Math 291H Computational Problem #1

Rajeev Atla

Problem 1

```python
import math

p = (1, 1, 0)
q = (0, 0, 2)
r = (1, 1, 1)


def dot_product(v1, v2):
    return v1[0] * v2[0] + v1[1] * v2[1] + v1[2] * v2[2]


def cross_product(v1, v2):
    x = v1[1] * v2[2] - v1[2] * v2[1]
    y = v1[2] * v2[0] - v1[0] * v2[2]
    z = v1[0] * v2[1] - v1[1] * v2[0]
    return (x, y, z)
```

Problem 2

```python


# question 2


def points_to_vec(p1, p2):
    v = tuple(map(lambda x, y: x - y, p1, p2))
    return v


pq = points_to_vec(p, q)
print(pq)

qr = points_to_vec(q, r)
print(qr)


def angle_dot(v1, v2):
    numerator = dot_product(v1, v2)
    denominator = norm(v1) * norm(v2)
    return math.acos(numerator / denominator)
```

From the dot product, we find $\theta \approx 2.802$. From the cross product, we find $\theta \approx 0.340$. Note that these angles add up to π.

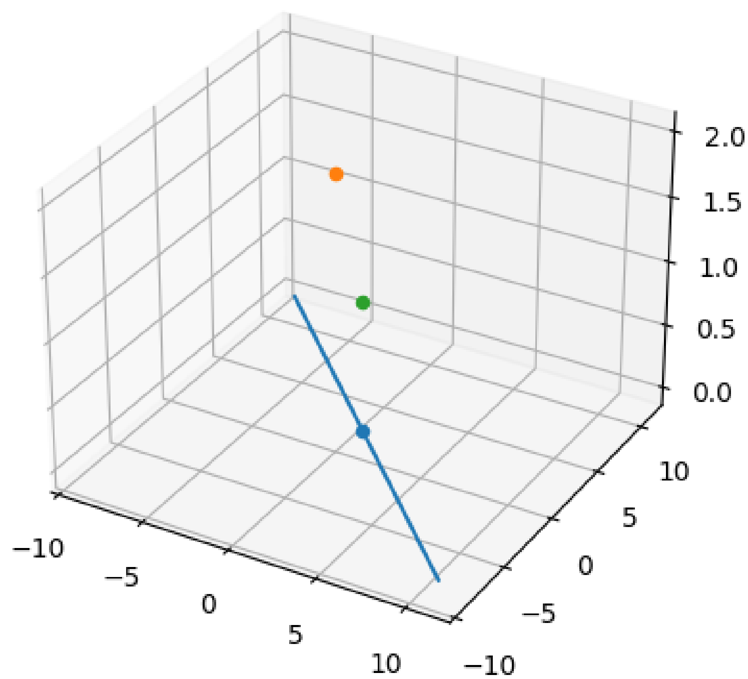
Problem 3

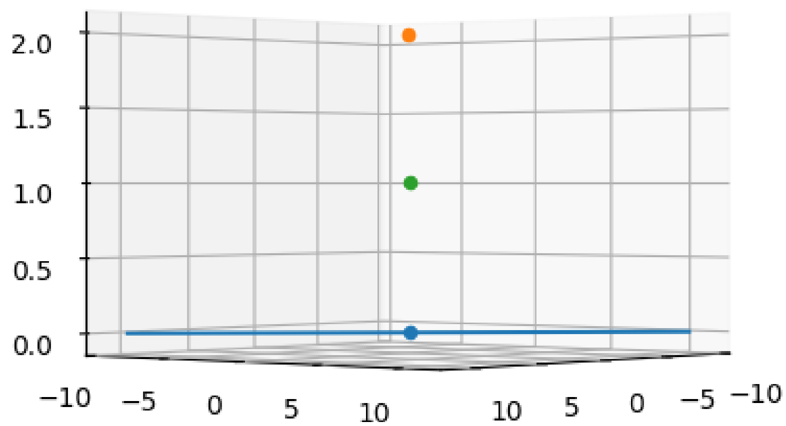
```
def angle_cross(v1, v2):
    numerator = norm(cross_product(v1, v2))
```

$\boldsymbol{v} = (-1, 1, 0)$ is a vector that is orthogonal to both $\boldsymbol{pq}$ and $\boldsymbol{qr}$.

Problem 4

```
    return math.asin(numerator / denominator)


print(angle_dot(pq, qr))
print(angle_cross(pq, qr))

v = cross_product(pq, qr)
print(v)

from mpl_toolkits import mplot3d
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
ax = plt.axes(projection="3d")


ax.scatter3D(p[0], p[1], p[2])
ax.scatter3D(q[0], q[1], q[2])
ax.scatter3D(r[0], r[1], r[2])


def get_line(t):
    return p[0] + t * v[0], p[1] + t * v[1], p[2] + t * v[2]
```


2.0
1.5
1.0
0.5
0.0
10
5
0
−5
−10
−10
−5
0
5
10


2.0
1.5
1.0
0.5
0.0
−10 −5 0 5 10
10 5 0 −5 −10

Problem 5

We use the fact that the equation of a plane can be written

$$\boldsymbol{v} \cdot \boldsymbol{x} + d = 0$$

Solving for d,

$$d = -\boldsymbol{v} \cdot x$$

```

t = np.linspace(-10, 10, 100)
```

We find that $d = 0$ Therefore,

$$-x + y + 0 \cdot z = 0$$

or

$$x = y$$