Sample Document using allan-eason.sty

Allan Pan and Eason Shao

March 22, 2022

Contents

Section 1 maths

https://mathxstudio.github.io/
https://mathxstudio.github.io/
https://mathxstudio.github.io/
https://mathxstudio.github.io/

§1.1 test subsection

Let $n \geq 3$ be a positive integer. Let $C_1, C_2, \ldots, C_n$ be unit circles in the plane, with centres $O_1, O_2, \ldots, O_n$ respectively. If no line meets more than two of the circles, prove that:

$$\sum_{1 \leq i < j \leq n} \frac{1}{O_i O_j} \leq \frac{(n-1)\pi}{4}.$$

For brevity, let d_{ij} be the length of O_{ij} and let $\angle(ijk)$ be shorthand for $\angle O_i O_j O_k$ (or its measure in radians).
First, we eliminate the circles completely and reduce the problem to angles using the following **Lemma**:

Lemma 1.1

For any indicies i, j, m we have the inequalities

$$\angle(imj) \geq \max\left(\frac{2}{d_{mi}}, \frac{2}{d_{mj}}\right) \quad \text{and} \quad \pi - \angle(imj) \geq \max\left(\frac{2}{d_{mi}}, \frac{2}{d_{mj}}\right)$$

Proof of Lemma 1.1

We first prove the former line. Consider the altitude from O_i to $O_m O_j$. The altitude must have length at least 2, otherwise its perpendicular bisector passes intersects all of C_i, C_m, C_j. Thus

$$2 \leq d_{mi} \sin \angle(imj) \leq \angle(imj)$$

proving the first line. The seconf line follows by considering the external angle formed by lines O_mO_i and O_mO_j instead of the internal one.$\square$

1.1

Lemma 1.2

another test lemma.

Proof of Lemma 1.2

proof of lemma.

1.2

Our idea now is for any index m we will make an estimate on $\sum_{\substack{1 \le i \le n \\ i \neq b}} \frac{1}{d_{bi}}$ for each index b. If the centers formed a convex polygon, this would be much simpler, but because we do not have this assumption some more care is needed.

Claim 1.1

Suppose O_a, O_b, O_c are consecutive verticies of the convex hull. Then

$$\frac{n-1}{n-2}\measuredangle(abc) \geq \frac{2}{d_{1b}} + \frac{2}{d_{2b}} + \ldots + \frac{2}{d_{nb}}$$

where the term $\frac{2}{d_{bb}}$ does not appear (obviously).

Proof of Claim 1.1

WLOG let's suppose $(a, b, c) = (2, 1, n)$ and that ...

another line of text...

Fact 1.1

Describe your fact.

Proof of Fact 1.1

Describe proof.

another line of text...

Theorem 1.1 (Test theorem)**.** *Here is a theorem. Here is a theorem. Here is a theorem. Here is a theorem. Here is a theorem. Here is a theorem.*

...

Now suppose there were r verticies in the convex hull. If we sum the first claim across all b on the hull, and the second across all b not on the hull (inside it), we get

$$\begin{aligned}\sum_{1\le i<j\le n} \frac{2}{d_{ij}} &= \frac{1}{2}\sum_{b}\sum_{i\ne b}\frac{2}{d_{bi}} \\ &\le \frac{1}{2}\cdot\frac{n-1}{n-2}((r-2)\pi+(n-2)\pi) \\ &= \frac{(n-1)\pi}{4}\end{aligned}$$

as needed (with $(r-2)\pi$ being the sum of all angles in the hull.

Remark. This is the sixth and last problem of IMO 2002, and is a difficult one. Allan put it here to test the latest style file.

Section 2 code

Hypothesis – test hypothesis.

Justification – type some justifications.

Algorithm

allanpy

```
# observation from the air

%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

v_car=5.611 # 20km/h on average in hk
v_eye=16 # Hz
alpha_lag=1.00
v_reload=alpha_lag*v_eye

pie=math.pi
r_a=13000
rho_car=0.001023
delta_d_car=v_car/v_reload
L_car=4.71769
C_eye=40960000
alpha_c1=1.00
C1=C_eye*alpha_c1
alpha_c2=0.90
C2=C_eye*alpha_c2
alpha_c3=0.80
C3=C_eye*alpha_c3

def alpha_clarity(x):
    if x>0 and x<sep_point:
        return float(0)
    elif x>=sep_point and x<=1:
        return $\displaystyle\left[ -\left( \frac{4096}{4095} \right) ^2 \
    elif x>1:
        return float(1)
output=[0 for i in range(len(dataport))]
for i in range(len(dataport)):
```

```
    output[i]=alpha_clarity(dataport[i])
dataport=np.arange(0,1.01,0.01)
sep_point=1/4096
```

Explanation

$$\text{data}_i = 4\pi\sqrt{r_a^2 - \text{datax}_i^2} \cdot \rho_{\text{car}} \cdot \Delta d_{\text{car}}$$

another line of text.

```
# observation from the air

%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

v_car=5.611 # 20km/h on average in hk
v_eye=16 # Hz
alpha_lag=1.00
v_reload=alpha_lag*v_eye

pie=math.pi
r_a=13000
rho_car=0.001023
delta_d_car=v_car/v_reload
L_car=4.71769
C_eye=40960000
alpha_c1=1.00
C1=C_eye*alpha_c1
alpha_c2=0.90
C2=C_eye*alpha_c2
alpha_c3=0.80
C3=C_eye*alpha_c3

```

```python
def alpha_clarity(x):
    if x>0 and x<sep_point:
        return float(0)
    elif x>=sep_point and x<=1:
        return float((-(4096/4095)**2)*((x-1)**2)+1)
    elif x>1:
        return float(1)
output=[0 for i in range(len(dataport))]
for i in range(len(dataport)):
    output[i]=alpha_clarity(dataport[i])
dataport=np.arange(0,1.01,0.01)
sep_point=1/4096
```

INSERTION-SORT(A)

for $j = 2$ **to** $A.length$
  $key = A[j]$
  // Insert $A[j]$ into the sorted sequence $A[1\,..\,j-1]$.
  $i = j - 1$
  while $i > 0$ and $A[i] > key$
    $A[i+1] = A[i]$
    $i = i - 1$
  $A[i+1] = key$

SEGMENTS-INTERSECT(p_1, p_2, p_3, p_4)

$d_1 = \text{DIRECTION}(p_3, p_4, p_1)$
$d_2 = \text{DIRECTION}(p_3, p_4, p_2)$
$d_3 = \text{DIRECTION}(p_1, p_2, p_3)$
$d_4 = \text{DIRECTION}(p_1, p_2, p_4)$
if $((d_1 > 0 \text{ and } d_2 < 0) \text{ or } (d_1 < 0 \text{ and } d_2 > 0))$ and
    $((d_3 > 0 \text{ and } d_4 < 0) \text{ or } (d_3 < 0 \text{ and } d_4 > 0))$
    return TRUE
elseif $d_1 == 0$ and ON-SEGMENT(p_3, p_4, p_1)
    return TRUE
elseif $d_2 == 0$ and ON-SEGMENT(p_3, p_4, p_2)
    return TRUE
elseif $d_3 == 0$ and ON-SEGMENT(p_1, p_2, p_3)
    return TRUE
elseif $d_4 == 0$ and ON-SEGMENT(p_1, p_2, p_4)
    return TRUE
else return FALSE

Section 3 colors

allanblue allanred allangreen allanpurple allancyan allanorange allanyellow allandarkblue

Section 4 cites

I love bibliography. [1]

References

[1] bibliography is important.