lab1

March 24, 2023

[ ]:

```python
import sympy
import sys
import numpy as np
import scipy
import math
```

[ ]:

```python
x_p = -10
x_q = -2
y_q = 1
print(x_p, x_q, y_q)

epsilon = 10**(-5)

p = np.array([x_p, 0])
P = (float)(np.linalg.norm(p))
q = np.array([x_q, y_q])
Q = (float)(np.linalg.norm(q))
print(p, q)
```

```
-10 -2 1
[-10   0] [-2  1]
```

[ ]:

```python
def Angle(theta):
    t = np.array([math.cos(theta), math.sin(theta)])
    tp = np.array([p[0] - t[0], p[1] - t[1]])
    tq = np.array([q[0] - t[0], q[1] - t[1]])
    ptv = math.acos(np.dot(tp, t) / (np.linalg.norm(tp) * np.linalg.norm(t)))
    qtv = math.acos(np.dot(tq, t) / (np.linalg.norm(tq) * np.linalg.norm(t)))
    # print(p, q, t, tp, tq)
    # print("angle:", ptv, qtv)
    return ptv, qtv
```

[ ]:

```python
theta1 = math.pi
theta2 = math.pi/2
theta3 = 0
# print(math.sin(theta1))
counter = 0
while True:
    theta3 = (theta1 + theta2)/2
    angle1, angle2 = Angle(theta3)
    print(theta1, theta2, theta3, angle1, angle2)
    if abs(angle1 - angle2) < epsilon:
        break
    # if counter > 10:
        # break
    if angle1 < angle2:
        theta1 = theta3
    else:
        theta2 = theta3
    print(theta1, theta2)
    # counter += 1

t = np.array([math.cos(theta3), math.sin(theta3)])
print(t)
```

```
3.141592653589793 1.5707963267948966 2.356194490192345 0.8613429528839481
0.562617536425785
3.141592653589793 2.356194490192345
3.141592653589793 2.356194490192345 2.748893571891069 0.4348378897119155
0.12812008839707373
3.141592653589793 2.748893571891069
3.141592653589793 2.748893571891069 2.945243112740431 0.217976690259403
0.4721009241980664
2.945243112740431 2.748893571891069
2.945243112740431 2.748893571891069 2.84706834231575 0.3266135646746481
0.30293730499939686
2.945243112740431 2.84706834231575
2.945243112740431 2.84706834231575 2.8961557275280905 0.2723386317956034
0.388390318625945
2.8961557275280905 2.84706834231575
2.8961557275280905 2.84706834231575 2.8716120349219203 0.29948800441344375
0.34586331369138307
2.8716120349219203 2.84706834231575
2.8716120349219203 2.84706834231575 2.859340188618835 0.313053888051292
0.32444768738848306
2.859340188618835 2.84706834231575
2.859340188618835 2.84706834231575 2.8532042654672924 0.3198345179906664
0.3137040158760982
2.859340188618835 2.8532042654672924
2.859340188618835 2.8532042654672924 2.856272227043064 0.3164443989645073
0.31907877270952656
2.856272227043064 2.8532042654672924
2.856272227043064 2.8532042654672924 2.8547382462551782 0.3181395077092385
0.3163921194517671
2.856272227043064 2.8547382462551782
```

```
2.856272227043064 2.8547382462551782 2.855505236649121 0.31729196561408574
0.31773562801133026
2.855505236649121 2.8547382462551782
2.855505236649121 2.8547382462551782 2.85512174145215 0.3177157397348032
0.3170639191342355
2.855505236649121 2.85512174145215
2.855505236649121 2.85512174145215 2.8553134890506353 0.3175038534422498
0.3173997849334601
2.855505236649121 2.8553134890506353
2.855505236649121 2.8553134890506353 2.855409362849878 0.31739790972005993
0.3175677093138142
2.855409362849878 2.8553134890506353
2.855409362849878 2.8553134890506353 2.8553614259502567 0.31745088162913626
0.3174837478338364
2.8553614259502567 2.8553134890506353
2.8553614259502567 2.8553134890506353 2.855337457500446 0.317477367547689
0.3174417665611788
2.8553614259502567 2.855337457500446
2.8553614259502567 2.855337457500446 2.8553494417253513 0.3174641245914111
0.3174627572418927
[-0.95931137  0.2823503 ]
```

[ ]:

```
r = np.array([q[0] - 2 * t[0]* (np.dot(t, q)  - 1),q[1] - 2 * t[1]* (np.dot(t,␣
 ↪q)  - 1) ])
r
```

[ ]:

```
array([0.3042142 , 0.32180981])
```

[ ]:

```

```