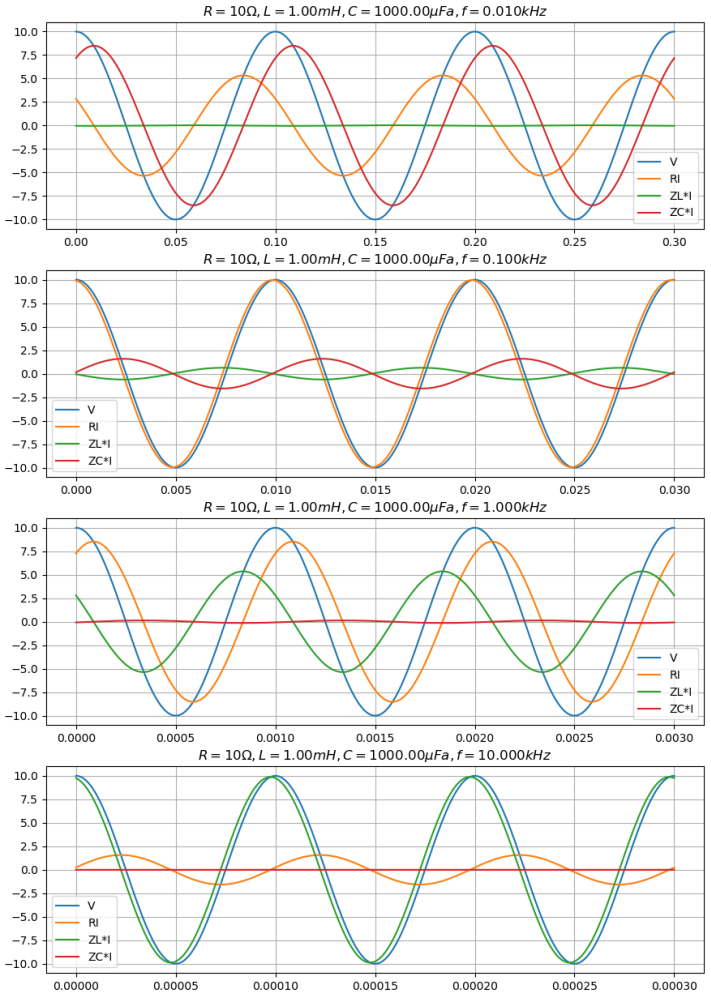
# Aula 10

Resolução de (sistemas) de equações lineares

(parte 2): mais alguns problemas que dão origem a equações lineares.



# Resolver é fácil...

Como os métodos de resolução estão bem padronizados, podemos quase sempre recorrer a funções pré-existentes:

```
x=np.linalg.solve(M,b)
```

Escolhendo outras opções se o problema for particular (exemplo matrizes esparsas).

A dificuldade está na construção do sistema de equações adequado para o problema.

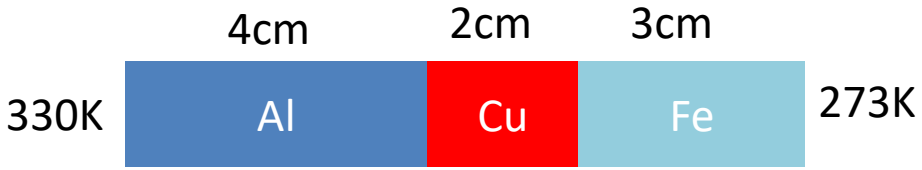
# Calcular a distribuição de temperatura numa barra composta 1D

| Metal | Condutividade $Wm^{-1}K^{-1}$ |
|---|---|
| Aluminio | 237 |
| Cobre | 401 |
| Ferro | 80 |

Admite-se que não existe fluxo lateral de calor.

Na direção longitudinal vale a lei de Fourier:

Fluxo de calor=-$k\frac{\partial T}{\partial x}$




Admite-se que a barra atingiu o equilíbrio térmico (T=const em cada ponto), logo Fluxo independente de $x$. Logo, em cada metal ($k = const$):

$$\frac{\partial T}{\partial x} = const$$

Isto é a temperatura varia linearmente, seguindo uma linha quebrada, com quebras nas transições entre materiais.

