import matplotlib.pyplot as plt

import math

# point cloud points sampled at equal intervals (24)

# starts at wiki gate 1 and ordered by rising elevation

points22 = [[-0.58299971, 12.66082001, -0.126706], [0.25814056, 12.43486309, -0.12684999],

[1.0097146, 12.00007153, -0.127079], [1.625175, 11.38345146, -0.126563],

[2.05816555, 10.63166714, -0.126628], [2.28238845, 9.79162979, -0.127224],

[2.28181314, 8.92331409, -0.126909], [2.05603147, 8.08386517, -0.12664001],

[1.62148714, 7.331388, -0.127205], [1.00670099, 6.71646881, -0.127313],

[0.25566006, 6.28195524, -0.12650999], [-0.58354139, 6.05611801, -0.12712499],

[-1.45354009, 6.05542564, -0.126553], [-2.29301953, 6.28169537, -0.12653001],

[-3.04554558, 6.71644878, -0.12664101], [-3.66146946, 7.33325195, -0.127333],

[-4.09496498, 8.08426666, -0.126644], [-4.31871605, 8.92384529, -0.127023],

[-4.31711388, 9.79423904, -0.126669], [-4.09208632, 10.63205338, -0.12721001],

[-3.65745878, 11.38477325, -0.126517], [-3.04275942, 11.99959183, -0.127316],

[-2.29086685, 12.43390179, -0.126361], [-1.45263338, 12.65971088, -0.127249]]

points28 = [[-0.58260727, 12.65959072, -0.28241101], [0.25699759, 12.43423843, -0.28284001],

[1.00955105, 11.99892044, -0.28244999], [1.6239233, 11.38328171, -0.28244799],

[2.05819488, 10.62919998, -0.282197], [2.28205657, 9.79082108, -0.28236201],

[2.28092837, 8.92148018, -0.28193501], [2.05518794, 8.08257103, -0.282711],

[1.62147665, 7.33114147, -0.281946], [1.0069387, 6.71612549, -0.28215799],

[0.25420094, 6.28062534, -0.28211799], [-0.58464217, 6.05478382, -0.282186],

[-1.45375395, 6.0544343, -0.28219101], [-2.29503107, 6.28127432, -0.28273201],

[-3.04593921, 6.7152524, -0.282675], [-3.66210866, 7.3320117, -0.28248501],

[-4.09523964, 8.08343697, -0.28242099], [-4.31931019, 8.92341518, -0.282197],

[-4.31815147, 9.79191589, -0.28207901], [-4.09119797, 10.6342144, -0.28285199],

[-3.65751266, 11.38440895, -0.28267699], [-3.04321814, 11.99855518, -0.28207201],

[-2.29169989, 12.43260384, -0.28255999], [-1.45092154, 12.65892506, -0.28195]]

points31 = [[-0.58279896, 12.65940475, -0.33619699], [0.25684166, 12.4337616, -0.33674201],

[1.00976133, 11.99857807, -0.33671099], [1.62369394, 11.38366318, -0.336384],

[2.05780983, 10.62991714, -0.33602899], [2.28172421, 9.79027653, -0.33614901],

[2.28075266, 8.92212009, -0.336777], [2.05505657, 8.08263588, -0.33611199],

[1.62087083, 7.33015394, -0.33612999], [1.00669789, 6.71609783, -0.336211],

[0.25461102, 6.28034353, -0.33675], [-0.58383727, 6.05462027, -0.336142],

[-1.45341206, 6.05417538, -0.33605701], [-2.29430938, 6.28086853, -0.336817],

[-3.04615569, 6.71535206, -0.33647299], [-3.66098475, 7.32951498, -0.336339],

[-4.09529781, 8.08292294, -0.33605599], [-4.31957865, 8.92395115, -0.336528],

[-4.31790829, 9.79197311, -0.33647901], [-4.09311771, 10.63043022, -0.33628899],

[-3.65824461, 11.3833313, -0.33596599], [-3.0443573, 11.99722576, -0.33642101],

[-2.29101324, 12.43290138, -0.33628401], [-1.45309782, 12.65841866, -0.33654499]]

points37 = [[-0.58373833, 12.65884972, -0.496685], [0.25585938, 12.43335915, -0.49620599],

[1.00941229, 11.99696064, -0.49621001], [1.6234622, 11.38186264, -0.49638301],

[2.05662155, 10.62987518, -0.49616799], [2.28146076, 9.79000854, -0.49671501],

[2.28039551, 8.91981411, -0.496126], [2.0542686, 8.08050346, -0.49601999],

[1.61989903, 7.32878494, -0.496723], [1.00640368, 6.714324, -0.49686],

[0.25397348, 6.27925587, -0.49680701], [-0.58626723, 6.05313396, -0.49647301],

[-1.45444274, 6.05351782, -0.49603301], [-2.29556465, 6.28043556, -0.49643299],

[-3.04638934, 6.71424294, -0.49649301], [-3.66107941, 7.32862854, -0.49687001],

[-4.09577274, 8.0817709, -0.49661201], [-4.31968737, 8.92141247, -0.496492],

[-4.31857109, 9.79117012, -0.49654701], [-4.09318924, 10.63008499, -0.496319],

[-3.65885925, 11.38182068, -0.49652299], [-3.04486799, 11.99636459, -0.49686],

[-2.29260159, 12.43149567, -0.49641401], [-1.45397377, 12.65742302, -0.49662599]]

points40 = [[-0.58277225, 12.65732098, -0.57818002], [0.25658751, 12.43192482, -0.57772601],

[1.00784159, 11.99698353, -0.57810801], [1.6230278, 11.38033295, -0.57774401],

[2.05676007, 10.62687874, -0.57794797], [2.28007698, 9.78898907, -0.57766098],

[2.27886939, 8.92064762, -0.57804501], [2.05347538, 8.08215046, -0.57790101],

[1.61877131, 7.32959509, -0.57718801], [1.0062089, 6.71600914, -0.577658],

[0.25289035, 6.28004837, -0.577784], [-0.58512306, 6.05498409, -0.57754499],

[-1.45431995, 6.05466795, -0.57794601], [-2.29353976, 6.28125, -0.57791001],

[-3.04758811, 6.71673965, -0.57792097], [-3.66022682, 7.32990074, -0.57730401],

[-4.0945673, 8.08263206, -0.57803297], [-4.31893349, 8.92217445, -0.57795],

[-4.31716251, 9.7918129, -0.57807302], [-4.09188938, 10.62883759, -0.57727802],

[-3.65765619, 11.38102531, -0.57750499], [-3.04411554, 11.99547005, -0.57787901],

[-2.29116058, 12.43060684, -0.57737201], [-1.45311594, 12.65635109, -0.578017]]

points43 = [[-0.58483696, 12.65541458, -0.64071703], [0.25420618, 12.43028259, -0.64158797],

[1.00613976, 11.99507427, -0.64151502], [1.62000275, 11.38046074, -0.640688],

[2.05377483, 10.62837124, -0.64144802], [2.27792883, 9.78877258, -0.64103198],

[2.27675772, 8.92096806, -0.64104402], [2.05098391, 8.08220577, -0.64092499],

[1.61768079, 7.33158493, -0.64105201], [1.00223303, 6.71568394, -0.64118701],

[0.25313973, 6.28261662, -0.64167702], [-0.58566403, 6.05675507, -0.64083803],

[-1.45615864, 6.05695105, -0.64149898], [-2.29244113, 6.2830658, -0.64080203],

[-3.04597902, 6.71794128, -0.64139301], [-3.65899014, 7.33180332, -0.64121801],

[-4.09265661, 8.08330154, -0.64107502], [-4.31652451, 8.92306137, -0.64095497],

[-4.31516409, 9.79051781, -0.64143097], [-4.09007311, 10.62789154, -0.64156598],

[-3.65392303, 11.3823576, -0.64092302], [-3.04050636, 11.99505901, -0.64138103],

[-2.29159164, 12.4279232, -0.64128703], [-1.45264626, 12.6542778, -0.64161402]]

points45 = [[-0.58468819, 12.65195847, -0.70790797], [0.25474262, 12.42625999, -0.70795399],

[1.00490284, 11.9921608, -0.70794398], [1.61870527, 11.37647057, -0.70798999],

[2.05040956, 10.62643433, -0.70770901], [2.27423358, 9.78820896, -0.708269],

[2.27270365, 8.92137241, -0.70782799], [2.0479188, 8.08511925, -0.70770103],

[1.61479259, 7.33456659, -0.70794398], [1.00249791, 6.720047, -0.70784497],

[0.25171494, 6.28570843, -0.70754099], [-0.58592129, 6.06035852, -0.70803303],

[-1.45464444, 6.06026983, -0.70814198], [-2.29060721, 6.28598452, -0.70804799],

[-3.04242992, 6.720047, -0.70820397], [-3.65548873, 7.33378839, -0.70802999],

[-4.08887672, 8.08434677, -0.70744503], [-4.31241894, 8.92191315, -0.70812899],

[-4.31102562, 9.79059601, -0.70825398], [-4.08629704, 10.62673283, -0.70825303],

[-3.65333509, 11.37693596, -0.70775503], [-3.03967381, 11.9906435, -0.70809501],

[-2.28998995, 12.42486, -0.70758998], [-1.45113897, 12.65097427, -0.70791698]]

points48 = [[-0.58600426, 12.64647484, -0.78083903], [0.25070047, 12.4218483, -0.78128898],

[1.00173235, 11.9872694, -0.78079897], [1.61315155, 11.37348366, -0.78137499],

[2.04479408, 10.62402248, -0.78098798], [2.26758742, 9.78854752, -0.78121102],

[2.26669407, 8.92297745, -0.78075403], [2.04205179, 8.0863924, -0.780604],

[1.60966516, 7.33761978, -0.78122598], [0.9990716, 6.72491407, -0.78129202],

[0.24383307, 6.28833008, -0.78131002], [-0.58746958, 6.06600332, -0.78056598],

[-1.45378256, 6.0660696, -0.780595], [-2.28879499, 6.29140902, -0.780406],

[-3.04198146, 6.72752762, -0.78100598], [-3.65064168, 7.33705378, -0.78043699],

[-4.08355951, 8.08701801, -0.78134], [-4.30653286, 8.92311478, -0.78102398],

[-4.30522537, 9.78948593, -0.78108299], [-4.08102608, 10.62472725, -0.78071302],

[-3.64831161, 11.37337399, -0.78109002], [-3.03562737, 11.98622131, -0.780586],

[-2.28793478, 12.41915131, -0.78047299], [-1.45100594, 12.64533424, -0.78071499]]

points51 = [[-0.58675671, 12.63925171, -0.85379398], [0.24896336, 12.41459846, -0.85420001],

[0.99700975, 11.98160744, -0.85363799], [1.60684109, 11.37010288, -0.854298],

[2.03841972, 10.62149429, -0.85407299], [2.26127124, 9.78737545, -0.85368901],

[2.25986075, 8.92283916, -0.85369802], [2.03626037, 8.08998203, -0.85344601],

[1.60394311, 7.34173441, -0.85407799], [0.99416304, 6.73036432, -0.85424399],

[0.2469933, 6.29736519, -0.85430801], [-0.5871985, 6.07300186, -0.85349703],

[-1.45159268, 6.07314301, -0.85439098], [-2.28612638, 6.29801846, -0.85388702],

[-3.03492832, 6.73061657, -0.85398698], [-3.64657331, 7.34325123, -0.85429698],

[-4.07838488, 8.09181023, -0.85376698], [-4.29997158, 8.92460346, -0.85399699],

[-4.29884052, 9.78773308, -0.85396802], [-4.07457972, 10.62157059, -0.85419297],

[-3.64150763, 11.37104034, -0.85353303], [-3.03231716, 11.980299, -0.85438401],

[-2.28485537, 12.41249084, -0.85349399], [-1.45031118, 12.63781929, -0.85404801]]

el22 = [1.015, 1, 0.979, 0.955, 0.939, 0.93, 0.906, 0.892, 0.901, 0.935, 0.981, 1.029, 1.049, 1.01, 1.005, 1.01, 0.999,

0.967, 0.912, 0.905, 0.905, 0.901, 0.925, 0.988]

el28 = [0.542, 0.527, 0.508, 0.49, 0.477, 0.461, 0.433, 0.421, 0.429, 0.462, 0.519, 0.565, 0.584, 0.537, 0.542, 0.547,

0.535, 0.499, 0.453, 0.437, 0.435, 0.442, 0.463, 0.516]

el31 = [0.448, 0.438, 0.408, 0.395, 0.383, 0.366, 0.337, 0.324, 0.335, 0.369, 0.428, 0.478, 0.488, 0.448, 0.45, 0.453,

0.448, 0.408, 0.363, 0.348, 0.338, 0.348, 0.37, 0.42]

el37 = [0.323, 0.31, 0.282, 0.268, 0.259, 0.232, 0.208, 0.195, 0.21, 0.255, 0.311, 0.359, 0.37, 0.326, 0.332, 0.335,

0.325, 0.29, 0.235, 0.215, 0.21, 0.22, 0.24, 0.29]

el40 = [0.322, 0.309, 0.278, 0.257, 0.249, 0.228, 0.203, 0.192, 0.203, 0.246, 0.305, 0.349, 0.359, 0.328, 0.327, 0.323,

0.323, 0.282, 0.234, 0.211, 0.205, 0.207, 0.234, 0.288]

el43 = [0.34, 0.324, 0.295, 0.277, 0.265, 0.247, 0.215, 0.202, 0.22, 0.268, 0.326, 0.37, 0.383, 0.335, 0.343, 0.335,

0.344, 0.297, 0.245, 0.225, 0.221, 0.224, 0.25, 0.307]

el45 = [0.367, 0.353, 0.321, 0.288, 0.275, 0.248, 0.218, 0.21, 0.227, 0.293, 0.357, 0.396, 0.403, 0.367, 0.358, 0.344,

0.354, 0.306, 0.254, 0.228, 0.224, 0.234, 0.262, 0.331]

el48 = [0.374, 0.353, 0.322, 0.285, 0.273, 0.248, 0.218, 0.206, 0.228, 0.296, 0.368, 0.403, 0.41, 0.373, 0.357, 0.343,

0.348, 0.306, 0.252, 0.231, 0.226, 0.231, 0.263, 0.328]

el51 = [0.374, 0.352, 0.323, 0.296, 0.284, 0.256, 0.223, 0.216, 0.231, 0.301, 0.368, 0.405, 0.415, 0.366, 0.366,

0.349, 0.361, 0.314, 0.258, 0.236, 0.234, 0.244, 0.275, 0.334]

def ang\_calc(point, center):

if point[0] > center[0] and point[1] > center[1]:

angle = math.degrees(math.atan(math.fabs((point[0] - center[0]) / (point[1] - center[1]))))

elif point[0] > center[0] and point[1] < center[1]:

angle = math.degrees(math.atan(math.fabs((point[1] - center[1]) / (point[0] - center[0])))) + 90

elif point[0] < center[0] and point[1] < center[1]:

angle = math.degrees(math.atan(math.fabs((point[0] - center[0]) / (point[1] - center[1])))) + 180

elif point[0] < center[0] and point[1] > center[1]:

angle = math.degrees(math.atan(math.fabs((point[1] - center[1]) / (point[0] - center[0])))) + 270

elif point[0] == center[0] and point[1] > center[1]:

angle = 0

elif point[0] == center[0] and point[1] < center[1]:

angle = 180

elif point[1] == center[1] and point[0] > center[0]:

angle = 90

elif point[1] == center[1] and point[0] < center[0]:

angle = 270

return angle

def retrieve(filename):

f = open(filename, "r")

new = f.readlines()

points = []

for p in new:

x = []

z = p.replace("[", "").replace("]", "").replace(",", "").split()

x += [float(z[0])]

x += [float(z[1])]

x += [float(z[2])]

points += [x]

f.close()

return points

def translate(value):

valueScaled = value / 360

return 0.5 + (valueScaled \* 24)

"""

new = []

for p in el22:

new += [p \* .0254]

el22 = list(new)

new = []

for p in el28:

new += [p \* .0254]

el28 = list(new)

"""

new = []

for p in el31:

new += [p \* .0254]

el31 = list(new)

new = []

for p in el37:

new += [p \* .0254]

el37 = list(new)

new = []

for p in el40:

new += [p \* .0254]

el40 = list(new)

new = []

for p in el43:

new += [p \* .0254]

el43 = list(new)

new = []

for p in el45:

new += [p \* .0254]

el45 = list(new)

new = []

for p in el48:

new += [p \* .0254]

el48 = list(new)

new = []

for p in el51:

new += [p \* .0254]

el51 = list(new)

# this variable concatenates all sampled point cloud and excel sheet data

# make sure that each elevation appears at the same position in both variables

# (or just in increasing order for simplicity)

points = points31 + points37 + points40 + points43 + points45 + points48 + points51

dist = el31 + el37 + el40 + el43 + el45 + el48 + el51

# starting parameters

xc = -1.0186031433218041

yc = 9.356643518673076

zc = -0.48031974931577454

rc = 3.3223712383492585

alpha = 0.0005 # learning rate

mse = 0.1

while mse > 0.000322238:

x\_change = 0

y\_change = 0

z\_change = 0

r\_change = 0

mse = 0

for i in range(0, len(points)):

mse += (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \*\* 2

x\_change += 2 \* (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \* (

(1 / 2) \* (1 / math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)))) \* (

2 \* (xc - points[i][0]))

y\_change += 2 \* (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \* (

(1 / 2) \* (1 / math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)))) \* (

2 \* (yc - points[i][1]))

z\_change += 2 \* (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \* (

-0.5 \* (1 / (math.sqrt(rc \*\* 2 - (zc - points[i][2]))))) \* (-2 \* (zc - points[i][2]))

r\_change += 2 \* (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \* (

-0.5 \* (1 / (math.sqrt(rc \*\* 2 - (zc - points[i][2]))))) \* (2 \* rc)

mse = math.sqrt(mse / len(points))

print(mse)

rmse = (1 / mse) \* 0.5

x\_change = (x\_change / len(points)) \* rmse

y\_change = (y\_change / len(points)) \* rmse

z\_change = (z\_change / len(points)) \* rmse

r\_change = (r\_change / len(points)) \* rmse

xc = xc - alpha \* x\_change

yc = yc - alpha \* y\_change

zc = zc - alpha \* z\_change

rc = rc - alpha \* r\_change

print(xc, yc, zc, rc)

for i in range(0, len(points)):

mse += (math.sqrt(((xc - points[i][0]) \*\* 2) + ((yc - points[i][1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - points[i][2]) \*\* 2)) - dist[i]) \*\* 2

rmse = math.sqrt(mse / len(points))

print(rmse)

y = [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0, 16.0, 17.0, 18.0, 19.0, 20.0,

21.0, 22.0, 23.0, 24.0]

x22 = []

x22lbl = []

p22 = retrieve("ele22.txt")

for p in p22:

x22 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x22lbl += [translate(ang\_calc(p, [xc, yc]))]

f22 = plt.figure()

plt.title("22\" elevation")

plt.scatter(x22lbl, x22, label="point cloud", s=1)

plt.plot(y, el22, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("22(range).png")

plt.close()

x28 = []

x28lbl = []

p28 = retrieve("ele28.txt")

for p in p28:

x28 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x28lbl += [translate(ang\_calc(p, [xc, yc]))]

f28 = plt.figure()

plt.title("28\" elevation")

plt.scatter(x28lbl, x28, label="point cloud", s=1)

plt.plot(y, el28, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("28(range).png")

plt.close()

x31 = []

x31lbl = []

p31 = retrieve("ele31.txt")

for p in p31:

x31 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x31lbl += [translate(ang\_calc(p, [xc, yc]))]

f31 = plt.figure()

plt.title("31\" elevation")

plt.scatter(x31lbl, x31, label="point cloud", s=1)

plt.plot(y, el31, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("31(range).png")

plt.close()

x37 = []

x37lbl = []

p37 = retrieve("ele37.txt")

for p in p37:

x37 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x37lbl += [translate(ang\_calc(p, [xc, yc]))]

f37 = plt.figure()

plt.title("37\" elevation")

plt.scatter(x37lbl, x37, label="point cloud", s=1)

plt.plot(y, el37, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("37(range).png")

plt.close()

x40 = []

x40lbl = []

p40 = retrieve("ele40.txt")

for p in p40:

x40 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x40lbl += [translate(ang\_calc(p, [xc, yc]))]

f40 = plt.figure()

plt.title("40\" elevation")

plt.scatter(x40lbl, x40, label="point cloud", s=1)

plt.plot(y, el40, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("40(range).png")

plt.close()

x43 = []

x43lbl = []

p43 = retrieve("ele43.txt")

for p in p43:

x43 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x43lbl += [translate(ang\_calc(p, [xc, yc]))]

f43 = plt.figure()

plt.title("43\" elevation")

plt.scatter(x43lbl, x43, label="point cloud", s=1)

plt.plot(y, el43, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("43(range).png")

plt.close()

x45 = []

x45lbl = []

p45 = retrieve("ele45.txt")

for p in p45:

x45 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x45lbl += [translate(ang\_calc(p, [xc, yc]))]

f45 = plt.figure()

plt.title("45\" elevation")

plt.scatter(x45lbl, x45, label="point cloud", s=1)

plt.plot(y, el45, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("45(range).png")

plt.close()

x48 = []

x48lbl = []

p48 = retrieve("ele48.txt")

for p in p48:

x48 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x48lbl += [translate(ang\_calc(p, [xc, yc]))]

f48 = plt.figure()

plt.title("48\" elevation")

plt.scatter(x48lbl, x48, label="point cloud", s=1)

plt.plot(y, el48, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("48(range).png")

plt.close()

x51 = []

x51lbl = []

p51 = retrieve("ele51.txt")

for p in p51:

x51 += [1 \* (math.sqrt(((xc - p[0]) \*\* 2) + ((yc - p[1]) \*\* 2)) - math.sqrt(

(rc \*\* 2) - ((zc - p[2]) \*\* 2)))]

x51lbl += [translate(ang\_calc(p, [xc, yc]))]

f51 = plt.figure()

plt.title("51\" elevation")

plt.scatter(x51lbl, x51, label="point cloud", s=1)

plt.plot(y, el51, color='green', label="physical")

plt.legend(loc='upper left')

plt.xlabel("Wicket Gate")

plt.ylabel("Distance (m)")

plt.savefig("51(range).png")

plt.close()

"""

x = []

y = []

for point in points:

x += [point[0]]

y += [point[1]]

fig, ax = plt.subplots()

circle2 = plt.Circle((xc, yc), rc, color='b', fill=True)

ax.plot(x, y, 'o')

ax.add\_artist(circle2)

fig.savefig('plotcircles2.png')

"""