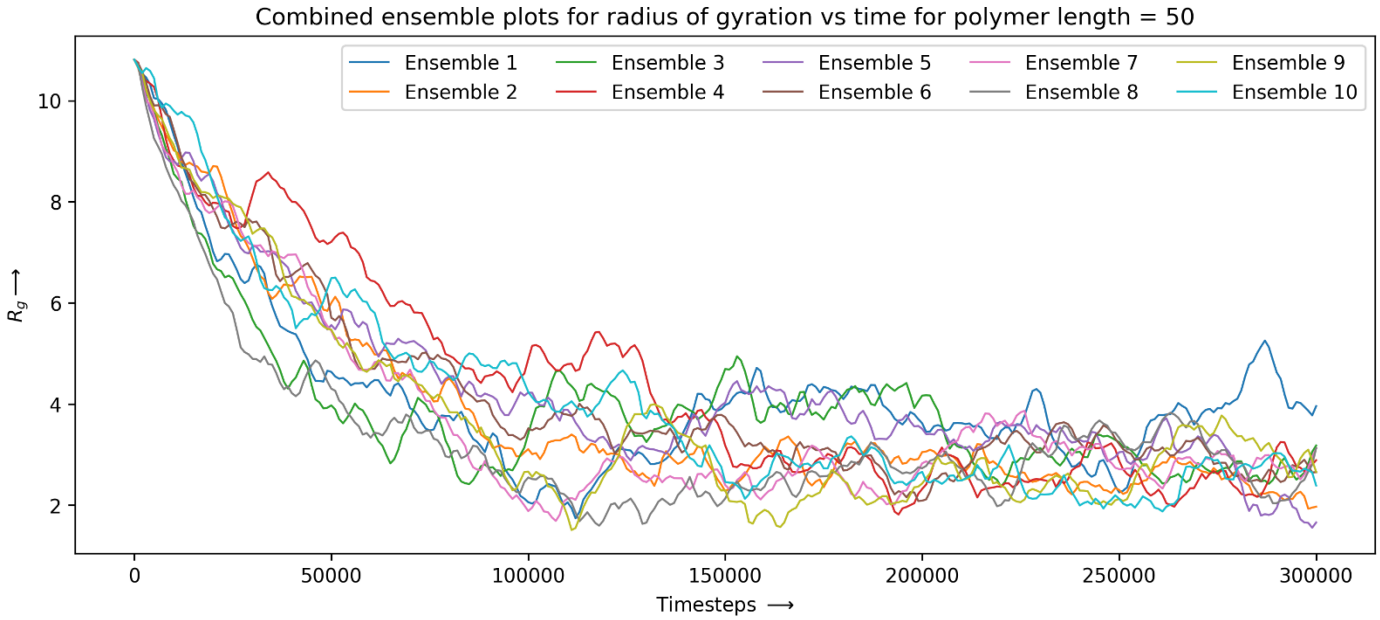
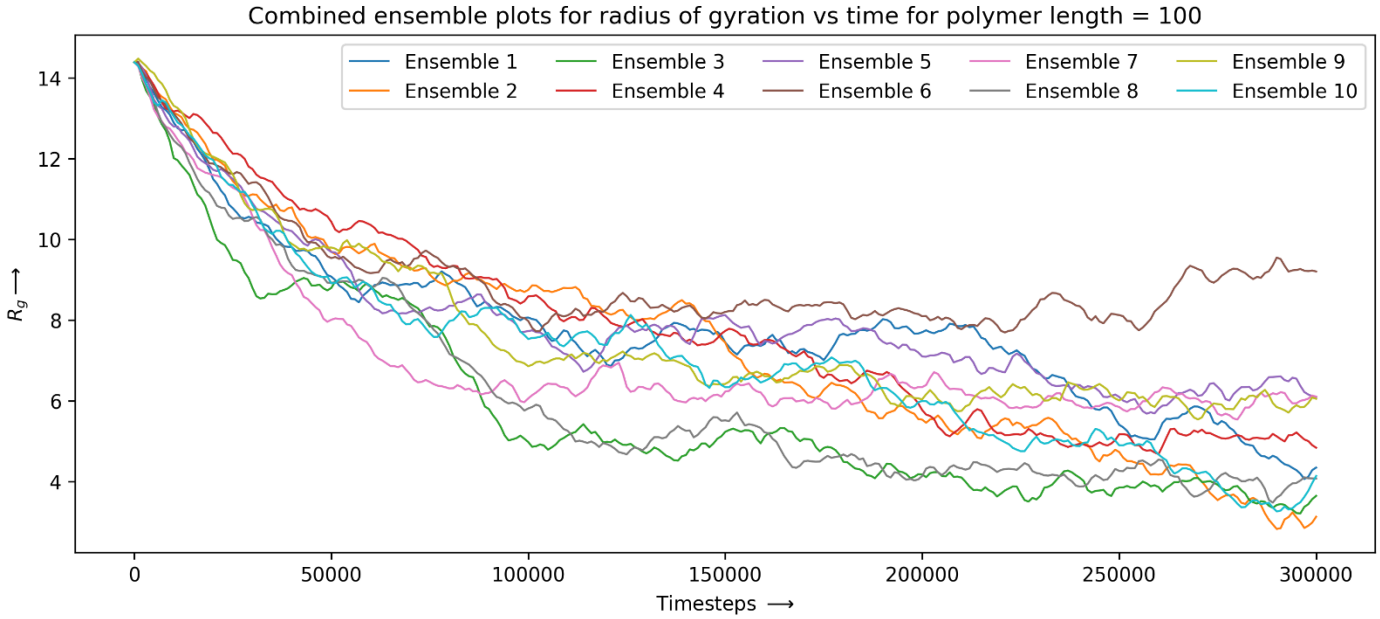
Below are the combined plots for different polymer lengths – these are here to give an idea of the system we are dealing with.



