Install Required Python Libraries:  
 for which we used pip command as package management system which used to install required package in the python environment.

!pip install pandas  
 !pip install numpy  
 !pip install matplotlib  
 !pip install scipy  
 !pip install peakutils

Import required libraries such as pandas for data analysis, Numpy for working on array to perform mathematical operations, matplotlib for the data visualization and Scipy for additional support to perform array operations.

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.signal import find\_peaks

Read the xrd data by pandas in which we use the command **read\_csv** and provide the file location as argument, which will read the file and convert it into pandas DataFrame.

data = pd.read\_csv('G:\msc project sources\XRD\Group 1 - Zinc Powder.csv')  
data

Angle (2theta) Intensity (Counts)  
0 20.03 1  
1 20.09 1  
2 20.15 1  
3 20.21 3  
4 20.27 2  
.. ... ...

Collect information about dataset such as data types.

data.info()

Column Dtype   
 ------------ ----------

Angle (2theta) float64  
Intensity (Counts) int64

Checking the data to filter out null spaces to avoid errors in array operations:

data.isnull().sum()

Angle (2theta) 0 –null values  
Intensity (Counts) 0 –null values  
dtype: int64

In above output we can see the dtype is int64 we must have convert it into the float data type to avoid further error in the mathematical ops. So,

data['Angle (2theta)'] = data['Angle (2theta)'].astype('float')

Now we are going to plot the XRD data for Zinc powder as Intensity vs 2-theta, for which we assume x variable as x axis and take its value as 2-theta

x=data['Angle (2theta)']  
y=data['Intensity (Counts)']  
plt.plot(data['Angle (2theta)'], data['Intensity (Counts)'])  
plt.xlabel('2-theta')  
plt.ylabel('Intensity')  
plt.title('XRD pattern for Zinc powder')  
plt.show()

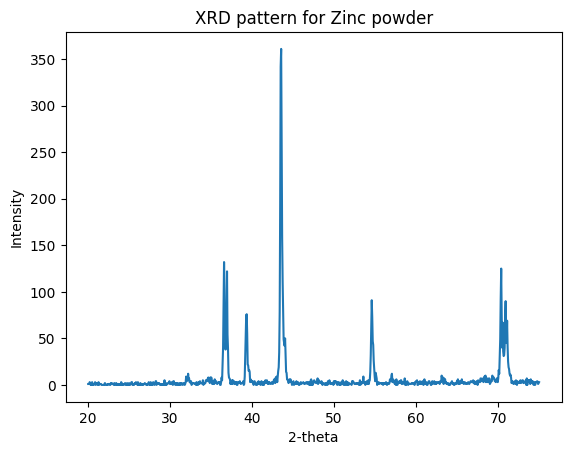


fig.1

#to detect the maximum intensity peaks  
peaks, \_ = find\_peaks(data['Intensity (Counts)'], height=60, distance=5)

#mark the detected peak and plot data  
plt.plot(data['Angle (2theta)'], data['Intensity (Counts)'])  
plt.plot(data['Angle (2theta)'][peaks], data['Intensity (Counts)'][peaks], '\*')  
plt.xlabel('2-theta')  
plt.ylabel('Intensity')  
plt.show()

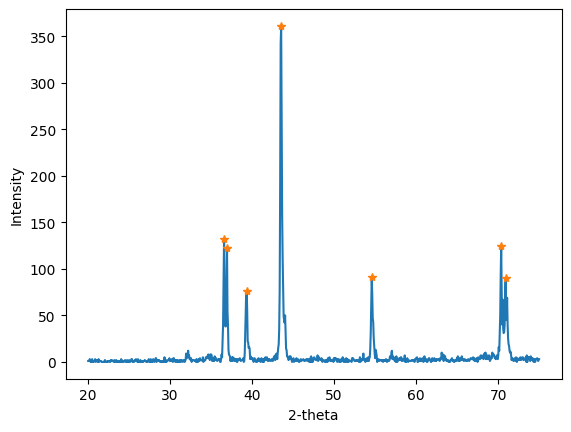


fig.2

Save the 2-theta values of detected peaks:

# Save the coordinates of the detected peaks in a new file  
  
peak\_coords = np.column\_stack((x[peaks], y[peaks])).astype(float)  
np.savetxt('peak\_file.csv', peak\_coords, delimiter=',', header='Theta,Intensity', comments='')  
  
# Show the plot  
  
plt.show()  
#detection of Theta w.r.t detected peak

Read the detected peak 2-theta values and save for further ops:

detected\_peak=pd.read\_csv('peak\_file.csv')  
x1=detected\_peak['Theta'].tolist()  
y1=detected\_peak['Intensity'].tolist()

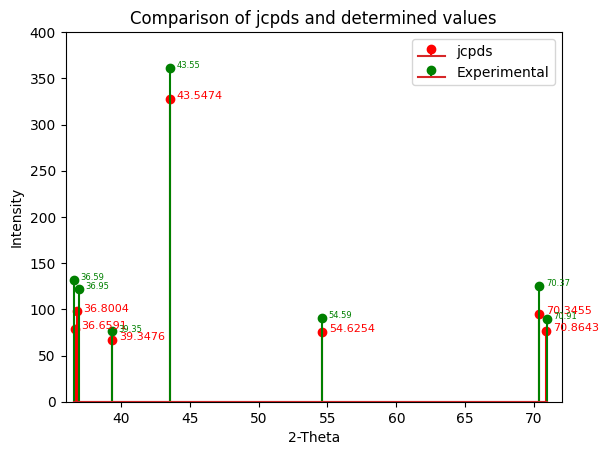
Read the jcpds data:

jcpds\_data=pd.read\_csv('G:\msc project sources\XRD\zinc powder.csv', encoding='unicode\_escape')  
#Theta=detected\_peak.iloc.drop('Intensity',axis=1,inplace=False  
  
# drop the column at index 1 (i.e. 'age' column) from the dataframe  
jcpds\_data1 = jcpds\_data.drop(jcpds\_data.columns[[2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,20,21]], axis=1)  
  
x=jcpds\_data.iloc[:,1].tolist() #jspds\_theta  
y=jcpds\_data.iloc[:,15].tolist() #jcpds intensity  
# print the resulting dataframe  
jcpds\_data1

No. Pos. [°2Th.] d-spacing [Å] Rel. Int. [%]  
0 1 36.6591 2.44942 24.10  
1 2 36.8004 2.44640 29.86  
2 3 39.3476 2.28803 20.52  
3 4 43.5474 2.07661 100.00  
4 5 54.6254 1.67877 23.00  
5 6 70.3455 1.33722 29.03  
6 7 70.8643 1.32870 23.40

Comparison of jcpds data with experimetal data

# Plot the data  
plt.stem(x,y,'ro',label='jcpds')  
plt.stem(x1,y1,'green',label='Experimental')  
  
# Add x-coordinate values to the peak right  
for i, j in zip(x, y):  
 plt.annotate(str(i), xy=(i, j), xytext=(5, 0), textcoords="offset points", ha='left', fontsize=8, color='red')  
  
for i, j in zip(x1, y1):  
 plt.annotate(str(i), xy=(i, j), xytext=(5, 0), textcoords="offset points", ha='left', fontsize=6, color='green')  
  
# Add legend  
plt.legend()  
  
# Set x-axis and y-axis limits  
plt.xlim(36,72)  
plt.ylim(0,400)  
  
plt.xlabel('2-Theta')  
plt.ylabel('Intensity')  
# Add title to the figure  
plt.title('Comparison of jcpds and determined values')  
  
# Display the plot  
plt.show()



Calculation for d\_spacing for detected peak theta value

# max\_indices = np.argmax(data['Intensity (Counts)'])  
# max\_2theta = data['Angle (2theta)'][max\_indices]  
wavelength = 1.5406 # Cu K-alpha radiation  
value=Theta['Theta'].to\_numpy()#[36.59, 36.95, 39.35, 43.55, 54.59, 70.37, 70.91]  
print("{0} \t {1} ".format("2theta","d-spacing"))  
d\_list=[]  
theta\_list=[]  
for i in value:  
 theta=i/2  
 #theta = 70.91 / 2  
 d\_spacing = wavelength / (2 \* np.sin(np.deg2rad(theta))) #brags law  
 d\_list.append(d\_spacing)  
 theta\_list.append(i)  
  
 #print("{0} {1} ".format(i,d\_spacing))  
  
d\_data={'2-Theta':theta\_list,'d-spacing':d\_list}   
dd=pd.DataFrame(d\_data)  
dd

2theta d-spacing

2-Theta d-spacing  
0 36.59 2.453891  
1 36.95 2.430805  
2 39.35 2.287899  
3 43.55 2.076489  
4 54.59 1.679780  
5 70.37 1.336820  
6 70.91 1.327958

we take corresponding hkl planes and perform operations on it and calculate the abc values

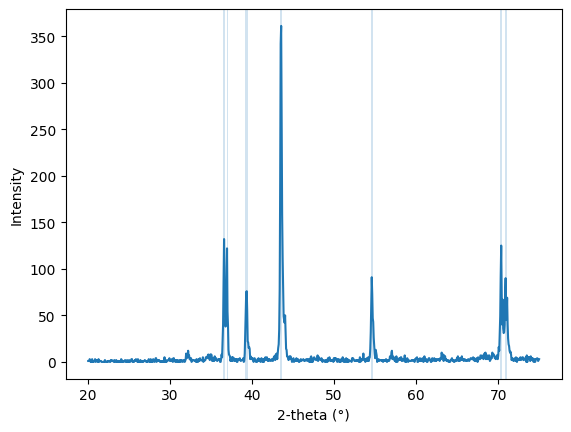
import pandas as pd  
  
# hkl data from jspds  
  
hkl\_data = ['002', '100', '101', '102', '110', '112', '201']  
  
# Create a dictionary to store the data for each column  
data1 = {'2theta':theta\_list,'d\_spacing':d\_list,'h': [], 'k': [], 'l': [], '(h^2+k^2+l^2)': [],'root':[],'abc':abc}  
  
# Iterate over each three-digit number and split it into its individual digits, square each digit, and calculate the sum of the squares  
for num in hkl\_data:  
 h, k, l = str(num)  
 h, k, l = int(h), int(k), int(l)  
 squared = h\*\*2 + k\*\*2 + l\*\*2  
 root=(squared)\*\*(0.5)  
 data1['h'].append(h)  
 data1['k'].append(k)  
 data1['l'].append(l)  
 data1['(h^2+k^2+l^2)'].append(squared)  
 data1['root'].append(root)  
  
# Create a DataFrame from the data dictionary  
df = pd.DataFrame(data1)  
  
# Save the DataFrame to a CSV file  
df.to\_csv('output.csv', index=False)  
  
#calculation for appx. abc values  
d2=df['d\_spacing'].tolist()  
root=df['root'].tolist()  
d2=np.array(d2)  
root=np.array(root)  
abc=np.multiply(d2,root).tolist()  
  
# highlight abc column  
highlight\_abc = lambda x: 'background-color: pink'  
styled\_df = df.style.applymap(highlight\_abc, subset=['abc'])

styled\_df

<pandas.io.formats.style.Styler at 0x283eeb425f0>

**To calculate FWHM and Grain size (D)**

from scipy.signal import find\_peaks  
from math import pi, cos  
  
# Load XRD data from CSV file  
xrd\_data = data  
  
# Plot XRD pattern  
plt.plot(xrd\_data['Angle (2theta)'], xrd\_data['Intensity (Counts)'])  
plt.xlabel('2-theta (°)')  
plt.ylabel('Intensity')  
  
# Find peaks  
peaks, \_ = find\_peaks(xrd\_data['Intensity (Counts)'], height=70)  
  
#save fwhm and grain size in new list  
FWHM=[]  
Grain\_size\_D=[]  
for peak\_index in peaks:  
 # Find left and right indices where intensity drops to half of the maximum  
 half\_max\_height = xrd\_data['Intensity (Counts)'][peak\_index] / 2  
 left\_index = peak\_index - 1  
 while xrd\_data['Intensity (Counts)'][left\_index] > half\_max\_height:  
 left\_index -= 1  
 right\_index = peak\_index + 1  
 while xrd\_data['Intensity (Counts)'][right\_index] > half\_max\_height:  
 right\_index += 1  
   
 # Calculate FWHM  
 fwhm = xrd\_data['Angle (2theta)'][right\_index] - xrd\_data['Angle (2theta)'][left\_index]  
 #print('Peak at', xrd\_data['Angle (2theta)'][peak\_index], 'degrees')  
 FWHM.append(fwhm)   
 # Calculate grain size using Scherrer's formula  
 k = 0.9 # shape factor  
 lambda\_ = 1.5406 # wavelength of X-ray (in Angstrom)  
 theta = xrd\_data['Angle (2theta)'][peak\_index] \* pi / 180 # convert to radians  
 grain\_size = k \* lambda\_ / (cos(theta) \* fwhm) # in nanometers  
 Grain\_size\_D.append(grain\_size)  
 #print('Grain size:', grain\_size, 'nm')  
   
 # Highlight peak on XRD plot  
 plt.axvspan(xrd\_data['Angle (2theta)'][left\_index], xrd\_data['Angle (2theta)'][right\_index], alpha=0.2)  
   
plt.show()  
df.insert(loc=2,column='FWHM',value=FWHM)  
df.insert(loc=3,column='Grain\_size (D)',value=Grain\_size\_D)  
# Define the style for index column  
styled\_df = df.style.set\_properties(subset=pd.IndexSlice[:, ['FWHM','Grain\_size (D)']],   
 \*\*{'background-color': 'yellow'})  
  
# Display the formatted dataframe  
display(styled\_df)



<pandas.io.formats.style.Styler at 0x283eeb085b0>

**Baseline correction**

import pandas as pd  
  
# Read the CSV file and store it in a dataframe  
df = pd.read\_csv(r'C:\Users\HP\Downloads\bfo\_xrd1.csv')

20 286  
0 20.02 286  
1 20.04 281  
2 20.06 276  
3 20.08 305  
4 20.10 302  
... ... ...  
2995 79.92 233  
2996 79.94 219  
2997 79.96 240  
2998 79.98 254  
2999 80.00 212  
  
[3000 rows x 2 columns]

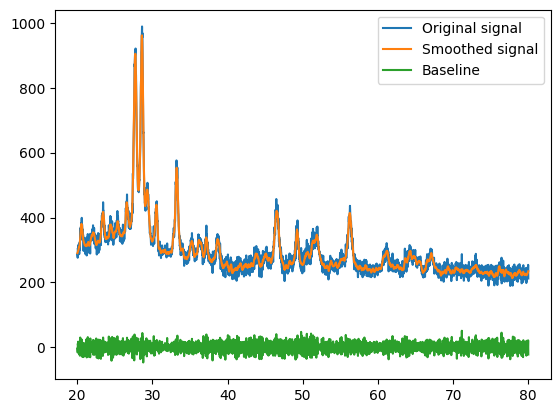
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.signal import savgol\_filter  
from scipy import sparse  
from scipy.sparse.linalg import spsolve  
import pandas as pd

# Load the XRD data from a CSV file  
xrd\_data = data  
x = data.iloc[:,0]  
y = data.iloc[:,1]  
  
# Extract the X and Y data from the CSV file  
  
# Apply the Savitzky-Golay filter to smooth the data and estimate the baseline  
baseline = savgol\_filter(y, 100, 2)  
  
# Subtract the baseline from the original data to obtain the adjusted data  
adjusted\_data = y - baseline  
  
# Save the adjusted data to a new CSV file  
adjusted\_data\_df = pd.DataFrame({'X': x, 'Y': adjusted\_data})  
adjusted\_data\_df.to\_csv('adjusted\_xrd\_data.csv', index=False)

base\_data=pd.read\_csv(r'C:\Users\HP\Downloads\adjusted\_xrd\_data.csv')  
base\_data

X Y  
0 20.02 -31.074438  
1 20.04 -36.611751  
2 20.06 -42.138685  
3 20.08 -13.655242  
4 20.10 -17.161420  
... ... ...  
2995 79.92 7.370522  
2996 79.94 -6.487265  
2997 79.96 14.658339  
2998 79.98 28.807334  
2999 80.00 -13.040280  
  
[3000 rows x 2 columns]

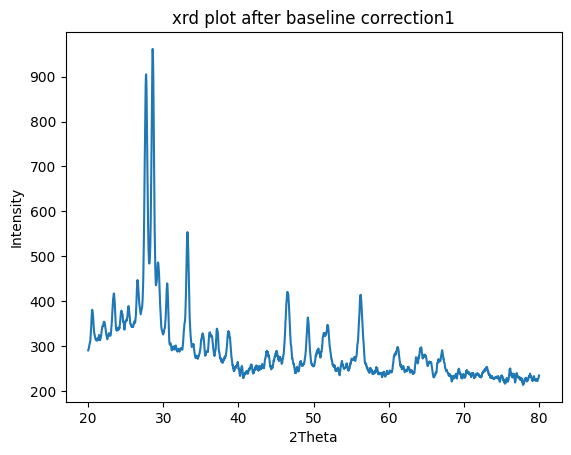
# Apply Savitzky-Golay filter to smooth the data  
y\_smooth = savgol\_filter(y, window\_length=21, polyorder=2)  
  
