

```

123         plt.plot(range(start_point,stop_point),data[N_index,T_index,start_point:stop_
            point],label='N = {0}, T =
            {1:.3f}'.format(N_range[N_index],T_range[T_index]))
124     plt.legend(loc='upper right')
125     plt.xlabel(xname)
126     plt.ylabel(yname)
127     plt.savefig('plots/{0} vs {1} N = {2} from {3} to
            {4}.pdf'.format(yname,xname,N_range[N_index],start_point,stop_point))
128     plt.figure()
129
130
131 # In[ ]:
132
133
134 # Function to calculate the average of a thermodynamic variable
135 def calculate_thermodynamic_variable(data, N_range, T_range, no_of_equilibrating_sweeps):
136     var = np.zeros((len(N_range),len(T_range)))
137
138     for N_index in range(len(N_range)):
139         for T_index in range(len(T_range)):
140             var[N_index,T_index] =
                np.sum(data[N_index,T_index,no_of_equilibrating_sweeps:])/len(data[N_index,T_
                    index,no_of_equilibrating_sweeps:])
141             var[N_index,:] = normalize_data(var[N_index,:],N_range[N_index])
142
143     return var
144
145
146 # In[ ]:
147
148
149 # Function to calculate the average of a derivative thermodynamic variable
150 def calculate_derivative_thermodynamic_average(data, N_range, T_range, power_of_T,
no_of_equilibrating_sweeps):
151     var = np.zeros((len(N_range),len(T_range)))
152
153     for N_index in range(len(N_range)):
154         for T_index in range(len(T_range)):
155             ave =
                np.sum(data[N_index,T_index,no_of_equilibrating_sweeps:])/len(data[N_index,T_
                    index,no_of_equilibrating_sweeps:])
156             squared_ave =
                np.sum(data[N_index,T_index,no_of_equilibrating_sweeps:]**2)/len(data[N_index
                    ,T_index,no_of_equilibrating_sweeps:])
157             var[N_index,T_index] = (squared_ave-ave**2)/(T_range[T_index]**power_of_T)
158             var[N_index,:] = normalize_data(var[N_index,:],N_range[N_index])
159
160     return var
161
162
163 # In[ ]:
164
165
166 # Function to plot thermodynamic variable against temperature
167 def plot_temperature_dependence(data, N_range, N_index_range, T_range, T_index_range,
xname, yname):
168     for N_index in N_index_range:
169
170         plt.plot(T_range[T_index_range[0]:(T_index_range[-1]+1)],data[N_index,T_index_ran
            ge[0]:(T_index_range[-1]+1)],'-o',label='N = {0}'.format(N_range[N_index]))
171         # +1 to include last element
172         plt.legend(loc='best')
173         plt.xlabel(xname)
174         plt.ylabel(yname)
175         plt.savefig('plots/{0} vs {1} from {2:.3f} to
            {3:.3f}.pdf'.format(yname,xname,T_range[T_index_range[0]],T_range[T_index_range[-1]]))

```