

Circuit RC soumis à différentes tensions

```
[1]: import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
```

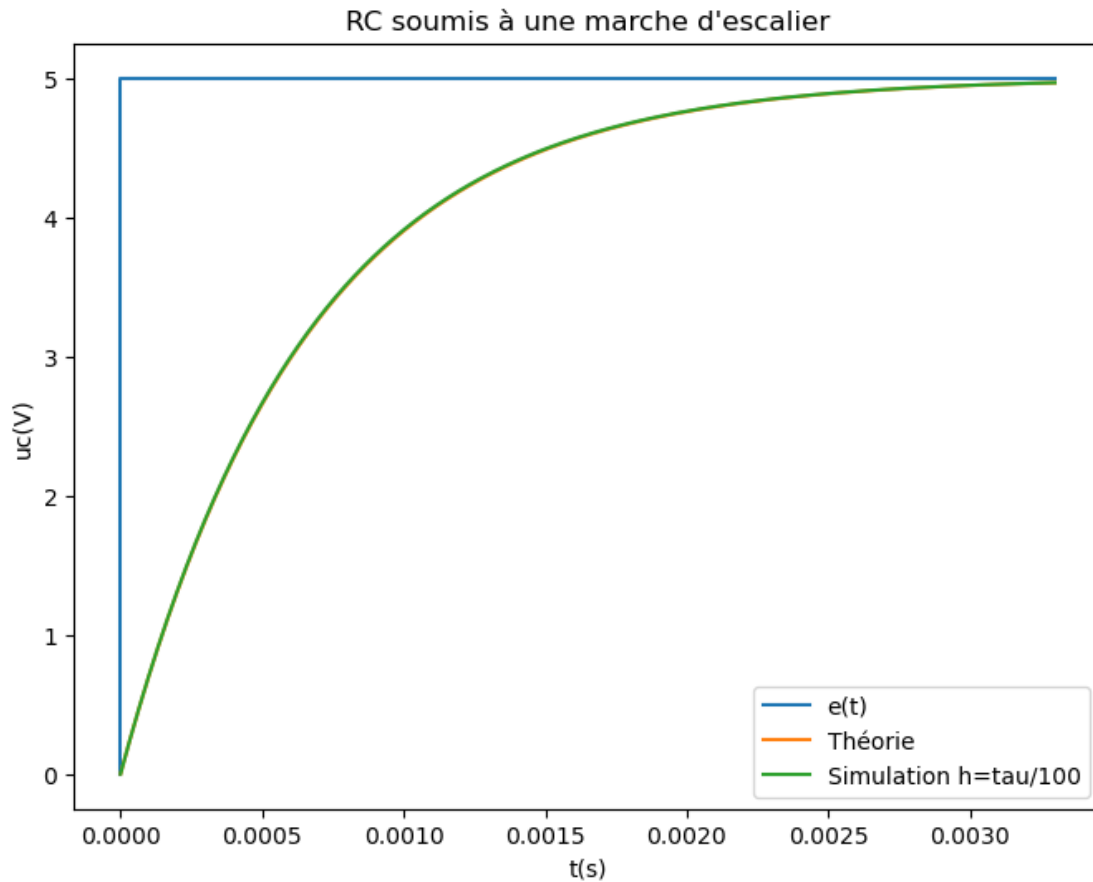
```
[2]: R,C=2200,300e-9
tau=R*C
```

1 Réponse à une marche d'escalier

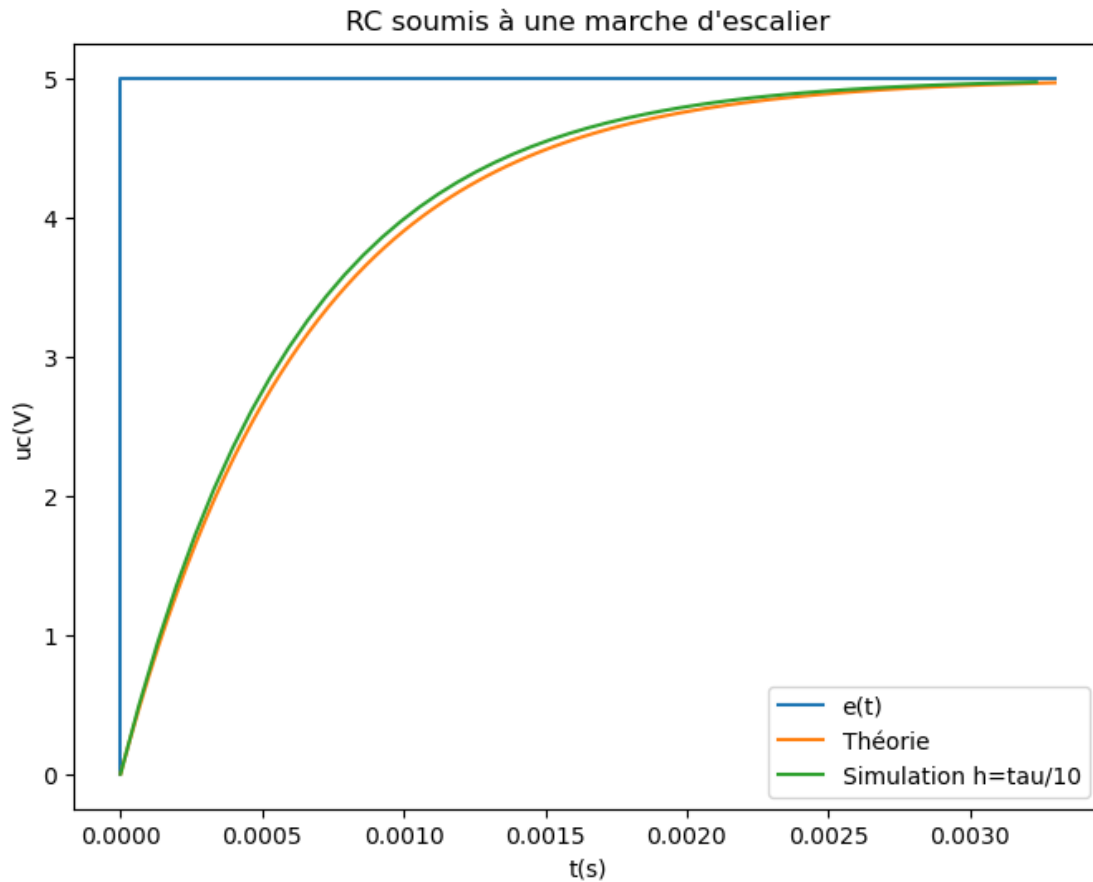
```
[3]: E=5
def derivee(uc,t):
    ducdt=(E-uc)/tau
    return ducdt
```

```
[4]: tab_tth=np.linspace(0,5*tau,100)
tab_uth=E*(1-np.exp(-tab_tth/tau))
```

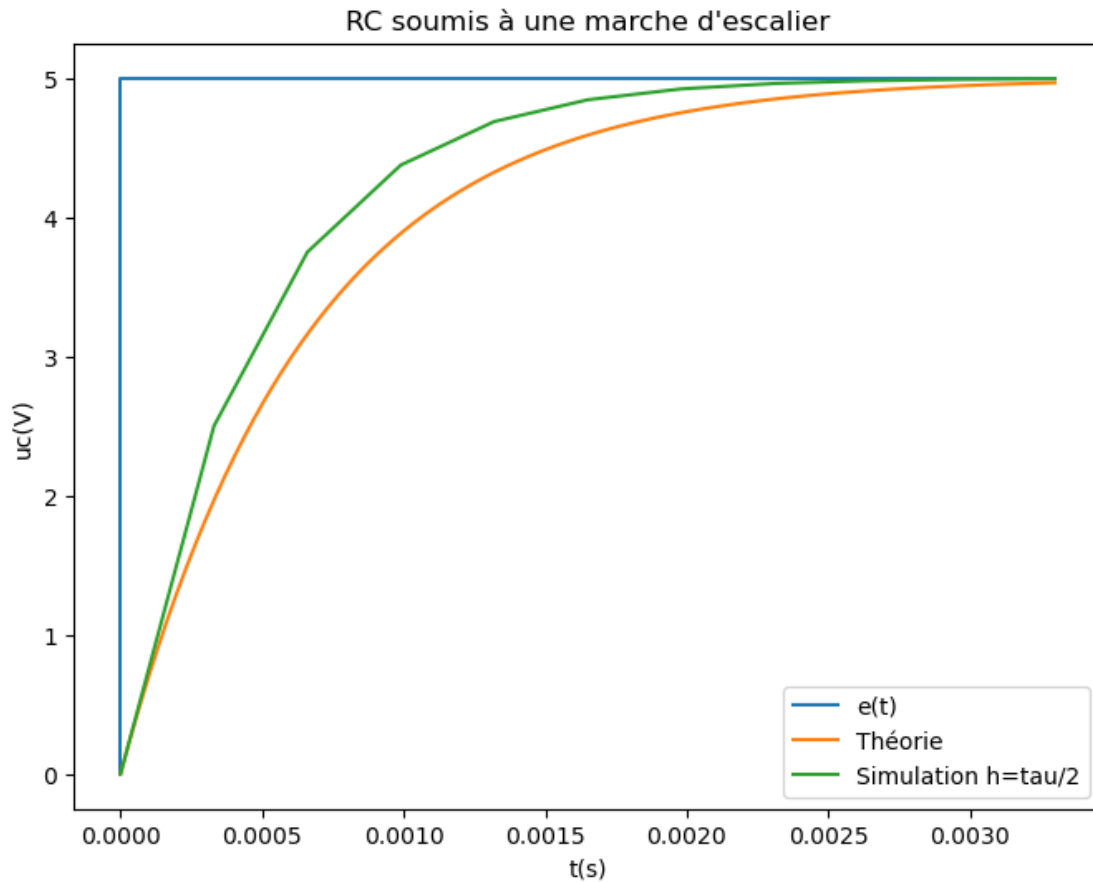
```
[5]: h=tau/100
tmax=5*tau
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_t[0],tab_uc[0]=0,0
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
plt.figure(figsize=(8,6))
plt.plot([0,0,tmax],[0,E,E],label="e(t)")
plt.plot(tab_tth,tab_uth,label="Théorie")
plt.plot(tab_t,tab_uc,label="Simulation h=tau/100")
plt.xlabel("t(s)")
plt.ylabel("uc(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()
```



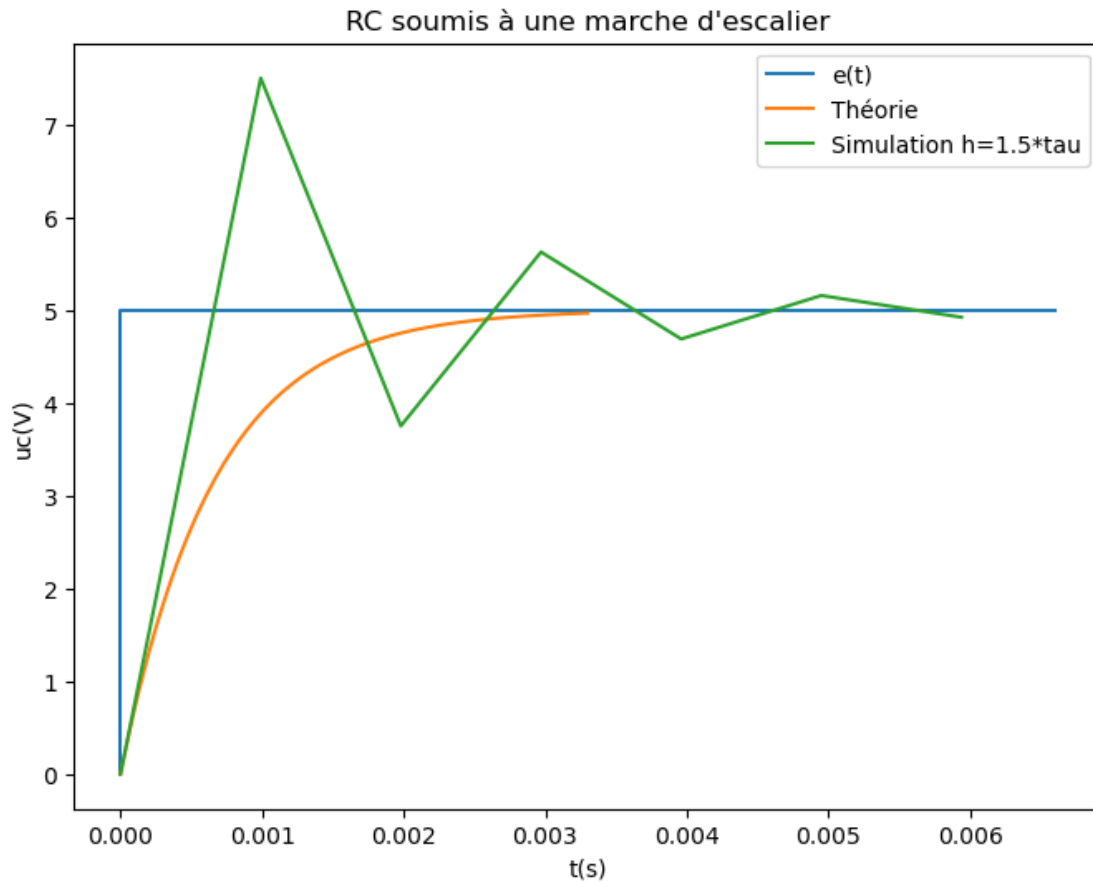
```
[6]: h=tau/10
tmax=5*tau
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_t[0],tab_uc[0]=0,0
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
plt.figure(figsize=(8,6))
plt.plot([0,0,tmax],[0,E,E],label="e(t)")
plt.plot(tab_tth,tab_uth,label="Théorie")
plt.plot(tab_t,tab_uc,label="Simulation h=tau/10")
plt.xlabel("t(s)")
plt.ylabel("uc(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()
```



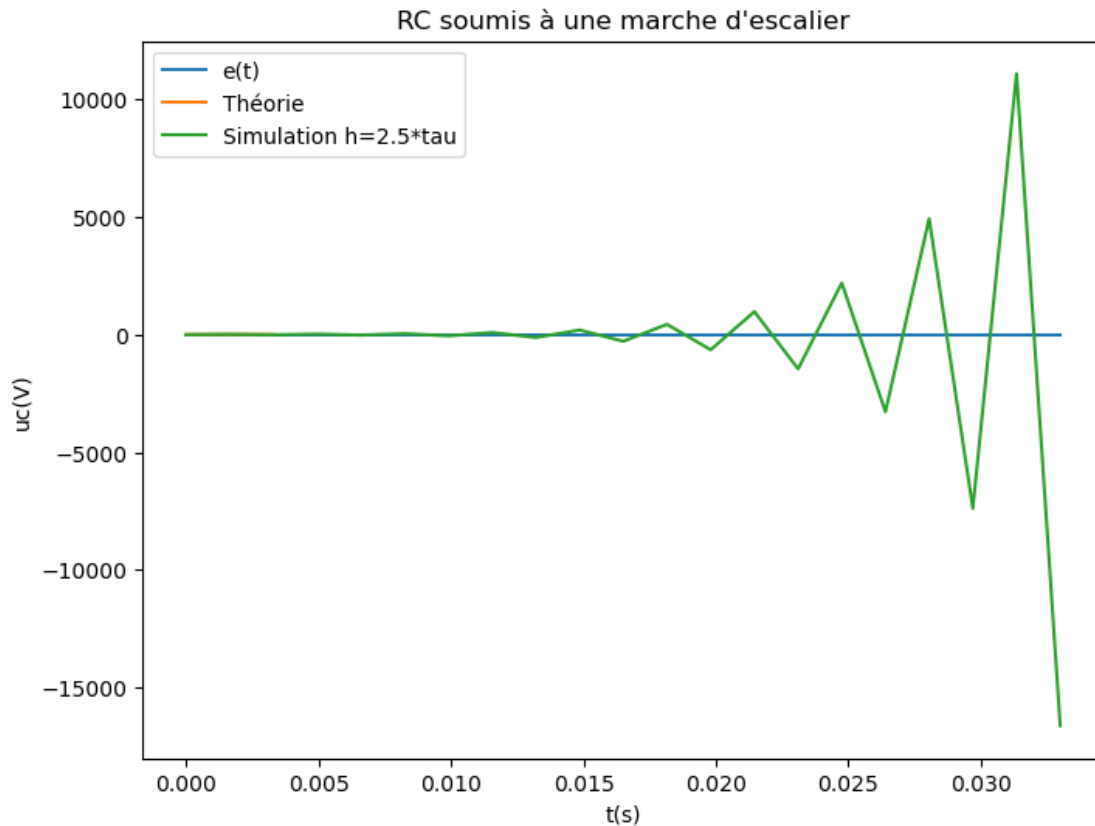
```
[7]: h=tau/2
tmax=5*tau
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_t[0],tab_uc[0]=0,0
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
plt.figure(figsize=(8,6))
plt.plot([0,0,tmax],[0,E,E],label="e(t)")
plt.plot(tab_tth,tab_uth,label="Théorie")
plt.plot(tab_t,tab_uc,label="Simulation h=tau/2")
plt.xlabel("t(s)")
plt.ylabel("uc(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()
```



```
[8]: h=1.5*tau
tmax=10*tau
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_t[0],tab_uc[0]=0,0
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
plt.figure(figsize=(8,6))
plt.plot([0,0,tmax],[0,E,E],label="e(t)")
plt.plot(tab_tth,tab_uth,label="Théorie")
plt.plot(tab_t,tab_uc,label="Simulation h=1.5*tau")
plt.xlabel("t(s)")
plt.ylabel("uc(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()
```



```
[9]: h=2.5*tau
tmax=50*tau
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_t[0],tab_uc[0]=0,0
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
plt.figure(figsize=(8,6))
plt.plot([0,0,tmax],[0,E,E],label="e(t)")
plt.plot(tab_tth,tab_uth,label="Théorie")
plt.plot(tab_t,tab_uc,label="Simulation h=2.5*tau")
plt.xlabel("t(s)")
plt.ylabel("uc(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()
```



