Solución Asignación 3
FISI6510

Guillermo Fidalgo

February 16, 2021

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from vpython import *
```

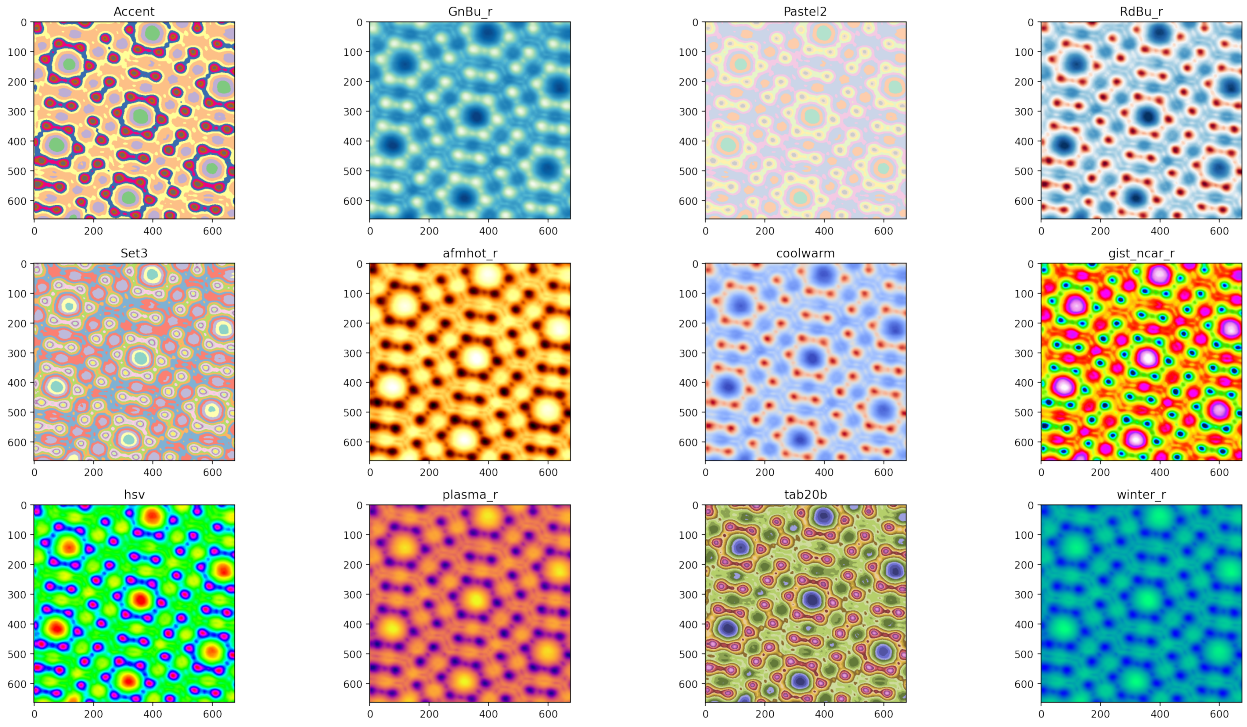
[2]:
```
cmaps=['Accent', 'Accent_r',
       'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r',
       'CMRmap', 'CMRmap_r',
       'Dark2', 'Dark2_r',
       'GnBu', 'GnBu_r', 'Greens', 'Greens_r', 'Greys', 'Greys_r',
       'OrRd', 'OrRd_r', 'Oranges', 'Oranges_r',
       'PRGn', 'PRGn_r', 'Paired', 'Paired_r', 'Pastel1', 'Pastel1_r',␣
 ↪'Pastel2', 'Pastel2_r', 'PiYG', 'PiYG_r',
       'PuBu', 'PuBuGn', 'PuBuGn_r', 'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd',␣
 ↪'PuRd_r', 'Purples', 'Purples_r',
       'RdBu', 'RdBu_r', 'RdGy', 'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu',␣
 ↪'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds',
       'Reds_r', 'Set1', 'Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r',␣
 ↪'Spectral', 'Spectral_r', 'Wistia',
       'Wistia_r', 'YlGn', 'YlGnBu', 'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r',␣
 ↪'YlOrRd', 'YlOrRd_r',
       'afmhot', 'afmhot_r', 'autumn', 'autumn_r', 'binary', 'binary_r', 'bone',␣
 ↪'bone_r', 'brg', 'brg_r',
       'bwr', 'bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm',␣
 ↪'coolwarm_r', 'copper',
       'copper_r', 'cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'gist_earth',␣
 ↪'gist_earth_r', 'gist_gray',
       'gist_gray_r', 'gist_heat', 'gist_heat_r', 'gist_ncar', 'gist_ncar_r',␣
 ↪'gist_rainbow', 'gist_rainbow_r',
       'gist_stern', 'gist_stern_r', 'gist_yarg', 'gist_yarg_r', 'gnuplot',␣
 ↪'gnuplot2', 'gnuplot2_r', 'gnuplot_r', 'gray', 'gray_r',
       'hot', 'hot_r', 'hsv', 'hsv_r', 'inferno', 'inferno_r', 'jet', 'jet_r',␣
 ↪'magma', 'magma_r', 'nipy_spectral', 'nipy_spectral_r',
       'ocean', 'ocean_r', 'pink', 'pink_r', 'plasma', 'plasma_r', 'prism',␣
 ↪'prism_r', 'rainbow', 'rainbow_r', 'seismic', 'seismic_r',
```

```
       'spring', 'spring_r', 'summer', 'summer_r', 'tab10', 'tab10_r', 'tab20',␣
 ↪'tab20_r', 'tab20b', 'tab20b_r', 'tab20c', 'tab20c_r',
       'terrain', 'terrain_r', 'turbo', 'turbo_r', 'twilight', 'twilight_r',␣
 ↪'twilight_shifted', 'twilight_shifted_r', 'viridis', 'viridis_r', 'winter',␣
 ↪'winter_r']
data=np.loadtxt("stm.txt")

m=cmaps[::15]

mcm=1
for i in range(1,13):
    if len(m)==4*i:
        mcm=i

plt.figure(figsize=(19,10),dpi=200)
for ix,x in enumerate(m,start=1):
    plt.subplot(mcm,4,ix)
    plt.title(x)
    plt.imshow(data,cmap=x)
plt.tight_layout()
plt.show()
```



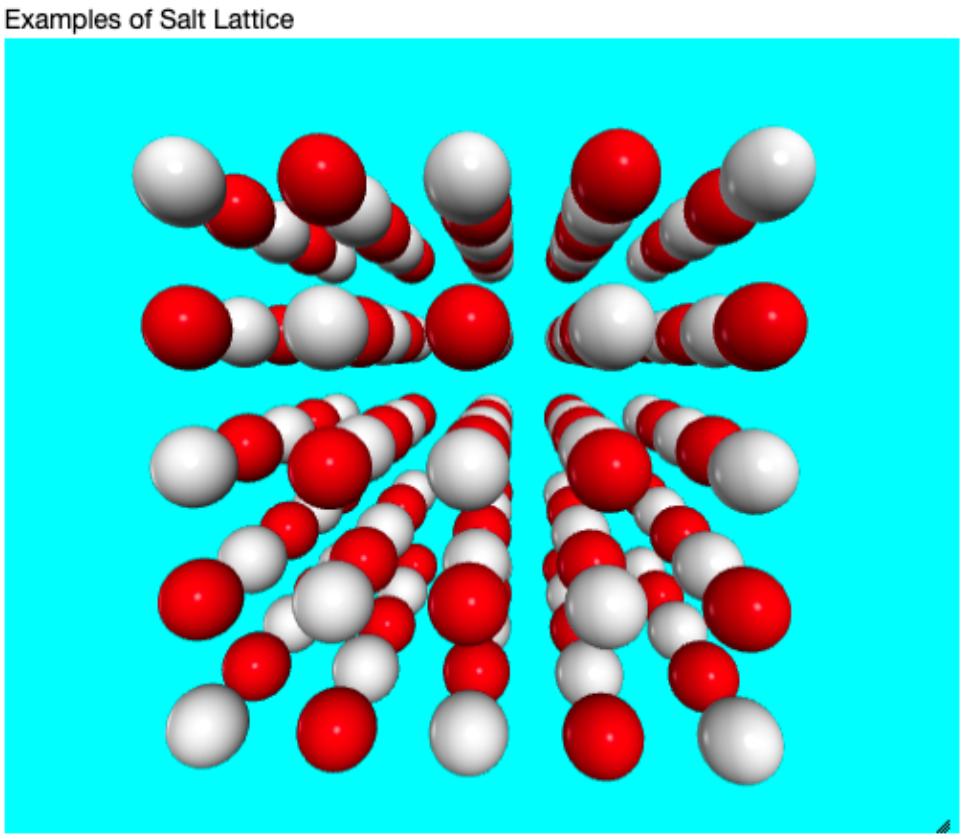
1 Salt Lattice

[3]:
```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


from vpython import sphere,vec,color, canvas
canvas(title='Examples of Salt Lattice',
     width=600, height=200, background=color.cyan)
R=.3
m=2
x=0
for i in range(-m,m+1):
    for j in range(-m,m+1):
        for k in range(-m,m+1):

            if x%2==0:
                sphere(radius=R,pos=vec(i,j,k),color=color.white)
            else:
                sphere(radius=R,pos=vec(i,j,k),color=color.red)
            x+=1
```



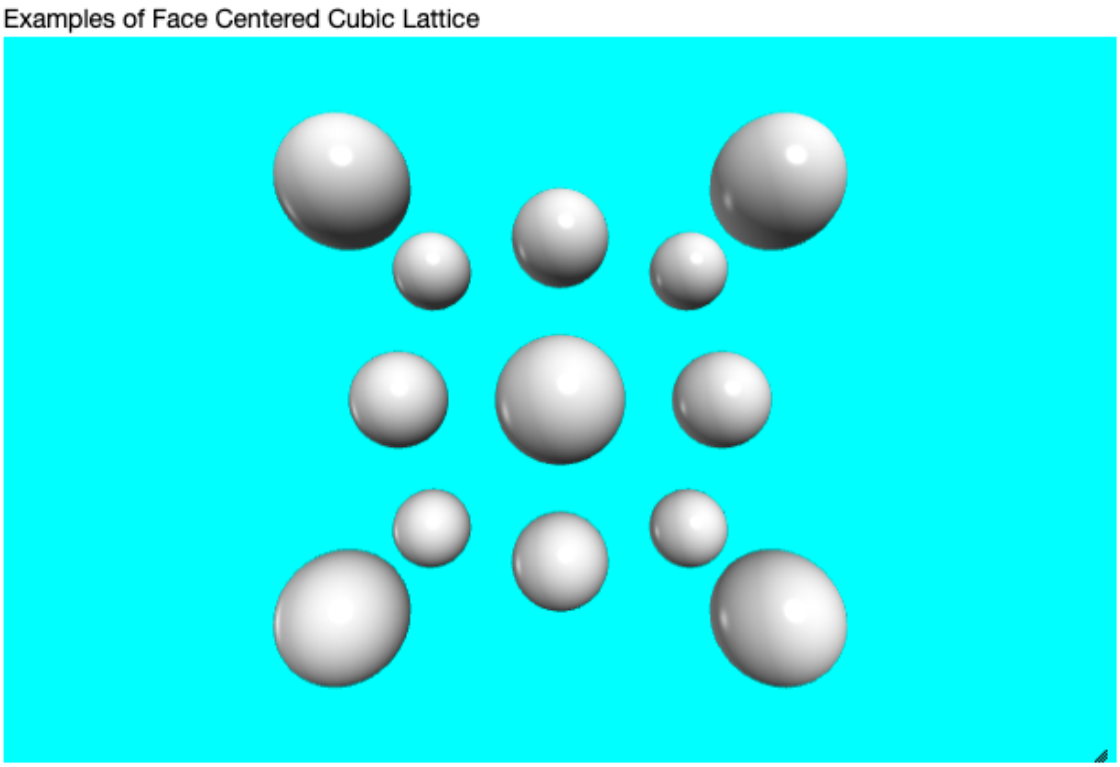
2 FCC Lattice

[4]:
```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


canvas(title='Examples of Face Centered Cubic Lattice',
     width=600, height=200, background=color.cyan)
R=.3
m=1
x=0
for i in range(-m,m+1):
    for j in range(-m,m+1):
        for k in range(-m,m+1):

            if x%2==0:
                sphere(radius=R,pos=vec(i,j,k),color=color.white)
            x+=1
```



3 Sistema Solar

[5]:
```
from vpython import *
from math import cos,sin,pi
from numpy import arange,empty,log, array,append


canvas(title='Solar system',
       width=900, height=600)

c1=5.0e6
Radio=array([2440,6052,6371,3386,69173/9,57316/9],float)*c1
Radio=append(Radio,695500*1e5)

# print(Radio)

c2=2.09e9
Radio_orbita=array([57.9,108.9,149.6,227.9,778.5,1433.4,0],float)*c2
Periodo=[88,224.7,365.3,687.0, 4331.6, 10759.2,0.01]
Periodo=array(Periodo)


s= empty(len(Radio),sphere)
for i in arange(len(s)):
    s[i]=sphere(make_trail=True)


colores = [color.orange,color.red,color.blue,color.cyan,color.green,color.
 ↪white,color.yellow]



# valores iniciales de cada esfera
for i in range(len(s)):
    s[i].color=colores[i]
    s[i].radius=Radio[i]
    s[i].pos=vec(Radio_orbita[i],0,0)




# Mover los planetas

# x=empty(len(s))
# y=empty(len(s))
omega=1/Periodo
```

```
for t in arange(0,1e3):
    for i in range(len(s)):
        rate(1000)
        x = Radio_orbita[i]*cos(omega[i]*t)
        y = Radio_orbita[i]*sin(omega[i]*t)
        s[i].pos = vec(x,y,0)
```

Solar system