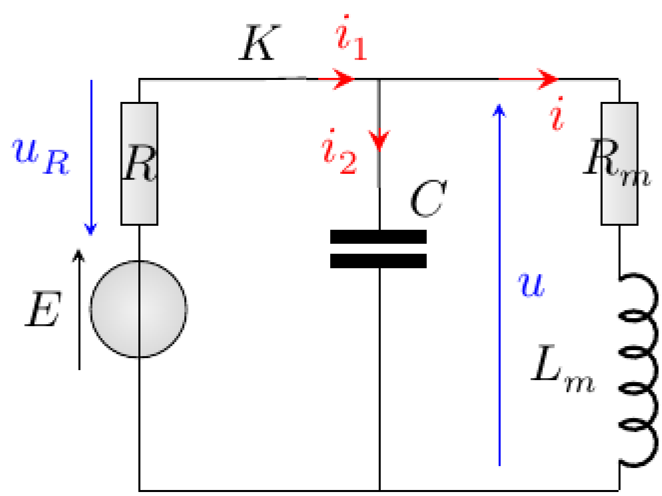
Moteur électrique protégé par des condensateurs

[1]:
```python
import numpy as np
import matplotlib.pyplot as plt
```

0.1 Q1:

Il y aurait une très grosse étincelle avec une forte surtension pouvant dégrader le moteur ou mettre le feu.

0.2 Q2:



En notant lorsque K est fermé:

On écrit les lois: $$\begin{cases} u + u_R = E \text{ (loi des mailles)} \\ i_1 = i_2 + i \text{ (loi des nœuds)} \\ u_R = R.i_1 \\ i_2 = C\dfrac{du}{dt} \\ u = R_m i + L_m\dfrac{di}{dt} \end{cases}$$

On a alors
$$R_m.i + L_m\frac{di}{dt} + R.i_1 = E$$
$$\text{donc } R_m.i + L_m\frac{di}{dt} + R.i_2 + R.i = E$$
$$\text{donc } R_m.i + L_m\frac{di}{dt} + RC\frac{du}{dt} + R.i = E$$
$$\text{donc } R_m.i + L_m\frac{di}{dt} + RCR_m\frac{di}{dt} + RCL_m\frac{d^2i}{dt^2} + R.i = E$$
$$\text{donc } RCL_m\frac{d^2i}{dt^2} + (L_m + RCR_m)\frac{di}{dt} + (R_m + R).i = E$$
Lorsque K est ouvert, on a un simple circuit R_mL_mC série.

0.3 Q3:

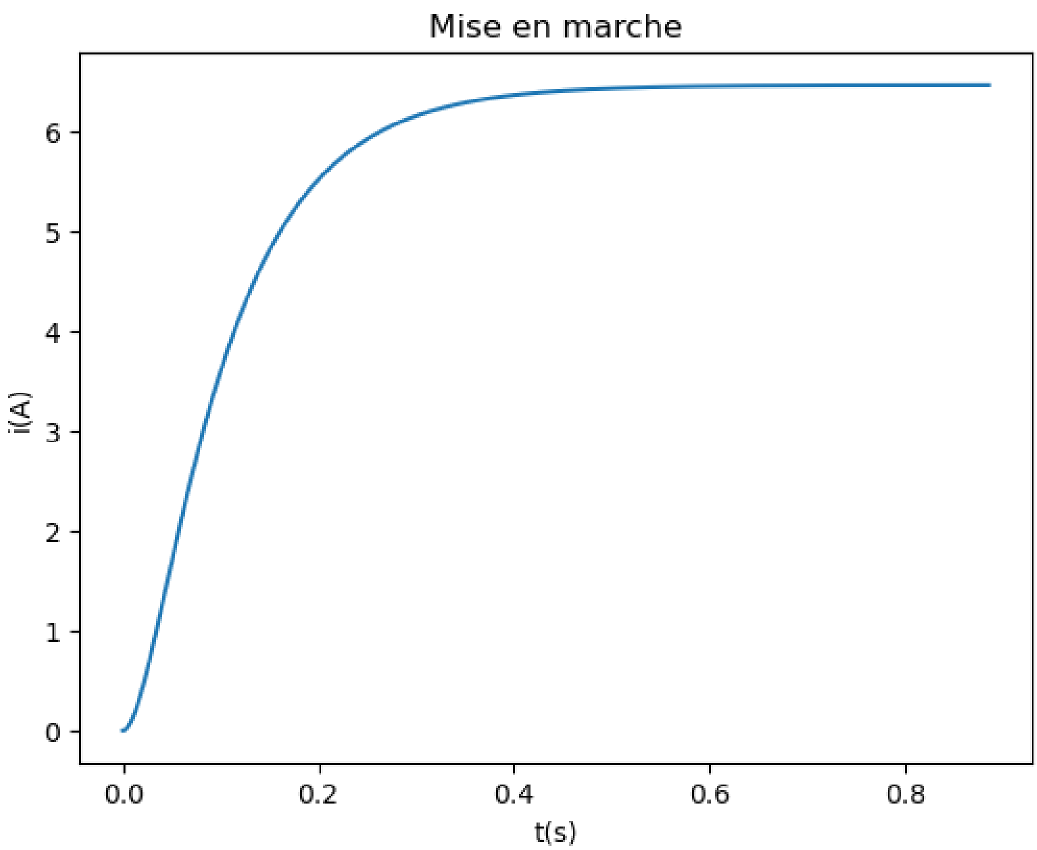
[2]:
```
Lm,Rm,R,E=3.1,32,2,220
Ccritique=4*Lm/Rm**2
print(Ccritique)
```

```
0.012109375
```

0.4 Q4:

[3]:
```
C=Ccritique
def deriveeseconde(i,didt,t):
    return (E-(Lm+R*Rm*C)*didt-(Rm+R)*i)/(R*C*Lm)
w0=np.sqrt((Rm+R)/(R*C*Lm))
T0=2*np.pi/w0
h=T0/1000
tmax=3*T0
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_i=np.zeros(N+1)
tab_didt=np.zeros(N+1)
tab_t[0],tab_i[0],tab_didt[0]=0,0,0
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_i[n+1]=tab_i[n]+h*tab_didt[n]
    tab_didt[n+1]=tab_didt[n]+h*deriveeseconde(tab_i[n],tab_didt[n],tab_t[n])
```

[4]:
```
plt.figure()
plt.plot(tab_t,tab_i)
plt.xlabel("t(s)")
plt.ylabel("i(A)")
plt.title("Mise en marche")
plt.show()
```



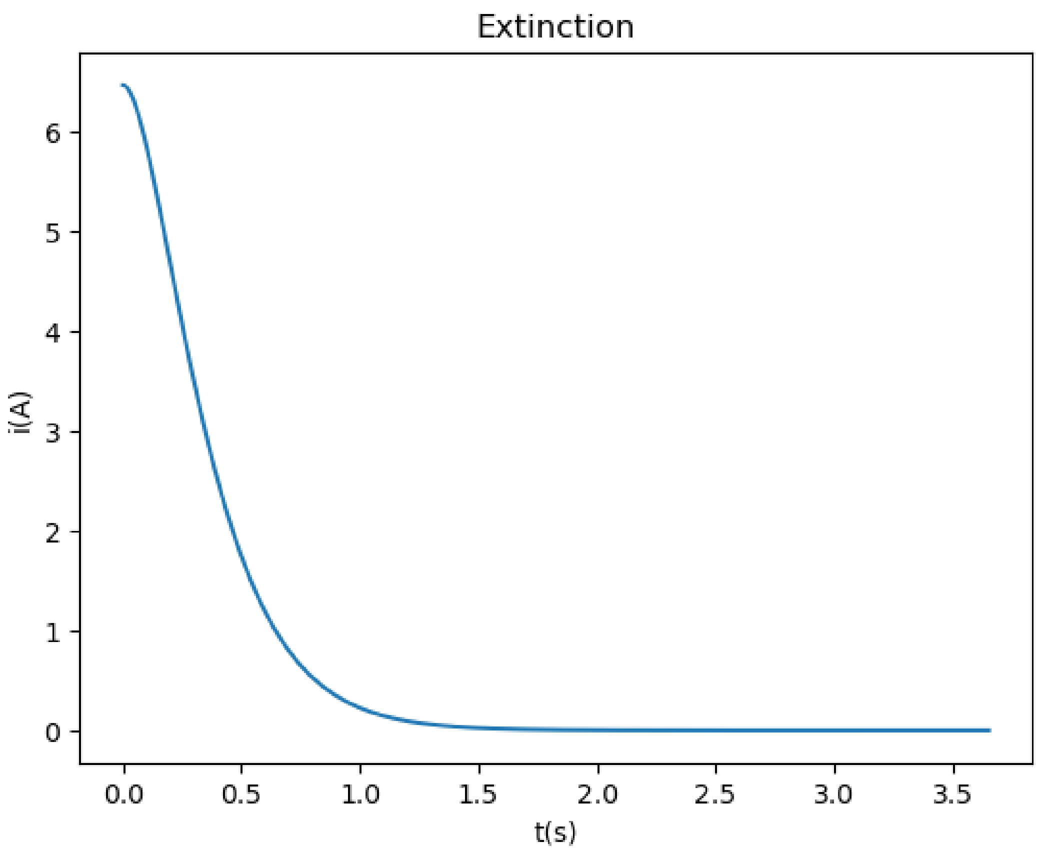
Le transitoire dure environ $0,3s$.