- pour $h < \tau$, la courbe a la bonne forme, d'autant plus précise que le pas est petit
- pour $\tau < h < 2\tau$, la courbe se met à osciller mais converge vers la bonne valeur
- pour $h > 2\tau$, la courbe diverge

2 Réponse à un créneau

```
[10]: E=5
def e(t):
    """ valeur du créneau de période T au temps t """
    if (t%T)<T/2:
        return E
    else:
        return -E
```

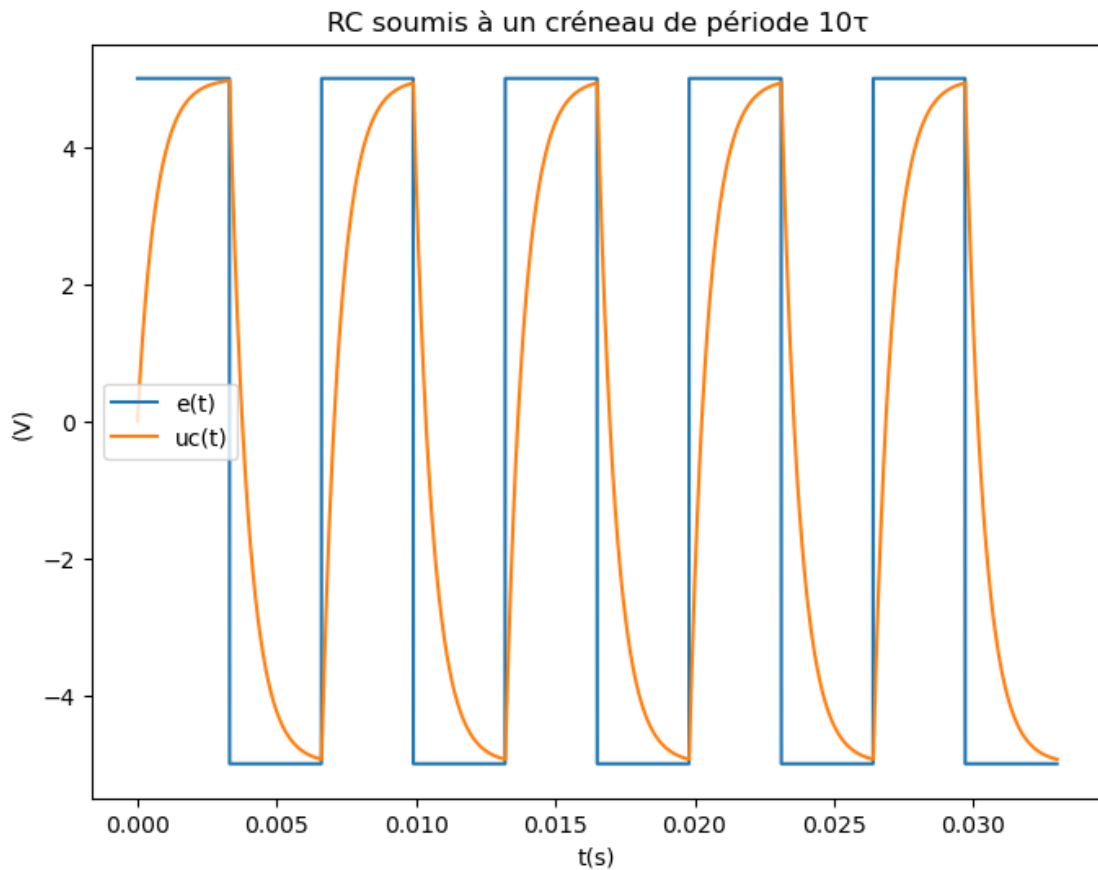
```
[11]: def derivee(uc,t):
        ducdt=(e(t)-uc)/tau
        return ducdt
```

```
[12]: h=tau/100
        T=10*tau
```

```

tmax=5*T
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_e=np.zeros(N+1)
tab_t[0],tab_uc[0],tab_e[0]=0,0,e(0)
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
    tab_e[i+1]=e(tab_t[i+1])
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_e,label="e(t)")
plt.plot(tab_t,tab_uc,label="uc(t)")
plt.xlabel("t(s)")
plt.ylabel("(V)")
plt.legend()
plt.title("RC soumis à un créneau de période 10 ")
plt.show()

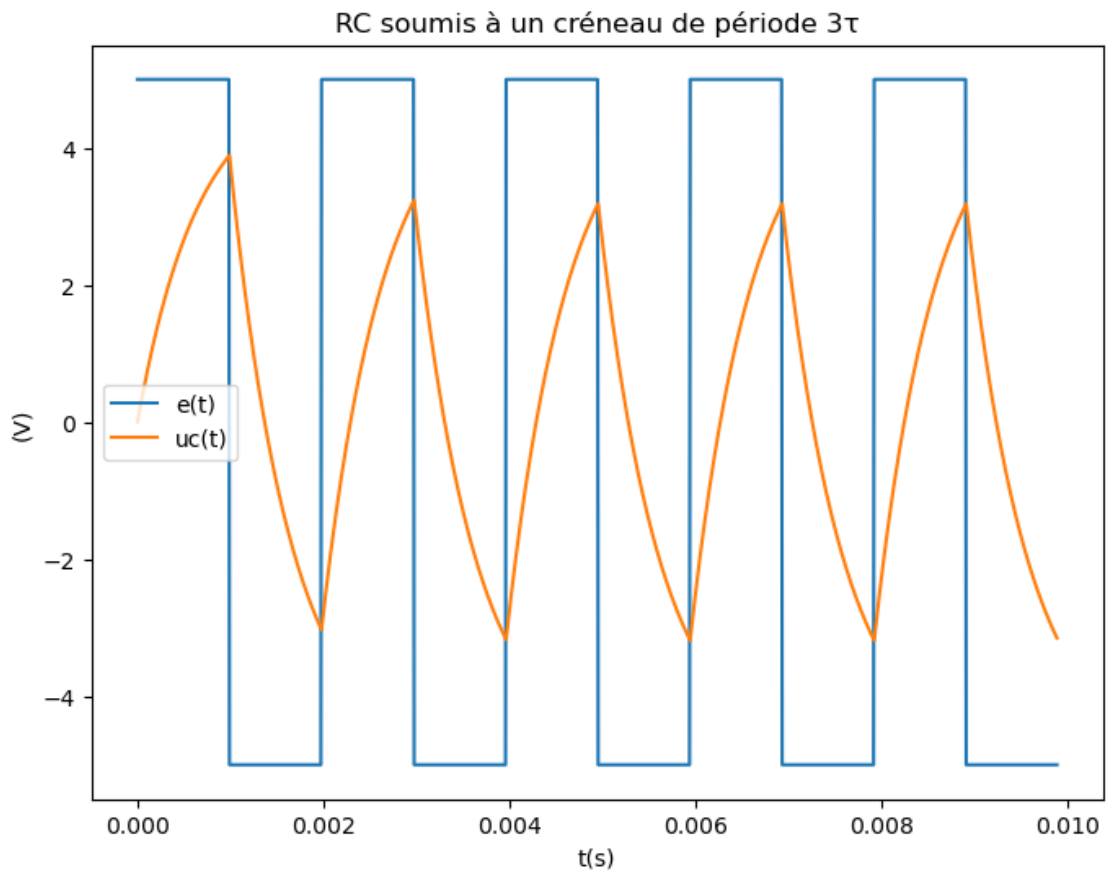
```



```

[13]: h=tau/100
      T=3*tau
      tmax=5*T
      N=int(tmax/h)
      tab_t=np.zeros(N+1)
      tab_uc=np.zeros(N+1)
      tab_e=np.zeros(N+1)
      tab_t[0],tab_uc[0],tab_e[0]=0,0,e(0)
      for i in range(N):
          tab_t[i+1]=tab_t[i]+h
          tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
          tab_e[i+1]=e(tab_t[i+1])
      plt.figure(figsize=(8,6))
      plt.plot(tab_t,tab_e,label="e(t)")
      plt.plot(tab_t,tab_uc,label="uc(t)")
      plt.xlabel("t(s)")
      plt.ylabel("(V)")
      plt.legend()
      plt.title("RC soumis à un créneau de période 3 ")
      plt.show()

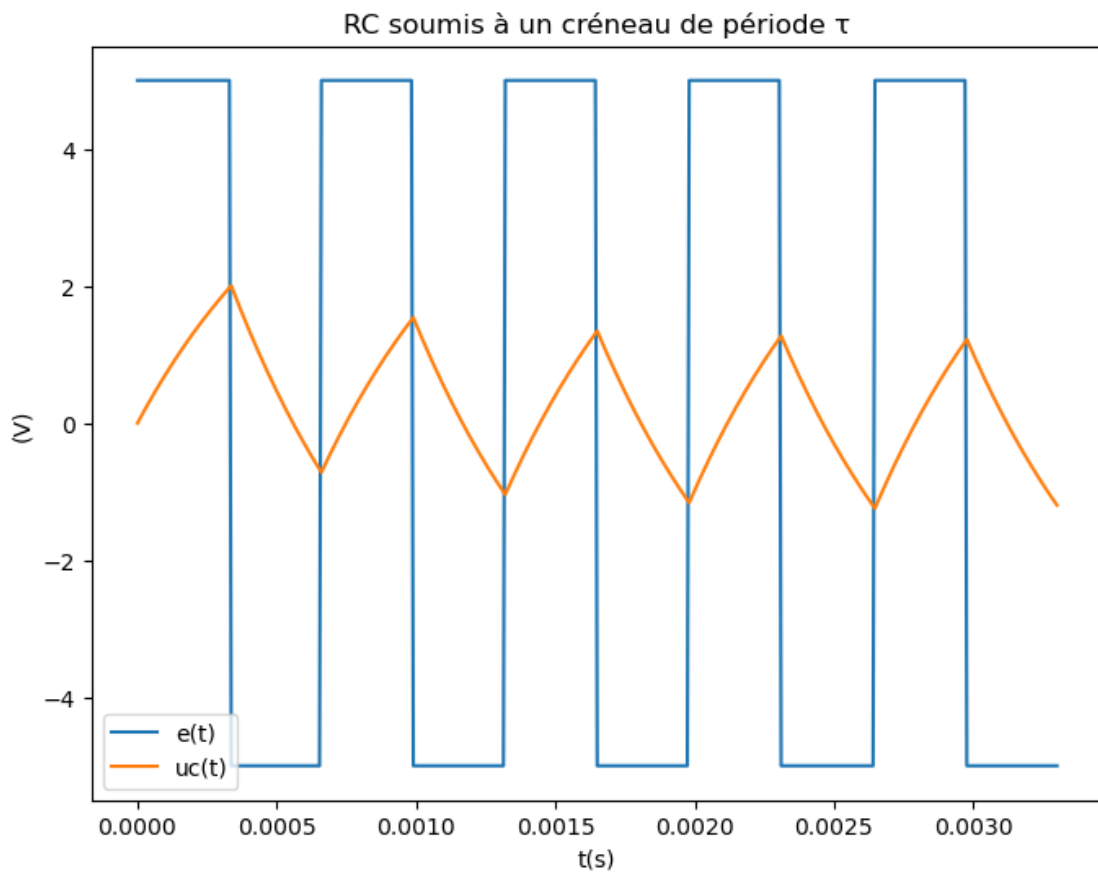
```



```

[14]: h=tau/100
T=tau
tmax=5*T
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_e=np.zeros(N+1)
tab_t[0],tab_uc[0],tab_e[0]=0,0,e(0)
for i in range(N):
    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
    tab_e[i+1]=e(tab_t[i+1])
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_e,label="e(t)")
plt.plot(tab_t,tab_uc,label="uc(t)")
plt.xlabel("t(s)")
plt.ylabel("(V)")
plt.legend()
plt.title("RC soumis à un créneau de période ")
plt.show()

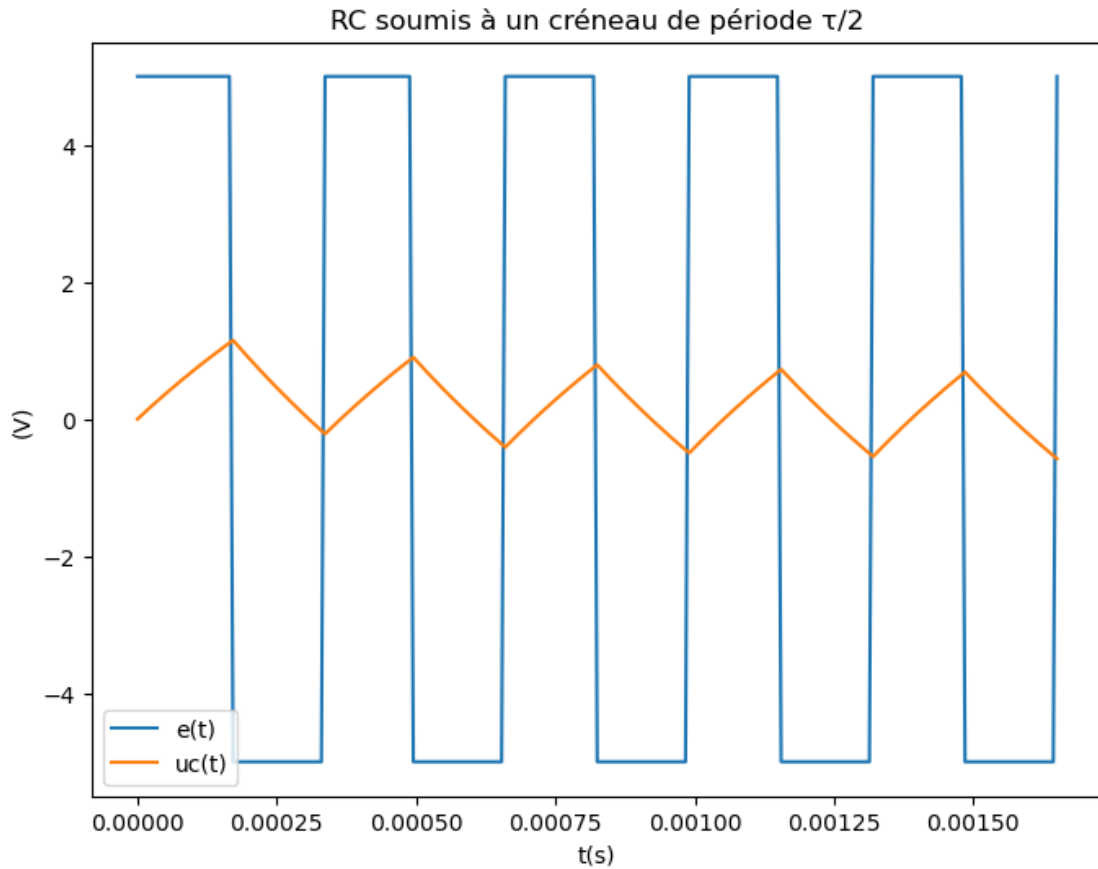
```



```

[15]: h=tau/100
      T=0.5*tau
      tmax=5*T
      N=int(tmax/h)
      tab_t=np.zeros(N+1)
      tab_uc=np.zeros(N+1)
      tab_e=np.zeros(N+1)
      tab_t[0],tab_uc[0],tab_e[0]=0,0,e(0)
      for i in range(N):
          tab_t[i+1]=tab_t[i]+h
          tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
          tab_e[i+1]=e(tab_t[i+1])
      plt.figure(figsize=(8,6))
      plt.plot(tab_t,tab_e,label="e(t)")
      plt.plot(tab_t,tab_uc,label="uc(t)")
      plt.xlabel("t(s)")
      plt.ylabel("(V)")
      plt.legend()
      plt.title("RC soumis à un créneau de période /2")
      plt.show()

```



3 Tension sinusoïdale

```
[28]: E=5
def e(t):
    return E*np.cos(2*np.pi*f*t)
```

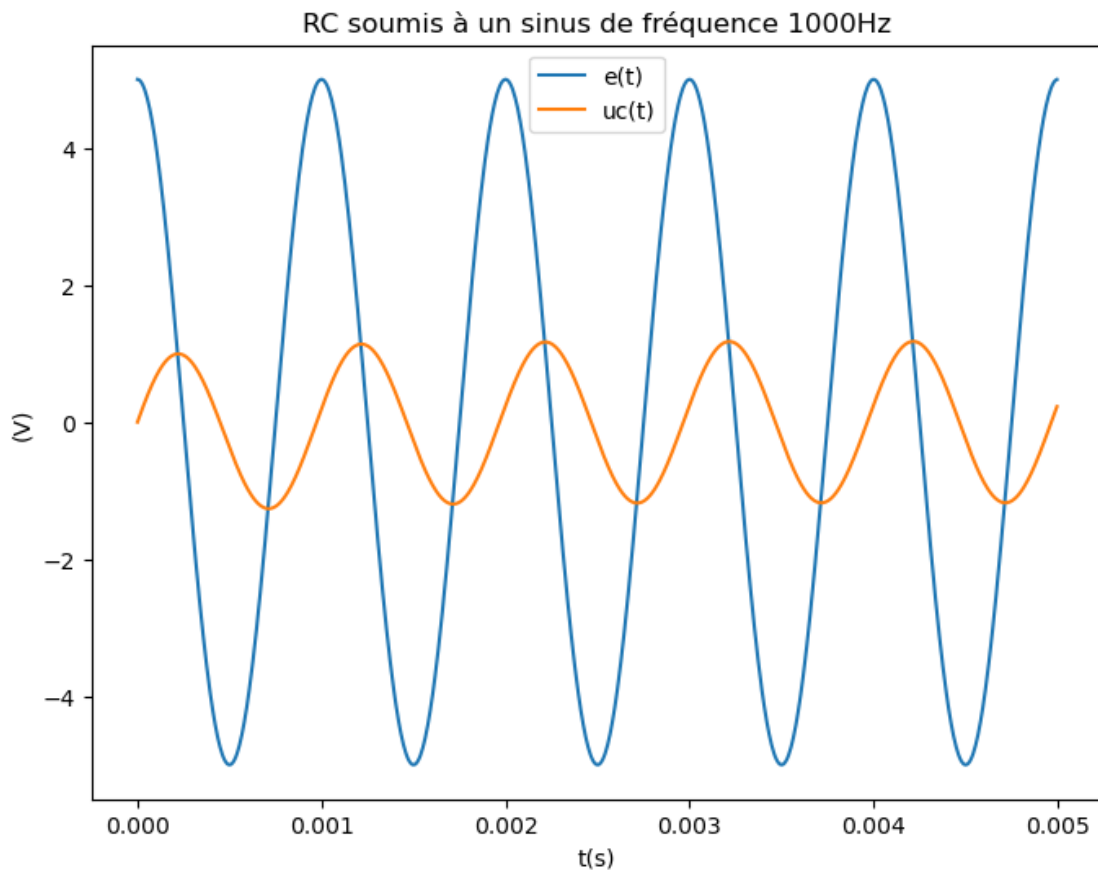
```
[29]: def derivee(uc,t):
    ducdt=(e(t)-uc)/tau
    return ducdt
```

```
[34]: h=tau/100
f=1000
tmax=5/f
N=int(tmax/h)
tab_t=np.zeros(N+1)
tab_uc=np.zeros(N+1)
tab_e=np.zeros(N+1)
tab_t[0],tab_uc[0],tab_e[0]=0,0,e(0)
for i in range(N):
```

```

    tab_t[i+1]=tab_t[i]+h
    tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
    tab_e[i+1]=e(tab_t[i+1])
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_e,label="e(t)")
plt.plot(tab_t,tab_uc,label="uc(t)")
plt.xlabel("t(s)")
plt.ylabel("V")
plt.legend()
plt.title("RC soumis à un sinus de fréquence "+str(f)+"Hz")
plt.show()

```



4 Utilisation de odeint

```

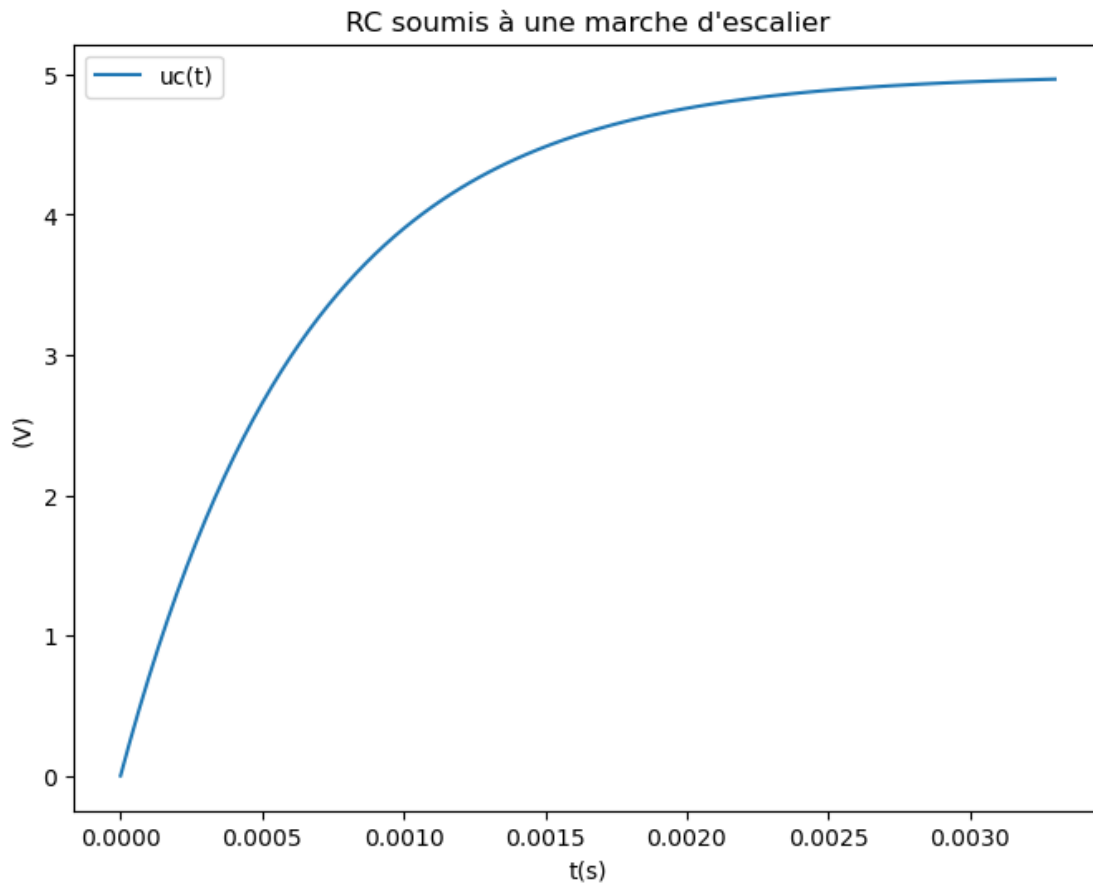
[16]: def derivee(inconnues,t):
    uc=inconnues[0]
    ducdt=(E-uc)/tau
    return [ducdt]

```

```

tab_t=np.linspace(0,5*tau,100)
ci=[0]
solution=odeint(derivee,ci,tab_t)
tab_uc=solution[:,0]
plt.figure(figsize=(8,6))
plt.plot(tab_t,tab_uc,label="uc(t)")
plt.xlabel("t(s)")
plt.ylabel("(V)")
plt.legend()
plt.title("RC soumis à une marche d'escalier")
plt.show()

```



5 Précision de la méthode d'Euler

```

[17]: E=5
def derivee(uc,t):
    ducdt=(E-uc)/tau
    return ducdt

```

```

Np=5
tab_h=np.zeros(Np)
tab_erreur=np.zeros(Np)
for p in range(1,Np+1):
    h=tau/10**p
    tab_h[p-1]=h
    tmax=5*tau
    N=int(tmax/h)
    tab_t=np.zeros(N+1)
    tab_uc=np.zeros(N+1)
    tab_ecart=np.zeros(N+1)
    tab_t[0],tab_uc[0],tab_ecart[0]=0,0,0
    for i in range(N):
        tab_t[i+1]=tab_t[i]+h
        tab_uc[i+1]=tab_uc[i]+h*derivee(tab_uc[i],tab_t[i])
        tab_ecart[i+1]=np.abs(tab_uc[i+1]-E*(1-np.exp(-tab_t[i+1]/tau)))
    tab_erreur[p-1]=max(tab_ecart)
print(tab_h)
print(tab_erreur)

```

```

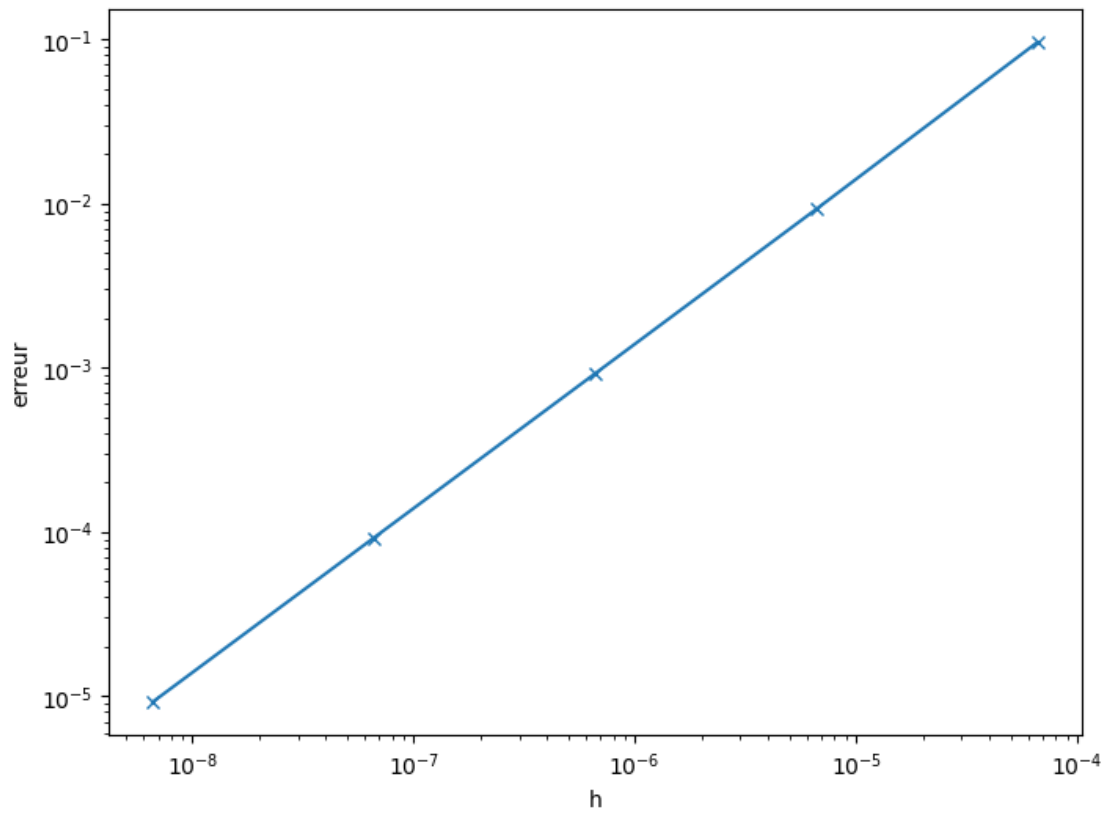
[6.6e-05 6.6e-06 6.6e-07 6.6e-08 6.6e-09]
[9.60050054e-02 9.23549949e-03 9.20082002e-04 9.19736928e-05
 9.19702886e-06]

```

```

[18]: plt.figure(figsize=(8,6))
plt.plot(tab_h,tab_erreur,marker="x")
plt.xscale("log")
plt.yscale("log")
plt.xlabel("h")
plt.ylabel("erreur")
plt.show()

```



[]: