**Biswas et al. 2023 (partial dyke with one tip)**

import pandas as pd  
from itertools import combinations  
from pathlib import Path  
from openpyxl import Workbook  
from openpyxl.utils.dataframe import dataframe\_to\_rows  
from openpyxl.utils import get\_column\_letter  
from openpyxl.styles import Font, Alignment  
  
input\_dir = Path.home() / 'Desktop' / 'data.xlsx'  
output\_dir = Path.home() / 'Desktop' / 'combi.xlsx'  
  
# read the excel file into a pandas dataframe  
df = pd.read\_excel(input\_dir)  
  
# extract the values from the first two columns as lists  
x\_values = df["X"].tolist()  
y\_values = df["Y/2"].tolist()  
  
# convert the values to floats  
x\_values = [float(x) for x in x\_values]  
y\_values = [float(y) for y in y\_values]  
  
# create all combinations of x\_values and y\_values  
comb1 = list(combinations(x\_values, 2))  
comb2 = list(combinations(y\_values, 2))  
  
# create a new dataframe with the combinations  
output\_df = pd.DataFrame({  
 "x1": [x[0] for x in comb1],  
 "x2": [x[1] for x in comb1],  
 "y1": [y[0] for y in comb2],  
 "y2": [y[1] for y in comb2]  
})  
  
workbook = Workbook()  
  
# create an Excel sheet object  
worksheet = workbook.active  
worksheet.title = "Sheet1"  
  
# write the output dataframe to the Excel sheet  
for r in dataframe\_to\_rows(output\_df, index=False, header=True):  
 worksheet.append(r)  
  
# write formulas to calculate the other columns  
headers = ["x1^2", "x2^2", "y1^2", "y2^2", "2a", "a", "A", "A^2", "b^2", "b", "B", "e^2", "e", "B/A"]  
formulas = [  
 "=A{row}^2", "=B{row}^2", "=C{row}^2", "=D{row}^2",  
 "=((H{row}\*E{row})-(G{row}\*F{row}))/((A{row}\*H{row})-(B{row}\*G{row}))",  
 "=I{row}/2", "=J{row}\*2", "=J{row}\*J{row}",  
 "=(L{row}\*G{row})/((I{row}\*A{row})-E{row})", "=M{row}^0.5", "=N{row}\*2",  
 "=1-(M{row}/L{row})", "=P{row}^0.5", "=O{row}/K{row}"  
]  
for i, header in enumerate(headers):  
 col = get\_column\_letter(i+5)  
 worksheet[col + "1"] = header  
 for row in range(2, len(output\_df) + 2):  
 formula = formulas[i].format(row=row)  
 worksheet[col + str(row)] = formula  
  
bold\_font = Font(bold=True, size=14)  
center\_alignment = Alignment(horizontal='center', vertical='center')  
for cell in worksheet[1]:  
 cell.font = bold\_font  
for row in worksheet.rows:  
 for cell in row:  
 cell.alignment = center\_alignment  
  
# save the workbook to a file  
workbook.save(filename=output\_dir)