# Discussão

A equação da condução de calor, de Fourier, é uma equação diferencial:

$$\frac{\partial T}{\partial t} = -\nabla.(-k\nabla T)$$

reduzindo-se, o caso estacionário, a:

$$\nabla^2 T = 0$$

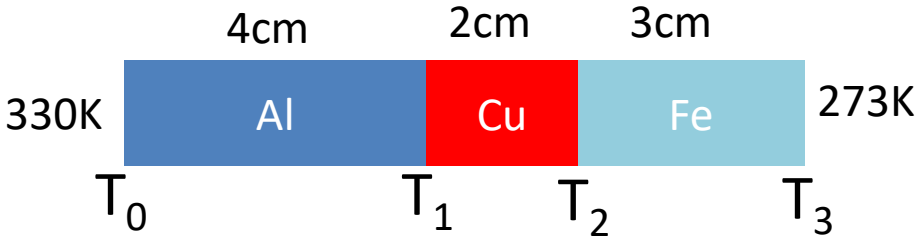
E, no caso unidimensional:

$$-\frac{\partial}{\partial x}\left(-k\frac{\partial T}{\partial x}\right) = 0$$

No caso em que $k$ é constante por troços, obtém-se (de forma exata) um sistema de equações lineares algébricas.

O caso geral, em que é preciso resolver a equação diferencial, será tratado mais tarde.

# Condução



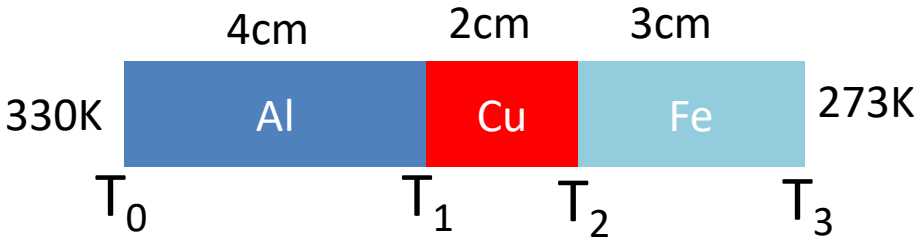

$$Fluxo\ de\ calor = k_{Al}\left(-\frac{\partial T}{\partial x}\right) \equiv k_{Al}\frac{T_0 - T_1}{x_1 - x_0} = k_{Cu}\frac{T_1 - T_2}{x_2 - x_1} = k_{Fe}\frac{T_2 - T_3}{x_3 - x_2}$$

$$= const$$

Sistema de equações lineares algébricas:

$$\begin{cases} -\frac{k_{Al}}{x_1 - x_0}T_1 - \frac{k_{Cu}}{x_2 - x_1}T_1 + \frac{k_{Cu}}{x_2 - x_1}T_2 = -\frac{k_{Al}}{x_1 - x_0}T_0 \\ \frac{k_{Cu}}{x_2 - x_1}T_1 - \frac{k_{Cu}}{x_2 - x_1}T_2 - \frac{k_{Fe}}{x_3 - x_2}T_2 = -\frac{k_{Fe}}{x_3 - x_2}T_3 \end{cases}$$

# Forma matricial



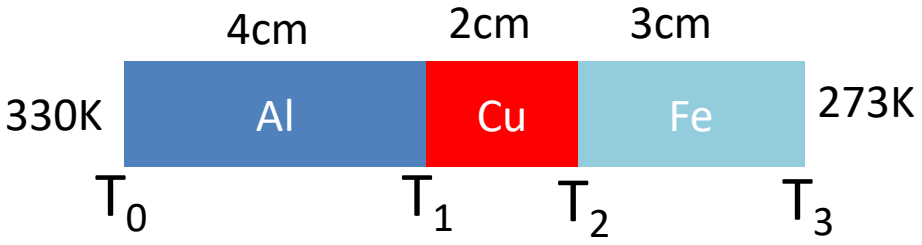

$$\begin{cases} -\dfrac{k_{Al}}{x_1 - x_0} T_1 - \dfrac{k_{Cu}}{x_2 - x_1} T_1 + \dfrac{k_{Cu}}{x_2 - x_1} T_2 = -\dfrac{k_{Al}}{x_1 - x_0} T_0 \\ \dfrac{k_{Cu}}{x_2 - x_1} T_1 - \dfrac{k_{Cu}}{x_2 - x_1} T_2 - \dfrac{k_{Fe}}{x_3 - x_2} T_2 = -\dfrac{k_{Fe}}{x_3 - x_2} T_3 \end{cases}$$

$$\begin{bmatrix} -\dfrac{k_{Al}}{x_1 - x_0} - \dfrac{k_{Cu}}{x_2 - x_1} & \dfrac{k_{Cu}}{x_2 - x_1} \\ \dfrac{k_{Cu}}{x_2 - x_1} & -\dfrac{k_{Cu}}{x_2 - x_1} - \dfrac{k_{Fe}}{x_3 - x_2} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} -\dfrac{k_{Al}}{x_1 - x_0} T_0 \\ -\dfrac{k_{Fe}}{x_3 - x_2} T_3 \end{bmatrix}$$

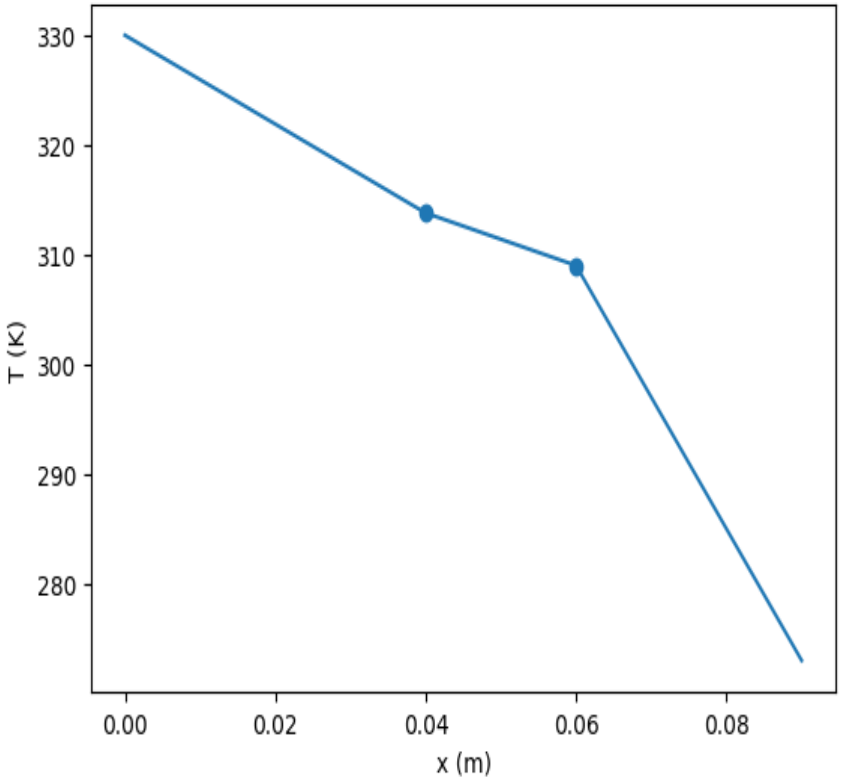
# Equilíbrio térmico numa barra

```
import numpy as np
import matplotlib.pyplot as plt
kAl=237;kCu=401;kFe=80;T0=330;T3=273
x0=0;x1=0.04;x2=0.06;x3=0.09
A=np.array([[-kAl/(x1-x0)-kCu/(x2-x1),kCu/(x2-x1)],\
    [kCu/(x2-x1),-kCu/(x2-x1)-kFe/(x3-x2)]],dtype=float)
b=np.array([-kAl/(x1-x0)*T0,-kFe/\
    (x3-x2)*T3],dtype=float)
T=np.linalg.solve(A,b)
print('T=',T)
plt.plot([x0,x1,x2,x3],[T0,T[0],T[1],T3])
plt.scatter([x1,x2],T)
plt.xlabel('x (m)')
plt.ylabel('T (K)')
```

# Solução

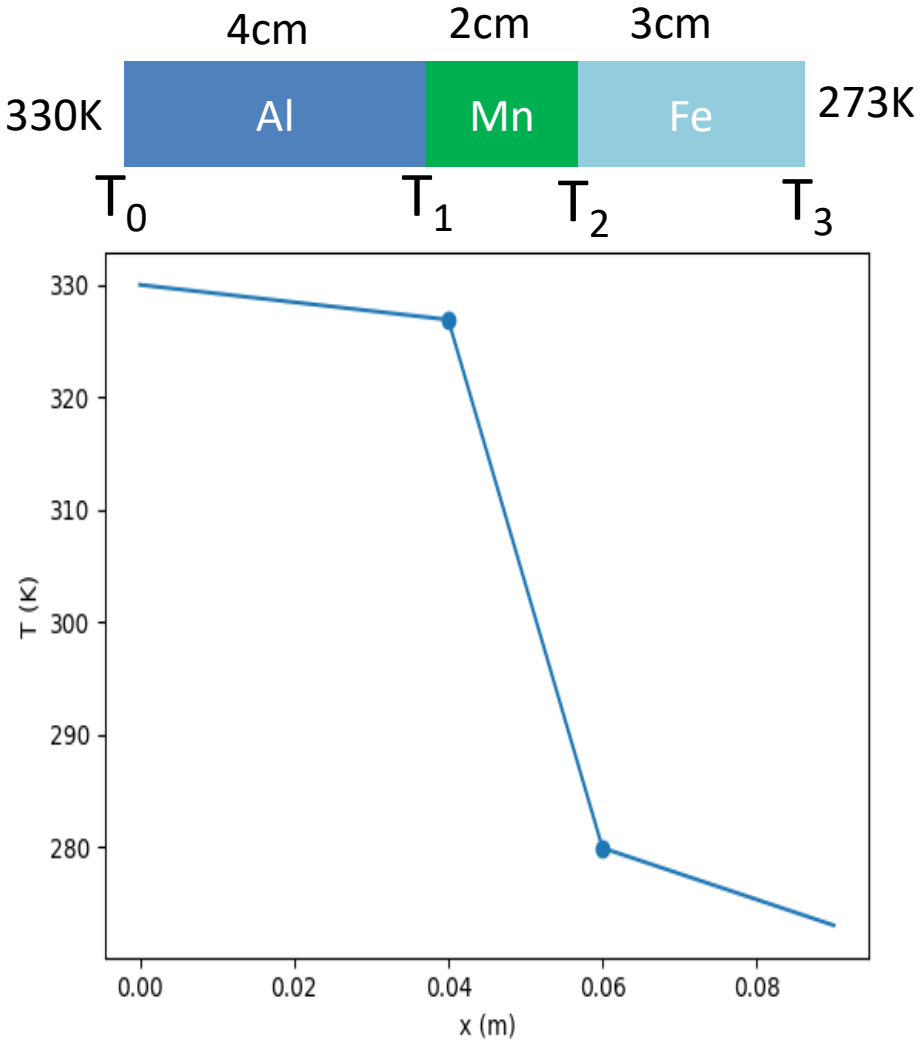



| Metal | Condutividade $Wm^{-1}K^{-1}$ |
|---|---|
| Aluminio | 237 |
| Cobre | 401 |
| Ferro | 80 |



# Solução 2

| Metal | Condutividade $Wm^{-1}K^{-1}$ |
|---|---|
| Aluminio | 237 |
| Manganésio | 7.81 |
| Ferro | 80 |



# Se a barra for constituída por n segmentos

$$\begin{bmatrix} -\frac{k_1}{\Delta x_1} - \frac{k_2}{\Delta x_2} & \frac{k_2}{\Delta x_2} & \dots & 0 & 0 \\ \frac{k_2}{\Delta x_2} & -\frac{k_2}{\Delta x_2} - \frac{k_3}{\Delta x_3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\frac{k_{n-2}}{\Delta x_{n-2}} - \frac{k_{n-1}}{\Delta x_{n-1}} & \frac{k_{n-1}}{\Delta x_{n-1}} \\ 0 & 0 & \dots & \frac{k_{n-1}}{\Delta x_{n-1}} & -\frac{k_{n-1}}{\Delta x_{n-1}} - \frac{k_n}{\Delta x_n} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \dots \\ T_{n-2} \\ T_{n-1} \end{bmatrix} = \begin{bmatrix} -\frac{k_1}{\Delta x_1} T_0 \\ 0 \\ \dots \\ 0 \\ -\frac{k_n}{\Delta x_n} T_n \end{bmatrix}$$

Trata-se de um sistema triadiagonal, cuja solução são as temperaturas nas interfaces $[T_1, \dots, T_{n-1}]$, dadas as temperaturas na fronteira $[T_0, T_n]$ e as condutividades $[k_1, \dots, k_n]$, com:

$$\Delta x_m = x_m - x_{m-1}$$

# Como temos tempo…

Vamos considerar um tema mais avançado, que também dá origem a equações lineares…

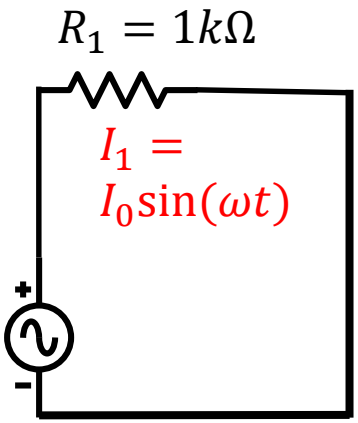
(tópico opcional)

# Circuitos com componentes lineares em corrente alterna (Resistor)

Equação algébrica linear:

Lei de Ohm

$$V_R = R_1 I$$

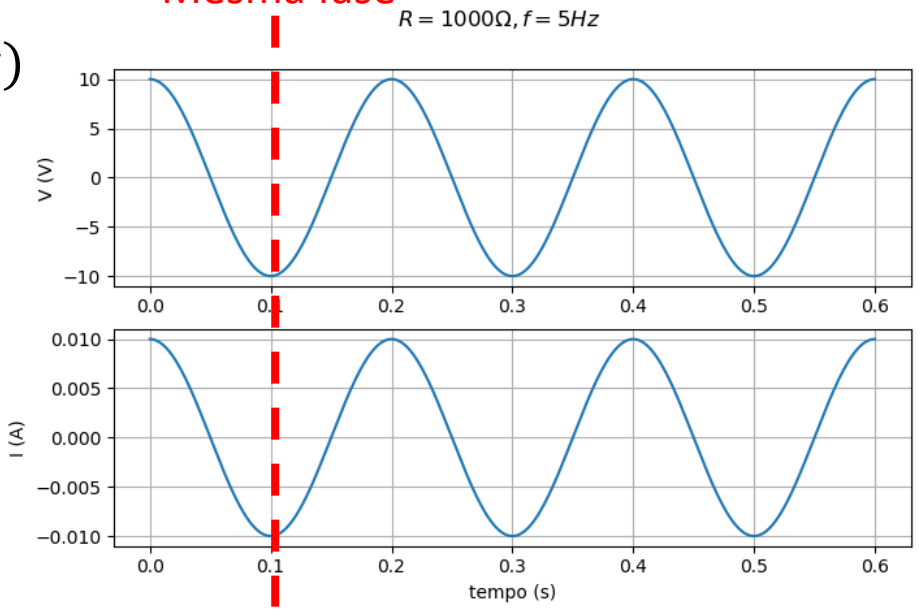



$$I_1 = \frac{V}{R_1} = I_0 \cos(\omega t)$$

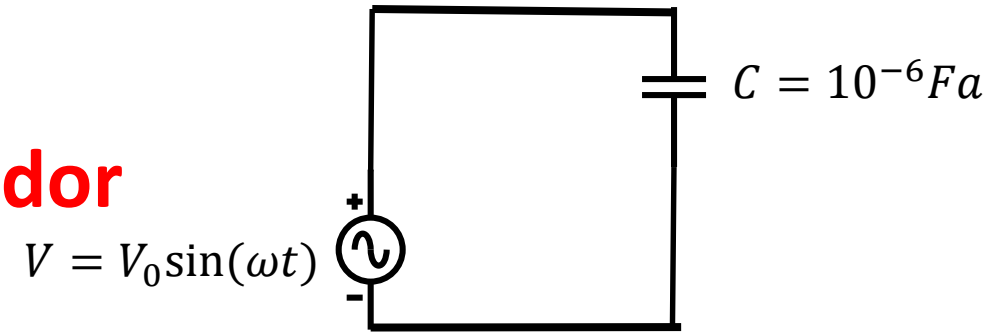
$$I_0 = \frac{V_0}{R_1} = 0.01A$$

$$\omega = 2\pi f$$

Mesma fase



# Circuito em corrente alterna num condensador




$$V_C = \frac{1}{C}\int I\, dt$$

Se

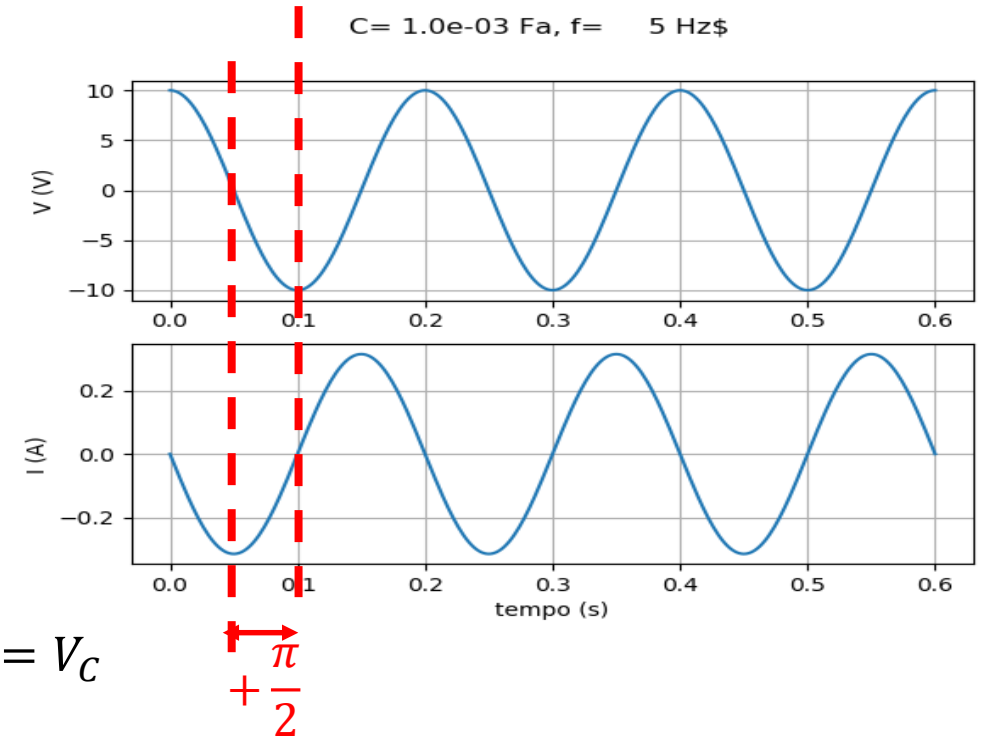
$$V_C = V_{C_0}\cos(\omega t)$$

Será:

$$I = -CV_0\omega \sin(\omega t)$$

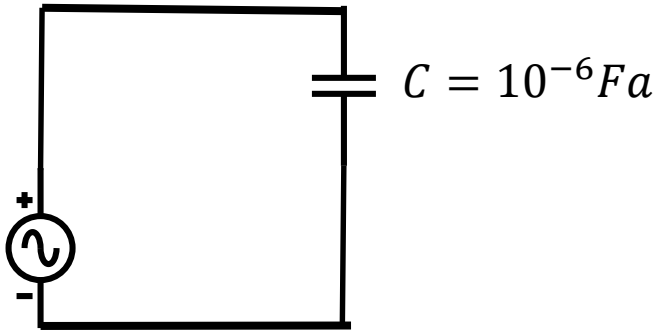
$$\frac{1}{C}\int -CV_0\omega \sin(\omega t)\ dt =$$

