In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

In [2]:

```
def funct(intial,t):
    #intial values for the parameters
    #can make these parameters time dependence
    delta=1
    omega=1
    v=1
    #ODE equations from Hamiltonian
    new1= omega*(intial[1]+intial[2])
    new2= omega*(intial[0]+intial[3])+delta*intial[1]
    new3= omega*(intial[0]+intial[3])+delta*intial[2]
    new4= omega*(intial[1]+intial[2])+ (v+2*delta)*intial[3]

    new_values= (new1,new2,new3,new4)
    new_values=np.array(np.concatenate( new_values , axis=None ))

    return new_values
```

In [3]:

```
#coeffcient value for each of the states
#can be changed
c_1 = 1
c_2 = 2
c_3 = 4
c_4 = 3


intial=(c_1,c_2,c_3,c_4)
intial=np.array(np.concatenate( intial , axis=None ))

#normalisation
intial_1=np.abs(intial)**2
normalisation=np.sum(intial_1)
#normalised coeffcient
intial_i=(intial)/(np.sqrt(normalisation))

#time step
t=np.linspace(0,1,100)
```

In [4]:

```
#solution to the ODE's
sol=odeint(funct,intial_i,t)
```

In [5]:

```
#normalise solution

norm=np.sqrt((sol[:,0])**2  +(sol[:,1])**2 +(sol[:,2])**2 +(sol[:,3])**2)
norm= 1/norm
norm=np.tile(norm,(4,1)).T
#solutions normalised
sol=sol*norm
```

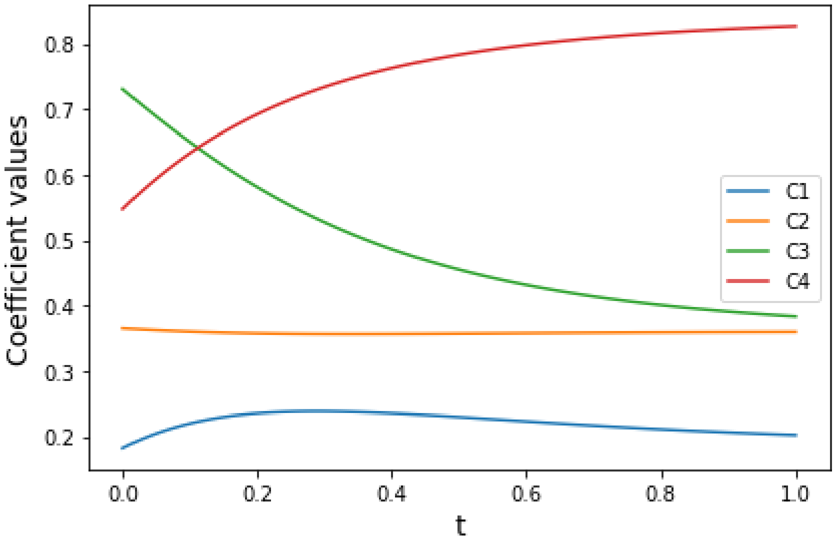
In [6]:

```
#size of array
go= np.size(intial)

fig = plt.plot()
for m in range(go):
    plt.plot(t,sol[:, (m)] ,label='C'+str(m+1))

plt.xlabel('t',fontsize=14)
plt.ylabel('Coefficient values',fontsize=14)
plt.legend(loc='best')

plt.tight_layout()
```



In [ ]: