TIF320 – Assignment 4

Computational Materials and Molecules
–
Kinetic Modeling

Nico Guth (nicog) and Erik Levin (eriklev)

March 10, 2023

Task 1: Determining lattice parameters

In the stable solid states of gold (Au), platinum (Pt) and rhodium (Rh) form face-centered-cubic (fcc) crystal lattices [1]. To determine the lattice parameters of these materials, bulk systems were constructed using the function *ase.build.bulk* with varying lattice parameters. The resulting potential energies, calculated using GPAW, are shown in fig. 1. The lattice parameters with the lowest energy are compared in table 1 with the experimentally found values. The results do agree generally well with the highest error being found in gold with an error of 2.4 % while the lowest was found in rhodium with a 1.0 % error.

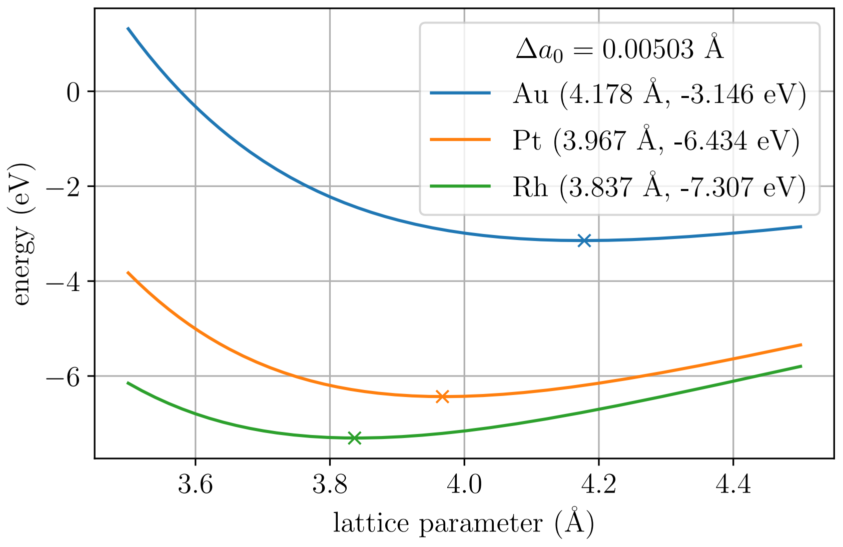


Figure 1: Scan for the minimum potential energy with varying lattice parameter a_0. Marked with crosses are the found minima with the values given in the legend. The scan was performed using a granularity of $\Delta a_0 = 0.005$ Å.

Table 1: Comparison of the lattice parameters found here through simulation and experimentally found values.

Material	simulated a_0 (Å)	experimental a_0 (Å)[1]
Au	4.178	4.08
Pt	3.967	3.92
Rh	3.837	3.80

Task 2: Cohesive Energies

Given the bulk energies E_{bulk} from task 1 and the atomic energies E_{at} from the task description, the cohesive energy is calculated through,

$$E_{\text{coh}} = E_{\text{at}} - E_{\text{bulk}} \,. \tag{1}$$

The results are listed in table 2.

Table 2: Energy values for gold, platinum and rhodium calculated through DFT and experimental values together with experimental values for the cohesive energy.

Material	E_{at} (eV)	E_{bulk} (eV)	E_{coh} (eV)	E_{Exp} (eV)[2]
Au	−0.158	−3.146	2.988	3.81
Pt	−0.704	−6.434	5.730	5.84
Rh	−1.128	−7.307	6.179	5.75

Here both an expected quantitative difference but also an unexpected qualitative difference could be found. For the quantitative differences, gold once again had the highest error with an underestimation of 21.6 % which was high compared to the errors of 1.9 % underestimation and 7.5 % overestimation for platinum and rhodium respectively. This indicates different structural properties in gold compared to platinum and rhodium, even though gold and platinum are closer in the periodic table. Even though a quantitative difference was to be expected to some degree, the qualitative difference was not. This difference occurs in the comparison between rhodium and platinum, as the experimental value places the cohesive energy of platinum higher than rhodium, meanwhile the computational method inverts this relationship. It does also increase the nominal difference from 0.09 eV to 0.449 eV, which is an increase in difference by almost a factor of 5. Overall these results indicate to some degree that the calculation might be wrong but is still used as the errors for some of the metals are still quite close to the experimental values.

Task 3: Construct the surfaces

The surface energy arrives as the molecules on the surface does not have a complete bound as compared to molecules in the bulk material [3]. As such, energies will be introduced at the surface compared to the bulk material. To calculate the surface energy of the system, it can be viewed as the extra energy compared to the bulk system. As such, it is calculated by the differences between the two, normalized by the area.

The slab was constructed using the function *ase.build.fcc111* with the lattice parameters given in task 1, a size of 3x3x3 and 6.0 Å of vacuum in the y-direction on both sides. Periodic boundary conditions were enabled. The potential energy was then calculated using GPAW.

For a slab, the equation for the surface energy γ is then defined as in equation (2) [4].

$$\gamma = \frac{E_{\text{slab}} - N \cdot E_{\text{atom}}}{2A} \tag{2}$$

Here the N represent the number of atoms in the slab and A is the area of the unit cell on the x-y-plane. The factor two comes from having two surfaces on one slab. E_{slab} is the energy of the slab with the surfaces. By using equation (2) we then acquired the values presented in table 3.

Table 3: Surface energies of gold, platinum and rhodium of the (111) surface together with the corresponding experimental values.

Material	simulated γ (eV/Å^2)	simulated γ (J/m^2)	experimental γ (J/m^2)[5]
Au	0.044 89	0.719	1.506
Pt	0.096 60	1.548	2.489
Rh	0.125 85	2.016	2.659

Here we can observe quite major differences compared to the experimental values, with the highest difference once again being for the gold. Here the value was only 47.7 % of its experimental value, meanwhile the platinum reached 62.2 % and rhodium 75.8 % compared to experimental data. All did in other words underestimate the surface energy with quite a big margin. But from the data, a conclusion about the dominant surface for an alloy could be drawn. In a triple alloy containing all metals, gold would likely be the surface metal as this is the lowest energy and would therefore minimize then energy of the alloy.

Task4

To calculate the energy of a molecule in gas form, we construct a big enough unit cell (12 Å side length) compared to the molecule (bond length ≈ 1.2 Å) and calculate the energy of the system using GPAW. The molecule is set up using the function *ase.build.molecule*. We also know that O_2 is spin-polarized due to electronic structure with unbounded electrons, which is not the case with CO and thus spin-polarization does not occur. The values within DFT then became -8.709 eV for O_2 and -14.186 eV for CO.

Task 5

To perform the vibrational analysis and find the entropy for a certain temperature and pressure, we can in this case use the ASE function for ideal gasses *ase.thermochemistry.IdealGasThermo*. This is due to the construction of the unit cell and the substances of oxygen and carbon. We can therefore approximate them to ideal gasses for the given temperature and pressure (300 K and 1 bar).

By then utilizing the ASE function, we obtain the vibrational energy from the biggest real eigenvalue (the others being rotational and translational). The results from this analysis can be seen in table (4). Here we can observe a good agreement with the experimental values for the entropies.

Table 4: Molecular energies, vibrational energies and entropy of O_2 and CO in a gas phase at 300 K and 1 bar.

Molecule	E_{mol} (eV)	E_{vib} (eV)	S (J/(mol K))	S_{exp} (J/(mol K))[6]
O_2	-8.709	0.189	205.80	205.15
CO	-14.186	0.204	198.14	197.66

Task 6: Adsorption of O and CO

In order to calculate the adsorption energy, we construct a surface using the function *ase.build.fcc111* and then place the adsorbate at the adsorption site using the function *ase.build.add_adsorbate*.

In the fcc (111) case, there are 4 possible adsorption sites. Those sites are visualized in fig. 2. Through research, we found that in general, the most stable adsorption sites for O are the fcc-hollow sites ("Generally the fcc hollow sites are the most stable adsorption sites for oxygen on metal (1 1 1) surfaces."[7]). For CO the most stable sites could not be determined with certainty because of contradictory results in different sources. However, it seems like most sources seem to state that the atop site is the most stable adsorption site for CO on Au[8], Pt[9] and Rh[10]. The orientation of CO seems to be most stable in an upright position. Therefore, we used the fcc-hollow site for O and the atop site for CO on all metal surfaces using the position argument in the function *ase.build.add_adsorbate*. The height is chosen such that the distance to the nearest surface atoms is approximately 2 Å. The combined system is then relaxed using GPAW and GPMin until the forces are less than 0.1 eV/Å.

The potential energy of the combined system E_{combined} is then used to calculate the adsorption energy through

$$E_{\mathrm{ads}} = E_{\mathrm{slab}} + E_{\mathrm{mol}} - E_{\mathrm{combined}} \tag{3}$$

where E_{slab} is taken to be the potential energy we got in task 3 (not γ) and E_{mol} is the energy we got in task 4 (only half of the O_2 energy for the O adsorption). The activation energy of

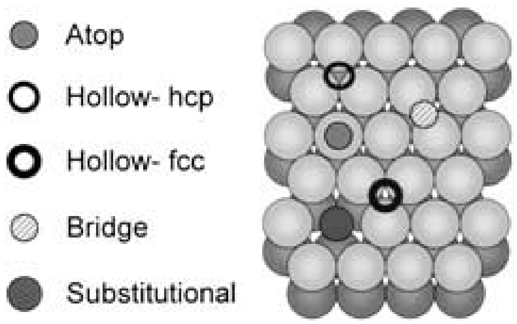


Figure 2: Different adsorption sites on a Cu(111) surface: atop, hollow-hcp and fcc, bridge and substitutional.[11]

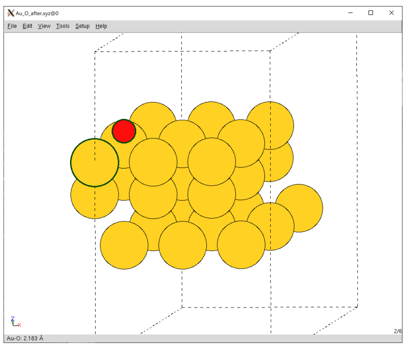
the reaction between CO and O, i.e. the activation energy for the formation of CO_2, is then calculated as

$$E_a = +0.3(E_{\text{ads}}(\text{CO}) + E_{\text{ads}}(\text{O})) + 0.22\,\text{eV}\,. \tag{4}$$

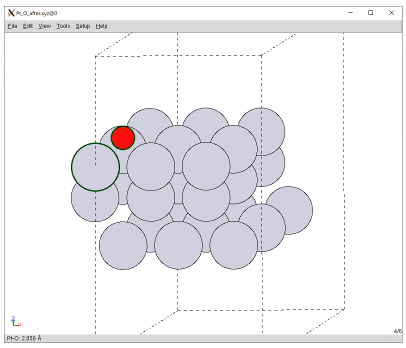
The resulting energies are listed in table 5.

Table 5: Adsorption energies E_{ads} for O and CO on gold, platinum and rhodium and the corresponding activation energies E_a for the formation of CO_2.

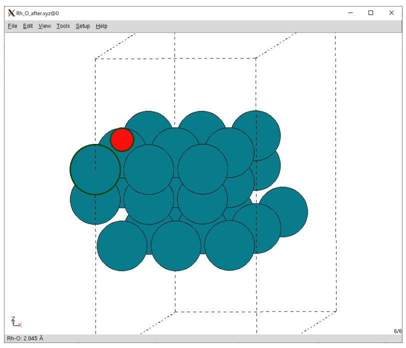
Metal	E_{ads}(O) (eV)	E_{ads}(CO) (eV)	E_a (eV)
Au	−0.314	0.150	0.171
Pt	1.044	1.601	1.014
Rh	1.755	1.728	1.265



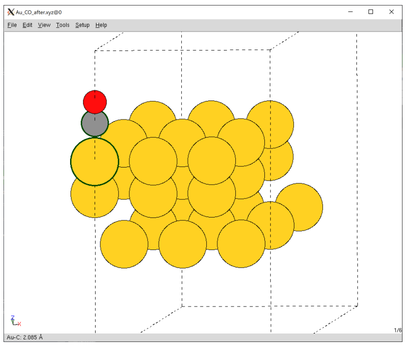
(a) $A_d = 2.183$ Å



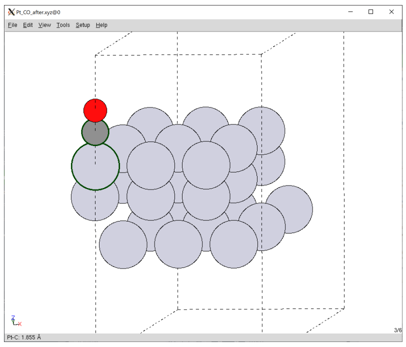
(b) $A_d = 2.050$ Å



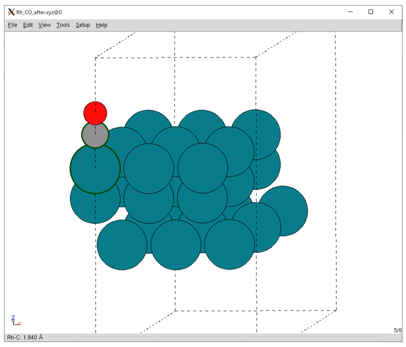
(c) $A_d = 2.045$ Å



(d) $A_d = 2.085$ Å



(e) $A_d = 1.855$ Å



(f) $A_d = 1.840$ Å

Figure 3: Relaxed structures for the adsorption energy calculation. The distances, A_d, between the molecule and the surface atom, represented in black, is written in the subcaptions of each figure.

Task 7: The Kinetic Analysis

The model of the CO-oxidization includes the following three fundamental chemical reactions

$$\mathrm{O_2(gas)} + 2* \underset{\mathrm{k_1}^-}{\overset{\mathrm{k_1}^+}{\rightleftharpoons}} \mathrm{O}* + \mathrm{O}* \tag{5}$$

$$\mathrm{CO\,(gas)} + * \underset{\mathrm{k_2}^-}{\overset{\mathrm{k_2}^+}{\rightleftharpoons}} \mathrm{CO}* \tag{6}$$

$$\mathrm{CO}* + \mathrm{O}* \xrightarrow{\mathrm{k_3}^+} \mathrm{CO_2(gas)} + 2* \tag{7}$$

with $*$ symbolizing a site on the metal surface that is unoccupied and A$*$ symbolizing a site that is occupied by A. The adsorption or desorption reactions occur at the rates

$$r_1^+ = p_{\mathrm{O_2}}\theta_*^2 k_1^+ \quad r_1^- = \theta_{\mathrm{O}}^2 k_1^- \tag{8}$$

$$r_2^+ = p_{\mathrm{CO}}\theta_* k_2^+ \quad r_2^- = \theta_{\mathrm{CO}} k_2^- \tag{9}$$

$$r_3^+ = \theta_{\mathrm{O}}\theta_{\mathrm{CO}} k_3^+ \tag{10}$$

where p is the partial pressure of the gas, θ is the fractional occupancy of reactant i and k the rate constants for either adsoprtion, k^+ or desorption k^-. r_3^+ is the formation rate of CO$_2$.

Knowing the adsorption rate constants k^+, the desorption rate constants k^- can be inferred by means of the equilibrium constants

$$K = \frac{pk^+}{k^-} = \exp\left(-\frac{S_{\mathrm{gas}}}{k_B} + \frac{E_{\mathrm{ads}}}{k_B T}\right) \tag{11}$$

where S_{gas} is the entropy of the molecule in the gas phase, E_{gas} is the corresponding energy and E_{ads} is the corresponding adsorption energy ($2 \cdot E_{\mathrm{ads}}(\mathrm{O})$ for K_1 and $E_{\mathrm{ads}}(\mathrm{CO})$ for K_2). T is the temperature in Kelvin and k_B is the Boltzmann constant.

The rate of the CO$_2$ formation can be written as

$$r_3^+ = f_3^+(\vec{\theta})\nu e^{-E_a/k_B T} \quad \text{with} \quad f_3^+(\vec{\theta}) = \theta_{\mathrm{O}}\theta_{\mathrm{CO}} \tag{12}$$

where we approximate $\nu = 1\,\mathrm{THz}$ and E_a is the activation energy of this reaction.

To find expressions for the fractional coverages of CO$*$ and O$*$ in equilibrium we use the fact that in equilibrium $\mathrm{d}\theta_i/\mathrm{d}t = 0$, but at the same time

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = \sum_j r_j(\vec{\theta})n_{i,j} \tag{13}$$

where j runs over all five reactions and $n_{i,j}$ is the stoichiometric coefficient of species i in reaction j, i.e. the number of reactants i in that reaction. This now gives us the following system of equations for the CO$*$ and O$*$ coverages.

$$\frac{\mathrm{d}\theta_{\mathrm{O}}}{\mathrm{d}t} = 2r_1^+ - 2r_1^- - r_3^3 = 0 \tag{14}$$

$$\frac{\mathrm{d}\theta_{\mathrm{CO}}}{\mathrm{d}t} = r_2^+ - r_2^- - r_3^3 = 0 \tag{15}$$

However, we assume that the formation rate of CO_2 r_3^+ is much lower than the adsorption or desorption rates and can therefore be neglected. Therefore, the system of equations becomes

$$\frac{\mathrm{d}\theta_\mathrm{O}}{\mathrm{d}t} = 2r_1^+ - 2r_1^- = 0 \tag{16}$$

$$\frac{\mathrm{d}\theta_\mathrm{CO}}{\mathrm{d}t} = r_2^+ - r_2^- = 0 \tag{17}$$

By then using the earlier stated definitions of r, we can rewrite the equations to,

$$2p_{\mathrm{O}_2}\theta_*^2 k_1^+ - 2\theta_\mathrm{O}^2 k_1^- = 0 \tag{18}$$

$$p_\mathrm{CO}\theta_* k_2^+ - \theta_\mathrm{CO} k_2^- = 0 \tag{19}$$

This can in turned be rewritten by using the definition of $K = \frac{pk^+}{k^-}$ and the assumption that $\theta_* = 1 - \theta_\mathrm{O} - \theta_\mathrm{CO}$ which gives

$$K_1(1 - \theta_\mathrm{O} - \theta_\mathrm{CO})^2 - \theta_\mathrm{O}^2 = 0 \tag{20}$$

$$K_2(1 - \theta_\mathrm{O} - \theta_\mathrm{CO}) - \theta_\mathrm{CO} = 0\,. \tag{21}$$

Solving this system of equations gives the following analytical expressions for the fractional coverages

$$\theta_\mathrm{O} = \frac{\sqrt{K_1}}{1 + K_2 + \sqrt{K_1}} \qquad \theta_\mathrm{CO} = \frac{K_2}{1 + K_2 + \sqrt{K_1}}\,. \tag{22}$$

To see the temperature dependence of the fractional coverages, eq. (22) was used in combination with eq. (11). The adsorption energies have been taken from table 5 and the entropies were calculated the same way as in task 5 but for each temperature (at 1 atm pressure). The results are shown in fig. 4. Finally, the temperature dependence of the formation rate of CO_2 on the three different metals is shown in fig. 5. Used was eq. (12) with E_a taken from table 5.

Answers and discussion

The formation rate in this case quantifies how much CO_2 is produced per second when a perfect (111) metal surface is in contact with air (CO+O_2 gas) at 1 atm pressure. The values that come out of eq. (12) without any modification represent how many CO_2 molecules are produced each second at one adsorption site on the surface. To convert this to a usable formation rate, the value was multiplied by how many sites per unit area are on the perfect (111) surfaces. This was approximated to be the same as the number of atoms on the surface, which is in our case 9 sites per area of the simulation cell (as shown in fig. 3). Additionally, the values are divided by the Avogadro constant to represent how many moles of CO_2 are formed each second per square meter of the surface. To estimate how much CO_2 is produced, the values in fig. 5 need to be multiplied by the area of the surface in square meters and the time in seconds.

The question about what is the *best* catalyst comes down to operating temperature range. That is, if we are considering the activity only and no factors such as economics or material

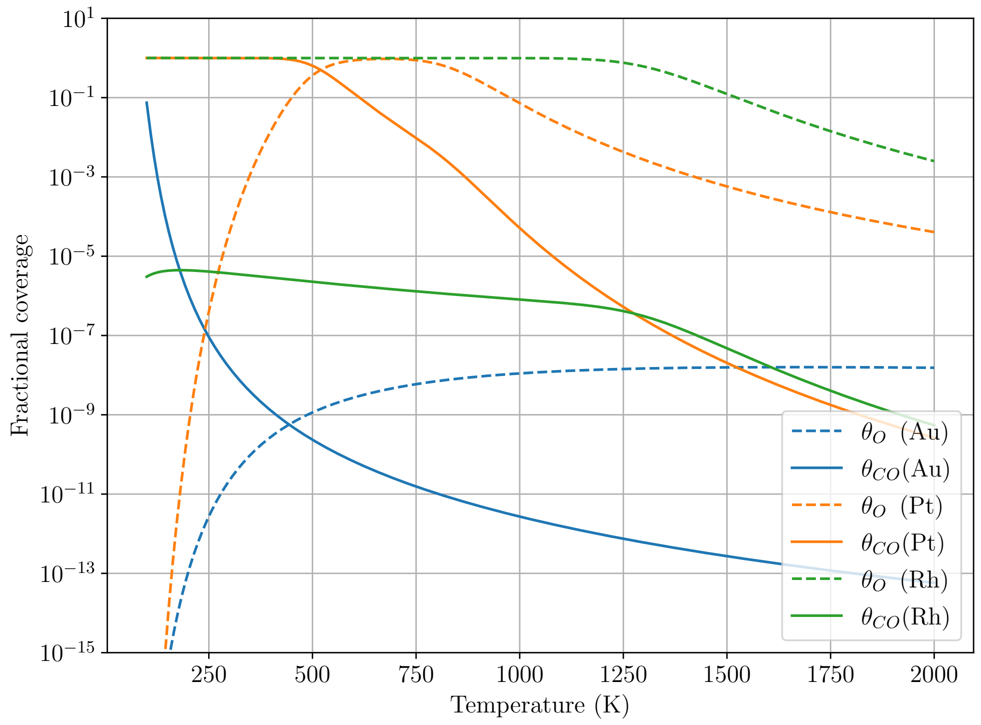


Figure 4: Temperature dependence of the fractional coverages of CO∗ and O∗ on the (111) surfaces of Au, Pt and Rh at 1 atm pressure. Notice the logarithmic scale.

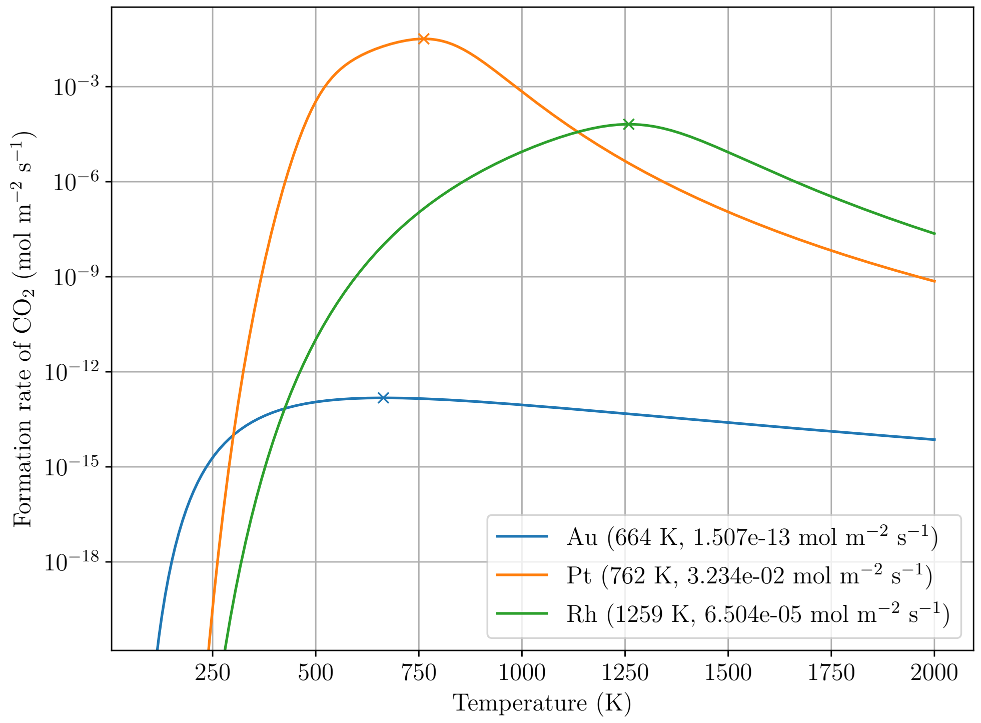


Figure 5: Temperature dependence of the formation rate of CO_2 on the (111) surfaces of Au, Pt and Rh at 1 atm pressure. Notice the logarithmic scale. Marked with and x are the maxima, which are quantified in the legend.

availability. In the broad range of non-extreme temperatures between 300-1100 K platinum has the highest formation rate per area and would therefore be preferred. This does however change for lower temperatures. There, the lower kinetics enable Au to surpass the other catalysts even if it is worse above 300 K. Mirrored behavior can be seen with Rh at temperatures above 1100 K, where it surpasses Pt.

The CO_2-desorption rate on Au is found experimentally to be on the order of $1\,10^16/\mathrm{m}^2/\mathrm{s}$ in temperatures between 200 and 400 K.[12] This is several orders of magnitude higher than our results for Au. The reasons for this discrepancy are unclear. The general temperature regions where each of the elements is the best catalyst roughly corresponds to experimental findings, where Au is the best for low temperatures, Pt is the best for intermediate temperatures and Rh is the best for high temperatures.

Table 6: Maxima in fig. 5 for the formation rates of CO_2 on the (111) surfaces of gold, platinum and rhodium.

Metal	$T_{\mathrm{max}}(\mathrm{K})$	$r^{+}_{3,\ \mathrm{max}}(\mathrm{mol}/(\mathrm{m}^2\,\mathrm{s}))$
Au	664	1.507×10^{-13}
Pt	762	3.234×10^{-2}
Rh	1259	6.504×10^{-5}

Task 8: Thoughts

The three peaks of the formation rate of CO_2 observed in fig. 5 are listed in table 6 for Au, Pt and Rh respectively. The reason behind the position and amplitude of the peaks originates from how strong the adsorbents bind to the metal surface. The distinct peaks and overall behavior of a heterogeneous catalyst, e.g. gas-solid interface, can be qualitatively explained by the so-called *Sabatier principle*. This principle states that a catalyst should not interact too weakly nor too strongly, but just right. This comes down to the fact that the reaction requires binding the molecule to the catalyst, which requires a certain interaction strength, but it should also be able to release the formed molecule. From the results, we can see that Au is the dominating catalyst up until 300 K. Up until this point, with Sabatier in mind, combined with the fact that the reaction requires a certain amount of energy, the so-called activation threshold is reached for gold at these temperatures but not for the other catalysts. This then changes and Pt takes over as the threshold has now been reached with the heat and the kinetics are higher but not too high. This continues until around the 1100 K point where we have moved far enough away from the optimal operating point of Pt and it is also in theory undergoing a phase shift in the most favorably state to from the fcc to a hcp which decreases the surface area and in turn the reaction rate. Instead, Rh is the dominating catalyst as the activation energy has been reached, albeit a bit later than for Pt. The explanation for decreasing processes can be several. First is once again the Sabatier principle where the optimum range for the catalyst is passed, the reaction rate will start to decline. Higher temperature implies stronger interactions and in that case, the interactions might be too strong, decreasing the reactants time on the surface. This then means that more reactants might dissociate before the catalyst process has taken place. In other words, with a simplified real world example, they are placing their hand on the stove so quickly that it does not burn before they take it away. More factors might however also affect for example the phase-shift for Pt is one where surface area decreases and so the overall reaction rate decreases.

The amplitude of these peaks also comes from this principle, Au is the best in a lower temperature region but is limited by the sluggish kinetics and the rate will therefore be decreased. Gold also binds strongly to CO and might therefore not release the reactants, and so the catalyst is limited in desorption. The Pt finds this balance and as such the highest amplitude as a good balance in interaction and speed has been reached. The Rh is not as good but still closer to this optimum and has therefore a higher peaked but looses more to decreased adsorption from weaker binding in stronger interactions.

There are multiple factors that could count as limitations to the accuracy of our predictions. Firstly, the attractive and repulsive forces between the adsorbed atoms/molecules were not taken into account. Repulsive forces would decrease the fractional coverage and attractive forces would increase the fractional coverage. The attractive forces between CO and O might lead to a higher reaction rate, since it is more likely to find CO and O at one place compared to the rate with only considering the average coverage of both species.

Secondly, we decided to only consider one type of adsorption sites for each species. In reality, however, the adsorbents can bind to various different sites (see fig. 2) and as such, combined with introduction of attraction and repulsion from earlier, could increase/decrease both raw rates as well as increase available sites per area. This could in turn both increase the total rates or if the repulsion forces are too strongly in play, slow the reactions and for example compete with oxygen for the same sites, so called CO-poisoning.

Thirdly, perfect (111) metal surfaces were assumed. This is an unrealistic approximation, since real surfaces always contain impurities, deviations from the perfect lattice structure and cannot be perfectly flat. On the one hand, the roughness of a real surface might increase the surface area and lead to a higher reaction rate. On the other hand, the impurities in the crystal might lead to a decrease in coverage and reaction rate, depending on the types of impurities.

Finally, the reactions considered in this method (eq. (5)) are not the only reactions that occur during CO-oxidation. There can be a lot more reactions that might compete. For example, if hydrogen, the universe most abundant element would be present, this could incur competing reactions and so the reaction rate for the process we want would be limited. Therefore, our method of calculation does only take perfect systems without impurities in either the reactants or on the surface into considerations while also ignoring some fundamental physics behind the process as the attraction/reaction processes.

A dream catalyst for this process is once again depending on the user case and need of reaction but taking a real world case into mind, the best catalyst would have a maximal reaction rate with the least amount of material. Now to maximize the rate, we need to maximize the surface area, which can be done with for example nanoparticles compared to the (111) facet construction. By alloying the nanoparticles, we can decrease use of the precious catalyst-material even further and in some cases even improve reactions. Furthermore, these nanoparticles should be stable and react to nothing else than CO and O_2, or rather strongly favor those reactions. However, it should also be tailored to the temperature ranges of operation to ensure no shift in phases can occur. With this maximized surface area and stable particles, the binding energy should in turn be tweaked to a perfect volcano plot position according to the mentioned Sabatier principle where the balance in binding has been optimized. With this combination of factors, some kind of Pt-nanoparticle alloy should be a possible candidate where we use as much surface area as possible, therefore improving rates and have some kind of function to make sure that the surface is stable and does not get poisoned by for example CO. In other words, the mono-metal (111) are not optimal, since doping might increase both the binding balance as well as economics. That is for a real world material, but a true dream catalyst would be made with some material so abundant that it was next to free and without any risk of harm to life. Furthermore, the material should itself bind only to the two reactants as mentioned above by default and no engineering would be required. Only to pick up the material and put it in the application as it already is on top of the volcano plot with a perfect balance in binding. That would be a true dream-catalyst, but we are being chained to the real physical world, which incurs a lot of limitations.

References

[1] Klaus Hermann. "Appendix E: Parameter Tables of Crystals". In: *Crystallography and Surface Structure*. John Wiley & Sons, Ltd, 2011, pp. 265–266. ISBN: 9783527633296. DOI: `https://doi.org/10.1002/9783527633296.app5`.

[2] Charles Kittel. "Table 1 Cohesive energies". In: *Introduction to Solid State Physics*. John Wiley & Sons, Ltd, 2004, p. 50. ISBN: 9780471415268.

[3] Emma Spooner. *What is Surface Energy? Formula & Definition*. 2023. URL: `https://www.ossila.com/pages/a-guide-to-surface-energy`.

[4] Materials Square. *#10 Adsorption Energy and Surface Energy Obtained through Slab Structure*. July 2019. URL: `https://www.materialssquare.com/blog/10-adsorption-energy-and-surface-energy-obtained-through-slab-structure`.

[5] L. Vitos et al. "The surface energy of metals". In: *Surface Science* 411.1 (1998), pp. 186–202. ISSN: 0039-6028. DOI: `https://doi.org/10.1016/S0039-6028(98)00363-X`.

[6] Jr. Donald R. Burgess. "Thermochemical Data". In: *NIST Chemistry WebBook*. National Institute of Standards and Technology, 2023. DOI: `https://doi.org/10.18434/T4D303`.

[7] Spencer D. Miller and John R. Kitchin. "Relating the coverage dependence of oxygen adsorption on Au and Pt fcc(111) surfaces through adsorbate-induced surface electronic structure effects". In: *Surface Science* 603.5 (2009), pp. 794–801. ISSN: 0039-6028. DOI: `https://doi.org/10.1016/j.susc.2009.01.021`.

[8] Yohaselly Santiago-Rodríguez et al. "Atomic and molecular adsorption on Au(111)". In: *Surface Science* 627 (2014), pp. 57–69. ISSN: 0039-6028. DOI: `https://doi.org/10.1016/j.susc.2014.04.012`.

[9] Denise C. Ford, Ye Xu, and Manos Mavrikakis. "Atomic and molecular adsorption on Pt(111)". In: *Surface Science* 587.3 (2005), pp. 159–174. ISSN: 0039-6028. DOI: `https://doi.org/10.1016/j.susc.2005.04.028`.

[10] M. Mavrikakis et al. "Atomic and molecular adsorption on Rh(111)". In: *The Journal of Chemical Physics* 117.14 (2002), pp. 6737–6744. DOI: `10.1063/1.1507104`.

[11] Michelle Spencer and Graeme Nyberg. “DFT modelling of hydrogen on CU(110)- and (111)-type clusters”. In: *Molecular Simulation* 28 (2002), pp. 807–825. DOI: 10.1080/08927020210000002502.

[12] BK Min et al. “Efficient CO oxidation at low temperature on Au (111)”. In: *The Journal of Physical Chemistry B* 110.40 (2006), pp. 19833–19838. URL: https://pubs.acs.org/doi/10.1021/jp0616213.

Appendix: Python script for task 1

```python
import time
start_time = time.time()

from tqdm.auto import tqdm
import numpy as np
from ase.build import bulk
from gpaw import GPAW, PW
from ase.parallel import world,parprint
from ase.units import eV

a_range = np.linspace(3.5,4.5, 200)
energy_arrs = {"Au":[],"Pt":[],"Rh":[]}
elements = ["Au", "Pt", "Rh"]

for element in elements:
    parprint(f"Scan through lattice parameters for {element}...")
    for a0 in tqdm(a_range, disable=(world.rank!=0)):
        atoms = bulk(element, 'fcc', a=a0)
        atoms.calc = GPAW(xc = 'PBE',
                          mode=PW(450),
                          kpts =(12, 12, 12),
                          txt = None)
        energy = atoms.get_potential_energy() * eV
        energy_arrs[element].append(energy)
        parprint(f'element = {element}; a0 = {a0:.2f} Å; energy = {energy:.5f}
   eV')

if world.rank == 0:
    data = list(zip(a_range, energy_arrs["Au"], energy_arrs["Pt"], energy_arrs[
   "Rh"]))
    np.savetxt("task1output/task1_energies.csv", data, header="# a0[Å], E_Au[eV
   ], E_Pt[eV], E_Rh[eV]", comments="", fmt= "%.10f, %.10f, %.10f, %.10f")
    print("Runtime",time.time()-start_time)
```

Appendix: Python script for task 2

```python
# %%
# https://wiki.fysik.dtu.dk/gpaw/tutorialsexercises/energetics/rpa_ex/rpa.html
import numpy as np

# from task 1 in eV:
E_bulk = {
    "Au": -3.146,
    "Pt": -6.434,
    "Rh": -7.307
}

# from the task 2 description in eV:
E_at = {
    "Au": -0.158,
    "Pt": -0.704,
    "Rh": -1.128
}

for elmt in ["Au", "Pt", "Rh"]:
    #E_coh = E_at[elmt] - 0.5*E_bulk[elmt]
    E_coh = E_at[elmt] - E_bulk[elmt]
    print(f"Cohesive energy for {elmt}: {E_coh:.3f} eV")


# %%
```

Appendix: Python script for task 3

```python
# %%
# https://wiki.fysik.dtu.dk/gpaw/tutorialsexercises/structureoptimization/
   surface/surface.html
# https://leppertlab.wordpress.com/2015/02/06/using-ase-to-create-surface-slabs
   /


from ase.build import fcc111
from gpaw import GPAW, PW
from ase.units import eV,J,m
from ase.parallel import world,parprint
from tqdm.auto import tqdm

mJ = 10**(-3) * J

E_bulk = {
    "Au": -3.146,
    "Pt": -6.434,
    "Rh": -7.307
}

a0 = {
    "Au": 4.178,
    "Pt": 3.967,
    "Rh": 3.837
}

for elmt in tqdm(["Au", "Pt", "Rh"], disable=(world.rank!=0)):
    slab = fcc111(elmt, a=a0[elmt], size=(3, 3, 3), vacuum=6.0)
    slab.pbc = True

    slab.calc = GPAW(xc = 'PBE',
                    mode=PW(450),
                        kpts =(4, 4, 1),
                        txt = f"task3output/gpaw_output_{elmt}.txt")

    E_surface = slab.get_potential_energy() * eV
    area = slab.get_volume() / slab.cell[2,2]
    gamma = (E_surface - E_bulk[elmt]*len(slab))/(2*area)

    parprint(f"Area ({elmt}): {area:.5f} Å")
    parprint(f"Slab Energy ({elmt}): {E_surface:.5f} eV")
    parprint(f"Surface Energy ({elmt}): {gamma:.5f} eV/Å^2")
    parprint(f"Surface Energy ({elmt}): {gamma*m**2/mJ:.5f} mJ/m^2")
```

Appendix: Python script for task 4

```
from ase import Atoms
from ase.build import molecule
from gpaw import GPAW, PW
from ase.units import eV
from ase.parallel import world,parprint

import json

with open("results.json", "r") as file:
    results = json.load(file)

for molecule_str in ["O2","CO"]:
    atoms = molecule(molecule_str, vacuum=6.0)
    atoms.pbc = True

    atoms.calc = GPAW(xc = 'PBE',
                        mode=PW(450),
                        kpts =(1, 1, 1),
                        spinpol=True,
                        txt = f"task4output/gpaw_output_{molecule_str}.txt")

    E = atoms.get_potential_energy()*eV

    results[f"E_mol_{molecule_str}"] = E
    parprint(f"Energy of {molecule_str}: {E:.5f} eV")


with open("results.json",'w') as json_file:
    json.dump(results, json_file, indent=4)
```

Appendix: Python script for task 5

```
import time
start_time = time.time()
from ase.build import molecule
from gpaw import GPAW,PW
from ase.optimize import QuasiNewton
from ase.vibrations import Vibrations
from ase.thermochemistry import IdealGasThermo
```

```python
from ase.parallel import world,parprint
from ase.units import eV, mol, J
from ase import Atoms
import json

spin = {"O2":1.0, "CO":0.0}
symmetry_number = {"O2":2, "CO":1}
magmom = {"O2":[1.7,1.7], "CO":[2.5,1.7]}

with open("results.json", "r") as file:
    results = json.load(file)

for molec in ["O2","CO"]:
    atoms = molecule(molec, vacuum=6.0)
    atoms.pbc = True
    atoms.calc = GPAW(xc = 'PBE',
                      mode=PW(450),
                      kpts =(1, 1, 1),
                      spinpol=True,
                      symmetry={'point_group': False}, # Turn off point-group symmetry
                      txt = f"task5output/gpaw_output_{molec}.txt")
    potentialenergy = atoms.get_potential_energy()*eV

    vib = Vibrations(atoms)
    vib.run()
    vib_energies = vib.get_energies()

    thermo = IdealGasThermo(vib_energies=vib_energies,
                            potentialenergy=potentialenergy,
                            atoms=atoms,
                            geometry='linear',
                            spin=spin[molec],
                            symmetrynumber = symmetry_number[molec])
    S = thermo.get_entropy(temperature=300, pressure=100000)

    results[f"entropy_{molec}"] = S
    results[f"E_mol_{molec}_task5"] = potentialenergy

    parprint(f"Vibrational energies of {molec}: {vib_energies}")
    parprint(f"Potential Energy of {molec}: {potentialenergy:.5f} eV")
    parprint(f"Entropy of {molec}: {S:.5f} eV/K")
    parprint(f"Entropy of {molec}: {S*mol/J:.5f} J/mol K")
    parprint(f"Total runtime: {time.time()-start_time}")


with open("results.json",'w') as json_file:
    json.dump(results, json_file, indent=4)
```

Appendix: Python script for task 6

```python
import time
start_time = time.time()

from ase.build import fcc111, add_adsorbate, molecule
from ase.units import eV,J,m
from ase.parallel import world,parprint
from tqdm.auto import tqdm
from ase.io import write
from gpaw import GPAW, PW
from ase.optimize import GPMin
from ase.constraints import FixAtoms

from pathlib import Path
Path("task6output/structures").mkdir(exist_ok=True)

a0 = {
    "Au": 4.178,
    "Pt": 3.967,
    "Rh": 3.837
}

combinations = [("Au","O"),("Pt","O"),("Rh","O"),
                ("Au","CO"),("Pt","CO"),("Rh","CO")]

for metal,adsorbate in tqdm(combinations, disable=(world.rank!=0)):
    parprint(f"Start with {metal}+{adsorbate} (current runtime: {time.time()-
    start_time} seconds)")
    atoms = fcc111(metal, a=a0[metal], size=(3, 3, 3), vacuum=6.0)
    atoms.pbc = True

    fixed_idxs = list(range(len(atoms)))

    if adsorbate == "O":
        add_adsorbate(atoms, "O", height=1.2, position="fcc")
    elif adsorbate == "CO":
        add_adsorbate(atoms, molecule("CO"), height=3.15, position="ontop")

    # Constrain all atoms except the adsorbate:
    atoms.constraints = [FixAtoms(indices=fixed_idxs)]

    write(f"task6output/structures/{metal}_{adsorbate}_before.xyz", atoms)

    atoms.calc = GPAW(xc = 'PBE',
                      mode=PW(450),
                      kpts =(4,4,1),
                      spinpol=True,
                      txt = f"task6output/gpaw_output_{metal}_{adsorbate}.txt")

    dyn = GPMin(atoms, trajectory=f'task6output/structures/relax_{metal}_{
    adsorbate}.traj', logfile=f'task6output/structures/relax_{metal}_{adsorbate
    }.log')
    dyn.run(fmax=0.1, steps=1000)

    E = atoms.get_potential_energy()

    parprint(f"Energy of {metal}+{adsorbate}: {E:.5f} eV")

    write(f"task6output/structures/{metal}_{adsorbate}_after.xyz", atoms)

parprint(f"Runtime: {time.time()-start_time} seconds")
```

```python





# %%
```

Appendix: Python script for task 7 (Entropy calculations)

```python
import time
start_time = time.time()
from ase.build import molecule
from gpaw import GPAW,PW
from ase.optimize import QuasiNewton
from ase.vibrations import Vibrations
from ase.thermochemistry import IdealGasThermo
from ase.parallel import world,parprint
from ase.units import eV, mol, J
from ase import Atoms
import json
import numpy as np
from tqdm import tqdm

spin = {"O2":1.0, "CO":0.0}
symmetry_number = {"O2":2, "CO":1}

T = np.linspace(100,2000,1901)

S_dict = {}

for molec in ["O2","CO"]:
    parprint(f"==========================")
    parprint(f"Start calculations for {molec}...")
    atoms = molecule(molec, vacuum=6.0)
    atoms.pbc = True
    atoms.calc = GPAW(xc = 'PBE',
                    mode=PW(450),
                    kpts =(1, 1, 1),
                    spinpol=True,
                    symmetry={'point_group': False}, # Turn off point-group symmetry
                    txt = None)
    potentialenergy = atoms.get_potential_energy()*eV

    vib = Vibrations(atoms)
    vib.run()
    vib_energies = vib.get_energies()

    thermo = IdealGasThermo(vib_energies=vib_energies,
                            potentialenergy=potentialenergy,
                            atoms=atoms,
                            geometry='linear',
                            spin=spin[molec],
                            symmetrynumber = symmetry_number[molec])

    parprint(f"Vibrational energies: {vib_energies}")
    parprint(f"Potential Energy: {potentialenergy:.5f} eV")
    parprint(f"Entropies for {molec}...")

    S_dict[molec] = []
    for T_i in tqdm(T, disable=(world.rank!=0)):
        S = thermo.get_entropy(temperature=T_i, pressure=100000, verbose=False)
        S_dict[molec].append(S)
        parprint(f"Entropy at {T_i:.1f} K: {S:.5f} eV/K \t = {S*mol/J:.5f} J/mol K")

if world.rank == 0:
    data = list(zip(T, S_dict["O2"], S_dict["CO"]))
    np.savetxt("task7output/task7_entropies.csv", data, header="# T[K], S_O2[eV/K], S_CO[eV/K]", comments="", fmt= "%.10f, %.10f, %.10f")
    print("Runtime",time.time()-start_time)
```

Appendix: Python script for task 7 (Calculations of coverages and formation rates)

```python
# %%
import numpy as np
import matplotlib.pyplot as plt
from scipy.constants import Boltzmann

# For latex interpretation of the figures (if latex is available)
from shutil import which
if which("latexmk") is not None:
    plt.rcParams.update({
        "text.usetex": True,
        "font.family": "Computer Modern"
    })
plt.rcParams.update({"font.size":14.0})

import json

with open("results.json", "r") as file:
    results = json.load(file)

k_B = 8.617333e-5 # eV/K

#S_O2 = results["entropy_O2"]
#S_CO = results["entropy_CO"]

T, S_O2, S_CO = np.genfromtxt("task7output/task7_entropies.csv", delimiter=",", unpack=True)

E_O2 = results["E_mol_O2"]
E_CO = results["E_mol_CO"]

# formulae
def K(T, S_gas, E_gas, E_ads):
    return np.exp(-S_gas/k_B + E_ads/(k_B*T))

```

```python

def theta_O(T, E_ads_O, E_ads_CO):
    K_1 = K(T, S_O2, E_O2, 2*E_ads_O)
    K_2 = K(T, S_CO, E_CO, E_ads_CO)
    return np.sqrt(K_1)/(np.sqrt(K_1)+K_2+1)

def theta_CO(T, E_ads_O, E_ads_CO):
    K_1 = K(T, S_O2, E_O2, 2*E_ads_O)
    K_2 = K(T, S_CO, E_CO, E_ads_CO)
    return K_2/(np.sqrt(K_1)+K_2+1)

def k_3(T, E_a):
    nu = 1e12
    return nu*np.exp(-E_a/(k_B*T))

def r_3(T, theta_O_, theta_CO_, E_a):
    return theta_O_*theta_CO_*k_3(T,E_a)

# %%
theta_O_dict = {}
theta_CO_dict = {}
r_3_atomic_units_dict = {}
r_3_dict = {}

for metal in ["Au","Pt","Rh"]:
    E_ads_O = results[f"E_ads_{metal}_O"]
    E_ads_CO = results[f"E_ads_{metal}_CO"]
    E_a = results[f"E_a_{metal}"]

    theta_O_ = theta_O(T, E_ads_O, E_ads_CO)
    theta_CO_ = theta_CO(T, E_ads_O, E_ads_CO)
    r_3_ = r_3(T, theta_O_, theta_CO_, E_a)

    sites_per_area = 9/results[f"surface_area_{metal}"] # Å^-2

    mol = 6.02214076e23 # atoms
    m = 1e10 # Å

    theta_O_dict[metal] = theta_O_
    theta_CO_dict[metal] = theta_CO_
    r_3_atomic_units_dict[metal] = r_3_ # s^-1
    r_3_dict[metal] = r_3_*sites_per_area / mol * 1e20 # mol m^-2 s^-1

# %%
# Find the maxima
for metal in ["Au","Pt","Rh"]:
    idx_max = np.argmax(r_3_dict[metal])
    results[f"r_3_max_{metal}[mol Ang^-2 s^-1]"] = r_3_dict[metal][idx_max]
    results[f"T(r_3_max_{metal})[K]"] = T[idx_max]

with open("results.json",'w') as json_file:
    json.dump(results, json_file, indent=4)

# %%
plt.figure(figsize=(8,6))
for i,metal in enumerate(["Au","Pt","Rh"]):
    plt.plot(T, theta_O_dict[metal], f"C{i}--", label=fr"$\theta_O\,$ ({metal})")
    plt.plot(T, theta_CO_dict[metal], f"C{i}-", label=fr"$\theta_{{CO}}$({metal})")
plt.yscale("log")
plt.xlabel("Temperature (K)")
plt.ylabel("Fractional coverage")
plt.ylim(1e-15,10)
plt.grid()
plt.legend(loc="lower right")
plt.tight_layout()
plt.savefig("plots/task7_theta.pdf")

plt.figure(figsize=(8,6))
for metal in ["Au","Pt","Rh"]:
    plt.plot(T, r_3_atomic_units_dict[metal], label=metal)
plt.ylim(bottom=r_3_atomic_units_dict["Au"].min()*100, top=r_3_atomic_units_dict["Pt"].max()*10)
plt.yscale("log")
plt.xlabel("Temperature (K)")
plt.ylabel(r"Formation rate of $\mathrm{CO}_2$ ($\mathrm{s}^{-1}$)")
plt.grid()
plt.legend()
plt.tight_layout()
plt.savefig("plots/task7_r_3_atomic_units.pdf")

plt.figure(figsize=(8,6))
for i,metal in enumerate(["Au","Pt","Rh"]):
    r_max = results[f"r_3_max_{metal}[mol Ang^-2 s^-1]"]
    T_max = results[f"T(r_3_max_{metal})[K]"]
    plt.plot(T, r_3_dict[metal], fr"C{i}-", label=f"{metal} ({T_max:.0f} K, {r_max:.3e} mol $\mathrm{{m}}^{{-2}}$ $\mathrm{{s}}^{{-1}}$)")
    plt.plot(T_max, r_max, f"C{i}x")
plt.ylim(bottom=r_3_dict["Au"].min()*100, top=r_3_dict["Pt"].max()*10)
plt.yscale("log")
plt.xlabel("Temperature (K)")
plt.ylabel(r"Formation rate of $\mathrm{CO}_2$ (mol $\mathrm{m}^{-2}$ $\mathrm{s}^{-1}$)")
plt.grid()
plt.legend(loc="lower right")
plt.tight_layout()
plt.savefig("plots/task7_r_3.pdf")



# %%
```