RLC Eq

January 25, 2021

1 Dampes RLC Circuit

1.0.1 solveing the diffirential equation

---

[1]:
```
import sympy as sp
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
%matplotlib inline
sp.init_printing()
```

[2]:
```
# Create symbols objects
q= sp.symbols("q", cls=sp.Function)
L, C, R, t, w, wp , Q= sp.symbols("L C_0 R t \\omega \\omega' Q", real=True)

# Create Eq. object
eq1 = sp.Eq(L * q(t).diff(t,2) + R * q(t).diff(t) + q(t)/C, 0)
eq1
```

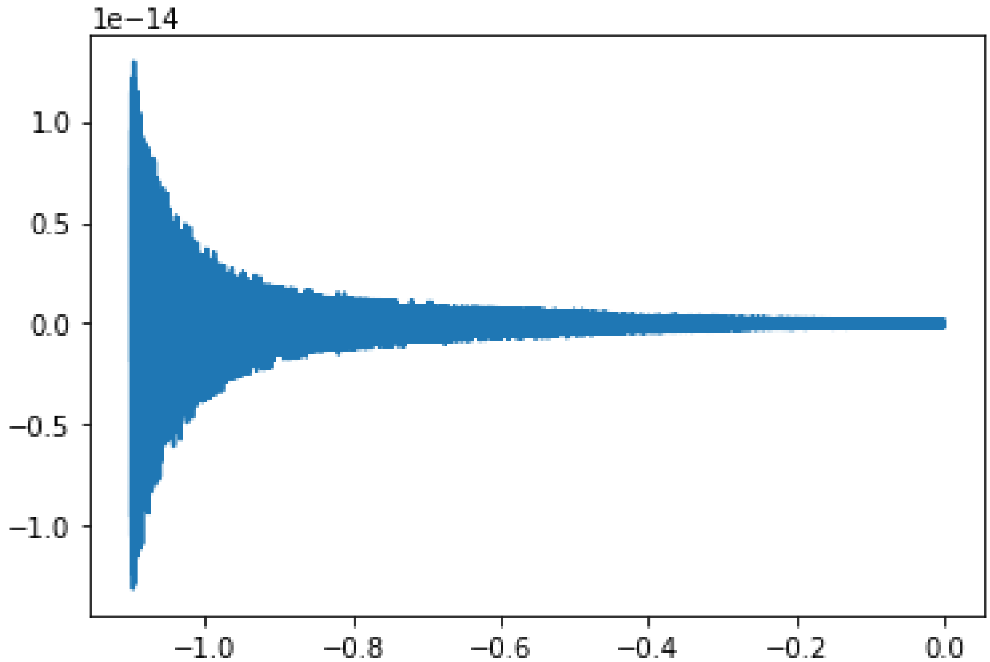
[2]:
$$L\frac{d^2}{dt^2}q(t) + R\frac{d}{dt}q(t) + \frac{q(t)}{C_0} = 0$$

[6]:
```
# Solving for q(t)
eq2 = sp.dsolve(eq1, q(t)).simplify()
# eq2 = eq2.subs({C:1, L:1, R:1})

# Find the constants value
sol = sp.solve([eq2.rhs.subs(t, 0)-Q, eq2.rhs.diff(t, substitute=0)-1])

eq2
```

[6]:
$$q(t) = C_1e^{-t\sqrt{-\frac{1}{C_0L}}} + C_2e^{t\sqrt{-\frac{1}{C_0L}}}$$

[122]:
```
func = sp.lambdify(t, eq3.rhs, "numpy")
xaxis = np.arange(0, 1.1, 0.000001)

plt.plot(-1*xaxis, func(xaxis).imag)
```

[122]: [<matplotlib.lines.Line2D at 0x7faa4a0034c0>]



[2]:
```
# Create symbols objects
q= sp.symbols("q", cls=sp.Function)
L, C, t, w, wp, Q = sp.symbols("L C_0 t \\omega \\omega' Q", real=True)

# Create Eq. object
eq1 = sp.Eq(L * q(t).diff(t,2) + q(t)/C, 0)
eq1
```

[2]:
$$L\frac{d^2}{dt^2}q(t) + \frac{q(t)}{C_0} = 0$$

1.1 Our conditions are

$ q(0)=C, , q'(0)=1 $

[3]:
```
# Solving for q(t)
eq2 = sp.dsolve(eq1, q(t))

sol = sp.solve([eq2.rhs.subs(t,0)-Q,
                eq2.rhs.diff(t).subs(t,0)])

# eq2 = eq2.subs(sol[0]).simplify()
```

```
eq2
```

[3]:

$$q(t) = C_1 e^{-t\sqrt{-\frac{1}{C_0 L}}} + C_2 e^{t\sqrt{-\frac{1}{C_0 L}}}$$

[8]:

```
eq3 = sp.lambdify(t, eq2.subs({C:1, L:1, Q:1}).rhs, "numpy")

data = np.arange(0,float(4 * sp.pi), 0.1)
plt.plot(data, eq3(data))
```

[8]: [<matplotlib.lines.Line2D at 0x7ff4b59e0490>]