

import numpy as np

from scipy.optimize import fsolve

import csv

def heat\_exchanger(shell\_flow\_rate, tube\_flow\_rate, shell\_inlet\_temp, tube\_inlet\_temp, shell\_density, tube\_density, Area, Cp\_s, Cp\_t, U\_dirty, initial\_guess):

def equations(variables):

Th, Tc = variables

eq1 = (tube\_flow\_rate \* tube\_density \* Cp\_t \* (Tc - tube\_inlet\_temp)) - (shell\_flow\_rate \* shell\_density \* Cp\_s \* (shell\_inlet\_temp - Th))

eq2 = tube\_flow\_rate \* tube\_density \* Cp\_t \* (Tc - tube\_inlet\_temp) - U\_dirty \* Area \* (((shell\_inlet\_temp - Tc) - (Th - tube\_inlet\_temp)) / (np.log((shell\_inlet\_temp - Tc) / (Th - tube\_inlet\_temp))))

#print(eq1, eq2)

return [eq1, eq2]

shell\_outlet\_temp, tube\_outlet\_temp = fsolve(equations, initial\_guess)

print(shell\_outlet\_temp, tube\_outlet\_temp)

return shell\_outlet\_temp, tube\_outlet\_temp

def forward\_flow(conditions):

results = {} # To store the Shell and Tube outlet temperatures as a dictionary

results["E103"] = heat\_exchanger(\*\*conditions["E103"]) # E103 HX's Shell and Tube outlet temps

conditions["E104AB"]["tube\_inlet\_temp"] = results["E103"][1] # Passing E103 HX's tube outlet to tube inlet of E104ABC HX

# Below is the continuation of the flow from left to right after E104ABC/AB

results["E104AB"] = heat\_exchanger(\*\*conditions["E104AB"])

conditions["E105"]["tube\_inlet\_temp"] = (results["E104AB"][1] - 2.74)

results["E105"] = heat\_exchanger(\*\*conditions["E105"])

conditions["E106A"]["tube\_inlet\_temp"] = (results["E105"][1] - 8)

results["E106A"] = heat\_exchanger(\*\*conditions["E106A"])

conditions["E107"]["tube\_inlet\_temp"] = results["E106A"][1]

results["E107"] = heat\_exchanger(\*\*conditions["E107"])

conditions["E108AB"]["tube\_inlet\_temp"] = results["E107"][1]

results["E108AB"] = heat\_exchanger(\*\*conditions["E108AB"])

conditions["E109"]["tube\_inlet\_temp"] = results["E108AB"][1]

results["E109"] = heat\_exchanger(\*\*conditions["E109"])

conditions["E110"]["tube\_inlet\_temp"] = results["E109"][1]

results["E110"] = heat\_exchanger(\*\*conditions["E110"])

conditions["E111AB"]["tube\_inlet\_temp"] = results["E110"][1]

results["E111AB"] = heat\_exchanger(\*\*conditions["E111AB"])

return results

def process\_flow(conditions):

results = forward\_flow(conditions)

conditions["E108AB"]["shell\_inlet\_temp"] = results["E111AB"][0] # Feeding back Shell outlet to Shell inlet of E108AB

results["E108AB"] = heat\_exchanger(\*\*conditions["E108AB"]) # Updating the results of outlet temperatures after the above op

conditions["E104AB"]["shell\_inlet\_temp"] = results["E108AB"][0] # Feeding back Shell outlet to Shell inlet of E108AB

conditions["E103"]["shell\_inlet\_temp"] = results["E105"][0]

results["E103"] = heat\_exchanger(\*\*conditions["E103"]) # Updating the results of outlet temperatures after the above op

return results, conditions

def write\_results\_to\_csv(results, iteration, csv\_writer):

for exchanger, (shell\_out, tube\_out) in results.items():

shell\_out = round(shell\_out,2)

tube\_out = round(tube\_out,2)

csv\_writer.writerow([iteration, exchanger, shell\_out, tube\_out])

def run\_simulation(initial\_conditions, num\_iterations=6):

conditions = initial\_conditions.copy()

#print(conditions)

all\_results = []

with open('simulation\_results\_changed\_Udirty.csv', mode='w', newline='') as file:

writer = csv.writer(file)

writer.writerow(['Iteration', 'Heat Exchanger', 'Shell Outlet Temp', 'Tube Outlet Temp'])

for i in range(num\_iterations):

#print(f"Iteration {i+1}")

results, conditions = process\_flow(conditions)

all\_results.append(results)

write\_results\_to\_csv(results, i+1, writer)

return all\_results

# Higher E111AB U\_dirty

initial\_conditions = {

"E103": {"shell\_flow\_rate": 207.9, "tube\_flow\_rate": 1035, "shell\_inlet\_temp": 146.6, "tube\_inlet\_temp": 95.15,

"shell\_density": 779.8, "tube\_density": 875, "Area": 703.1,"Cp\_s":0.566088529455742,"Cp\_t":0.534213494442481,"U\_dirty":225.9, "initial\_guess": (100, 120)},

"E104AB": {"shell\_flow\_rate": 207.9, "tube\_flow\_rate": 519.7, "shell\_inlet\_temp": 212.73, "tube\_inlet\_temp": None,

"shell\_density": 877.16, "tube\_density": 875, "Area": 1955.8,"Cp\_s":0.67204501,"Cp\_t":0.556748477033156,"U\_dirty":96.7, "initial\_guess": (120, 140)},

"E105": {"shell\_flow\_rate": 211, "tube\_flow\_rate": 1039.5, "shell\_inlet\_temp": 208.8, "tube\_inlet\_temp": None,

"shell\_density": 749, "tube\_density": 875, "Area": 644.9,"Cp\_s":0.630306413745319,"Cp\_t":0.578571841535755,"U\_dirty":357.5, "initial\_guess": (145, 155)},

"E106A": {"shell\_flow\_rate": 175, "tube\_flow\_rate": 474, "shell\_inlet\_temp": 214.1, "tube\_inlet\_temp": None,

"shell\_density": 655.7, "tube\_density": 875, "Area": 819.8,"Cp\_s":0.455504463120254,"Cp\_t":0.582896434020995,"U\_dirty":122.4, "initial\_guess": (155, 165)},

"E107": {"shell\_flow\_rate": 179.8, "tube\_flow\_rate": 947.9, "shell\_inlet\_temp": 257.2, "tube\_inlet\_temp": None,

"shell\_density": 661, "tube\_density": 875, "Area": 537,"Cp\_s":0.855037095821607,"Cp\_t":0.597633677673339,"U\_dirty":328.45, "initial\_guess": (170, 180)},

"E108AB": {"shell\_flow\_rate": 199.5, "tube\_flow\_rate": 474, "shell\_inlet\_temp": 268.8, "tube\_inlet\_temp": None,

"shell\_density": 851.9, "tube\_density": 875.1, "Area": 1052.3,"Cp\_s":0.907771557,"Cp\_t":0.622017192077636,"U\_dirty":148.6, "initial\_guess": (200, 220)},

"E109": {"shell\_flow\_rate": 248.9, "tube\_flow\_rate": 947.9, "shell\_inlet\_temp": 298.9, "tube\_inlet\_temp": None,

"shell\_density": 734.9, "tube\_density": 875.1, "Area": 580.8,"Cp\_s":0.538406851140075,"Cp\_t":0.644237101745605,"U\_dirty":217.87, "initial\_guess": (210, 240)},

"E110": {"shell\_flow\_rate": 399.1, "tube\_flow\_rate": 947.9, "shell\_inlet\_temp": 299.5, "tube\_inlet\_temp": None,

"shell\_density": 677.3, "tube\_density": 875.1, "Area": 402.6,"Cp\_s":0.997496898964167,"Cp\_t":0.658227702331542,"U\_dirty":353.7, "initial\_guess": (230, 270)},

"E111AB": {"shell\_flow\_rate": 199.5, "tube\_flow\_rate": 474.0, "shell\_inlet\_temp": 349.3, "tube\_inlet\_temp": None,

"shell\_density": 815.3, "tube\_density": 875.1, "Area": 1153.8,"Cp\_s":0.794846381463514,"Cp\_t":0.684942805480957,"U\_dirty":168.4, "initial\_guess": (245, 260)},

}  
  
  
simulation\_results = run\_simulation(initial\_conditions)

for i, results in enumerate(simulation\_results):

print(f"\nIteration {i+1} Results:\n")

for exchanger, (shell\_out, tube\_out) in results.items():

print(f"{exchanger}: Shell out = {shell\_out:.2f}, Tube out = {tube\_out:.2f}")