1. **# ----------------------------------**
2. **# Exercise 01 of Course "Introduction to Wind Turbine Aerodynamics".**
3. **# ----------------------------------**
4. **# Import Libraries**
5. **import numpy as np**
6. **import math**
7. **import os**
8. **import pandas as pd**
9. **import matplotlib.pyplot as plt**
10. **import seaborn as sns**
11. **# a) find R, lambda\_D and z**
12. **# Define the Variables and Values**
13. **R = 63 # [m]**
14. **Lamda\_D = 7 # [-]**
15. **Z = 3 # [-]**
16. **# b) find optimal AoA and cL**
17. **# Read and extract data from Excel File (NACA64\_A17)**
18. **AirfoilFile = pd.read\_excel('C:/Users/s/Desktop/VS\_Codes/Aerodynamic/Lift\_to\_Drag\_Ratio.xlsx')**
19. **print (AirfoilFile)**
20. **# find maximum value of Lift to Drag\_ratio**
21. **Maximum\_Value\_of\_Lift\_to\_Drag\_Ratio = AirfoilFile['Lift to drag ratio'].max()**
22. **print (Maximum\_Value\_of\_Lift\_to\_Drag\_Ratio)**
23. **# Entire row details of maximum Lift to drag ratio**
24. **Row\_Values = AirfoilFile[AirfoilFile['Lift to drag ratio'] == Maximum\_Value\_of\_Lift\_to\_Drag\_Ratio]**
25. **print(Row\_Values)**
26. **# Take Alfa\_A and CL directly from Row\_Values**
27. **Alfa\_A = Row\_Values['Degree'].iloc[0] # Angle of Attack**
28. **CL = Row\_Values['Cl'].iloc[0] # Lift Coefficient**
29. **print(Alfa\_A)**
30. **print(CL)**
31. **# c) find twist and chord**
32. **# Read NRELoffshBsline5MW\_AeroDyn\_Equil\_noTwr excel file**
33. **NRELoffshrbsline5MW = pd.read\_excel('C:/Users/s/Desktop/VS\_Codes/Aerodynamic/NRELOffshrBsline5MW\_AeroDyn\_Equil\_noTwr.xlsx')**
34. **print (NRELoffshrbsline5MW)**
35. **# Extract RNodes values from Excel files**
36. **r\_values = NRELoffshrbsline5MW['RNodes']**
37. **print (r\_values)**
38. **# Define functions of Distrubution twist angle**
39. **def Distrubution\_Twist\_Angle(R,r, Lamda\_D, Alfa\_A):**
40. **return math.degrees(math.atan((2/3) \* (R / (r \* Lamda\_D)))) - Alfa\_A**
41. **# Define Loops for repait r values**
42. **result\_Distrubution\_Twist\_angle = []**
43. **for r in r\_values:**
44. **result\_D\_T\_A = Distrubution\_Twist\_Angle(R, r, Lamda\_D, Alfa\_A)**
45. **print(f"For r = {r}: Result = {result\_Distrubution\_Twist\_angle}")**
46. **result\_Distrubution\_Twist\_angle.append(result\_D\_T\_A)**
47. **print(result\_Distrubution\_Twist\_angle)**
48. **# Define function Chord Length**
49. **def Chord\_Length(R,r, Lamda\_D, Z, CL):**
50. **return (1/Z) \* (8/9) \* ((2 \* math.pi \* R)/CL) \* (1/ (Lamda\_D \* ((Lamda\_D \* (r/R))\*\*2 + (4/9) )\*\* (1/2)))**
51. **result\_Chord\_Length = []**
52. **for r in r\_values:**
53. **r\_C\_Length = Chord\_Length(R, r, Lamda\_D, Z, CL)**
54. **print(f"For r = {r}: Result = {result\_Chord\_Length}")**
55. **result\_Chord\_Length.append(r\_C\_Length)**
56. **print(result\_Chord\_Length)**
57. **# Read Exercise\_01 excel file**
58. **Exercise\_01 = pd.read\_excel('C:/Users/s/Desktop/VS\_Codes/Aerodynamic/Exercise01.xlsx')**
59. **print (Exercise\_01)**
60. **# Chord length values are being exported for Exercise\_01 excel file**
61. **for j in range(len(result\_Chord\_Length)):**
62. **Exercise\_01.iloc[0 + j, 1] = result\_Chord\_Length[j]**
63. **# Distrubution Twist angle values are being exported for Exercise\_01 excel file**
64. **for k in range(len(result\_Distrubution\_Twist\_angle)):**
65. **Exercise\_01.iloc[0 + k, 3] = result\_Distrubution\_Twist\_angle[k]**
66. **print (Exercise\_01)**
67. **# Updated Values back to the original Excel file (Exercise\_01)**
68. **Exercise\_01.to\_excel('C:/Users/s/Desktop/VS\_Codes/Aerodynamic/Exercise01.xlsx')**
69. **# Plotting Comparison of chord Length**
70. **plt.figure()**
71. **graph = sns.lineplot(x= 'r',y= 'Chord Betz', data = Exercise\_01, color = 'Orange', marker = 'o', label = 'Chord Betz')**
72. **graph = sns.lineplot(x= 'r',y= 'Chord NREL', data = Exercise\_01, color = 'Yellow', marker = 'o', label = 'Chord NREL')**
73. **plt.xticks([2.87,5.60,8.33,11.75,15.85,19.95,24.05,28.15,32.25,36.35,40.45,44.55,48.65,52.75,56.17,58.90,61.63])**
74. **plt.title('Comparison Chord Length')**
75. **plt.xlabel('r')**
76. **plt.ylabel('Chord Length')**
77. **plt.legend()**
78. **# Plotting Comparison of Twist angle**
79. **plt.figure()**
80. **graph\_2 = sns.lineplot(x= 'r',y= 'Twist Betz', data = Exercise\_01, color = 'Orange', marker = 'o', label = 'Twist Betz')**
81. **graph\_2 = sns.lineplot(x= 'r',y= 'Twist NREL', data = Exercise\_01, color = 'Yellow', marker = 'o', label = 'Twist NREL')**
82. **plt.xticks([2.87,5.60,8.33,11.75,15.85,19.95,24.05,28.15,32.25,36.35,40.45,44.55,48.65,52.75,56.17,58.90,61.63])**
83. **plt.title('Comparison Twist Angle')**
84. **plt.xlabel('r')**
85. **plt.ylabel('Twist Angle')**
86. **plt.legend()**
87. **# Show the plot**
88. **plt.show()**