cloud formation1

April 24, 2021

[31]:
```
R=8.314551 # Universal gas constant
rho_s=2163 #Density of NaCl
rho_w=1000 #Density of pure water
D_dry1=10*10**-9 #Diameter of dry NaCl
D_dry2=40*10**-9
D_dry3=100*10**-9
D_dry4=80*10**-9
T=288 # Temperature in Kelivin
M_s=0.05844 #Molecular weight of NaCl
M_w=0.01802 # Molecular weight of Water
```

[32]:
```
w_t=[i for i in range(1,46)] # % of solute in the solotion
#Density of aq solution
A=[7.41,-3.741*10**-2,2.525*10**-3,-2.060*10**-5]
rho_aq=[997.1+A[0]*w_t[i]+A[1]*w_t[i]**2+A[2]*w_t[i]**3+A[3]*w_t[i]**4 for i in␣
↪range(len(w_t))]
#Water activity
C=[-6.366*10**-3,8.624*10**-5,-1.154*10**-5,1.518*10**-7]
a_w=[1+C[0]*w_t[i]+C[1]*w_t[i]**2+C[2]*w_t[i]**3+C[3]*w_t[i]**4 for i in␣
↪range(len(w_t))]
#Surface tension of aqueous solution
delta_aq=[0.072+((0.029*w_t[i])/(100-w_t[i])) for i in range(len(w_t))]
```

[33]:
```
g=[(100*rho_s/(w_t[i]*rho_aq[i]))**(1/3) for i in range(len(w_t))] #growth␣
↪factor
D_aq1=[g[i]*D_dry1 for i in range(len(w_t))] #Diameter of aqueous solution
D_aq2=[g[i]*D_dry2 for i in range(len(w_t))] #Diameter of aqueous solution
D_aq3=[g[i]*D_dry3 for i in range(len(w_t))] #Diameter of aqueous solution
D_aq4=[g[i]*D_dry4 for i in range(len(w_t))] #Diameter of aqueous solution
D_aq3[14]
```

[33]: 2.352692225422668e-07

[34]:
```
import math
m_aq1=[rho_aq[i]*(math.pi/6)*D_aq1[i]**3 for i in range(len(w_t))] #Mass of␣
↪aqueous solution in kg
```

```
m_aq2=[rho_aq[i]*(math.pi/6)*D_aq2[i]**3 for i in range(len(w_t))] #Mass of␣
 ↪aqueous solution in kg
m_aq3=[rho_aq[i]*(math.pi/6)*D_aq3[i]**3 for i in range(len(w_t))] #Mass of␣
 ↪aqueous solution in kg
m_aq4=[rho_aq[i]*(math.pi/6)*D_aq4[i]**3 for i in range(len(w_t))] #Mass of␣
 ↪aqueous solution in kg
```

[35]:
```
m_w1 =[((100-w_t[i])/100)*m_aq1[i] for i in range(len(w_t)) ] # Mass of water␣
 ↪in the aqueous solution in kg
m_w2 =[((100-w_t[i])/100)*m_aq2[i] for i in range(len(w_t)) ] # Mass of water␣
 ↪in the aqueous solution in kg
m_w3 =[((100-w_t[i])/100)*m_aq3[i] for i in range(len(w_t)) ] # Mass of water␣
 ↪in the aqueous solution in kg
m_w4 =[((100-w_t[i])/100)*m_aq4[i] for i in range(len(w_t)) ] # Mass of water␣
 ↪in the aqueous solution in kg


m_s1 =[w_t[i]/100*m_aq1[i] for i in range(len(w_t)) ] # Mass of NaCl in the␣
 ↪aqueous solution in kg
m_s2 =[w_t[i]/100*m_aq2[i] for i in range(len(w_t)) ] # Mass of NaCl in the␣
 ↪aqueous solution in kg
m_s3 =[w_t[i]/100*m_aq3[i] for i in range(len(w_t)) ] # Mass of NaCl in the␣
 ↪aqueous solution in kg
m_s4 =[w_t[i]/100*m_aq4[i] for i in range(len(w_t)) ] # Mass of NaCl in the␣
 ↪aqueous solution in kg
```

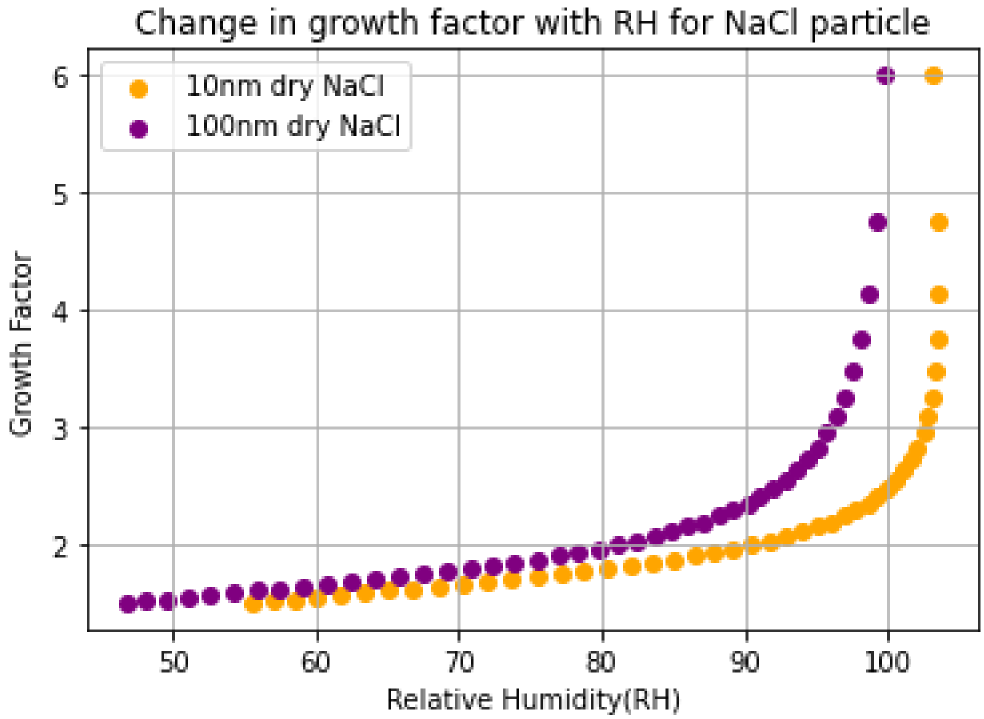
[36]:
```
RH_1=[100*a_w[i] for i in range(len(w_t))]
RH_21=[RH_1[i]*math.exp((4*delta_aq[i]*M_w)/(rho_w*R*T*D_aq1[i])) for i in␣
 ↪range(len(w_t))]
RH_22=[RH_1[i]*math.exp((4*delta_aq[i]*M_w)/(rho_w*R*T*D_aq2[i])) for i in␣
 ↪range(len(w_t))]
RH_23=[RH_1[i]*math.exp((4*delta_aq[i]*M_w)/(rho_w*R*T*D_aq3[i])) for i in␣
 ↪range(len(w_t))]
RH_24=[RH_1[i]*math.exp((4*delta_aq[i]*M_w)/(rho_w*R*T*D_aq4[i])) for i in␣
 ↪range(len(w_t))]
RH_23
```

[36]:
```
[99.73231276183478,
 99.20803590998659,
 98.65992702513905,
 98.09654307381639,
 97.51747219773264,
 96.91996882302804,
 96.3005462702816,
 95.65553649433922,
```

```
 94.9813324067807,
 94.2745080451049,
 93.53188423189593,
 92.75056718111799,
 91.92797255173298,
 91.0618411606548,
 90.15024966415416,
 89.19161807313284,
 88.18471520464117,
 87.12866274770325,
 86.02293837518366,
 84.8673781850279,
 83.66217866188593,
 82.4078982910499,
 81.10545891788685,
 79.75614691996371,
 78.36161424130675,
 76.92387932589156,
 75.44532797875972,
 73.92871417694887,
 72.377160847959,
 70.7941606302511,
 69.18357662794749,
 67.54964317024103,
 65.89696658486132,
 64.23052599417923,
 62.555674142077805,
 60.878138259526075,
 59.2040209768282,
 57.539801290760394,
 55.892335595245896,
 54.268858784849975,
 52.67698544120876,
 51.12471111355553,
 49.62041370579231,
 48.17285498410762,
 46.79118222099736]
```

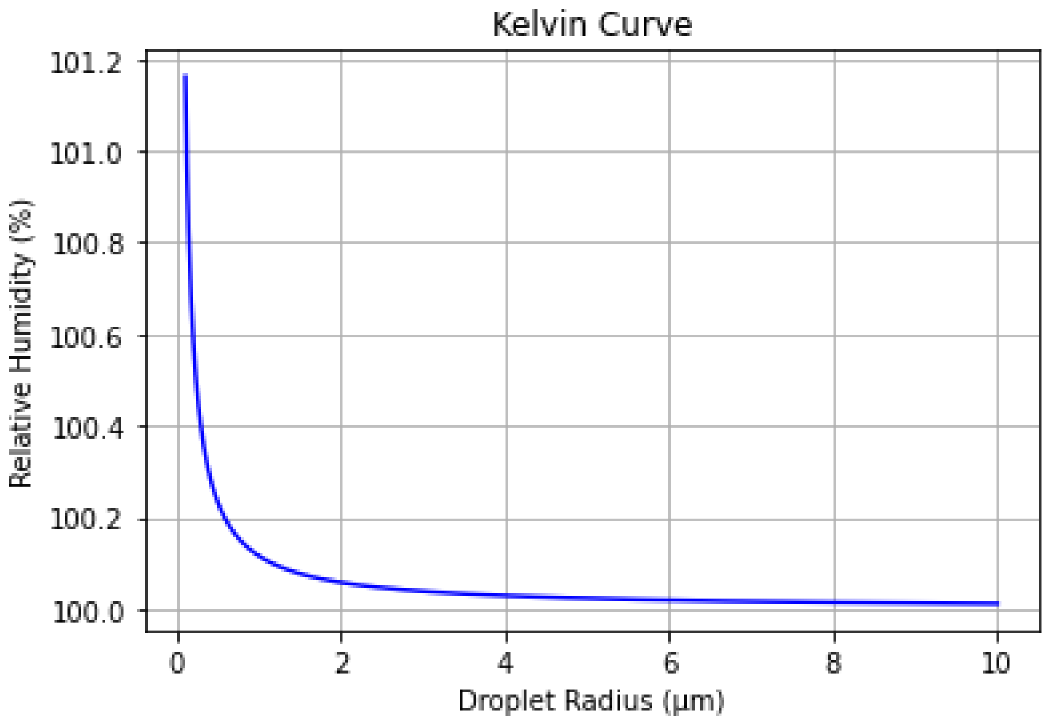
[37]:
```
import matplotlib.pyplot as plt
plt.scatter(RH_21,g, color='orange') #Growth factor for 10nm dry particle
#plt.plot(RH_22,g, color='blue')
plt.scatter(RH_23,g, color='purple') #Growth factor for 90nm dry particle
plt.xlabel('Relative Humidity(RH)')
plt.legend(['10nm dry NaCl', '100nm dry NaCl'], loc='upper left')
plt.ylabel('Growth Factor')
plt.grid(True)
plt.title('Change in growth factor with RH for NaCl particle')
```

[37]: Text(0.5, 1.0, 'Change in growth factor with RH for NaCl particle')



[38]:
```python
#Plotting Kelvin equation
import numpy as np
delta=0.076
M_w=0.0182
rho_w=1000
R=8.314551

r=[i/100 for i in range(10,1001)]
RH=[100*math.exp((2*delta*M_w)/(rho_w*R*T*r[i]*10**-6)) for i in range(len(r))]

plt.plot(r,RH,color='blue')
plt.grid(True)
plt.xlabel('Droplet Radius (µm)')
plt.ylabel('Relative Humidity (%)')
plt.title('Kelvin Curve')
```

[38]: Text(0.5, 1.0, 'Kelvin Curve')



[39]:
```
n=3.3*10**28
k=1.38*10**-23
e_s=1
e1=0.5
e2=1.5
i=2
R=[i/100 for i in range(1,1001)]

Delta_E1=[4*math.pi*(R[i])**2*delta-(4/3)*math.pi*(R[i])**3*n*k*T*math.log(e1/
 ↪e_s) for i in range(len(R))]
Delta_E2=[4*math.pi*(R[i])**2*delta-(4/3)*math.pi*(R[i])**3*n*k*T*math.log(e2/
 ↪e_s) for i in range(len(R))]


#plt.plot(R,Delta_E1,color='red')
#plt.plot(R,Delta_E2,color='blue')
```

[40]:
```
import numpy as np
import math

i=2
i2=3
M_w=0.01802
```

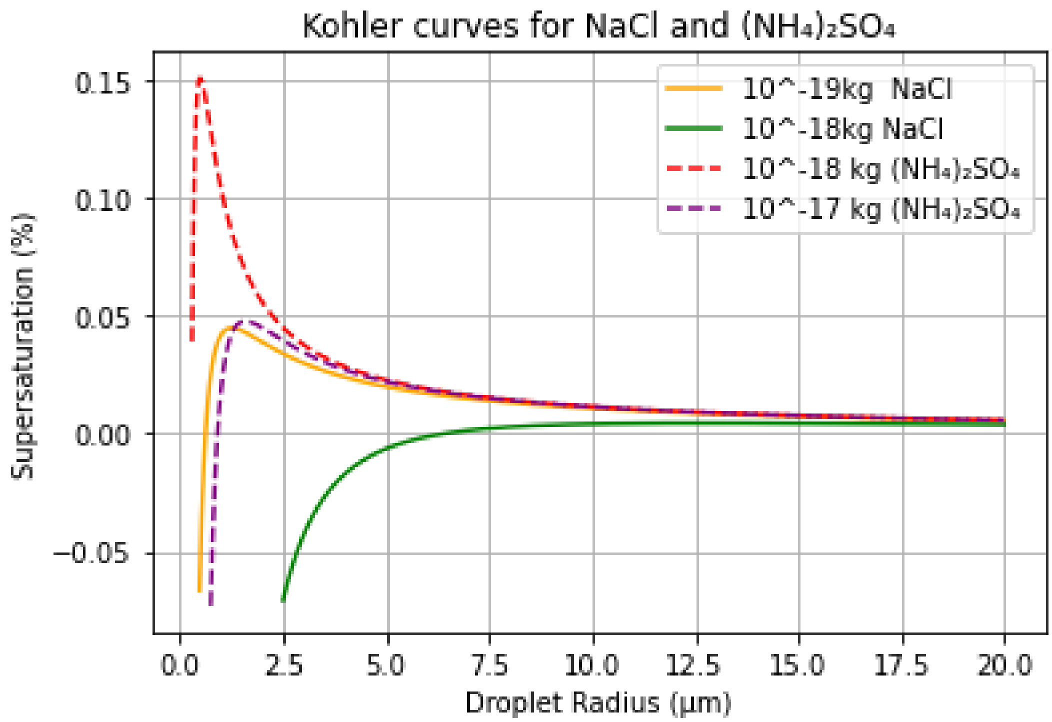
```
M_s=0.05844
M_s2=0.13214
m1=10**-19
m2=10**-18
m3=10**-17
T=288
R=[i/100 for i in range(1,2001)]
delta=0.076
rho_w=1000
R_c=8.314551
RH1=[100*((math.exp((2*delta*M_w)/(rho_w*R_c*T*R[i]*10**-6))*(1+((i*m1*M_w)/
 ↪(M_s*((4/3)*math.pi*(R[i]*10**-6)**3*rho_w-m1))))**-1)-1) for i in␣
 ↪range(len(R))]
RH2=[100*((math.exp((2*delta*M_w)/(rho_w*R_c*T*R[i]*10**-6))*(1+((i*m2*M_w)/
 ↪(M_s*((4/3)*math.pi*(R[i]*10**-6)**3*rho_w-m2))))**-1)-1) for i in␣
 ↪range(len(R))]
RH3=[100*((math.exp((2*delta*M_w)/(rho_w*R_c*T*R[i]*10**-6))*(1+((i2*m2*M_w)/
 ↪(M_s2*((4/3)*math.pi*(R[i]*10**-6)**3*rho_w-m2))))**-1)-1) for i in␣
 ↪range(len(R))]
RH4=[100*((math.exp((2*delta*M_w)/(rho_w*R_c*T*R[i]*10**-6))*(1+((i2*m3*M_w)/
 ↪(M_s2*((4/3)*math.pi*(R[i]*10**-6)**3*rho_w-m3))))**-1)-1) for i in␣
 ↪range(len(R))]
```

[41]:
```
R1=[i/100 for i in range(48,len(RH1))]
R2=[i/100 for i in range(250,len(RH1))]
R3=[i/100 for i in range(30,len(RH1))]
R4=[i/100 for i in range(75,len(RH1))]
R5=[i/100 for i in range(len(RH1))]

s1=[RH1[i] for i in range(48,len(RH1))]
s2=[RH2[i] for i in range(250,len(RH2))]
s3=[RH3[i] for i in range(30,len(RH2))]
s4=[RH4[i] for i in range(75,len(RH2))]
s5=[0 for i in range(len(RH2))  ]

plt.plot(R1,s1, color='orange')
plt.plot(R2,s2, color='green')
plt.plot(R3,s3,'--' ,color='red')
plt.plot(R4,s4,'--' ,color='purple')
#plt.plot(R5,s5,color='blue')

plt.legend(['10^-19kg  NaCl', '10^-18kg NaCl','10^-18 kg (NH ) SO ','10^-17 kg␣
 ↪(NH ) SO '], loc='upper right')
#plt.plot(R,RH2, color='blue')
plt.grid(True)
plt.xlabel('Droplet Radius (μm)')
plt.ylabel('Supersaturation (%)')
```

```
plt.title('Kohler curves for NaCl and (NH ) SO ')
```

[41]:

```
Text(0.5, 1.0, 'Kohler curves for NaCl and (NH ) SO ')
```



[42]:

```
Happy_New_Year=['\x1b[6;30;46m' + 'HAPPY NEW YEAR 2021 FRIENDS' + '\x1b[0m' for␣
 ↪i in range(5)]
Happy_New_Year1=['\x1b[6;30;43m' + 'HAPPY NEW YEAR 2021 FRIENDS' + '\x1b[0m'␣
 ↪for i in range(5)]
#A=[print(Happy_New_Year[i]) for i in range(5)]
B=[print(Happy_New_Year1[i]) for i in range(5)]
```

```
HAPPY NEW YEAR 2021 FRIENDS
HAPPY NEW YEAR 2021 FRIENDS
HAPPY NEW YEAR 2021 FRIENDS
HAPPY NEW YEAR 2021 FRIENDS
HAPPY NEW YEAR 2021 FRIENDS
```

[ ]:

[ ]: