Exercise_2_Butane_assessment_data_science

January 29, 2023

[8]:
```
import numpy as np
from scipy.io import loadmat
import ase
from ase import visualize

def view (x, s):
    system = ase.Atoms(positions = x.T , symbols=s )
    return visualize.view(system, viewer='x3d')


def frobenius(A):
    s=0
    for i in range(0,len(A[:,0]),1):
        for j in range(0,len(A[0,:]),1):
            s += A[i,j]*A[i,j]
    return s


butane=loadmat('butane.mat')
x1=butane['x1']
x2=butane['x2']
x3=butane['x3']

view(x1,butane['s'])
```

[8]:
```
<IPython.core.display.HTML object>
```

[9]:
```
view(x2,butane['s'])
```

[9]:
```
<IPython.core.display.HTML object>
```

[10]:
```
view(x3,butane['s'])
```

[10]:
```
<IPython.core.display.HTML object>
```

[11]:
```
Ux1x2, Sx1x2, Vtx1x2= np.linalg.svd(x2 @ x1.T) #x1-Qx2
Ux1x3, Sx1x3, Vtx1x3= np.linalg.svd(x3 @ x1.T) #x1-Qx3
Ux2x3, Sx2x3, Vtx2x3= np.linalg.svd(x3 @ x2.T) #x2-Qx3
```

```
Qx1x2= Vtx1x2.T @ Ux1x2.T #x1-Qx2
Qx1x3= Vtx1x3.T @ Ux1x3.T #x1-Qx3
Qx2x3= Vtx2x3.T @ Ux2x3.T #x2-Qx3


minx1x2=frobenius(x1-Qx1x2@x2)
minx1x3=frobenius(x1-Qx1x3@x3)
minx2x3=frobenius(x2-Qx2x3@x3)

print(minx1x2) #minimun of the frobenoìius norm of x1-Qx2
print(minx1x3) #minimun of the frobenoìius norm of x1-Qx3
print(minx2x3) #minimun of the frobenoìius norm of x2-Qx3
```

```
2.855986792362352e-29
0.827619335642786
0.827619335642785
```

[ ]: