Geo106E
Fundamentals of Programming
Lab-10

2018-2019
Spring Semester

LabWork 13.1 Line Intersection (File name « Lab13.1.py »)

Coordinates of point for two line segments are given below.

line1 = ((0,0),(2,2))
line2 = ((0.5, 2),(2.5, 1))

Calculate the coordinates of intersection point using line_intersection module.

Module

```
import numpy as np

def line_from_points(p1, p2):
    if p1 == p2:
        raise ValueError("Points are Not Distinct...")
    else:
        x1, y1 = p1[0], p1[1]
        x2, y2 = p2[0], p2[1]
    # line for two points
    # ((y - y1) / (x - x1)) = ((y2-y1)/(x2-x1))
    # check if line is vertical x2 == x1
    if x2 == x1:
        # ax + by + c = 0
        # ax + by = -c
        # line is vertical -> x = -c
        a, b, c = 1, 0, -x1
    else:
        m = ((y2-y1)/(x2-x1))
        a, b, c = (-m), 1, (m*x1-y1)
    return a, b, c

def line_intersection(line1, line2):
    a1, b1, c1 = line_from_points(line1[0], line1[1])
    a2, b2, c2 = line_from_points(line2[0], line2[1])
    a = np.array([[a1, b1], [a2, b2]])
    b = np.array([-c1, -c2])
    x, y = np.linalg.solve(a, b)
    return x, y
```

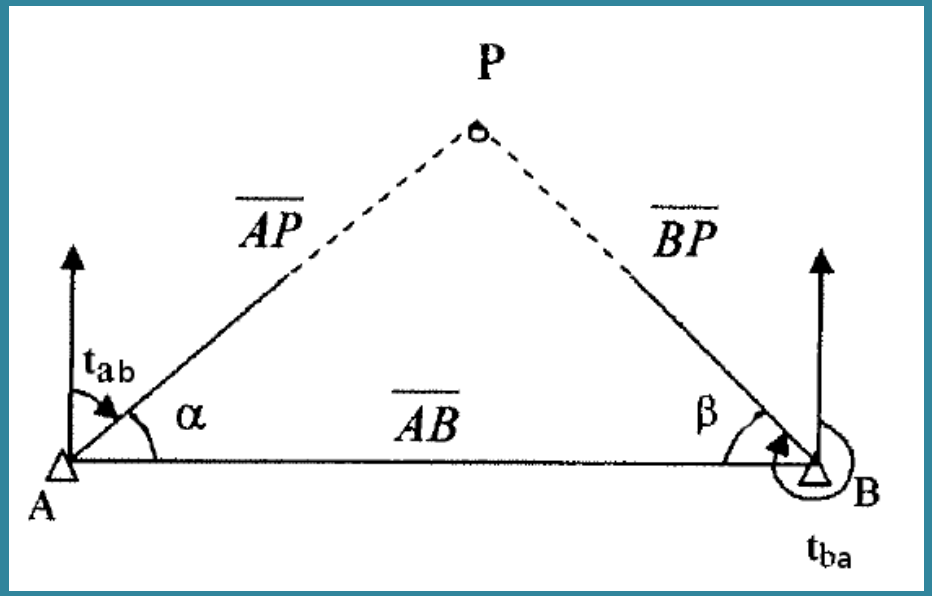
Solution

```
line1 = ((0.5, 0.5), (1.5, 0.5))
line2 = ((1, 0), (1, 2))

intersection = line_intersection(line1, line2)
print(intersection)
```

Expected output

(1.0, 0.5)

LabWork 13.2 Intersection
(File name « Lab13.2.py »)



Given Parameters:

Internal Angles: α, β

$\alpha = 38.4325^g$	$\beta = 73.4894^g$

Coordinates of Known Points:
Y_A, X_A and Y_B, X_B

N.N	Y	X
A	11 314.51m	26 242.67m
B	12 450.24	25 318.11

```
import intersection

# First point
Xa = 26242.67
Ya = 11314.51
# Second Point
Xb = 25318.11
Yb = 12450.24
# Mesaurements
alpha = 38.4325
beta  = 73.4894

Yp, Xp = intersection.intersection(Xa, Ya, Xb, Yb, alpha, beta, 'left')
print("Yp:", format(Yp,".2f"), "m")
print("Xp:", format(Xp,".2f"), "m")
```

Requested Parameters:

Coordinates of Unknown Point P:
Y_P , X_P

Expected Output

Y_P: 12673.35 m
X_P: 26134.33 m

LabWork 13.3 Resection
(File name « Lab13.3.py »)

Use the «resection.py» code to solve the coordinates of unknown point. Here is an exaple input and output:

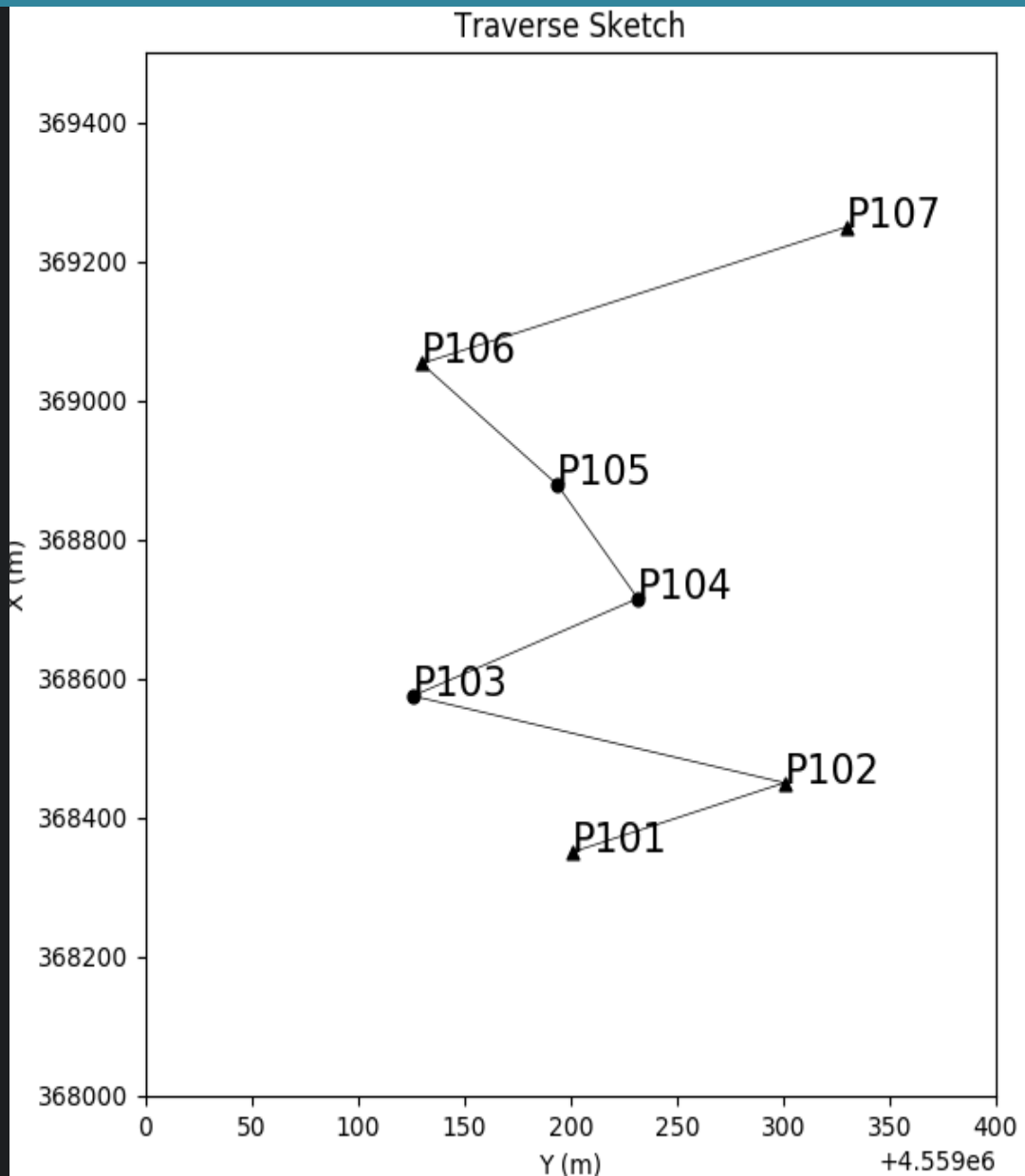
INPUT:
x1:1000 m
y1:1000 m
x2:1008 m
y2:1006 m
dist1:6 m
dist2:8 m
beta:100 grad

OUTPUT:
Azimuth from P1 to P2: 40.9666 grad
Horizontal distance from P1 to P2: 10.00 meter
alfa: 40.9666 grad
theta: 59.0334 grad
Check the Angles
Azimuth from P1 to A: 100.0000 grad
Coordinates of A: X: 1000.00 meter Y: 1006.00 meter
Azimuth from P2 to P1: 240.9666 grad
Azimuth from P2 to A: 200.0000 grad
Control of Coordinates of Point A
Coordinates of A: X: 1000.00 Y: 1006.00 meter

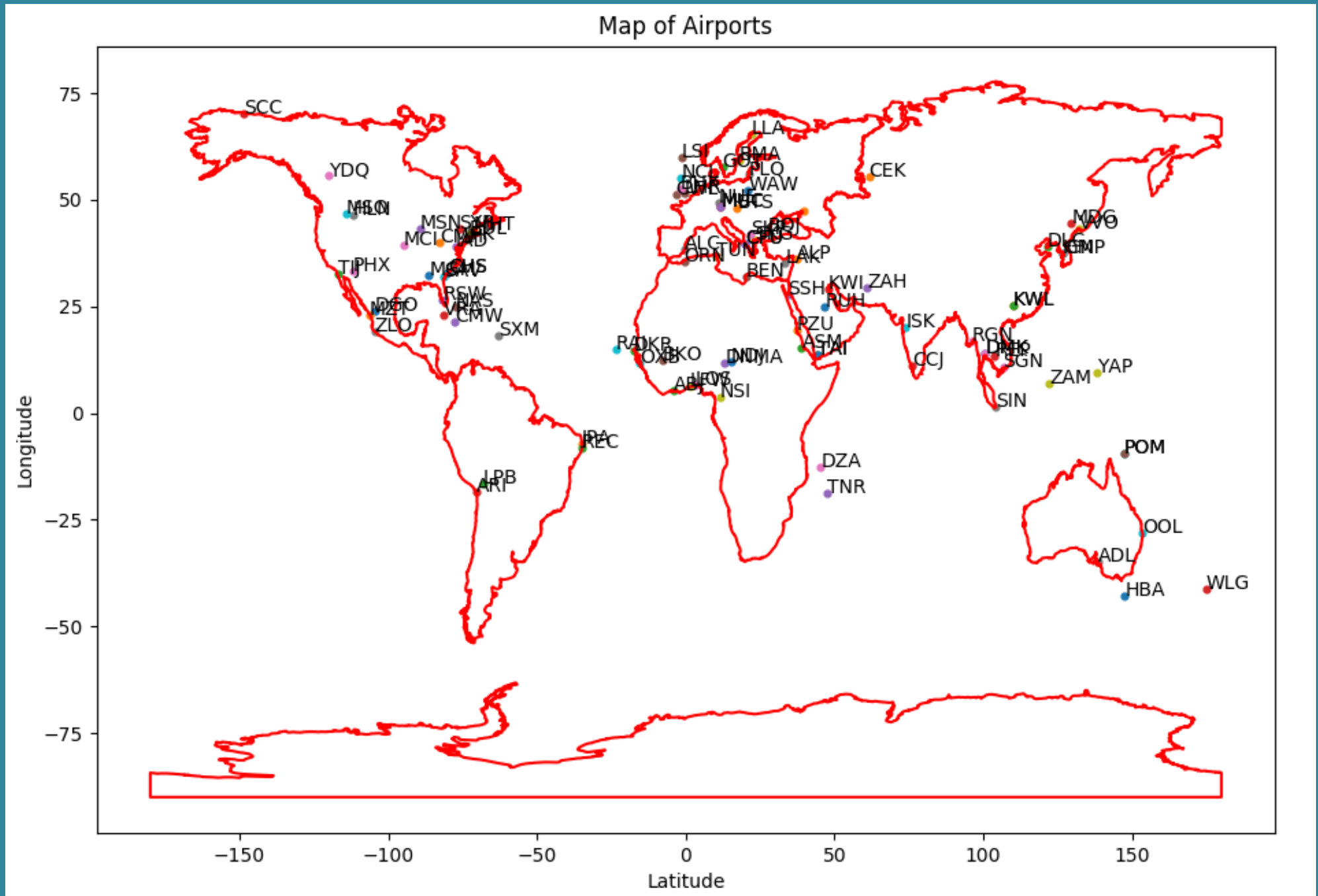
LabWork 13.4 Plotting (File name « Lab13.4.py »)

Create a simple traverse sketch for coordinates given in «traverse.txt» P101, P102, P106 and P107 are known points in closed-linked traverse

```
import matplotlib.pyplot as plt
import numpy as np

# Import coordinates of Continent borders
pid, y, x = np.loadtxt("traverse.txt",
                       delimiter=" ",
                       skiprows=1,
                       unpack=True,
                       dtype= [('PID','U10'),('Y',float),('X',float)])

# Create a figure
plt.figure("Traverse Sketch")
plt.title("Traverse Sketch") # figure title
plt.xlabel("Y (m)") # Label of x axis
plt.ylabel("X (m)") # Label of y axis

axes = plt.gca()
axes.set_xlim([4559000,4559400])
axes.set_ylim([368000,369500])

plt.scatter(x[0],y[0], marker="^", c="black", s = 24)
plt.annotate(pid[0], (x[0],y[0]), fontsize=16)
plt.scatter(x[1],y[1], marker="^", c="black", s = 24)
plt.annotate(pid[1], (x[1],y[1]), fontsize=16)
plt.scatter(x[2],y[2], marker="o", c="black", s = 24)
plt.annotate(pid[2], (x[2],y[2]), fontsize=16)
plt.scatter(x[3],y[3], marker="o", c="black", s = 24)
plt.annotate(pid[3], (x[3],y[3]), fontsize=16)
plt.scatter(x[4],y[4], marker="o", c="black", s = 24)
plt.annotate(pid[4], (x[4],y[4]), fontsize=16)
plt.scatter(x[5],y[5], marker="^", c="black", s = 24)
plt.annotate(pid[5], (x[5],y[5]), fontsize=16)
plt.scatter(x[6],y[6], marker="^", c="black", s = 24)
plt.annotate(pid[6], (x[6],y[6]), fontsize=16)

plt.plot(x,y, c="black", linewidth=.5)
```



LabWork 13.5 Plotting
(File name « Lab13.5.py »)



```
import random
import matplotlib.pyplot as plt
import numpy as np
# Import coordinates of Continent borders
x_europe, y_europe = np.loadtxt("europe.txt", delimiter=",", unpack=True)
x_asia, y_asia = np.loadtxt("asia.txt", delimiter=",", unpack=True)
x_africa, y_africa = np.loadtxt("africa.txt", delimiter=",", unpack=True)
x_americaN, y_americaN = np.loadtxt("north_america.txt", delimiter=",", unpack=True)
x_americaS, y_americaS = np.loadtxt("south_america.txt", delimiter=",", unpack=True)
x_americaS, y_americaS = np.loadtxt("south_america.txt", delimiter=",", unpack=True)
x_austria, y_austria = np.loadtxt("austria.txt", delimiter=",", unpack=True)
x_antarctica, y_antarctica = np.loadtxt("antarctica.txt", delimiter=",", unpack=True)
# Import coordinates of Airports
x,y,name,code = np.loadtxt("airports.txt",
    delimiter=",",
    unpack=True,
    dtype=[('Lon', float), ('Lat', float), ('name', 'U10'), ('code', 'U10')])
# Create a figure
plt.figure("Map of Airports")
plt.title("Map of Airports") # figure title
plt.xlabel("Latitude") # Label of x axis
plt.ylabel("Longitude") # Label of y axis
# Plot continents
plt.plot(x_europe, y_europe, linestyle='-', color="red")
plt.plot(x_asia, y_asia, linestyle='-', color="red")
plt.plot(x_africa, y_africa, linestyle='-', color="red")
plt.plot(x_americaN, y_americaN, linestyle='-', color="red")
plt.plot(x_americaS, y_americaS, linestyle='-', color="red")
plt.plot(x_austria, y_austria, linestyle='-', color="red")
plt.plot(x_antarctica, y_antarctica, linestyle='-', color="red")
# Plot airports
numbers = [random.randint(0,len(name)) for i in range(100)] # 100 random numbers
for i in numbers: # limited to 100 for speed
    plt.scatter(x[i],y[i], s=12)
    plt.annotate(code[i], (x[i],y[i]), fontsize=10) # label of airport
```