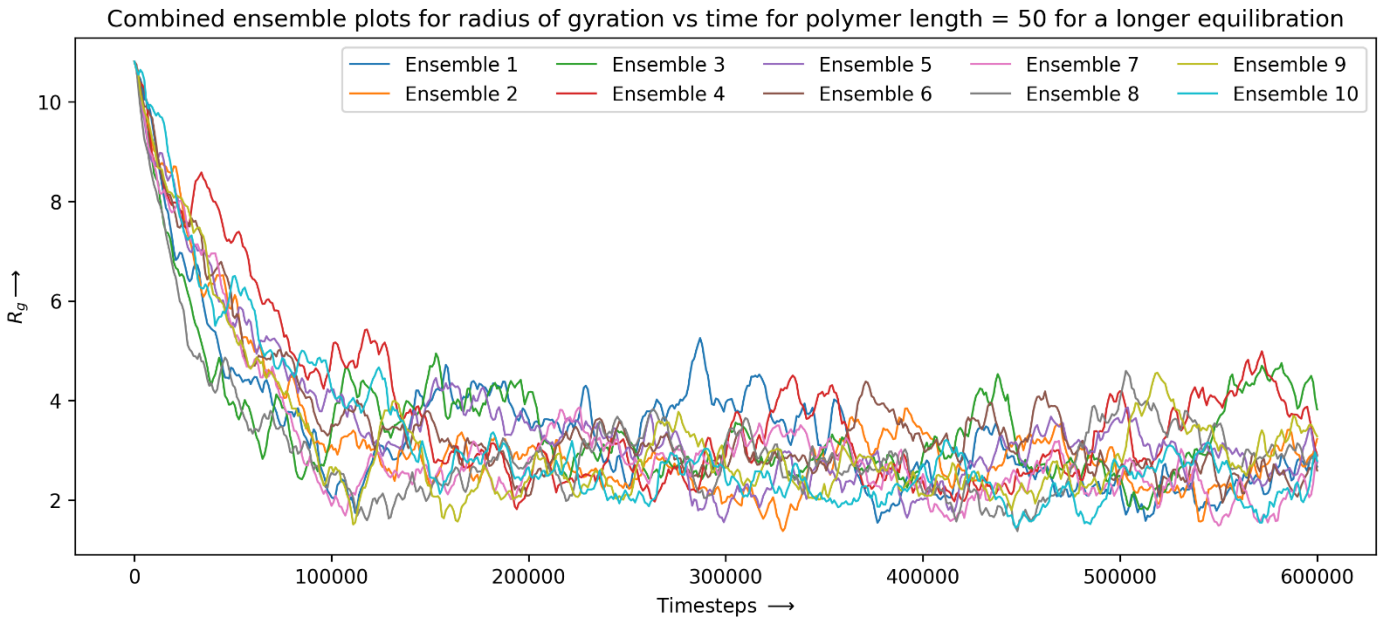


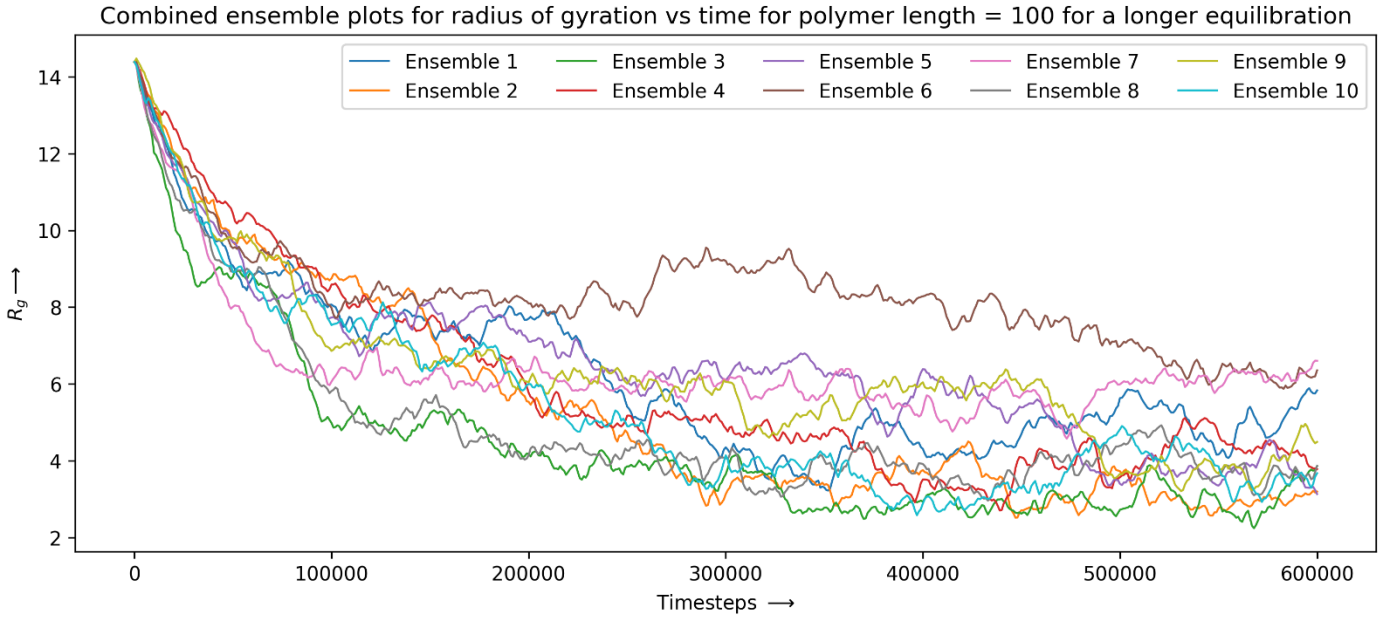






We see that the system does not seem to have reached a steady state for polymer lengths = 50 and 100. So, the simulation was run for 600,000 steps instead of 300,000 for these two cases.





In theory, the time-average values for a longer time will not change for a system in steady state, and neither will the ensemble-average if different timesteps are sampled within the steady state period. With this reason, I have hence not run additional longer simulations for N = 20, 30, 40

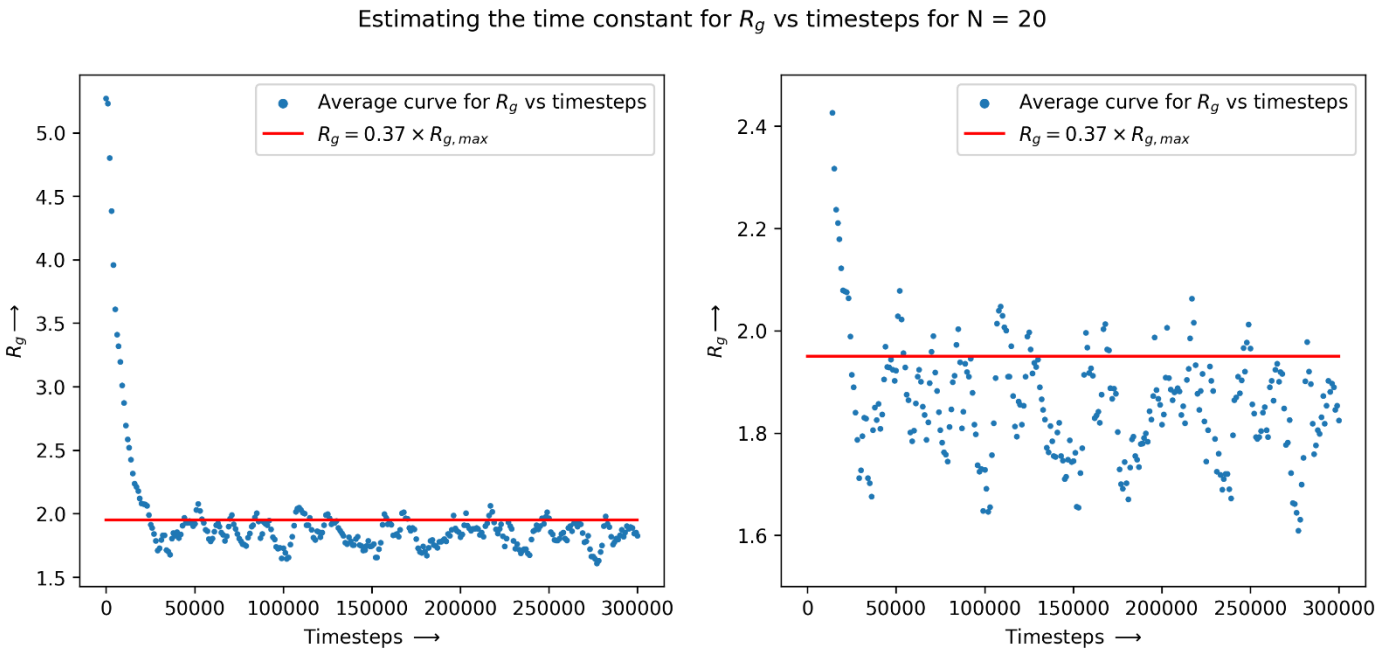
Q1.

To find the time constant after which the system can be considered equilibrated, the following function was defined:

```
def tau(y):
    t = 0
    for k in range(len(y)-1):
        if np.round(y[k], 4) < np.round(0.37*y[0], 4) and np.round(y[k+1], 4) > np.round(0.37*y[0], 4):
            t = data[k][0]
            break
    return t
```

I was facing an overflow error while attempting to fit the data to an exponential, so instead the above function was used to find the time at which the curve reaches 37% of its maximum value (this is a consequence of the first order decay exponential, since $e^{-1} = 0.37$). Here I have not imposed the condition that the returned index is that of the first time the curve crosses

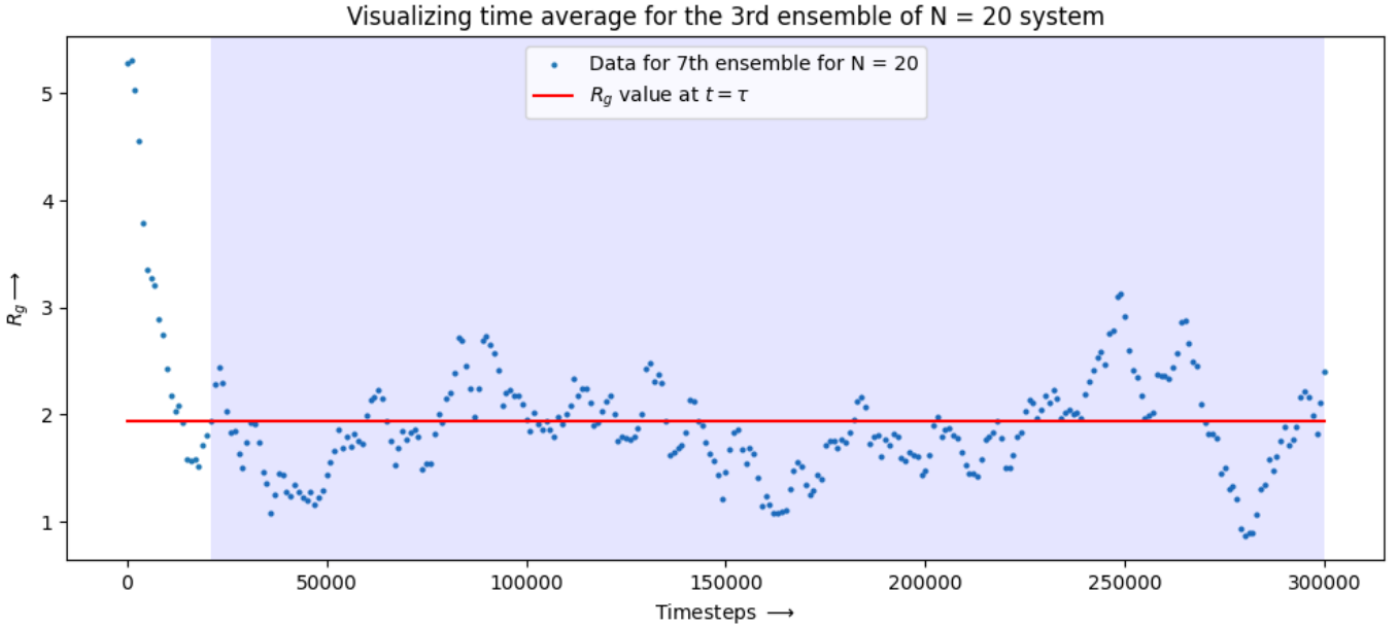
$0.37 \times R_{g,max}$ since the curve can do so easily due to fluctuations in the simulation. It is important to note that **I have used an average of all ensembles**, with my reasoning being that, if I am to choose a common time point across all ensembles, it might be better to find a 'common' time constant. This also saves time while not being entirely inaccurate. The following plots illustrates this point, with the time constant being 43000 timesteps:



Having obtained a working value of the time constant τ, I used this to calculate the ensemble average. This is done by averaging over the value of $R_g(t)$ at $t = 5\tau$, since by this time the system can be said to have equilibrated. $\langle R_g \rangle$ is thus calculated to be 1.9259 LJ units.

Q2.

For time-averaging, I will choose the 3rd ensemble for reasons I will go into in Q4. The time constant is estimated as 30000 timesteps. This can be visualized in the plot as follows, where the blue shaded region is the region in which the data points are sampled for calculation:



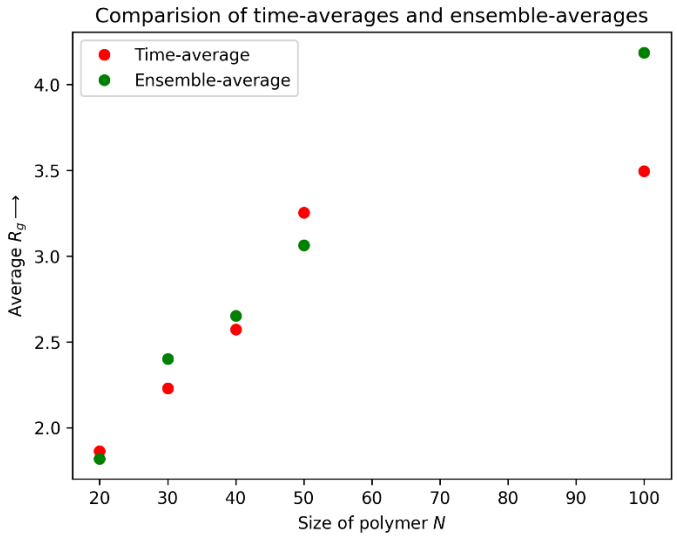
The time-average $\overline{R_g}$ is thus calculated to be 1.7975 LJ units.

Q3.

$\langle R_g \rangle = 1.9259$ and $\overline{R_g} = 1.7975$. These are reasonably close to each other, and hence the system is in accordance with the Ergodic hypothesis, which states that $\langle R_g \rangle = \overline{R_g}$.

Q4.

For the time-average, I have the taken the 3rd ensemble in each case of N in order to be consistent with Q2. This is because, if I chose any other ensemble, I would get an index out of bounds error for some case of N – this might be due to that particular ensemble for that N-size polymer system still having considerable fluctuations. This could be potentially corrected by choosing a longer equilibration time if the Ergodic hypothesis' verification was to be watertight. For N = 40, the 3rd ensemble gave a ridiculous time average (9.9183), so only for this case, I have taken the 2nd ensemble

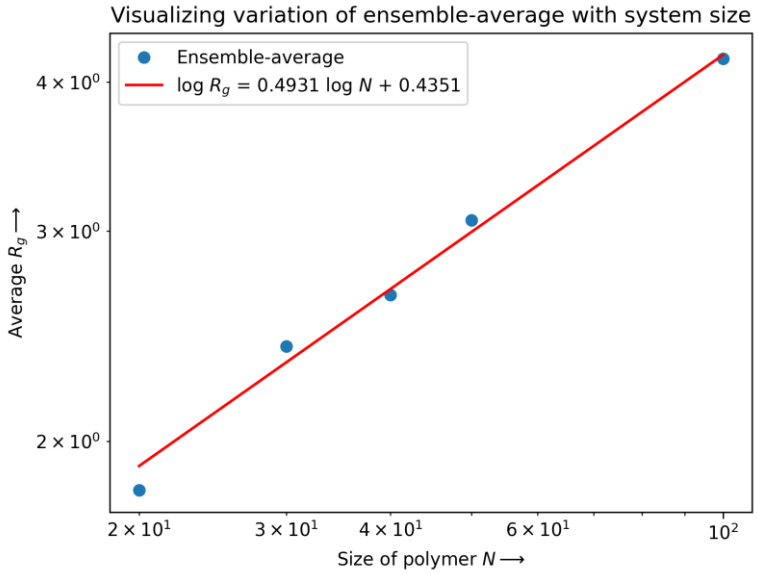


N	Time-average	Ensemble-average
20	1.8635	1.8192
30	2.2304	2.4017
40	2.5721	2.6514
50	3.2542	3.0631
100	3.4951	4.1872

We see that both these values are pretty close for most of the systems. For N = 100, I think the particular ensemble might not have reached a steady state.

Q5.

Following is a plot of $\langle R_g \rangle$ vs N on the log scale, and a linear fit to the data:



From $\langle R_g \rangle^2 = \frac{1}{6} N b^2$, we can get $\log\langle R_g \rangle = \frac{1}{2} \log N + \log C$ for some constant C. This implies that on a log scale, $\langle R_g \rangle$ and N have a linear relationship with the slope being 0.5. This is exactly what we see in our plot.