ECE457B - Assignment 3

Sheen Thusoo | 20728766

Question 1

The following libraries were used for this question: matplotlib, numpy, scipy, and skfuzzy.
The membership function, μ_A, was sketched and the fuzziness measures given by M_1, M_2, and M_3 were also represented with the shaded areas in these plots as shown below in Figures 1, 2, and 3. The value of these measures was also calculated using scipy's `integrate.quad` function and is shown in these figures.

a.

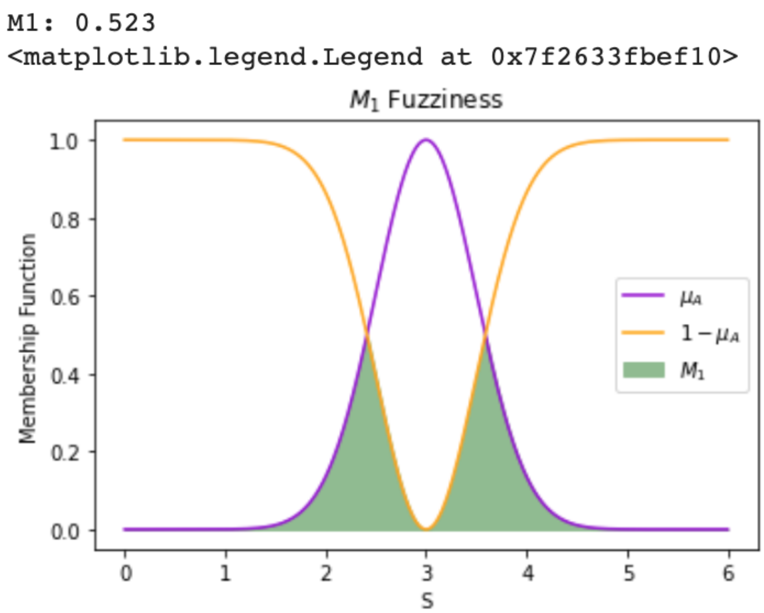


Figure 1: M_1 Fuzziness Measure Shaded and Integral Value (top)

b.

```
M2: 0.523
<matplotlib.legend.Legend at 0x7f2633632f90>
```

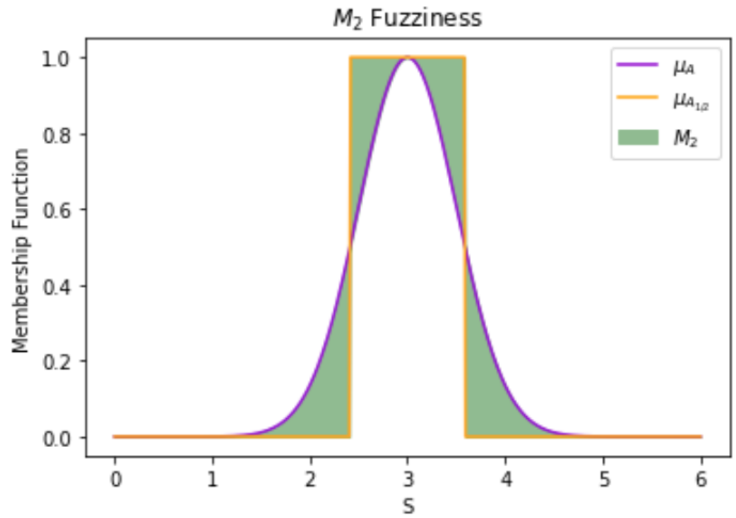


Figure 2: M_2 Fuzziness Measure Shaded and Integral Value (top)

c.

```
M3: 4.953
<matplotlib.legend.Legend at 0x7f26335cdbd0>
```

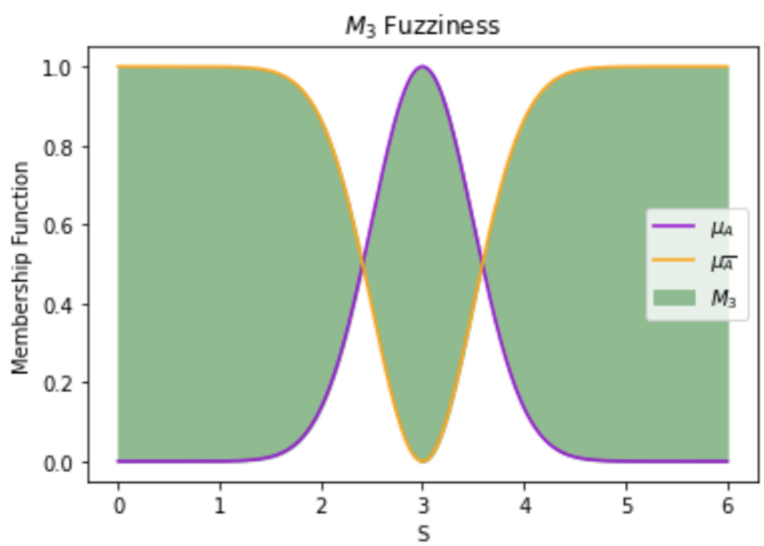


Figure 3: M_3 Fuzziness Measure Shaded and and Integral Value (top)

i.

From the figures and values shown above, a relationship between these three fuzziness measures can be established. First, from the integral values,

$$M_1 = 0.523, \quad M_2 = 0.523$$

Thus,

$$M_1 = M_2$$

From Figure 3, M_3 seems to be larger than the previous two measures as the shaded area is larger. The shaded area covers almost the entire support set excluding two areas in the middle regions which resemble the shaded area in Figure 1, from the M_1 measure. Thus,

$$M_3 = 6 - 2M_1$$

This is also confirmed by the value of M_3 which is 4.953, as

$$M_3 = 4.953 = 6 - 2(0.523)$$

Thus, the relationship between the three fuzziness measures can be described with the following equation:

$$M_1 = M_2 = \frac{1}{2}\left(S \times 1 - M_3\right)$$

ii.

These measures can be used to represent the degree of fuzziness of a membership function as all three of these functions are different measures of fuzziness.

M_1 measures the closeness of a membership value to a grade of 0.5. When a value is close to 0.5, the M_1 measure will be high. M_2 measures the distance from the membership function to the ½-cut (alpha-cut where alpha =½) where the alpha-cut is formed by the elements of the set whose membership function grades are greater than or equal to alpha. M_1 and M_2 yield the same values. Lastly, M_3 measures the inverse of the distance from the membership function to its complement. When the distance is lower, it implies that the membership function grade would be close to 0.5 which indicates fuzziness. Thus, we take the inverse of this, so that the M_3 function value is high when the grade is close to 0.5.

Question 2

To show that the expressions are true, truth tables can be shown as shown below.

Table 1: Truth Table for Complement Relation

χ_A	$\chi_{A'}$	$1 - \chi_A$
1	0	$1 - 1 = 0$
0	1	$1 - 0 = 1$

Table 2: Truth Table for Union Relation

χ_A	χ_B	$\chi_{A\cup B}$	$max(\chi_A, \chi_B)$
0	0	0	$max(0, 0) = 0$
0	1	1	$max(0, 1) = 1$
1	0	1	$max(1, 0) = 1$
1	1	1	$max(1, 1) = 1$

Table 3: Truth Table for Intersection Relation

χ_A	χ_B	$\chi_{A\cap B}$	$min(\chi_A, \chi_B)$
0	0	0	$min(0, 0) = 0$
0	1	0	$min(0, 1) = 0$
1	0	0	$min(1, 0) = 0$
1	1	1	$min(1, 1) = 1$

Table 4: Truth Table for Implication Relation

χ_A	χ_B	$\chi_{A\to B}(x,y)$	$min\left[1, \left\{1 - \chi_A(x) + \chi_B(y)\right\}\right]$
0	0	1	$min(1, 1 - 0 + 0) = 1$
0	1	1	$min(1, 1 - 0 + 1) = 1$
1	0	0	$min(1, 1 - 1 + 0) = 0$
1	1	1	$min(1, 1 - 1 + 1) = 1$

The truth tables in Tables 1, 2, 3 and 4 show that these fuzzy operators are equivalent to the crisp set equations. Thus, the characteristic function of a crisp set is analogous to the membership function of a fuzzy set. Since binary logic holds, these operators are valid for fuzzy sets.

Question 3

The following libraries were used for this question: matplotlib and numpy.

a.

The linguistic hedge "very" is appropriately represented using its membership function. The state of "very fast speed" indicates a faster speed than the state of "fast speed" as adding the word "very" indicates that the speed is fast at a higher degree. The membership function associated with it, $\mu_F(v - v_0)$, where v_0 is greater than 0, shifts the membership function of "fast speed" to the right by 50 rev/s. This allows the membership functions at faster speeds to be higher.

Furthermore, the linguistic hedge "presumably" is not appropriately represented using its membership function. The word "presumably" is used to convey that the fastness of the speed is likely not known for certain. Its associated membership function, $\mu_F^2(v)$, squares the membership values, thereby decreasing them (since they are decimals). This in turn narrows the membership function for "presumably fast speed" and reduces confidence for lower speeds (we want to reduce confidence for higher speeds instead). Since it is more unsure of the membership values, the function should expand the function using a fractional exponent like $\mu_F^{1/2}(v)$; this increases the fuzziness.

b.

The membership function of "very fast speed" was determined by shifting the membership function of "fast speed" to the right by 50 rev/s. Since the universe increases by 10, the values had to be shifted by 5 steps (to resultantly shift by 50 rev/s). The membership function of "presumably fast speed" was determined by squaring each membership value in the "fast speed" function. The plot of the 3 membership functions over the discrete universe V in shown in Figure 4 below.

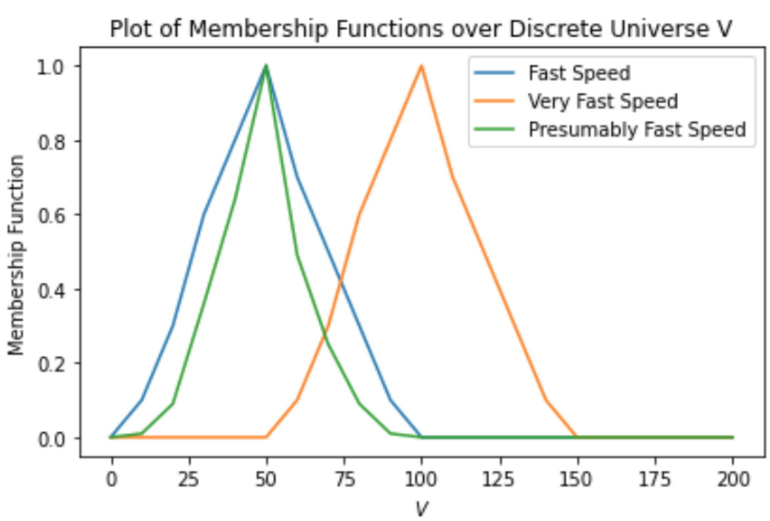


Figure 4: Plot of Membership Functions over Universe V

Question 4

We can show that $max[0,\ x\ +\ y\ -\ 1]$ is a t-norm by showing that it fulfills all the properties of t-norms.

T-norm Properties:

(1) It is non-decreasing in each argument

$$xTc\ \le yTd$$

$$\text{If}\quad x \le c,\quad y \le d\quad \text{then,}$$

$$x\ +\ y\ \le\ c +\ d\quad \text{so,}$$

$$max[0,\ x +\ y\ -\ 1]\ \le max[0,\ c\ +\ d\ -\ 1]$$

Therefore, this property is satisfied.

(2) It satisfies commutativity

$$xTy\ =\ yTx$$

$$max[0,\ x\ +\ y\ -\ 1]\ = max[0,\ y +\ x\ -\ 1]$$

Therefore, this property is satisfied.

(3) It satisfies associativity

$$(xTy)Tz\ =\ xT(yTz)$$

$$max\ [0,\ max[0,\ x\ +\ y -\ 1]\ +\ z\ -\ 1]\ =\ max[0,\ x\ +\ max[0,\ y\ +\ z\ -\ 1]\ -\ 1]$$

Therefore, this property is satisfied.

(4) It satisfies the boundary conditions

$$xT1\ =\ x$$

$$max[0,\ x\ +\ 1 -\ 1]\ =\ max[0,\ x]$$

$$max[0,\ a]\ =\ a,\quad since\quad 0 \le\ a\ \le 1$$

$$xT0\ =\ 0$$

$$max[0,\ x\ +\ 0 -\ 1]\ = max[0,\ x\ -\ 1]$$

$$max[0,\ x - 1] = 0,\quad since\ \ 0 \le\ x\ \le 1$$

Therefore, this property is satisfied. Since the four properties are satisfied, it is a t-norm.

The corresponding t-conorm (s-norm) is

$$xSy\ =\ 1\ -\ (1 - x)T(1 - y)$$

$$xSy\ = 1\ -\ max[0,\ (1 - x)\ +\ (1 - y)\ -\ 1]$$

$$xSy\ = 1\ -\ max[0,\ 1 - x\ - y]$$

Question 5

The following libraries were used for this question: matplotlib and numpy.

a.

First, I investigated the meaning of the lambda parameter by varying its value (keeping *a* and *n* constant) and plotting the membership function as shown in Figure 5 a, b, c, d.

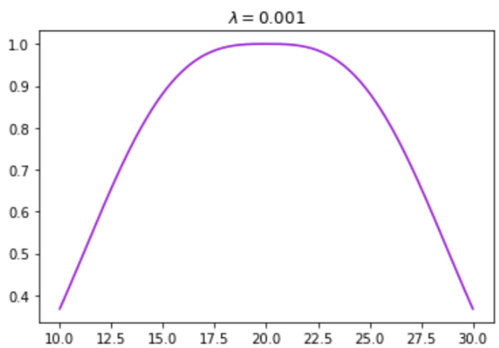


Figure 5a: $\lambda = 0.001$

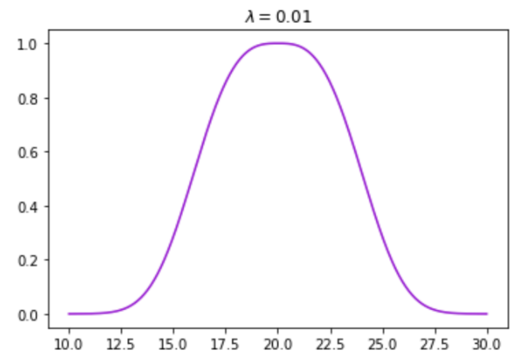


Figure 5b: $\lambda = 0.01$

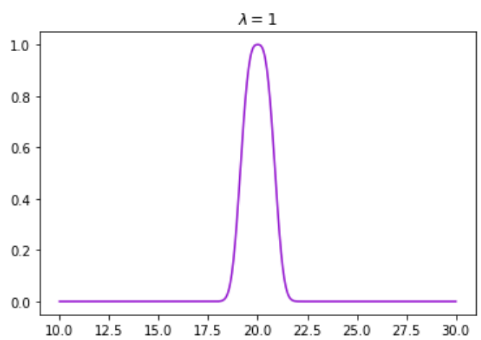


Figure 5c: $\lambda = 1$

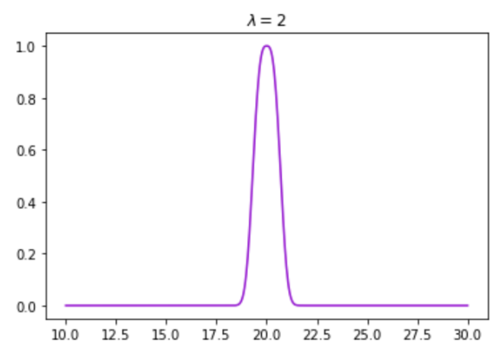


Figure 5d: $\lambda = 2$

From these figures, it is clear that the lambda parameter affects the width of the membership function. As lambda increases, the width of the membership decreases and as lambda decreases, the width of the membership function increases.

Next, I investigated the meaning of the *a* parameter by varying its value and plotting the membership function as shown in Figure 6 a, b, c, d.

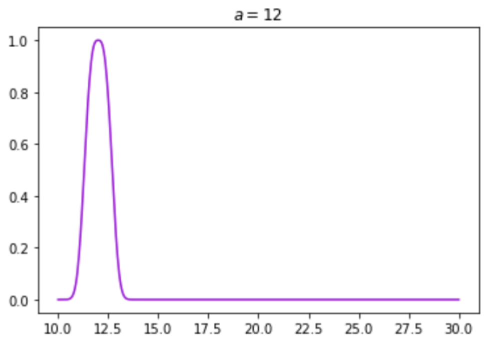


Figure 6a: $a = 12$

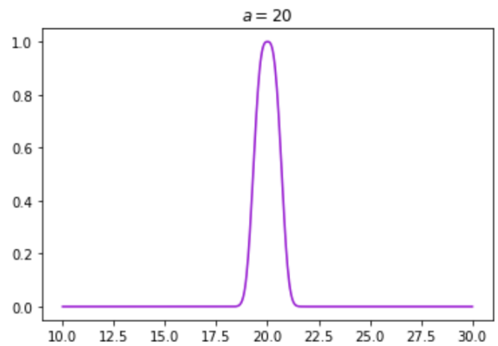


Figure 6b: $a = 20$

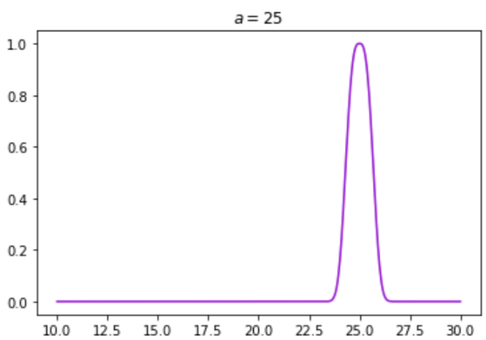


Figure 6c: $a = 25$

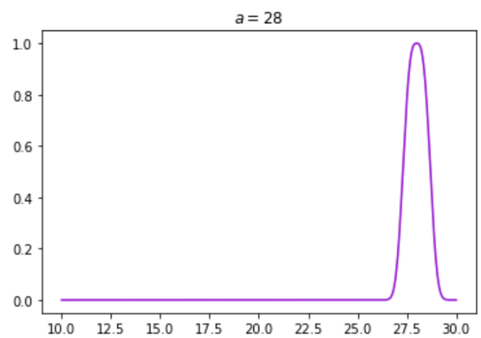


Figure 6d: $a = 28$

From these figures, it is clear that the *a* parameter changes the location of the midpoint (or peak) of the membership function. As *a* increases, the membership function shifts right so that its peak is at the value of *a* and as *a* decreases, the function shifts left. For example, Figure 6d shows that when $a = 28$, the peak of the membership function is located at 28.

Finally, I investigated the meaning of the *n* parameter by varying its value and plotting the membership function as shown in Figure 7 a, b, c, d.

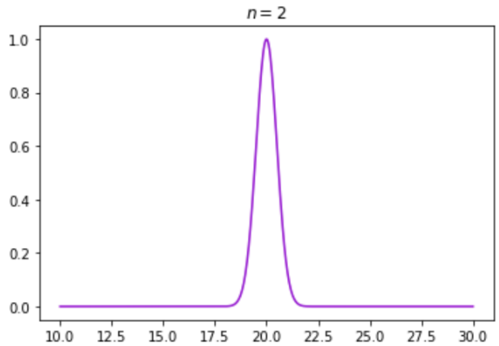


Figure 7a: $n = 2$

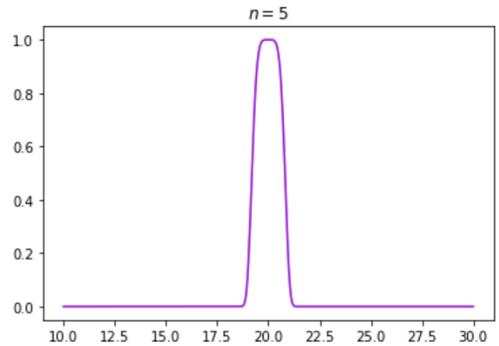


Figure 7b: $n = 5$

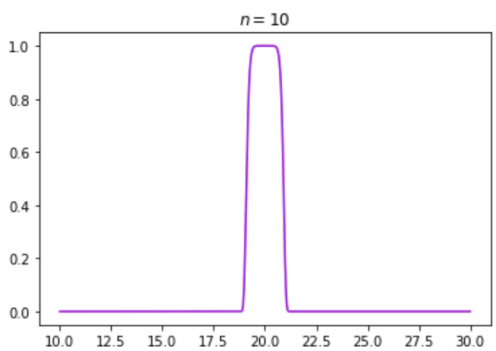


Figure 7c: $n = 10$

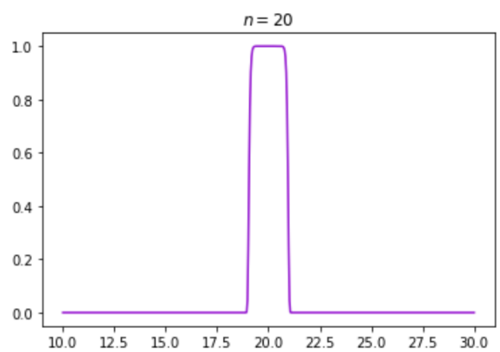


Figure 7d: $n = 20$

From these figures, it is clear that the *n* parameter changes the curvature of the peak of the membership function. As *n* increases, the peak of the membership function becomes more flat as seen in Figure 7d; however, as *n* decreases, the peak becomes more curved.

In particular, the fuzziness of a system may be represented using these parameters. The fuzzy modifier "very" could be used to describe something in a high degree such as a very fast speed or a very cold room. In this case, the parameter *a* could be used to shift the function right or left depending on the fuzzy descriptor. For example, in the case of "very fast speed", the *a* parameter could be increased to shift the membership function to the right. On the other hand, in the case of "very slow speed", the *a* parameter could be decreased to shift the membership function to the left. Similarly, the "somewhat" modifier could be used to describe something at a lower degree. For example, in the case of "somewhat fast speed", the membership function could be shifted to the left by decreasing the *a* parameter..

Furthermore, “very” could be used to emphasize confidence. In this case, the lambda parameter could be increased to decrease the width of the membership function which makes it more crisp. Also, the *n* parameter could be increased to increase crispness of the set as well (more flat function). The modifier “somewhat” could be used to describe the lack of confidence so in this case we could decrease the lambda and *n* parameters to increase the fuzziness of the set.

b.

The numerical values found for the parameters for the “cold”, “hot”, and “comfortable” states are shown below.

Cold State: $\lambda = 0.06$, $n = 2$, $a = 10$

Hot State: $\lambda = 0.06$, $n = 2$, $a = 30$

Comfortable State: $\lambda = 0.1$, $n = 4$, $a = 21$

The parameters set for the cold state were set based on normal room temperatures for indoor areas. For instance, the a parameter was set at 10 as this shifts the peak of the membership value to 10°C. This ensures that 10°C and temperatures close to this have a high membership value; this makes sense as 10-18°C is cold for indoor areas. The lambda value was set to 0.06 and the *n* parameter was set to 2 to ensure that the membership functions width was larger and its peak was more curved to cover more temperatures that are considered cold.

The parameters set for the hot state were set similarly. In fact, the only difference is the value of the *a* parameter which was set to 30 to shift the membership function to the right. This is because 30°C is a very hot temperature for indoor areas. However, some people find hotter temperatures more suitable. For instance, someone could consider 25°C to be warm rather than hot.

The parameters set for the comfortable state were set to reflect the values of normal room temperature which are around 21-22°C. Thus, the *a* parameter was set to 21 as 21°C is a common comfortable temperature. The lambda parameter was increased to 0.1 and the *n* parameter was set higher at 4 to make this set more crisp as there are only a small range of temperatures that are considered comfortable.

These membership functions were graphed as shown in Figure 8 below.

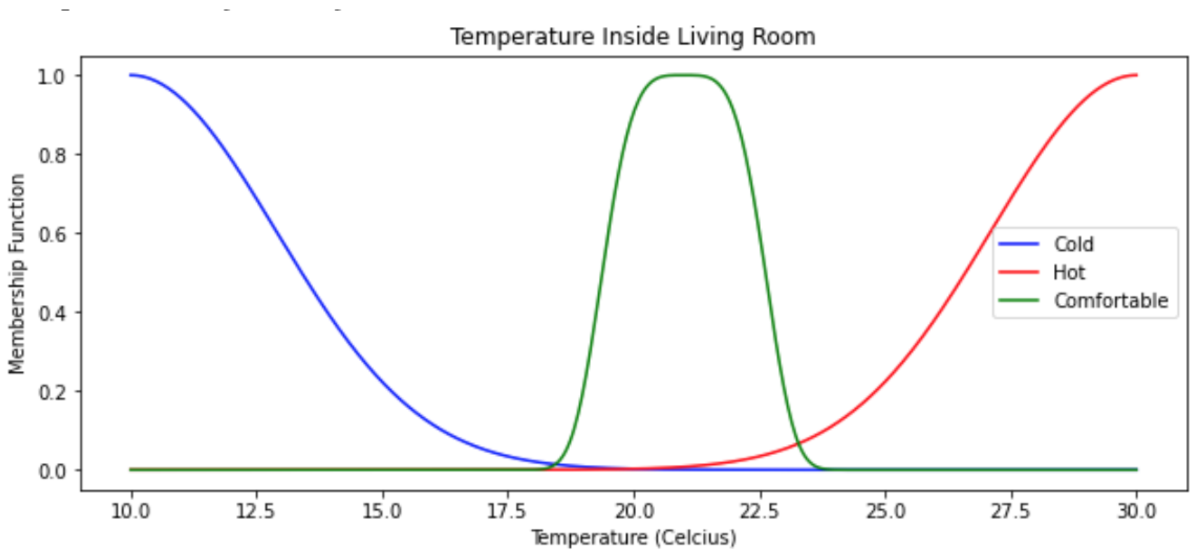


Figure 8: Plot of Cold, Hot and Comfortable Temperature Membership Functions

ECE457B_Assignment3_20728766

April 4, 2022

1 ECE 457B - Assignment 3

1.1 By: Sheen Thusoo, 20728766

1.1.1 Problem 1

[1]:
```
!pip install -U scikit-fuzzy
```

```
Collecting scikit-fuzzy
  Downloading scikit-fuzzy-0.4.2.tar.gz (993 kB)
     |                    | 993 kB 4.3 MB/s
Requirement already satisfied: numpy>=1.6.0 in
/usr/local/lib/python3.7/dist-packages (from scikit-fuzzy) (1.21.5)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.7/dist-
packages (from scikit-fuzzy) (1.4.1)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.7/dist-
packages (from scikit-fuzzy) (2.6.3)
Building wheels for collected packages: scikit-fuzzy
  Building wheel for scikit-fuzzy (setup.py) … done
  Created wheel for scikit-fuzzy: filename=scikit_fuzzy-0.4.2-py3-none-any.whl
size=894089
sha256=6b4c1114cf40a4608a5eca548f72367a556fbb614c862d6b2e5b101356805057
  Stored in directory: /root/.cache/pip/wheels/d5/74/fc/38588a3d2e3f34f74588e6da
a3aa5b0a322bd6f9420a707131
Successfully built scikit-fuzzy
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.4.2
```

[2]:
```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt
import scipy.integrate as integrate
```

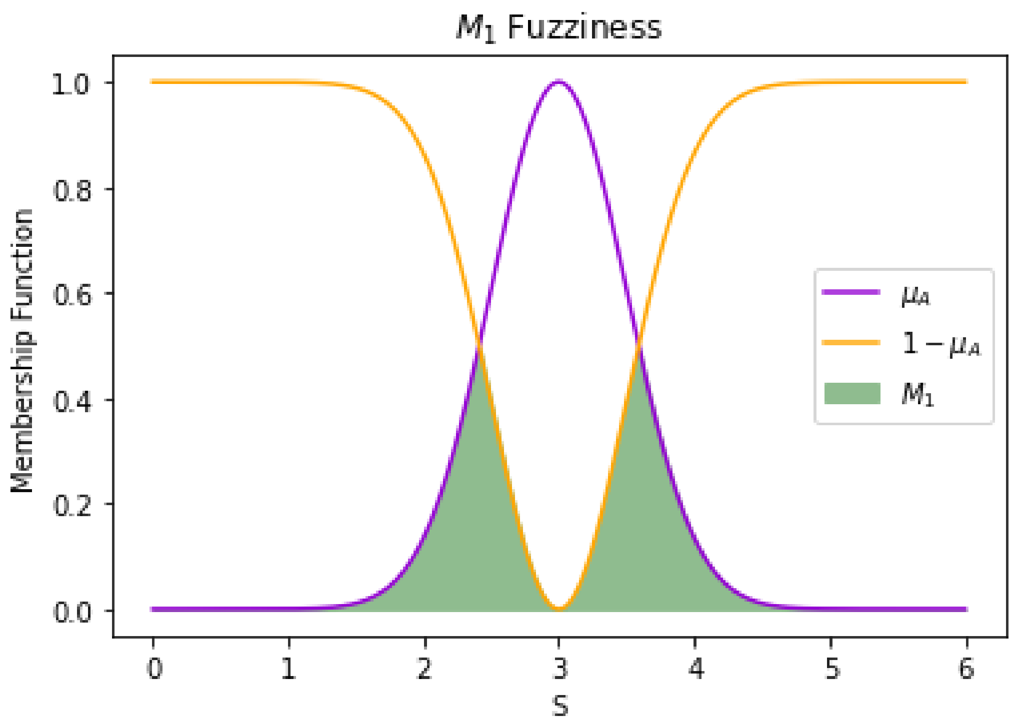
[3]:
```
s = np.linspace(0,6,500) # Support Set
lambd, n, a = 2, 2, 3

# Membership Function
```

```
def mu_a(x):
  return np.exp( -lambd* np.power( (x - a), n) )

def minus_mu_a(x):
  return 1 - mu_a(x)

def alpha_cut(x):
  return fuzz.lambda_cut(x, 1/2)
```

[4]:
```
# Part a)
def m1(x):
  return mu_a(x) if mu_a(x) <= 0.5 else (1 - mu_a(x))

membership_func = mu_a(s)
membership_minus = np.array([ minus_mu_a(x) for x in s ])
M1 = np.array([ m1(x) for x in s ])

print('M1: {:.3f}'.format(integrate.quad(m1, 0, 6)[0]))

fig, ax = plt.subplots()
ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
ax.plot(s, membership_minus, label='$1 - \mu_A$', color='orange')
ax.fill_between(s, M1, color='darkseagreen', alpha=1, label='$M_1$')
ax.set_title('$M_1$ Fuzziness')
ax.set_xlabel('S')
ax.set_ylabel('Membership Function')
ax.legend(loc='best')
```

```
M1: 0.523
```

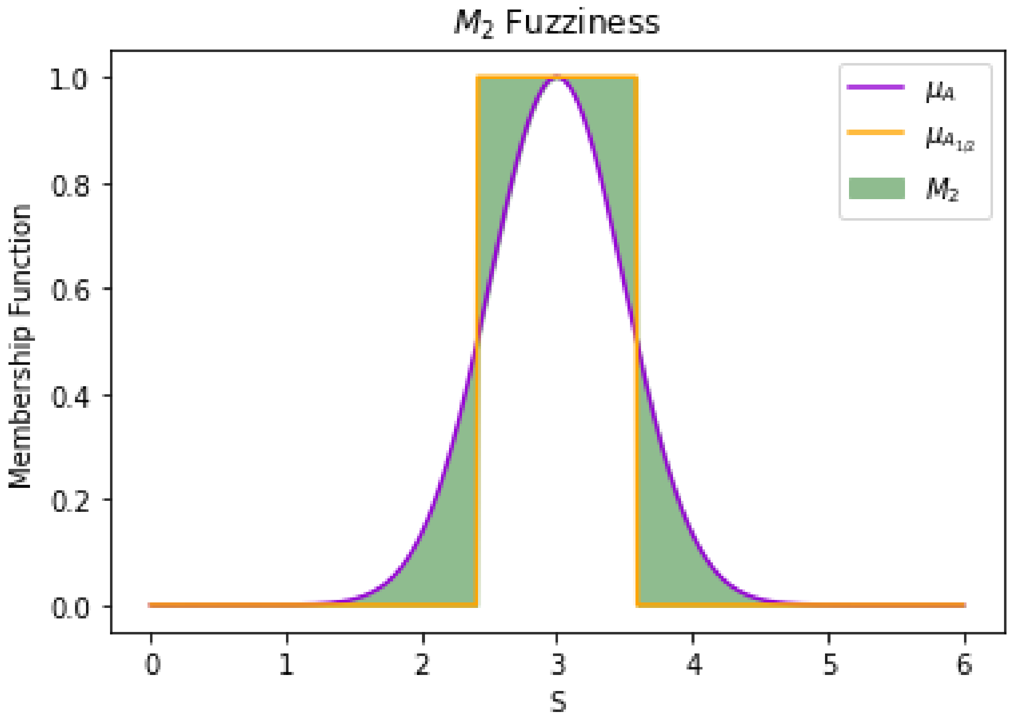
[4]: <matplotlib.legend.Legend at 0x7fb6a0af3a10>




```
[5]: # Part b)
def m2(x):
  half_mu_a = 1 if mu_a(x) >= 0.5 else 0
  return np.abs(mu_a(x) - half_mu_a)

M2 = np.array([ m2(x) for x in s ])
print('M2: {:.3f}'.format(integrate.quad(m2, 0, 6)[0]))

fig, ax = plt.subplots()
ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
ax.plot(s, alpha_cut(membership_func), label='$\mu_{A_{1/2}}$', color='orange')
ax.fill_between(s, membership_func, alpha_cut(membership_func),␣
 ↪color='darkseagreen', label='$M_2$')
ax.set_title('$M_2$ Fuzziness')
ax.set_xlabel('S')
ax.set_ylabel('Membership Function')
ax.legend(loc='best')
```

```
M2: 0.523
```

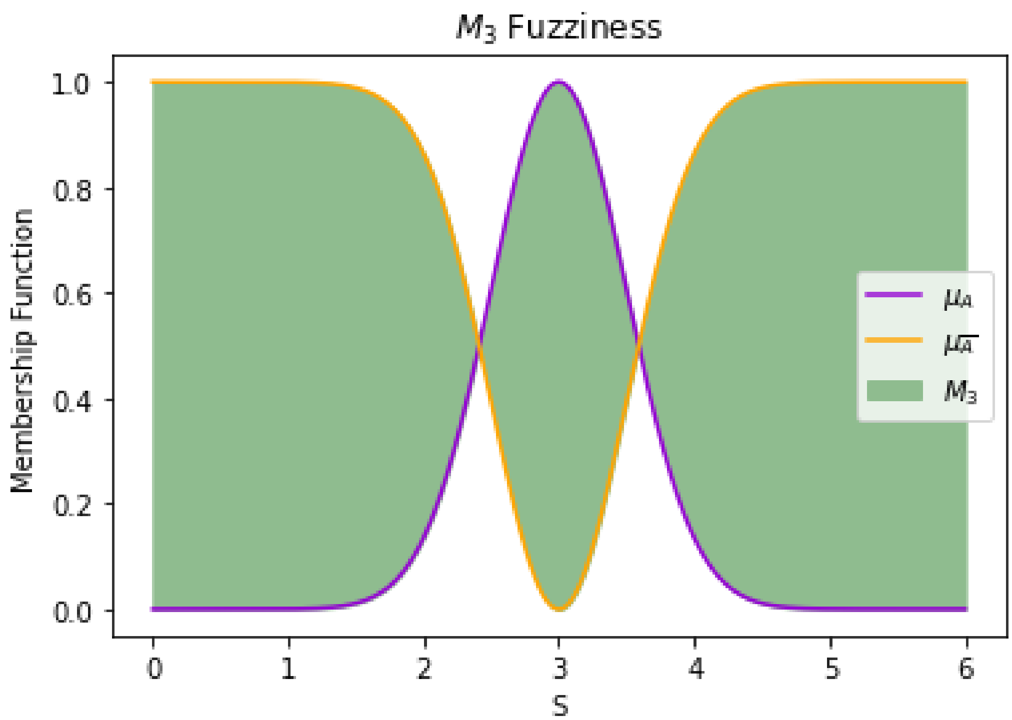
```
[5]: <matplotlib.legend.Legend at 0x7fb6a09d59d0>
```




```
[6]: # Part c)
def m3(x):
  return np.abs(mu_a(x) - minus_mu_a(x))

M3 = np.array([ m3(x) for x in s ])
print('M3: {:.3f}'.format(integrate.quad(m3, 0, 6)[0]))

fig, ax = plt.subplots()
ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
ax.plot(s, minus_mu_a(s), label='$\mu_{\overline{A}}$', color='orange')
ax.fill_between(s, membership_func, minus_mu_a(s), color='darkseagreen',␣
 ↪label='$M_3$')
ax.set_title('$M_3$ Fuzziness')
ax.set_xlabel('S')
ax.set_ylabel('Membership Function')
ax.legend(loc='best')
```

```
M3: 4.953
```

```
[6]: <matplotlib.legend.Legend at 0x7fb6b27c0050>
```



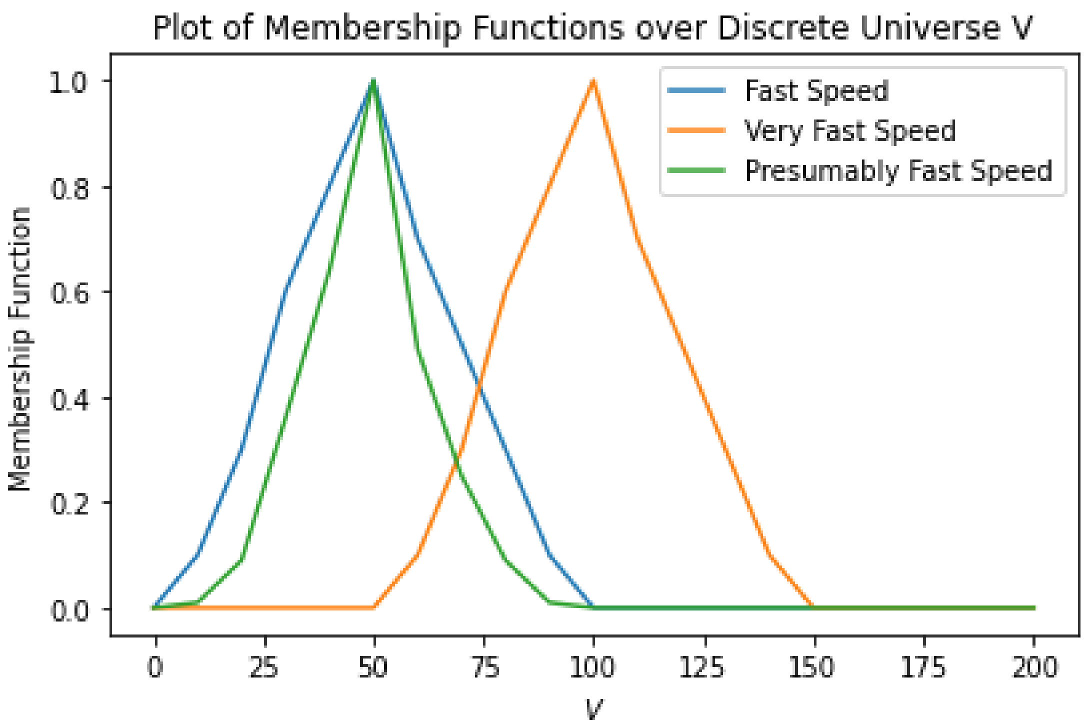
1.1.2 Problem 2

[7]:
```
# No Code Required
```

1.1.3 Problem 3

[8]:
```
V = np.arange(0, 201, 10)
F = np.array([0, 0.1, 0.3, 0.6, 0.8, 1.0, 0.7, 0.5, 0.3, 0.
 ↪1,0,0,0,0,0,0,0,0,0,0,0])

mu_F_1 = np.roll(F, 5) # shift right by 50
mu_F_2 = F **2 # square

fig, ax = plt.subplots(figsize=(6.5,4))
ax.plot(V, F, label='Fast Speed')
ax.plot(V, mu_F_1, label='Very Fast Speed')
ax.plot(V, mu_F_2, label='Presumably Fast Speed')
ax.set_title('Plot of Membership Functions over Discrete Universe V')
ax.set_xlabel('$V$')
ax.set_ylabel('Membership Function')
ax.legend(loc='best')
```

[8]: <matplotlib.legend.Legend at 0x7fb6a03b79d0>



1.1.4 Problem 4

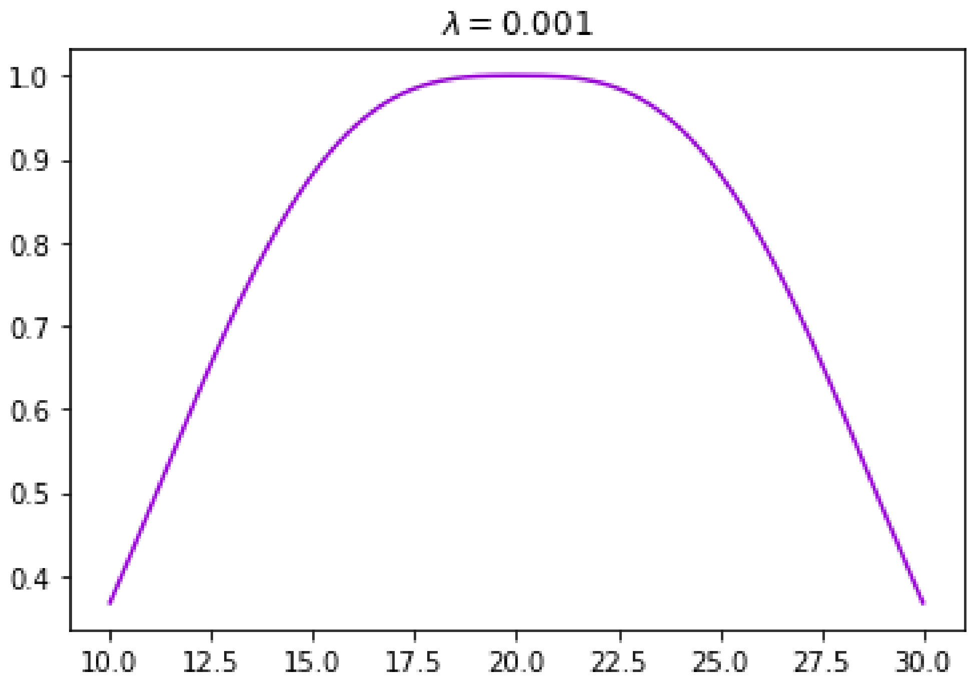
[9]:
```
# No Code Required
```

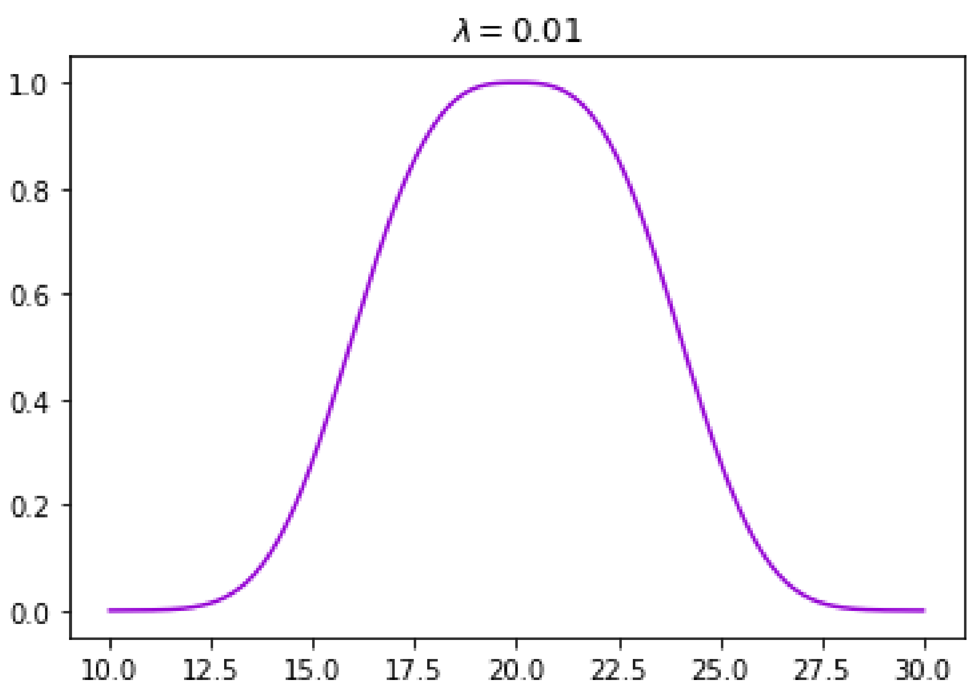
1.1.5 Problem 5

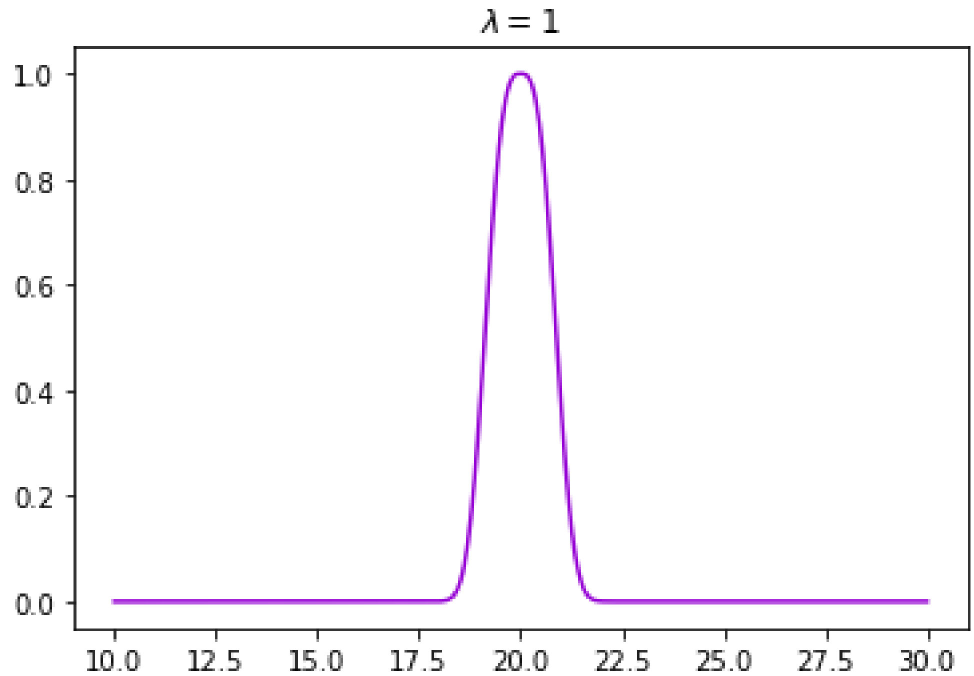
Part a)

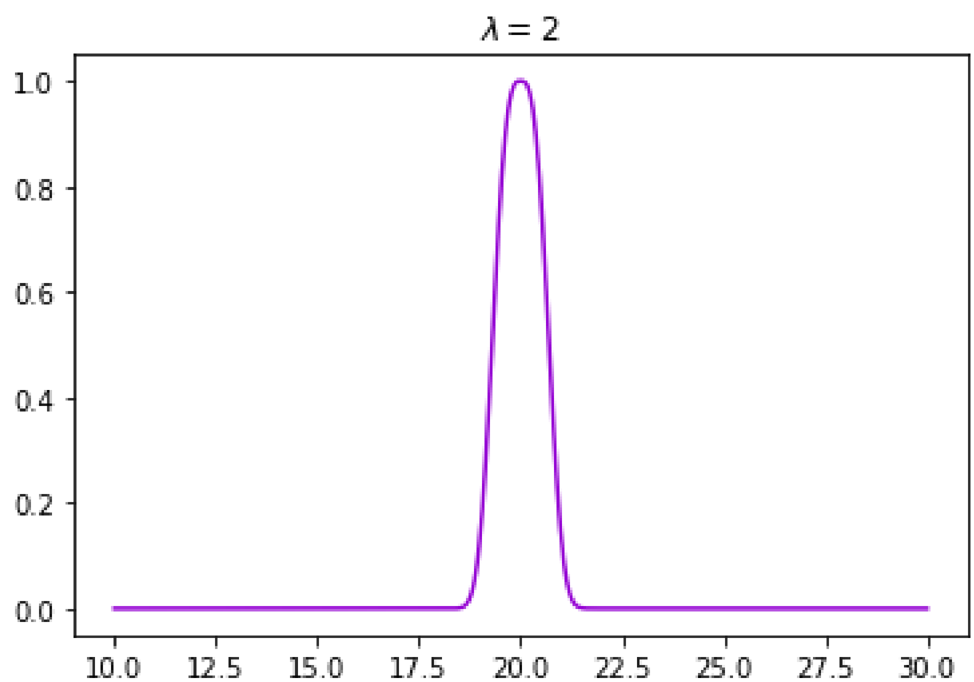
[10]:
```
lambd_param = [0.001, 0.01, 1, 2]

# Membership Function
def mu_a(x, lambd, a, n):
  return np.exp( -lambd* np.abs( np.power( (x - a), n) ) )

# Support Set
s = np.linspace(10,30,500)

for num in lambd_param:
  membership_func = mu_a(s, num, a=20, n=3)
```

```
fig, ax = plt.subplots()
ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
ax.set_title('$\lambda = {}$'.format(num))
```

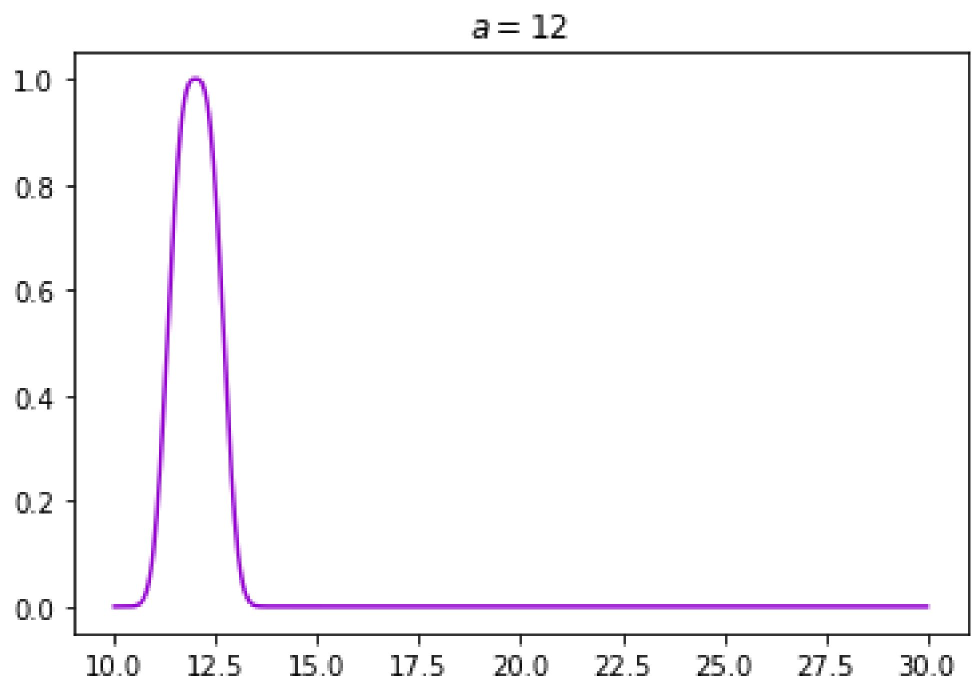


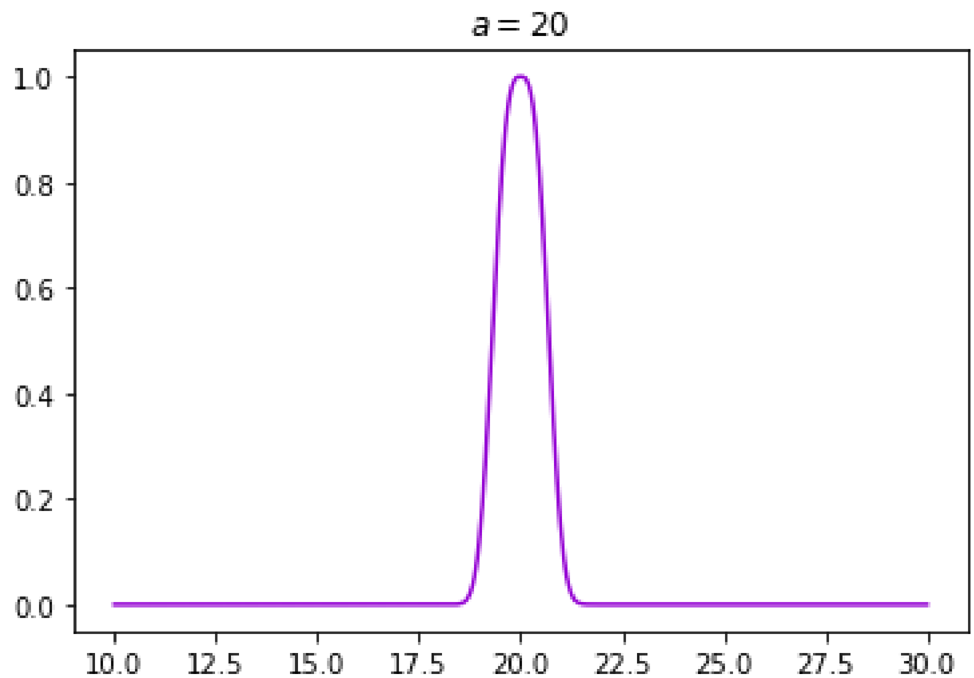


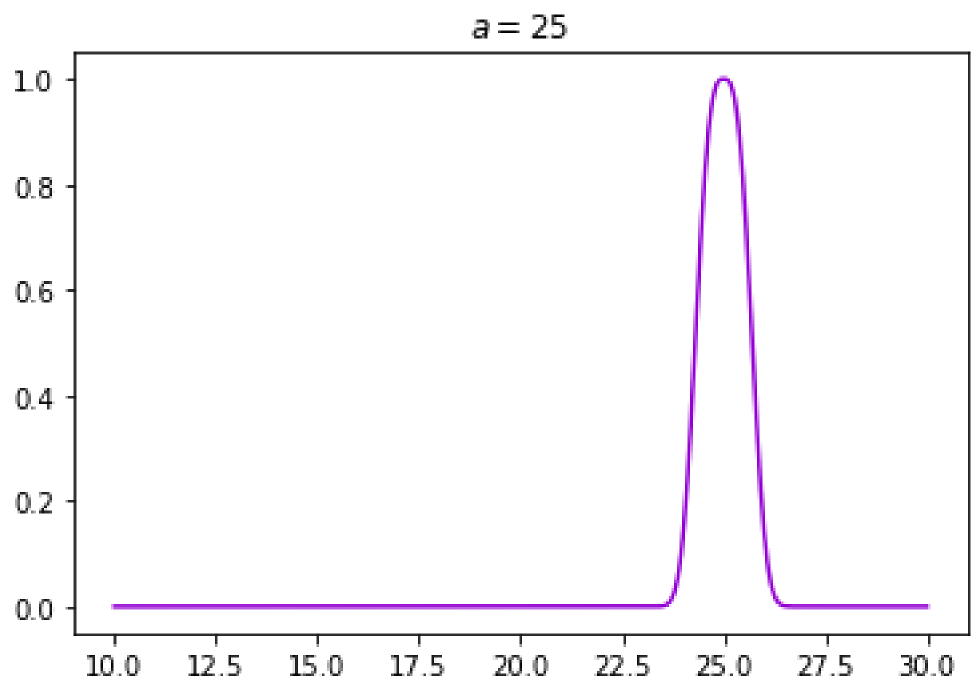

$\lambda = 1$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

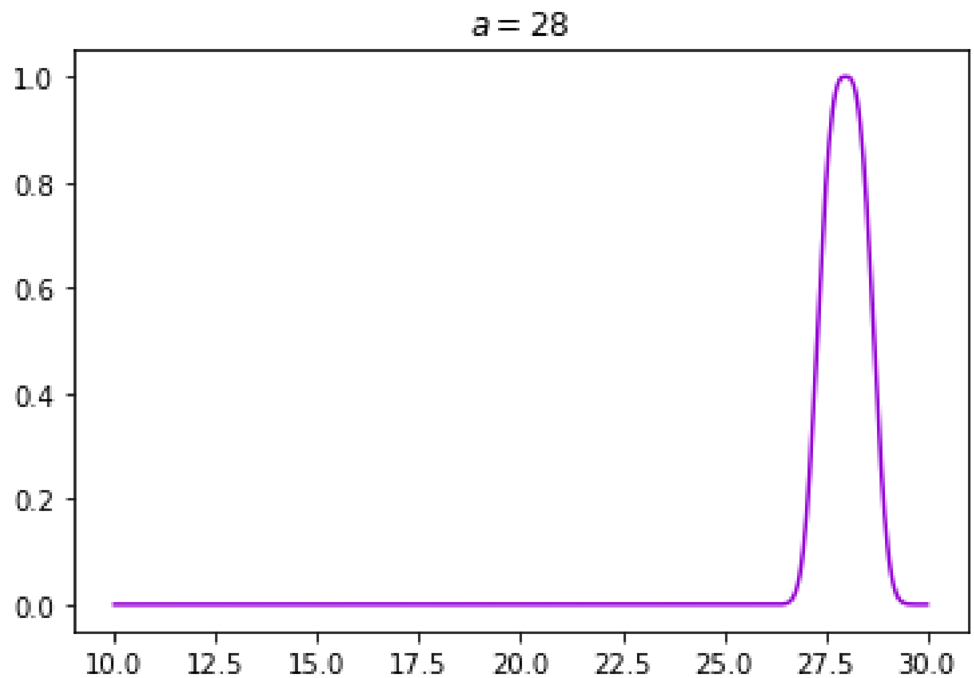

$\lambda = 2$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

[11]:
```
a_param = [12, 20, 25, 28]
for num in a_param:
  membership_func = mu_a(s, lambd=2, a=num, n=3)

  fig, ax = plt.subplots()
  ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
  ax.set_title('$a = {}$'.format(num))
```

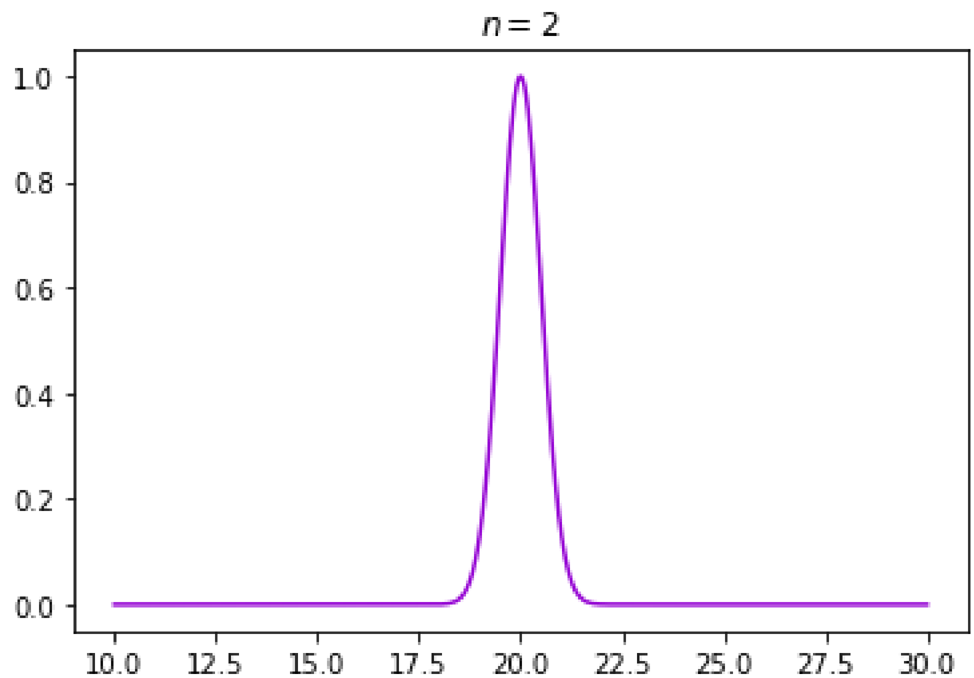


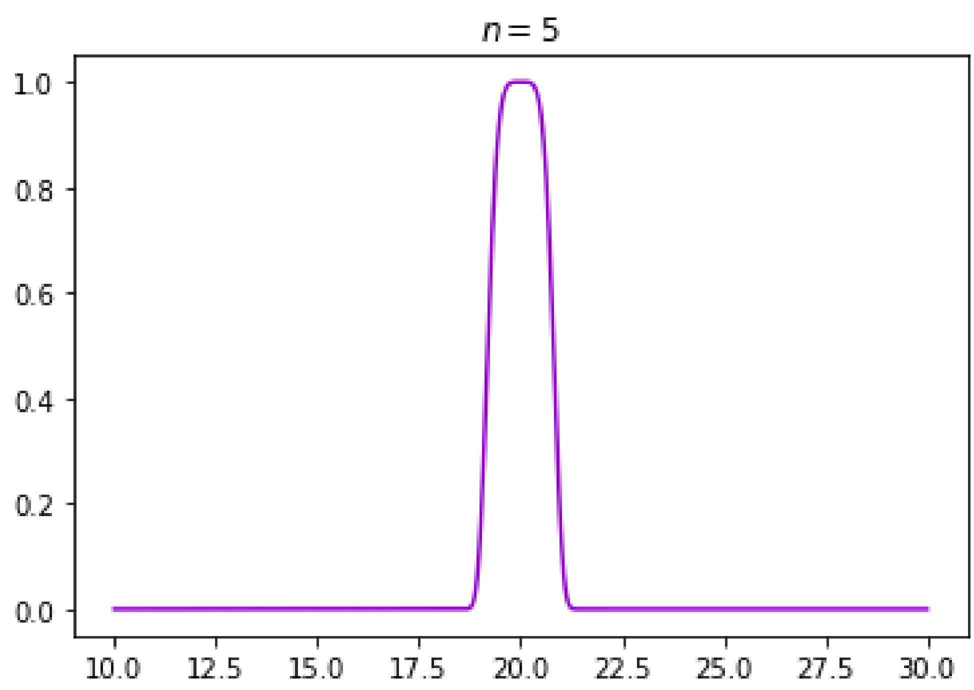

$a = 20$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

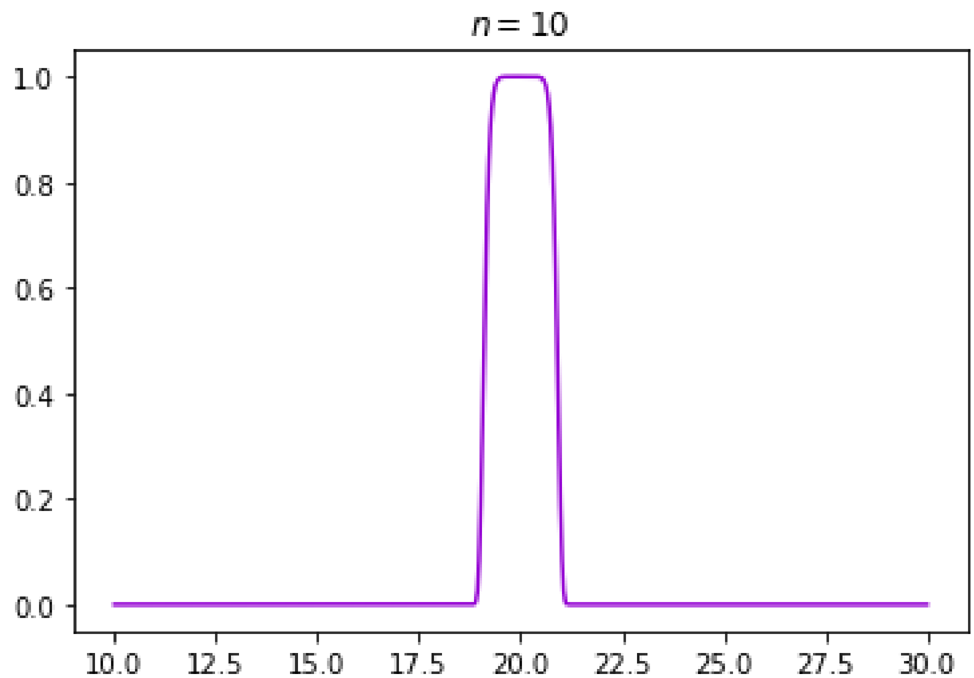

$a = 25$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

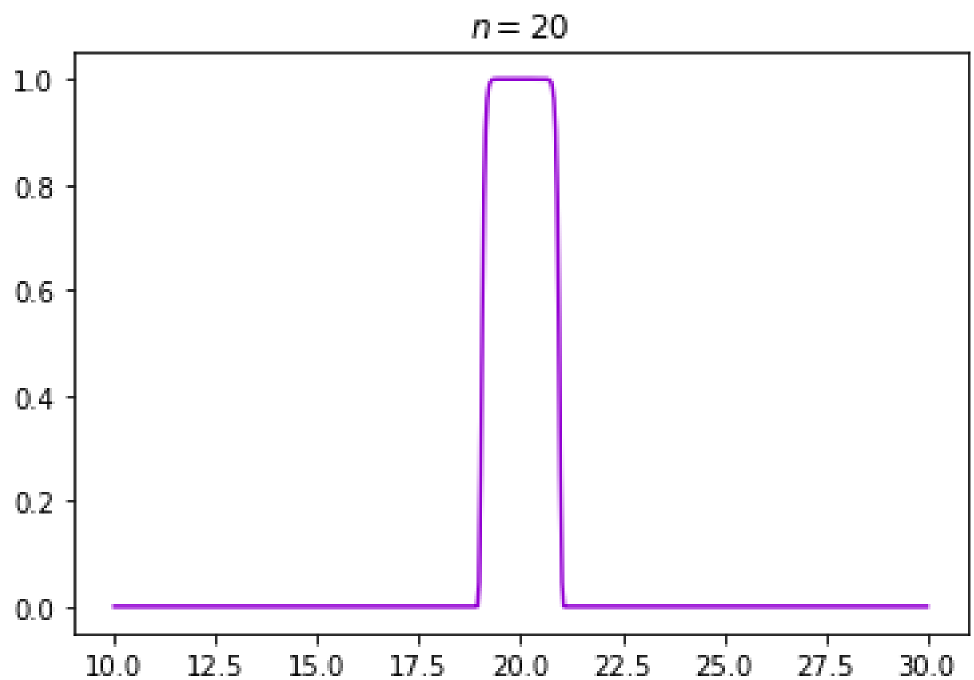


[12]:
```
n_param = [2, 5, 10, 20]
for num in n_param:
  membership_func = mu_a(s, lambd=2, a=20, n=num)

  fig, ax = plt.subplots()
  ax.plot(s, membership_func, label='$\mu_A$', color='darkviolet')
  ax.set_title('$n = {}$'.format(num))
```


$n = 2$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


$n = 5$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


$n = 10$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0


$n = 20$
1.0
0.8
0.6
0.4
0.2
0.0
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

Part b)

[13]:
```python
cold = mu_a(s, lambd=0.06, a=10, n=2)
hot = mu_a(s, lambd=0.06, a=30, n=2)
comfortable = mu_a(s, lambd=0.1, a=21, n=4)

fig, ax = plt.subplots(figsize=(10.5,4.25))
ax.plot(s, cold, label='Cold', color='blue')
ax.plot(s, hot, label='Hot', color='red')
ax.plot(s, comfortable, label='Comfortable', color='green')
ax.set_title('Temperature Inside Living Room')
ax.set_xlabel('Temperature (Celcius)')
ax.set_ylabel('Membership Function')
ax.legend(loc='best')
```

[13]: <matplotlib.legend.Legend at 0x7fb69ff20a50>