hw2

February 25, 2021

[1]:
```
import numpy as np
from math import exp
from math import factorial
```

[2]:
```
#problem 2
```

[3]:
```
r_a = [10, 20, 30, 5, 5]
r_d = [10, 15, 15, 10, 15]
t = [60, 60, 60, 180, 30]
```

[4]:
```
a_t = np.cumsum([r_a[i] * t[i] for i in range(5)])
```

[5]:
```
d_t = np.cumsum([r_d[i] * t[i] for i in range(5)])
```

[6]:
```
a_t
```

[6]:
```
array([ 600, 1800, 3600, 4500, 4650])
```

[7]:
```
d_t
```

[7]:
```
array([ 600, 1500, 2400, 4200, 4650])
```

[8]:
```
np.cumsum(t)
```

[8]:
```
array([ 60, 120, 180, 360, 390])
```

[9]:
```
a_t - d_t
```

[9]:
```
array([   0,  300, 1200,  300,    0])
```

[10]:
```
delay = [((a_t[i] + a_t[i+1]) * t[i+1] / 2) - ((d_t[i] + d_t[i+1]) * t[i+1] /␣
↪2) for i in range(4)]
```

[11]:
```
delay = [0] + delay
```

[12]:
```
delay = np.cumsum(delay)
```

[13]:
```
delay
```

[13]:
```
array([     0.,   9000.,  54000., 189000., 193500.])
```

[14]:
```
ave_d = [delay[i]/a_t[i] for i in range(5)]
```

[15]:
```
ave_d
```

[15]:
```
[0.0, 5.0, 15.0, 42.0, 41.61290322580645]
```

[16]:
```
#problem 6
```

[17]:
```
exp(-20)
```

[17]:
```
2.061153622438558e-09
```

[18]:
```
l = 20
t = 1
n = 5
exp(-l*t)*(l*t)**n/factorial(n)
```

[18]:
```
5.496409659836154e-05
```

[19]:
```
l = 20
t = 1
n = 5
1-np.sum([exp(-l*t)*(l*t)**i/factorial(i) for i in range(5)])
```

[19]:
```
0.9999830552560699
```

[20]:
```
#problem 3
```

[21]:
```
r_a = [30, 30]
r_d = [20, 40]
t = [2, 2]
```

[22]:
```
a_t = np.cumsum([r_a[i] * t[i] for i in range(2)])
```

[23]:
```
d_t = np.cumsum([r_d[i] * t[i] for i in range(2)])
```

[24]:
```
a_t
```

[24]:
```
array([ 60, 120])
```

[25]:
```
d_t
```

[25]:
```
array([ 40, 120])
```

[26]:
```
a_t - d_t
```

[26]:
```
array([20,  0])
```

[27]:
```
delay = [((a_t[i] + a_t[i+1]) * t[i+1] / 2) - ((d_t[i] + d_t[i+1]) * t[i+1] /␣
↪2) for i in range(1)]
```

[28]:
```
delay = [0] + delay
```

[29]:
```
delay = np.cumsum(delay)
```

[30]:
```
delay
```

[30]:
```
array([ 0., 20.])
```

[31]:
```
ave_d = [delay[i]/a_t[i] for i in range(2)]
```

[32]:
```
ave_d
```

[32]:
```
[0.0, 0.16666666666666666]
```

[33]:
```
ave_d[-1]*60
```

[33]:
```
10.0
```

[34]:
```
#problem 4
```

[35]:
```
r_a = [6, 6, 6]
r_d = [0, 15, 10]
t = [2, 1, 0.75]
```

[36]:
```
a_t = np.cumsum([r_a[i] * t[i] for i in range(len(t))])
```

[37]:
```
d_t = np.cumsum([r_d[i] * t[i] for i in range(len(t))])
```

[38]:
```
a_t
```

[38]:
```
array([12. , 18. , 22.5])
```

[39]:
```
d_t
```

[39]:
```
array([ 0. , 15. , 22.5])
```

[40]:
```
a_t - d_t
```

[40]:
```
array([12.,  3.,  0.])
```

[41]:
```
delay = [((a_t[i] + a_t[i+1]) * t[i+1] / 2) - ((d_t[i] + d_t[i+1]) * t[i+1] /␣
↪2) for i in range(len(t)-1)]
```

[42]:
```
delay = [0] + delay
```

[43]:
```
delay = np.cumsum(delay)
```

[44]:
```
delay
```

[44]:
```
array([0.   , 7.5  , 8.625])
```

[45]:
```
ave_d = [delay[i]/a_t[i] for i in range(len(t))]
```

[46]:
```
ave_d
```

[46]:
```
[0.0, 0.4166666666666667, 0.38333333333333336]
```

[47]:
```
ave_d[-1]*60
```

[47]:
```
23.0
```

[48]:
```
#problem 5
```

[49]:
```
r_a = [45, 45]
r_d = [30, 60]
t = [1, 1]
```

[50]:
```
a_t = np.cumsum([r_a[i] * t[i] for i in range(len(t))])
```

[51]:
```
d_t = np.cumsum([r_d[i] * t[i] for i in range(len(t))])
```

[52]:
```
a_t
```

[52]:
```
array([45, 90])
```

[53]:
```
d_t
```

[53]:
```
array([30, 90])
```

[54]:
```
a_t - d_t
```

[54]:
```
array([15,  0])
```

[55]:
```
delay = [((a_t[i] + a_t[i+1]) * t[i+1] / 2) - ((d_t[i] + d_t[i+1]) * t[i+1] /␣
↪2) for i in range(len(t)-1)]
```

[56]:
```
delay = [0] + delay
```

[57]:
```
delay = np.cumsum(delay)
```

[58]:
```
delay
```

[58]:
```
array([0. , 7.5])
```

[59]:
```
ave_d = [delay[i]/a_t[i] for i in range(len(t))]
```

[60]:
```
ave_d
```

[60]:
```
[0.0, 0.08333333333333333]
```

[61]:
```
ave_d[-1]*60
```

[61]:
```
5.0
```

[62]:
```
r_a = [10, 10]
r_d = [0, 30]
t = [2, 1]
```

[63]:
```
a_t = np.cumsum([r_a[i] * t[i] for i in range(len(t))])
```

[64]:
```
d_t = np.cumsum([r_d[i] * t[i] for i in range(len(t))])
```

[65]:
```
a_t
```

[65]:
```
array([20, 30])
```

[66]:
```
d_t
```

[66]:
```
array([ 0, 30])
```

[67]:
```
a_t - d_t
```

[67]:
```
array([20,  0])
```

[68]:
```
delay = [((a_t[i] + a_t[i+1]) * t[i+1] / 2) - ((d_t[i] + d_t[i+1]) * t[i+1] /␣
↪2) for i in range(len(t)-1)]
```

[69]:
```
delay = [0] + delay
```

[70]:
```
delay = np.cumsum(delay)
```

[71]:
```
delay
```

[71]:
```
array([ 0., 10.])
```

[72]:
```
ave_d = [delay[i]/a_t[i] for i in range(len(t))]
```

[73]:
```
ave_d
```

[73]: [0.0, 0.3333333333333333]

[74]:
```
ave_d[-1]*60
```

[74]: 20.0