# Subtract the smoothed signal from the original signal to obtain the baseline  
baseline = y - y\_smooth  
  
# Plot the original signal and the baseline-corrected signal  
import matplotlib.pyplot as plt  
plt.plot(x, y, label='Original signal')  
plt.plot(x, y\_smooth, label='Smoothed signal')  
plt.plot(x, baseline, label='Baseline')  
plt.legend()  
plt.show()  
  
# Save the baseline-corrected data to a new CSV file  
data['Baseline'] = baseline  
data['y\_smooth']=y\_smooth  
data.to\_csv('xrd\_data\_baseline\_corrected.csv', index=False)



peaks, \_ = find\_peaks(y, height=230, distance=50)  
baseline = savgol\_filter(y, window\_length=51, polyorder=3)  
y\_flat = y - baseline

**Data plot after correction of baseline**

plt.plot(x, y\_smooth, label='Smoothed signal')  
plt.title('xrd plot after baseline correction1')  
plt.xlabel("2Theta")  
plt.ylabel("Intensity")  
plt.show()  
baseline = savgol\_filter(y\_smooth, window\_length=250, polyorder=2)  
y\_data\_corrected = y\_smooth - baseline



!pip install pymatgen

from mp\_api.client import MPRester  
from pymatgen.analysis.diffraction.xrd import XRDCalculator  
from pymatgen.symmetry.analyzer import SpacegroupAnalyzer  
  
with MPRester(api\_key='ygczzVeeUPfPlnBGRhMxZ0dRZWcX75df') as mpr:  
 # first retrieve the relevant structure  
 structure = mpr.get\_structure\_by\_material\_id('mp-79')  
  
# important to use the conventional structure to ensure  
# that peaks are labelled with the conventional Miller indices  
sga = SpacegroupAnalyzer(structure)  
conventional\_structure = sga.get\_conventional\_standard\_structure()  
  
# this example shows how to obtain an XRD diffraction pattern  
# these patterns are calculated on-the-fly from the structure  
calculator = XRDCalculator(wavelength='CuKa')  
pattern = calculator.get\_pattern(conventional\_structure)

{"model\_id":"bbf5cb2f02984eb4b90f66d40c012db3","version\_major":2,"version\_minor":0}

pattern

DiffractionPattern  
$2\Theta$: [36.89000473 39.81431701 44.10456105 55.39518347 71.4791533 72.27969535  
 78.51230026 84.02122503 85.8427416 88.71408618]  
Intensity: [ 37.27656114 22.51094754 100. 16.40157294 21.73371872  
 14.00057617 3.65531482 18.30605221 2.17192822 12.09650591]

from mp\_api.client import MPRester  
  
with MPRester(api\_key='ygczzVeeUPfPlnBGRhMxZ0dRZWcX75df') as mpr:  
 surface\_properties\_doc = mpr.surface\_properties.get\_data\_by\_id('mp-79')

c:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\mp\_api\client\mprester.py:147: UserWarning: mpcontribs-client not installed. Install the package to query MPContribs data, or construct pourbaix diagrams: 'pip install mpcontribs-client'  
 warnings.warn(

{"model\_id":"b8e1e2e850664e9b9e73c90289661928","version\_major":2,"version\_minor":0}

surface\_properties\_doc

MPDataDoc<SurfacePropDoc>(  
surfaces=[SurfaceEntry(miller\_index=[2, -1, -1, 2], surface\_energy\_EV\_PER\_ANG2=0.049858712834864614, surface\_energy=0.7988246101579123, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 4.59376062\n\_cell\_length\_b 5.66297186\n\_cell\_length\_c 28.26609457\n\_cell\_angle\_alpha 82.19687645\n\_cell\_angle\_beta 90.00000937\n\_cell\_angle\_gamma 90.00000940\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn24\n\_cell\_volume 728.51493978\n\_cell\_formula\_units\_Z 24\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.663833 0.070876 0.393521 1\n Zn Zn2 1 0.665003 0.547588 0.140193 1\n Zn Zn3 1 0.241334 0.021849 0.462445 1\n Zn Zn4 1 0.154320 0.626595 0.187243 1\n Zn Zn5 1 0.654320 0.873405 0.312757 1\n Zn Zn6 1 0.741334 0.478151 0.037555 1\n Zn Zn7 1 0.746882 0.264214 0.474686 1\n Zn Zn8 1 0.656226 0.703566 0.228574 1\n Zn Zn9 1 0.165003 0.952412 0.359807 1\n Zn Zn10 1 0.163833 0.429123 0.106479 1\n Zn Zn11 1 0.156226 0.796434 0.271426 1\n Zn Zn12 1 0.246882 0.235786 0.025314 1\n Zn Zn13 1 0.845680 0.373405 0.312757 1\n Zn Zn14 1 0.758666 0.978151 0.037555 1\n Zn Zn15 1 0.334997 0.452412 0.359807 1\n Zn Zn16 1 0.336167 0.929123 0.106479 1\n Zn Zn17 1 0.753118 0.764214 0.474686 1\n Zn Zn18 1 0.843774 0.203566 0.228574 1\n Zn Zn19 1 0.836167 0.570877 0.393521 1\n Zn Zn20 1 0.834997 0.047588 0.140193 1\n Zn Zn21 1 0.343774 0.296434 0.271426 1\n Zn Zn22 1 0.253118 0.735786 0.025314 1\n Zn Zn23 1 0.258666 0.521849 0.462445 1\n Zn Zn24 1 0.345680 0.126595 0.187243 1\n", work\_function=4.155235970301791, efermi=-0.5666, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[1, 0, -1, 1], surface\_energy\_EV\_PER\_ANG2=0.04375600971447873, surface\_energy=0.7010485312367011, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 2.65911549\n\_cell\_length\_b 5.60856078\n\_cell\_length\_c 26.59114700\n\_cell\_angle\_alpha 103.71299538\n\_cell\_angle\_beta 60.00001858\n\_cell\_angle\_gamma 76.28698106\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn10\n\_cell\_volume 302.38962581\n\_cell\_formula\_units\_Z 10\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.021684 0.277725 0.068084 1\n Zn Zn2 1 0.958651 0.817277 0.026375 1\n Zn Zn3 1 0.016166 0.241203 0.172466 1\n Zn Zn4 1 0.025301 0.746664 0.120384 1\n Zn Zn5 1 0.992566 0.254545 0.276287 1\n Zn Zn6 1 0.007485 0.745445 0.223708 1\n Zn Zn7 1 0.974710 0.253328 0.379615 1\n Zn Zn8 1 0.983807 0.758807 0.327538 1\n Zn Zn9 1 0.041362 0.182725 0.473626 1\n Zn Zn10 1 0.978267 0.722281 0.431918 1\n", work\_function=3.9063081115052793, efermi=-0.5292, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[2, 1, -3, 0], surface\_energy\_EV\_PER\_ANG2=0.05028248677993825, surface\_energy=0.8056142169732693, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 4.99382900\n\_cell\_length\_b 7.00939255\n\_cell\_length\_c 21.20770200\n\_cell\_angle\_alpha 79.09969358\n\_cell\_angle\_beta 90.00000000\n\_cell\_angle\_gamma 90.00000000\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn24\n\_cell\_volume 728.95453608\n\_cell\_formula\_units\_Z 24\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.750000 0.611097 0.022391 1\n Zn Zn2 1 0.750000 0.258627 0.092354 1\n Zn Zn3 1 0.750000 0.952734 0.044105 1\n Zn Zn4 1 0.250000 0.783255 0.019021 1\n Zn Zn5 1 0.250000 0.443693 0.092400 1\n Zn Zn6 1 0.250000 0.106085 0.060291 1\n Zn Zn7 1 0.750000 0.575785 0.147364 1\n Zn Zn8 1 0.750000 0.236541 0.225660 1\n Zn Zn9 1 0.750000 0.911262 0.181291 1\n Zn Zn10 1 0.250000 0.761893 0.147801 1\n Zn Zn11 1 0.250000 0.426003 0.227710 1\n Zn Zn12 1 0.250000 0.100719 0.186346 1\n Zn Zn13 1 0.750000 0.573997 0.272290 1\n Zn Zn14 1 0.750000 0.238107 0.352199 1\n Zn Zn15 1 0.750000 0.899281 0.313654 1\n Zn Zn16 1 0.250000 0.763459 0.274340 1\n Zn Zn17 1 0.250000 0.424215 0.352637 1\n Zn Zn18 1 0.250000 0.088738 0.318709 1\n Zn Zn19 1 0.750000 0.556307 0.407600 1\n Zn Zn20 1 0.750000 0.216745 0.480979 1\n Zn Zn21 1 0.750000 0.893915 0.439709 1\n Zn Zn22 1 0.250000 0.741373 0.407646 1\n Zn Zn23 1 0.250000 0.388903 0.477609 1\n Zn Zn24 1 0.250000 0.047267 0.455895 1\n", work\_function=3.9211903528085656, efermi=-0.5504, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[1, 1, -2, 0], surface\_energy\_EV\_PER\_ANG2=0.05747595203239761, surface\_energy=0.920866231099918, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 5.03691300\n\_cell\_length\_b 4.56831700\n\_cell\_length\_c 21.20319700\n\_cell\_angle\_alpha 90.00000000\n\_cell\_angle\_beta 90.00000000\n\_cell\_angle\_gamma 90.00000000\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn16\n\_cell\_volume 487.89012771\n\_cell\_formula\_units\_Z 16\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.750000 0.377768 0.997388 1\n Zn Zn2 1 0.750000 0.868057 0.053437 1\n Zn Zn3 1 0.250000 0.622232 0.997388 1\n Zn Zn4 1 0.250000 0.131943 0.053437 1\n Zn Zn5 1 0.750000 0.355801 0.124269 1\n Zn Zn6 1 0.750000 0.854268 0.185650 1\n Zn Zn7 1 0.250000 0.644199 0.124269 1\n Zn Zn8 1 0.250000 0.145732 0.185650 1\n Zn Zn9 1 0.750000 0.354268 0.251850 1\n Zn Zn10 1 0.750000 0.855801 0.313231 1\n Zn Zn11 1 0.250000 0.645732 0.251850 1\n Zn Zn12 1 0.250000 0.144199 0.313231 1\n Zn Zn13 1 0.750000 0.368057 0.384063 1\n Zn Zn14 1 0.750000 0.877768 0.440112 1\n Zn Zn15 1 0.250000 0.631943 0.384063 1\n Zn Zn16 1 0.250000 0.122232 0.440112 1\n", work\_function=4.060610224816135, efermi=-0.6139, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[1, 0, -1, 2], surface\_energy\_EV\_PER\_ANG2=0.04412010415371362, surface\_energy=0.7068819669983859, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 2.65966882\n\_cell\_length\_b 7.26939376\n\_cell\_length\_c 27.06148620\n\_cell\_angle\_alpha 85.56823435\n\_cell\_angle\_beta 90.00001077\n\_cell\_angle\_gamma 68.53873172\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn16\n\_cell\_volume 485.25351327\n\_cell\_formula\_units\_Z 16\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.209158 0.790842 0.116105 1\n Zn Zn2 1 0.433444 0.066556 0.050324 1\n Zn Zn3 1 0.969346 0.530654 0.176573 1\n Zn Zn4 1 0.716750 0.283250 0.241289 1\n Zn Zn5 1 0.533194 0.966806 0.193585 1\n Zn Zn6 1 0.770126 0.229874 0.128750 1\n Zn Zn7 1 0.222523 0.777477 0.009862 1\n Zn Zn8 1 0.019078 0.480922 0.059736 1\n Zn Zn9 1 0.229874 0.770126 0.371250 1\n Zn Zn10 1 0.466806 0.033194 0.306415 1\n Zn Zn11 1 0.980922 0.519078 0.440264 1\n Zn Zn12 1 0.777477 0.222523 0.490138 1\n Zn Zn13 1 0.566556 0.933444 0.449676 1\n Zn Zn14 1 0.790842 0.209158 0.383895 1\n Zn Zn15 1 0.283250 0.716750 0.258711 1\n Zn Zn16 1 0.030654 0.469346 0.323427 1\n", work\_function=4.2631710537985255, efermi=-0.6426, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[2, 1, -3, 1], surface\_energy\_EV\_PER\_ANG2=0.05217617082022478, surface\_energy=0.835954378787201, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 5.63908065\n\_cell\_length\_b 6.76664547\n\_cell\_length\_c 20.66784977\n\_cell\_angle\_alpha 84.07041388\n\_cell\_angle\_beta 88.23642285\n\_cell\_angle\_gamma 68.13902878\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn24\n\_cell\_volume 727.98462949\n\_cell\_formula\_units\_Z 24\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.956565 0.807576 0.919058 1\n Zn Zn2 1 0.353128 0.486541 0.881283 1\n Zn Zn3 1 0.549343 0.105783 0.962741 1\n Zn Zn4 1 0.458554 0.797250 0.920727 1\n Zn Zn5 1 0.660728 0.416521 0.983922 1\n Zn Zn6 1 0.072853 0.117307 0.960244 1\n Zn Zn7 1 0.169109 0.847698 0.797067 1\n Zn Zn8 1 0.500947 0.503052 0.753226 1\n Zn Zn9 1 0.771611 0.177106 0.837946 1\n Zn Zn10 1 0.670425 0.849149 0.792256 1\n Zn Zn11 1 0.879442 0.494499 0.874735 1\n Zn Zn12 1 0.279480 0.172424 0.833650 1\n Zn Zn13 1 0.220520 0.827576 0.666350 1\n Zn Zn14 1 0.620558 0.505501 0.625265 1\n Zn Zn15 1 0.829575 0.150851 0.707744 1\n Zn Zn16 1 0.728389 0.822894 0.662054 1\n Zn Zn17 1 0.999053 0.496948 0.746774 1\n Zn Zn18 1 0.330891 0.152302 0.702933 1\n Zn Zn19 1 0.427147 0.882693 0.539756 1\n Zn Zn20 1 0.839272 0.583479 0.516078 1\n Zn Zn21 1 0.041446 0.202750 0.579273 1\n Zn Zn22 1 0.950657 0.894217 0.537259 1\n Zn Zn23 1 0.146872 0.513459 0.618717 1\n Zn Zn24 1 0.543435 0.192424 0.580942 1\n", work\_function=3.933140504685586, efermi=-0.5572, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[2, 2, -4, 1], surface\_energy\_EV\_PER\_ANG2=0.051946457678011634, surface\_energy=0.8322739686674226, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 4.57904800\n\_cell\_length\_b 10.33350030\n\_cell\_length\_c 21.25762700\n\_cell\_angle\_alpha 97.38709375\n\_cell\_angle\_beta 90.00000000\n\_cell\_angle\_gamma 77.19913270\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn32\n\_cell\_volume 972.29700544\n\_cell\_formula\_units\_Z 32\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.613178 0.600483 0.032817 1\n Zn Zn2 1 0.069496 0.163033 0.091485 1\n Zn Zn3 1 0.530346 0.217167 0.030966 1\n Zn Zn4 1 0.042675 0.721880 0.033877 1\n Zn Zn5 1 0.425373 0.873211 0.072923 1\n Zn Zn6 1 0.705800 0.411646 0.105866 1\n Zn Zn7 1 0.178880 0.455053 0.039838 1\n Zn Zn8 1 0.851023 0.976591 0.032959 1\n Zn Zn9 1 0.814203 0.614934 0.205083 1\n Zn Zn10 1 0.072731 0.106271 0.217281 1\n Zn Zn11 1 0.569011 0.117138 0.154421 1\n Zn Zn12 1 0.298272 0.636726 0.146822 1\n Zn Zn13 1 0.441318 0.864028 0.214588 1\n Zn Zn14 1 0.702645 0.357963 0.227760 1\n Zn Zn15 1 0.210095 0.375805 0.165422 1\n Zn Zn16 1 0.924824 0.872153 0.151415 1\n Zn Zn17 1 0.789905 0.624195 0.334578 1\n Zn Zn18 1 0.075176 0.127847 0.348585 1\n Zn Zn19 1 0.558682 0.135972 0.285412 1\n Zn Zn20 1 0.297355 0.642037 0.272240 1\n Zn Zn21 1 0.430989 0.882862 0.345579 1\n Zn Zn22 1 0.701728 0.363274 0.353178 1\n Zn Zn23 1 0.185797 0.385066 0.294917 1\n Zn Zn24 1 0.927269 0.893729 0.282719 1\n Zn Zn25 1 0.821120 0.544947 0.460162 1\n Zn Zn26 1 0.148977 0.023409 0.467040 1\n Zn Zn27 1 0.574627 0.126789 0.427076 1\n Zn Zn28 1 0.294200 0.588354 0.394134 1\n Zn Zn29 1 0.469654 0.782833 0.469035 1\n Zn Zn30 1 0.957325 0.278120 0.466123 1\n Zn Zn31 1 0.386822 0.399517 0.467183 1\n Zn Zn32 1 0.930504 0.836967 0.408515 1\n", work\_function=4.128799878057071, efermi=-0.6368, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[1, 1, -2, 1], surface\_energy\_EV\_PER\_ANG2=0.058244034240247794, surface\_energy=0.9331722641956242, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 4.56722100\n\_cell\_length\_b 5.67682500\n\_cell\_length\_c 21.22394400\n\_cell\_angle\_alpha 103.51315689\n\_cell\_angle\_beta 90.00000000\n\_cell\_angle\_gamma 66.27989539\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn16\n\_cell\_volume 487.10759464\n\_cell\_formula\_units\_Z 16\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.007292 0.275147 0.067006 1\n Zn Zn2 1 0.456403 0.364180 0.021341 1\n Zn Zn3 1 0.459886 0.789186 0.107760 1\n Zn Zn4 1 0.947095 0.808381 0.046051 1\n Zn Zn5 1 0.017856 0.255990 0.201114 1\n Zn Zn6 1 0.515362 0.260362 0.139090 1\n Zn Zn7 1 0.476029 0.751697 0.234203 1\n Zn Zn8 1 0.974701 0.754380 0.171564 1\n Zn Zn9 1 0.025299 0.245620 0.328436 1\n Zn Zn10 1 0.523971 0.248303 0.265797 1\n Zn Zn11 1 0.484638 0.739638 0.360910 1\n Zn Zn12 1 0.982144 0.744010 0.298886 1\n Zn Zn13 1 0.052905 0.191619 0.453949 1\n Zn Zn14 1 0.540114 0.210814 0.392240 1\n Zn Zn15 1 0.543597 0.635820 0.478659 1\n Zn Zn16 1 0.992708 0.724853 0.432994 1\n", work\_function=3.939817043546375, efermi=-0.5502, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[2, 1, -3, 2], surface\_energy\_EV\_PER\_ANG2=0.05557558092787579, surface\_energy=0.8904189307102564, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 7.00370555\n\_cell\_length\_b 5.64909904\n\_cell\_length\_c 26.55704500\n\_cell\_angle\_alpha 103.59489641\n\_cell\_angle\_beta 79.07097978\n\_cell\_angle\_gamma 69.20080658\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn30\n\_cell\_volume 910.59413604\n\_cell\_formula\_units\_Z 30\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.028291 0.253341 0.062157 1\n Zn Zn2 1 0.354761 0.257189 0.094152 1\n Zn Zn3 1 0.689224 0.282089 0.029821 1\n Zn Zn4 1 0.643772 0.830660 0.009393 1\n Zn Zn5 1 0.989154 0.797223 0.029873 1\n Zn Zn6 1 0.330486 0.734244 0.056559 1\n Zn Zn7 1 0.032723 0.216454 0.162312 1\n Zn Zn8 1 0.355932 0.228446 0.197534 1\n Zn Zn9 1 0.688754 0.230589 0.131258 1\n Zn Zn10 1 0.661068 0.737199 0.101576 1\n Zn Zn11 1 0.999407 0.733766 0.129956 1\n Zn Zn12 1 0.327668 0.725744 0.165454 1\n Zn Zn13 1 0.007877 0.266764 0.266672 1\n Zn Zn14 1 0.333670 0.284505 0.300744 1\n Zn Zn15 1 0.688729 0.227711 0.230927 1\n Zn Zn16 1 0.666330 0.715495 0.199256 1\n Zn Zn17 1 0.992123 0.733236 0.233328 1\n Zn Zn18 1 0.311271 0.772289 0.269073 1\n Zn Zn19 1 0.000593 0.266234 0.370044 1\n Zn Zn20 1 0.338932 0.262801 0.398424 1\n Zn Zn21 1 0.672332 0.274256 0.334546 1\n Zn Zn22 1 0.644068 0.771554 0.302466 1\n Zn Zn23 1 0.967277 0.783546 0.337688 1\n Zn Zn24 1 0.311246 0.769411 0.368742 1\n Zn Zn25 1 0.010846 0.202777 0.470127 1\n Zn Zn26 1 0.356228 0.169340 0.490607 1\n Zn Zn27 1 0.669514 0.265756 0.443441 1\n Zn Zn28 1 0.645239 0.742811 0.405848 1\n Zn Zn29 1 0.971709 0.746659 0.437843 1\n Zn Zn30 1 0.310776 0.717911 0.470179 1\n", work\_function=4.041841841549309, efermi=-0.5335, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[1, 0, -1, 0], surface\_energy\_EV\_PER\_ANG2=0.032960744647671514, surface\_energy=0.5280893247464475, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 2.66098286\n\_cell\_length\_b 4.92216800\n\_cell\_length\_c 26.60982500\n\_cell\_angle\_alpha 90.00000000\n\_cell\_angle\_beta 60.00001065\n\_cell\_angle\_gamma 90.00000000\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn10\n\_cell\_volume 301.83611752\n\_cell\_formula\_units\_Z 10\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.299827 0.250000 0.040034 1\n Zn Zn2 1 0.798563 0.750000 0.040288 1\n Zn Zn3 1 0.268487 0.250000 0.146302 1\n Zn Zn4 1 0.766125 0.750000 0.146775 1\n Zn Zn5 1 0.251135 0.250000 0.249773 1\n Zn Zn6 1 0.748865 0.750000 0.250227 1\n Zn Zn7 1 0.233875 0.250000 0.353225 1\n Zn Zn8 1 0.731513 0.750000 0.353698 1\n Zn Zn9 1 0.201437 0.250000 0.459712 1\n Zn Zn10 1 0.700173 0.750000 0.459966 1\n", work\_function=4.0750544273143205, efermi=-0.554, area\_fraction=0.5587169254603455, has\_wulff=True), SurfaceEntry(miller\_index=[2, 0, -2, 1], surface\_energy\_EV\_PER\_ANG2=0.05064493839965916, surface\_energy=0.8114213318657781, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 2.65716304\n\_cell\_length\_b 10.33397767\n\_cell\_length\_c 27.47409520\n\_cell\_angle\_alpha 102.80017596\n\_cell\_angle\_beta 90.00001879\n\_cell\_angle\_gamma 82.61335132\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn24\n\_cell\_volume 729.24307035\n\_cell\_formula\_units\_Z 24\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.436237 0.127525 0.036708 1\n Zn Zn2 1 0.227301 0.545398 0.064335 1\n Zn Zn3 1 0.928438 0.143124 0.142985 1\n Zn Zn4 1 0.693290 0.613420 0.161846 1\n Zn Zn5 1 0.829257 0.341486 0.088020 1\n Zn Zn6 1 0.615135 0.769730 0.095465 1\n Zn Zn7 1 0.336736 0.326528 0.987246 1\n Zn Zn8 1 0.033683 0.932634 0.056507 1\n Zn Zn9 1 0.449285 0.101431 0.223884 1\n Zn Zn10 1 0.172935 0.654130 0.248188 1\n Zn Zn11 1 0.932129 0.135743 0.315957 1\n Zn Zn12 1 0.697909 0.604182 0.324110 1\n Zn Zn13 1 0.827065 0.345870 0.251812 1\n Zn Zn14 1 0.550715 0.898569 0.276116 1\n Zn Zn15 1 0.302091 0.395818 0.175890 1\n Zn Zn16 1 0.067872 0.864257 0.184043 1\n Zn Zn17 1 0.384865 0.230270 0.404535 1\n Zn Zn18 1 0.170743 0.658514 0.411980 1\n Zn Zn19 1 0.966317 0.067366 0.443493 1\n Zn Zn20 1 0.663264 0.673472 0.512754 1\n Zn Zn21 1 0.772699 0.454602 0.435665 1\n Zn Zn22 1 0.563763 0.872475 0.463292 1\n Zn Zn23 1 0.306710 0.386580 0.338154 1\n Zn Zn24 1 0.071562 0.856876 0.357015 1\n", work\_function=4.169229806601448, efermi=-0.6002, area\_fraction=0.0, has\_wulff=True), SurfaceEntry(miller\_index=[0, 0, 0, 1], surface\_energy\_EV\_PER\_ANG2=0.02086611403535226, surface\_energy=0.334311988057284, is\_reconstructed=False, structure="# generated using pymatgen\ndata\_Zn\n\_symmetry\_space\_group\_name\_H-M 'P 1'\n\_cell\_length\_a 2.66132000\n\_cell\_length\_b 2.66131987\n\_cell\_length\_c 39.35363200\n\_cell\_angle\_alpha 90.00000000\n\_cell\_angle\_beta 90.00000000\n\_cell\_angle\_gamma 120.00002648\n\_symmetry\_Int\_Tables\_number 1\n\_chemical\_formula\_structural Zn\n\_chemical\_formula\_sum Zn8\n\_cell\_volume 241.38457274\n\_cell\_formula\_units\_Z 8\nloop\_\n \_symmetry\_equiv\_pos\_site\_id\n \_symmetry\_equiv\_pos\_as\_xyz\n 1 'x, y, z'\nloop\_\n \_atom\_site\_type\_symbol\n \_atom\_site\_label\n \_atom\_site\_symmetry\_multiplicity\n \_atom\_site\_fract\_x\n \_atom\_site\_fract\_y\n \_atom\_site\_fract\_z\n \_atom\_site\_occupancy\n Zn Zn1 1 0.333333 0.666667 0.280938 1\n Zn Zn2 1 0.666667 0.333333 0.344526 1\n Zn Zn3 1 0.333333 0.666667 0.406038 1\n Zn Zn4 1 0.666667 0.333333 0.468340 1\n Zn Zn5 1 0.333333 0.666667 0.531660 1\n Zn Zn6 1 0.666667 0.333333 0.593962 1\n Zn Zn7 1 0.333333 0.666667 0.655474 1\n Zn Zn8 1 0.666667 0.333333 0.719062 1\n", work\_function=3.9185972726184413, efermi=-0.37246069, area\_fraction=0.44128307453965454, has\_wulff=True)],  
weighted\_surface\_energy\_EV\_PER\_ANG2=0.027623588865645848,  
weighted\_surface\_energy=0.44257866583614763,  
surface\_anisotropy=0.21740376274172288,  
pretty\_formula='Zn',  
shape\_factor=5.85949104225898,  
weighted\_work\_function=4.0060125330563965,  
has\_reconstructed=False,  
task\_id='mp-79',  
structure=Structure Summary  
Lattice  
 abc : 2.626730472467679 2.626730472467679 5.207234  
 angles : 90.0 90.0 120.00000000000001  
 volume : 31.114924432303443  
 A : 1.3133652362338395 -2.274815318051711 0.0  
 B : 1.3133652362338395 2.274815318051711 0.0  
 C : 0.0 0.0 5.207234  
 pbc : True True True  
PeriodicSite: Zn (1.3134, -0.7583, 3.9054) [0.6667, 0.3333, 0.7500]  
PeriodicSite: Zn (1.3134, 0.7583, 1.3018) [0.3333, 0.6667, 0.2500],  
fields\_not\_requested=[]  
)

mpr.get\_structures  
structure

Structure Summary  
Lattice  
 abc : 2.61436012 2.614359482682322 4.87316102  
 angles : 90.0 90.0 119.99998275785532  
 volume : 28.845108913918352  
 A : 2.61436012 0.0 0.0  
 B : -1.30717906 2.26410212 0.0  
 C : 0.0 -0.0 4.87316102  
 pbc : True True True  
PeriodicSite: Zn (-0.0000, 1.5094, 1.2183) [0.3333, 0.6667, 0.2500]  
PeriodicSite: Zn (1.3072, 0.7547, 3.6549) [0.6667, 0.3333, 0.7500]