0.5 Q5:

[5]:

```
C=Ccritique
def deriveeseconde(i,didt,t):
    return (-(Rm*C)*didt-i)/(C*Lm)
w0=np.sqrt(1/(Lm*C))
T0=2*np.pi/w0
h=T0/1000
tmax=3*T0
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_i=np.zeros(N+1)
tab_didt=np.zeros(N+1)
tab_t[0],tab_i[0],tab_didt[0]=0,E/(R+Rm),0
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_i[n+1]=tab_i[n]+h*tab_didt[n]
    tab_didt[n+1]=tab_didt[n]+h*deriveeseconde(tab_i[n],tab_didt[n],tab_t[n])
```

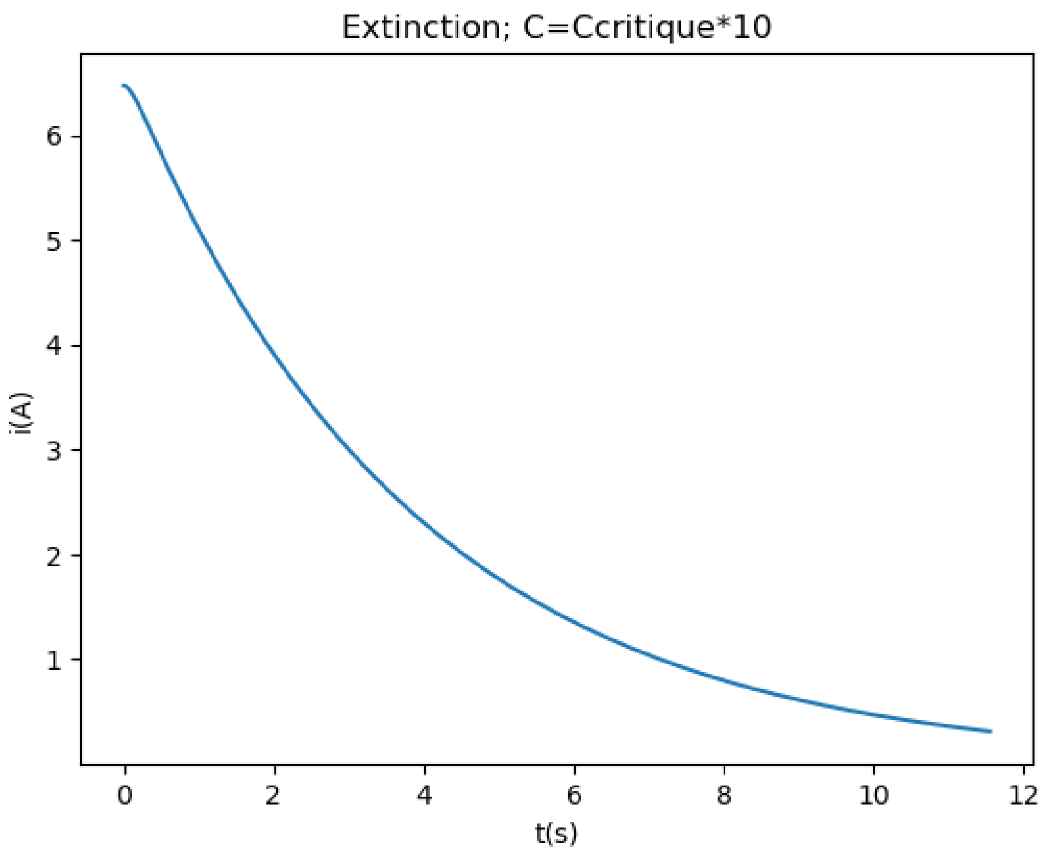
[6]:
```
plt.figure()
plt.plot(tab_t,tab_i)
plt.xlabel("t(s)")
plt.ylabel("i(A)")
plt.title("Extinction")
plt.show()
```



Temps d'extinction: $0,6s$

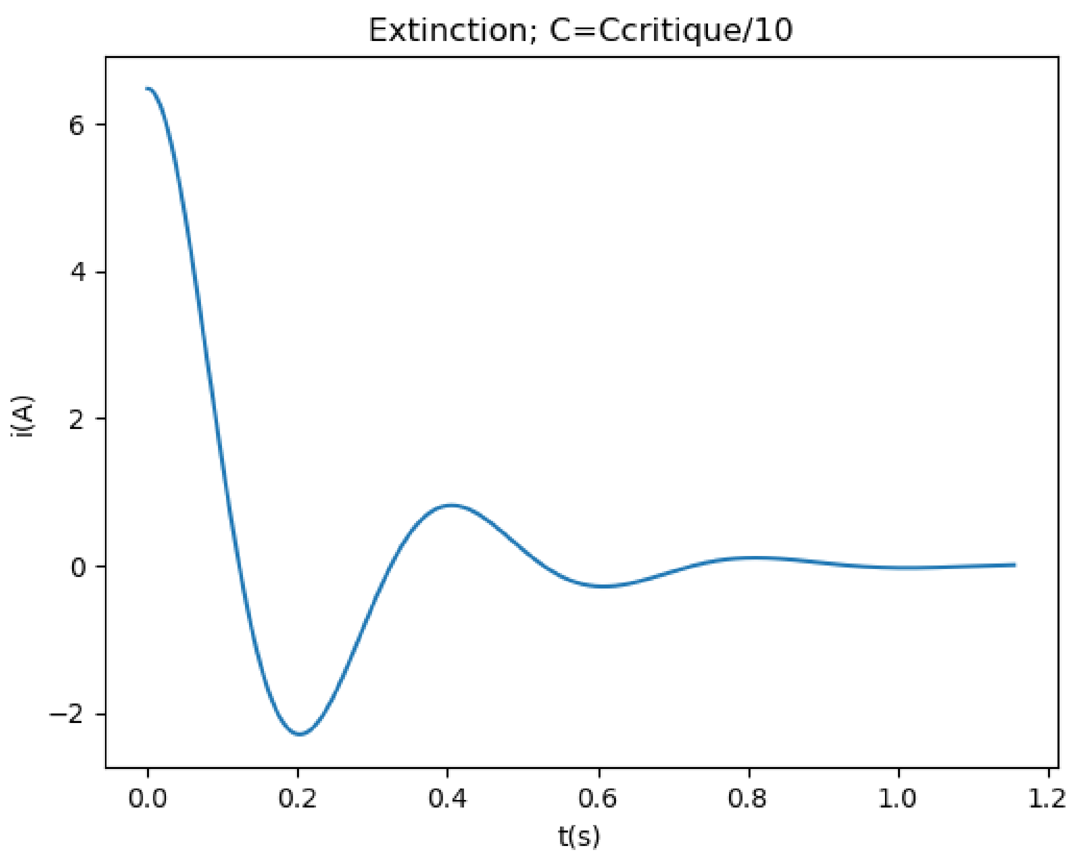
[7]:
```
C=10*Ccritique
def deriveeseconde(i,didt,t):
    return (-(Rm*C)*didt-i)/(C*Lm)
w0=np.sqrt(1/(Lm*C))
T0=2*np.pi/w0
h=T0/1000
tmax=3*T0
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_i=np.zeros(N+1)
tab_didt=np.zeros(N+1)
```

```
tab_t[0],tab_i[0],tab_didt[0]=0,E/(R+Rm),0
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_i[n+1]=tab_i[n]+h*tab_didt[n]
    tab_didt[n+1]=tab_didt[n]+h*deriveeseconde(tab_i[n],tab_didt[n],tab_t[n])
plt.figure()
plt.plot(tab_t,tab_i)
plt.xlabel("t(s)")
plt.ylabel("i(A)")
plt.title("Extinction; C=Ccritique*10")
plt.show()
```



[8]:
```
C=Ccritique/10
def deriveeseconde(i,didt,t):
    return (-(Rm*C)*didt-i)/(C*Lm)
w0=np.sqrt(1/(Lm*C))
T0=2*np.pi/w0
h=T0/1000
tmax=3*T0
N=int(tmax/h)
```

```
tab_t=np.zeros(N+1)
tab_i=np.zeros(N+1)
tab_didt=np.zeros(N+1)
tab_t[0],tab_i[0],tab_didt[0]=0,E/(R+Rm),0
for n in range(N):
    tab_t[n+1]=tab_t[n]+h
    tab_i[n+1]=tab_i[n]+h*tab_didt[n]
    tab_didt[n+1]=tab_didt[n]+h*deriveeseconde(tab_i[n],tab_didt[n],tab_t[n])
plt.figure()
plt.plot(tab_t,tab_i)
plt.xlabel("t(s)")
plt.ylabel("i(A)")
plt.title("Extinction; C=Ccritique/10")
plt.show()
```



[ ]:

```
```