

Statistical Mechanics Final Project: Study of the competitive
adsorption of CO and CH₄ on Cu (111)

Candidate Number: 1090111

January 2025

Question 0

Finish the tutorial started in Class 4 and report the results here, i.e. obtain the CO coverage on Cu(111) from KMC simulations at 300 K and pressures ranging from 0.01 bar to 10^6 bar, and compare it with the Langmuir adsorption isotherm.

The KMC simulation data corresponding to the pressure range of 0.01 - 10^6 bar, equi-spaced on a logarithmic scale, were read into an Excel spreadsheet to determine the CO coverage on Cu(111). The data were then pulled from the Excel sheet by Python in order to be compared to the Langmuir adsorption isotherm given by:

$$\theta_{CO} = \frac{KP}{1 + KP} \quad (1)$$

The code for this process is found below.

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 # Read the average coverage of CO on Cu(111) from the Excel sheet
6 data = pd.ExcelFile('~\Desktop\TCC_SM_\Project\KMC\KMC_CO.xlsx')
7 KMC_CO_data_sheet = data.parse('Sheet1')
8 pressure_column = KMC_CO_data_sheet.columns[0]
9 coverage_column = KMC_CO_data_sheet.columns[3]
10
11 # Selecting rows 6-14 (inclusive)
12 filtered_KMC_CO_data = KMC_CO_data_sheet.iloc[4:15]
13 filtered_KMC_CO_data = filtered_KMC_CO_data[[pressure_column, coverage_column]]
14
15 # Rename columns
16 filtered_KMC_CO_data.columns = ['p', 'Coverage per site']
17
18 # Define constants used
19 delta_g = 0.106 # eV
20 T = 300 # K
21 boltzmann = 8.6173e-05 # eV / K
22 P = np.logspace(-2, 6, num=500)
23
24 # Calculate theta using the Langmuir isotherm expression
25 constant = -delta_g / (T * boltzmann)
26 K = np.exp(constant)
27 theta = (K * P) / (1 + (K * P))
28
29 # Plot the results
30 plt.figure(figsize=(10, 8))
31 plt.plot(P, theta, label='Langmuir isotherm', color='red')
32 plt.plot(filtered_KMC_CO_data['p'], filtered_KMC_CO_data['Coverage per site'], label='KMC Data',
33 color='red', marker='o', linestyle='')
34 plt.xlabel('Pressure (bar)', fontsize=14, weight='bold')
35 plt.ylabel('Surface coverage', fontsize=14, weight='bold')
36 plt.title('Surface coverage vs pressure', fontsize=16, weight='bold')
37
38 # Custom x-axis scale
39 plt.xscale("log")
40
41 # Manually specify major ticks
42 major_ticks = [1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3, 1e4, 1e5, 1e6]
43 plt.xticks(major_ticks, [f"$10^{{{int(np.log10(tick))}}}$" for tick in major_ticks])
44
45 # Grid and layout
46 plt.grid(True, which="both", linestyle="--", linewidth=0.5)
47 plt.legend(loc='upper left', fontsize=8)
```

```

48 plt.tight_layout()
49 plt.show()

```

The results of both methods were then plotted on the same graph to show how the simulated KMC data for CO coverage on Cu(111) generally tend to follow the Langmuir adsorption isotherm trends. This plot is found below.

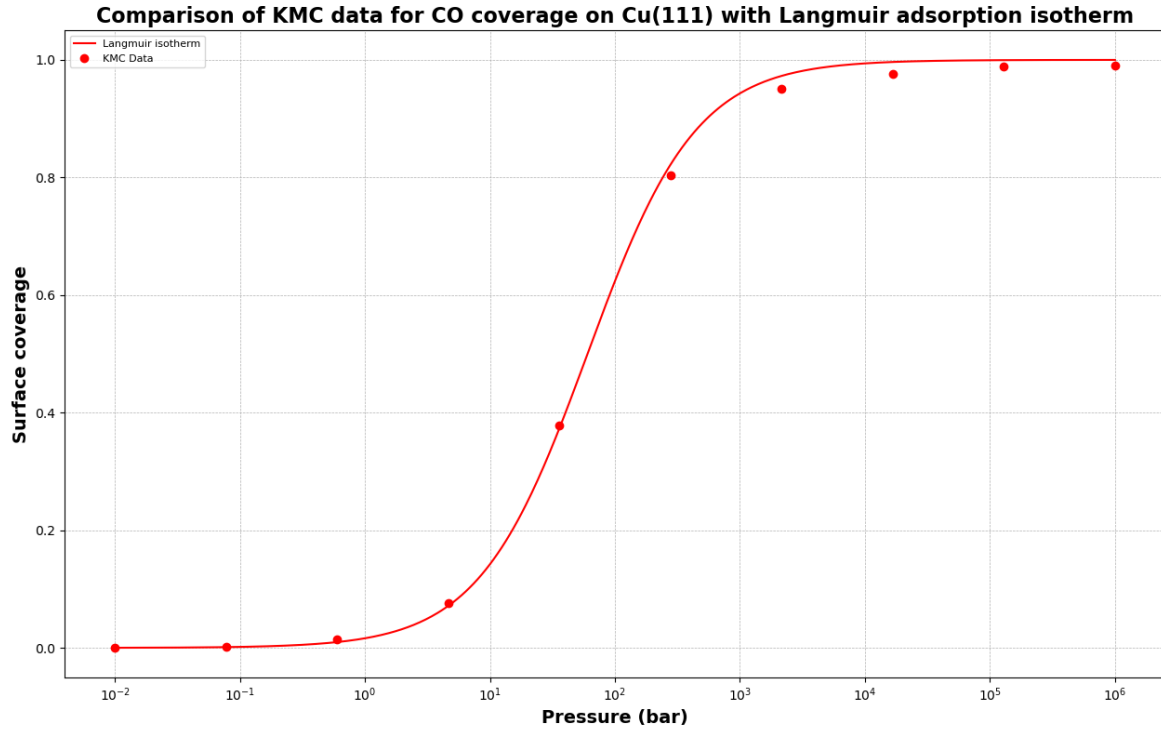


Figure 1: Comparison of KMC data for CO coverage on Cu(111) with the Langmuir adsorption isotherm.

As expected, surface coverage is close to zero for very low pressures for both the Langmuir isotherm and the KMC data. When P is very small, $KP \ll 1$. Hence, $\theta \approx KP$ and since P is low, θ will also be small due to its linear dependence on P . At low pressure, there are fewer adsorbate molecules in the vicinity of the surface, resulting in fewer collisions and adsorption events, keeping coverage low.

When the pressure reaches 1 bar, the surface coverage slowly rises due to an increase in adsorbate molecules arriving at and sticking to the surface as seen in Figure 1. However, once the pressure reaches 10 bar, surface coverage sees a sharp increase. This is due to there being more adsorbate molecules per unit volume, causing more frequent collisions with the surface. At moderate to high pressures, the KP term becomes large and so $\theta \approx 1$. Consequently, at extremely high pressures of approximately 10^3 , nearly all surfaces are covered and the curve starts to level off.

Question A

Following the same procedure used for CO in the tutorial of Class 2, calculate ΔE_{ads} of CH₄ on Cu(111), and show that CH₄ physisorbs on Cu(111).

The code for this question is as follows:

```
1  from copy import deepcopy
2  from ase.build import add_adsorbate, fcc111, molecule
3  from ase.calculators.emt import EMT
4  from ase.constraints import FixAtoms
5  from ase.optimize import QuasiNewton
6  from ase.visualize import view
7  from ase.thermochemistry import IdealGasThermo, HarmonicThermo
8
9  # Get the energy of a CH4 molecule
10 CH4_mol = molecule("CH4")
11 CH4_mol.calc = EMT()
12 energy_CH4_gas = CH4_mol.get_potential_energy()
13
14 # Create a Cu(111) slab and get energy
15 slab = fcc111("Cu", size=(4,4,2), vacuum=10.0)
16 slab.calc = EMT()
17 energy_slab = slab.get_potential_energy()
18
19 # Run geometry optimisation of CH4 on Cu(111) slab and print adsorption energy
20 CH4_ads = deepcopy(slab)
21 add_adsorbate(slab=CH4_ads, adsorbate=CH4_mol, height=4.0, position=(3.82, 2.21))
22 constraint_CH4 = FixAtoms(mask=[atom.symbol == "Cu" for atom in CH4_ads])
23 CH4_ads.set_constraint(constraint_CH4)
24 dyn = QuasiNewton(CH4_ads, trajectory="CH4_Cu(111).traj")
25 dyn.run(fmax=0.05)
26 energy_CH4_ads = CH4_ads.get_potential_energy()
27 view(CH4_ads)
28
29 # Calculate adsorption energy
30 adsorption_energy_CH4 = energy_CH4_ads - (energy_slab + energy_CH4_gas)
31 print(f"Adsorption energy of CH4 on Cu(111): {adsorption_energy_CH4: .3f} eV")
```

This will give the following output:

```
1  Adsorption energy of CH4 on Cu(111): -0.239 eV
```

The optimised configuration of CH₄ on Cu(111) is shown below:

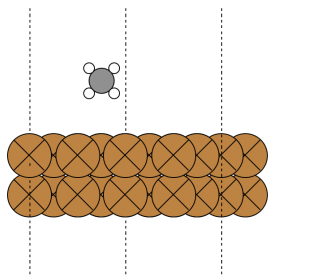


Figure 2: Physisorption of CH₄ on Cu(111) in the XZ-plane.

From Figure 2, it is clear that the CH₄ molecule is physisorbed because it is not directly attached to the Cu (111) surface. The gap between the Cu(111) surface and the CH₄ molecule is due to the weak Van der Waal's forces as a result of physisorption.

For contrast, the optimised configuration of CO on Cu(111) is shown below:

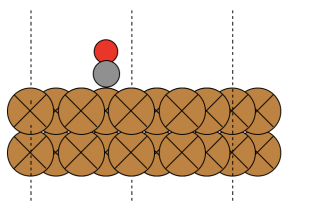


Figure 3: Chemisorption of CO on Cu(111) in the XZ-plane.

The lower adsorption energy of CO over CH₄ means it is more tightly bound to the Cu(111) surface and an energy barrier needs to be overcome in order for this adsorbed species to leave the lattice site. From figure 3, it is clear that CO chemisorbs onto the Cu(111) surface as the CO molecule is visibly attached to the surface as a result of (weak) chemisorption.

Question B

Calculate $\Delta G_{ads}^{CH_4,o}(T)$ of CH₄ on Cu(111) at 300 K and 1 bar. As for the case of CO, use the harmonic oscillator model and assume that $G_{slab+CH_4}(T, P) \approx A_{slab+CH_4}(T)$ and $G_{slab}(T, P) \approx E_{slab}$.

In addition to the code in Part A above, the code below will calculate the adsorption free energy for methane on a Cu(111) surface.

```

1  # Calculate Gibb's Free energy of adsorption of CH4 on Cu(111) at 300K and 1 bar
2  vib_energy_CH4_gas = [0.3843, 0.3840, 0.3840, 0.3685, 0.1881, 0.1879,
3                        0.1595, 0.1593, 0.1592]
4  vib_energy_CH4_ads = [0.3815, 0.3758, 0.3758, 0.3625, 0.1850, 0.1848,
5                        0.1589, 0.1584, 0.1559, 0.0161, 0.0161, 0.0112, 0.0061, 0.0061, 0.0061]
6  thermo_CH4_gas = IdealGasThermo(vib_energies=vib_energy_CH4_gas,
7                                  geometry="nonlinear",
8                                  potentialenergy=energy_CH4_gas,
9                                  atoms=CH4_molec,
10                                 symmetrynumber=12,
11                                 spin=0)
12  thermo_CH4_ads = HarmonicThermo(vib_energies=vib_energy_CH4_ads, potentialenergy=energy_CH4_ads)
13  temp = 300
14  pressure = 1.0e+5
15
16  g_CH4_gas = thermo_CH4_gas.get_gibbs_energy(temperature=temp, pressure=pressure, verbose=False)
17  g_CH4_ads = thermo_CH4_ads.get_helmholtz_energy(temperature=temp, verbose=False)
18  g_slab = energy_slab
19  Pa_to_bar = 1.0e-5
20  adsorption_free_energy_CH4 = g_CH4_ads - (g_slab + g_CH4_gas)
21  print(f"Adsorption free energy of CH4 on Cu(111):
22  {adsorption_free_energy_CH4: .3f} eV")

```

This will give the following output:

```

1  Adsorption free energy of CH4 on Cu(111): 0.061 eV

```

CH₄ has a lower Gibbs Free Energy of adsorption than CO under these conditions, so CH₄ is thermodynamically more favourable to adsorb. As a result, CH₄ occupies more surface sites on average, leading to higher surface coverage. In turn, this affects the selectivity of CO over CH₄. This argument is proven in parts E and F where KMC and MMC simulations will demonstrate higher surface coverage of CH₄ on a Cu(111) surface over CO.

Question C

The selectivity of CO over $\text{CH}_4(g)$, $S_{\text{CO}/\text{CH}_4}$, can be estimated from the equilibrium constants for adsorption, K_i , as follows:

$$S_{\text{CO}/\text{CH}_4} = \frac{K_{\text{CO}}}{K_{\text{CH}_4}} = \frac{\exp(-\frac{\Delta G_{\text{ads}}^{\text{CO},o}}{k_B T})}{\exp(-\frac{\Delta G_{\text{ads}}^{\text{CH}_4,o}}{k_B T})} \quad (2)$$

where

$$K_i = \frac{1}{P^o} \exp(-\frac{\Delta G_{\text{ads}}^{i,o}}{k_B T}) \quad (3)$$

Plot $S_{\text{CO}/\text{CH}_4}$ as a function of T in the range T = 100-1000 K and discuss the results.

The selectivity of CO over CH_4 as a function of T can be found in Figure 4, where the selectivity has been plotted on both a linear and a logarithmic scale to highlight the distinct trends at different temperatures. The behaviour of the selectivity curve is influenced by the temperature dependence of the equilibrium constants, which are controlled by the Gibbs Free Energy of adsorption ($\Delta G_{\text{ads}}^{i,o}$) of CO and CH_4 at 0.106 and 0.061 eV respectively.

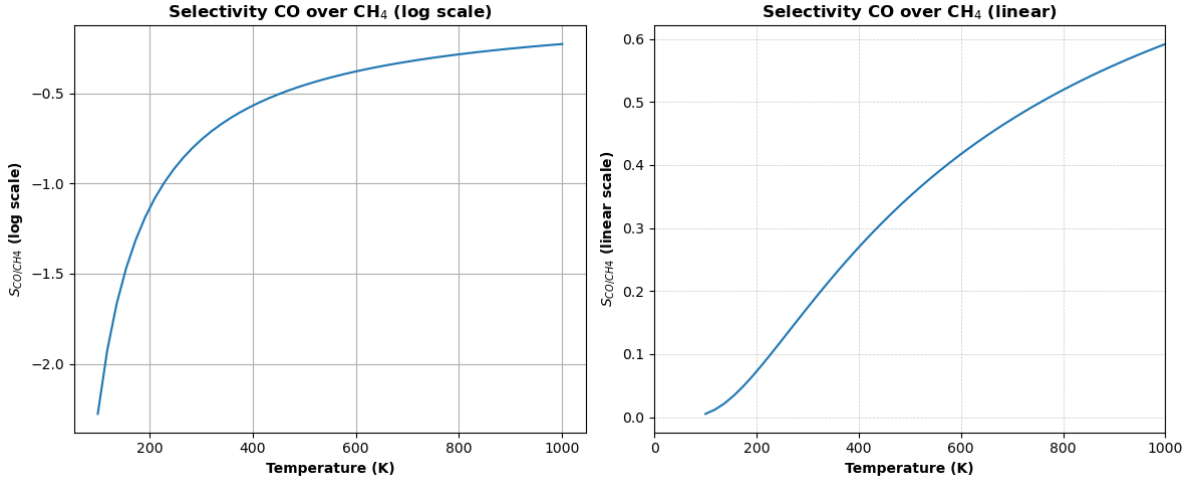


Figure 4: Selectivity of CO over CH_4 as a function of T

In the temperature region of 100 - 300 K, a sharp increase in selectivity is observed. This is because $k_b T$ is small at low temperatures, making $\exp(-\Delta G_{\text{ads}}^{i,o}/k_b T)$ highly sensitive to small changes in T. The free energy difference between CO and CH_4 , with $\Delta G_{\text{ads}}^{\text{CO},o}$ being larger than $\Delta G_{\text{ads}}^{\text{CH}_4,o}$, causes K_{CO} to increase more significantly than K_{CH_4} as T rises. Consequently, the selectivity increases sharply.

As the temperature increases from 300 - 800 K, the thermal energy $k_b T$ becomes comparable to the Gibbs Free Energy of adsorption. Hence, the sensitivity of the exponent to T is reduced. Both K_{CO} and K_{CH_4} are still changing, but they are changing slower and the selectivity may decrease slightly if CH_4 's coverage becomes more competitive. Eventually, K_{CO} and K_{CH_4} reach near-asymptotic values, giving a plateau in selectivity as seen in Figure 4, plot 1.

In summary, CO has a stronger interaction with the CU(111) surface, reflected by its larger Gibbs Free Energy of adsorption, making it more dominant at low temperatures. At higher temperatures, thermal desorption weakens both coverages, so the final selectivity plateaus at a value of approximately 0.6.

Question D

Under the Langmuir adsorption isotherm for competitive adsorption of multiple species, the surface coverage θ_i of species i is given by:

$$\theta_i = \frac{K_i P_i}{1 + \sum_j K_j P_j} \quad (4)$$

Demonstrate that, for competitive CO and CH₄ adsorption, the following relationship between selectivity and coverage exists:

$$\frac{\theta_{CO}}{\theta_{CH_4}} = S_{CO/CH_4} * \left(\frac{P_{CO}}{P_{CH_4}} \right) \quad (5)$$

Take $i = \text{CO}$ and $j = \text{CH}_4$ such that $\sum_k K_k P_k = K_{CO} P_{CO} + K_{CH_4} P_{CH_4}$. Hence,

$$\theta_{CO} = \frac{K_{CO} P_{CO}}{1 + (K_{CO} P_{CO} + K_{CH_4} P_{CH_4})} \quad (6)$$

and

$$\theta_{CH_4} = \frac{K_{CH_4} P_{CH_4}}{1 + (K_{CO} P_{CO} + K_{CH_4} P_{CH_4})} \quad (7)$$

giving:

$$\frac{\theta_{CO}}{\theta_{CH_4}} = \frac{K_{CO} P_{CO}}{1 + (K_{CO} P_{CO} + K_{CH_4} P_{CH_4})} * \frac{1 + (K_{CO} P_{CO} + K_{CH_4} P_{CH_4})}{K_{CH_4} P_{CH_4}} \quad (8)$$

The terms originating from the summation cancel out, leading to

$$\frac{\theta_{CO}}{\theta_{CH_4}} = \frac{K_{CO} P_{CO}}{K_{CH_4} P_{CH_4}} \quad (9)$$

but since $S_{CO/CH_4} = K_{CO}/K_{CH_4}$, we can rewrite equation 9 to give

$$\frac{\theta_{CO}}{\theta_{CH_4}} = S_{CO/CH_4} * \left(\frac{P_{CO}}{P_{CH_4}} \right) \quad (10)$$

as stated in the question.

Question E

Using the ZacroS code, run a KMC simulation to calculate the CO/CH₄ selectivity on a Cu(111) surface at 300 K and $P_{CO} = P_{CH_4} = 5$ bar using the given input files.

Running the KMC simulation at 300 K and 5 bar resulted in obtaining an output file from which the average surface coverage for CO and CH₄ can be calculated.

The average surface coverage was calculated using the equation below:

$$\langle \theta_i \rangle = \frac{\text{sum of number of adsorbed CO* species}}{\text{number of lattice sites} * \text{number of entries}} \quad (11)$$

Applying this equation to CO and CH₄ gives

$$\langle \theta_{CO} \rangle = \frac{2777}{100 * 500} = 0.05554 \quad (12)$$

$$\langle \theta_{CH_4} \rangle = \frac{14957}{100 * 500} = 0.29914 \quad (13)$$

Now use equation (10) to calculate the CO/CH₄ selectivity on a Cu(111) surface at 300 K.

$$\frac{0.05554}{0.29914} = S_{CO/CH_4} * \left(\frac{5}{5}\right) \quad (14)$$

$$0.1857 = S_{CO/CH_4} \quad (15)$$

Question F

Write a Metropolis Monte Carlo code to calculate the CO/CH₄ selectivity on a Cu(111) surface at 300 K and $P_{CO} = P_{CH_4} = 5$ bar.

- Calculate the average coverages $\langle \theta_{CO} \rangle$ and $\langle \theta_{CH_4} \rangle$ from the production phase.
- Compute the selectivity S_{CO/CH_4} and compare it with the selectivity calculated from the ratio of the equilibrium constants and with the selectivity obtained from the KMC simulation.

The code for this question can be found below:

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  import random
4
5  # Set random seeds for reproducibility reasons
6  random.seed(13)
7  np.random.seed(13)
8
9  # Define constants
10 T = 300                # temperature in K
11 G_ads_CO = 0.106      # CO adsorption free energy
12 G_ads_CH4 = 0.061     # CH4 adsorption free energy
13 P_CO = 5.0            # Partial pressure CO in bar
14 P_CH4 = 5.0           # Partial pressure CH4 in bar
15 P0 = 1.0              # Reference pressure in bar
16 kB = 8.617333262145e-05 # Boltzmann constant (ev / K)
17 kBT = kB * T          # Thermal energy
18

```



```

19 # Determine adsorption probabilities
20 P_ads_CO = min(1, (P_CO / P0) * np.exp(-G_ads_CO / kBT))
21 P_ads_CH4 = min(1, (P_CH4 / P0) * np.exp(-G_ads_CH4 / kBT))
22
23 # Determine the desorption probabilities
24 P_des_CO = min(1, np.exp(G_ads_CO / kBT))
25 P_des_CH4 = min(1, np.exp(G_ads_CH4 / kBT))
26
27 # Determine the exchange probabilities
28 # Exchange from CO to CH4
29 delta_G_CO_to_CH4 = G_ads_CH4 - G_ads_CO
30 P_exchange_CO_CH4 = min(1, np.exp(-delta_G_CO_to_CH4 / kBT))
31
32 # Exchange from CH4 to CO
33 delta_G_CH4_to_CO = G_ads_CO - G_ads_CH4
34 P_exchange_CH4_CO = min(1, np.exp(-delta_G_CH4_to_CO / kBT))
35
36 # Build the lattice
37 L = 10
38 lattice = np.zeros((L, L), dtype=int)
39
40 # Set up steps and storage of data
41 N_steps = 10**6
42 N_snapshot = 10**3
43 equilibration_steps = N_steps // 2 # define equilibration phase
44 production_steps = N_steps // 2 # define production phase
45
46 snapshots = [] # store snapshots of the lattice
47 CO_count = [] # count the CO molecules during the production stage
48 CH4_count = [] # count the CH4 molecules during the production stage
49 selectivity_list = []
50
51 for step in range(1, N_steps + 1):
52     # first select a random site to work on
53     random_row = np.random.randint(0, L)
54     random_column = np.random.randint(0, L)
55     current_state = lattice[random_row, random_column]
56
57     # determine the possible actions based on the current state
58     # empty site; decide between CO or CH4 adsorption
59     if current_state == 0:
60         action = random.choices(['adsorb_CO', 'adsorb_CH4'], weights=[0.5, 0.5], k=1)[0]
61
62         if action == 'adsorb_CO':
63             rand_val = random.random()
64             if rand_val < P_ads_CO:
65                 # if random generated number is less than CO adsorption probability, change lattice site to 1
66                 lattice[random_row, random_column] = 1
67         elif action == 'adsorb_CH4':
68             rand_val = random.random()
69             if rand_val < P_ads_CH4:
70                 lattice[random_row, random_column] = 2
71                 # if random generated number is less than CH4 adsorption probability, change lattice site to 2
72
73     # site occupied by CO; decide between CO desorption or exchange with CH4
74     elif current_state == 1:
75         action = random.choices(['desorb_CO', 'exchange_to_CH4'], weights=[0.5, 0.5], k=1)[0]
76
77         if action == 'desorb_CO':
78             rand_val = random.random()
79             if rand_val < P_des_CO:
80                 lattice[random_row, random_column] = 0
81                 # if random generated number is less than CO desorption probability, go back to an empty lattice

```

```

82         elif action == 'exchange_to_CH4':
83             rand_val = random.random()
84             if rand_val < P_exchange_CO_CH4:
85                 lattice[random_row, random_column] = 2
86                 # if random generated number is less than probability of exchanging CO with CH4, CH4 is adsorped
87
88             # site occupied by CH4; decide between CH4 desorption or exchange with CO
89         elif current_state == 2:
90             action = random.choices(['desorb_CH4', 'exchange_to_CO'], weights=[0.5, 0.5], k=1)[0]
91
92             if action == 'desorb_CH4':
93                 rand_val = random.random()
94                 if rand_val < P_des_CH4:
95                     lattice[random_row, random_column] = 0
96                     # if random generated number is less than CH4 desorption probability, go back to an empty lattice
97             elif action == 'exchange_to_CO':
98                 rand_val = random.random()
99                 if rand_val < P_exchange_CH4_CO:
100                     lattice[random_row, random_column] = 1
101                     # if random generated number is less than probability of exchanging CH4 with CO, CO is adsorped
102
103             # take snapshots at the specified intervals
104             if step % N_snapshot == 0:
105                 snapshots.append(lattice.copy())
106
107             # when entering the second half of the Monte Carlo steps,
108             # store the collected data from the production phase
109             if step > equilibration_steps:
110                 CO = np.sum(lattice == 1)          # sum all the CO molecules on the lattice
111                 CH4 = np.sum(lattice == 2)         # sum all the CH4 molecules on the lattice
112                 CO_count.append(CO)
113                 CH4_count.append(CH4)
114
115                 if CH4 > 0:
116                     selectivity = CO / CH4
117                 else:
118                     selectivity = np.inf # Avoid division by zero
119
120                 selectivity_list.append(selectivity)
121
122             # Now calculate the average surface coverages
123             avg_CO = np.mean(CO_count)
124             avg_CH4 = np.mean(CH4_count)
125
126             # Calculate selectivity based on average surface coverage
127             finite_selectivity = [s for s in selectivity_list if np.isfinite(s)]
128             if finite_selectivity:
129                 avg_selectivity = np.mean(finite_selectivity)
130             else:
131                 avg_selectivity = np.inf
132
133             # print the CO and CH4 surface coverage and the selectivity
134             print(f"Average CO surface coverage: {avg_CO:.4f}")
135             print(f"Average CH4 surface coverage: {avg_CH4:.4f}")
136             print(f"Selectivity (CO/CH4): {avg_selectivity:.4f}")

```

Determining the average coverages $\langle \theta_{CO} \rangle$ and $\langle \theta_{CH_4} \rangle$ and the selectivity using the Metropolis Monte Carlo simulation gives the following output:

```

1 Average CO surface coverage: 5.3107
2 Average CH4 surface coverage: 30.4486
3 Selectivity (CO/CH4): 0.1802

```

The selectivity obtained above can be compared to the one calculated from the ratio of the equilibrium constants from the plot in Figure 4 using the code below:

```

1  # Determine the selectivity at x = 300 K
2  x_target = 300
3  y_target = np.interp(x_target, T, np.log10(S_CO_CH4))
4
5  # Print the selectivity of CO over CH4 at 300 K
6  print(f"Selectivity (linear scale) at {x_target} K: {10**y_target:.4f}")

```

Giving the following output:

```

1  Selectivity (linear scale) at 300 K: 0.1740

```

A useful side-by-side overview of the different methods used and their selectivity values can be found in Table 1.

Table 1: Comparison of selectivity values for different methods			
Method	Temperature (K)	Pressure (bar)	Selectivity
Equilibrium Constants	300	1	0.1740
Kinetic Monte Carlo	300	5	0.1857
Metropolis Monte Carlo	300	5	0.1802

Clearly, the selectivity is very similar for the three different methods.

The difference in values does not stem from error per se, but rather from the fact that the pressure used in the two Monte Carlo simulations was set at 5 bar, whereas the pressure for the method using the equilibrium constants was set at 1 bar. The discrepancy in values is therefore due to the fact that at higher pressure, surface coverage generally increases and as a result so does selectivity.