$$-\omega V_0 \int \sin(\omega t)\, dt = V_0 \cos(\omega t) = V_C$$




Desta forma, não se trata de uma relação linear (sin não é proporcional ao cos).

# Corrente alterna com números complexos




$\cos(\omega t) = Re\{e^{i\omega t}\}$, pois (fórmula de Euler)

$$e^{i\theta} = \cos(\theta) + i\,\sin(\theta)$$

$$V = Re\{V_0 e^{i\omega t}\} = \frac{1}{C}\int I\,dt \Longrightarrow I = -CV_0\,\sin(\omega t) = -CV_0\left(Re\{-ie^{i\omega t}\}\right)$$

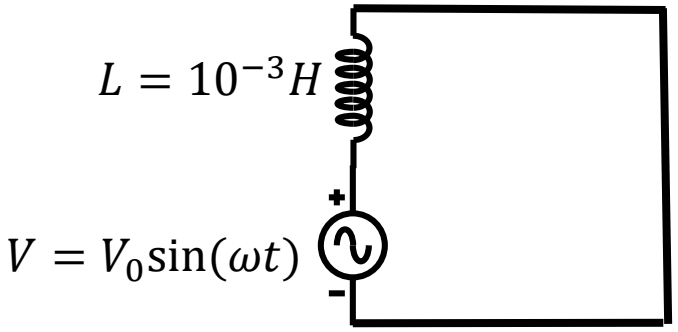
Ou seja:

$$V = -\frac{i}{\omega C} I = Z_C I$$

O que reproduz a lei de Ohm (lei linear) mas com uma impedância complexa

$$Z_C = -\frac{i}{\omega C}$$

# corrente alterna num indutor

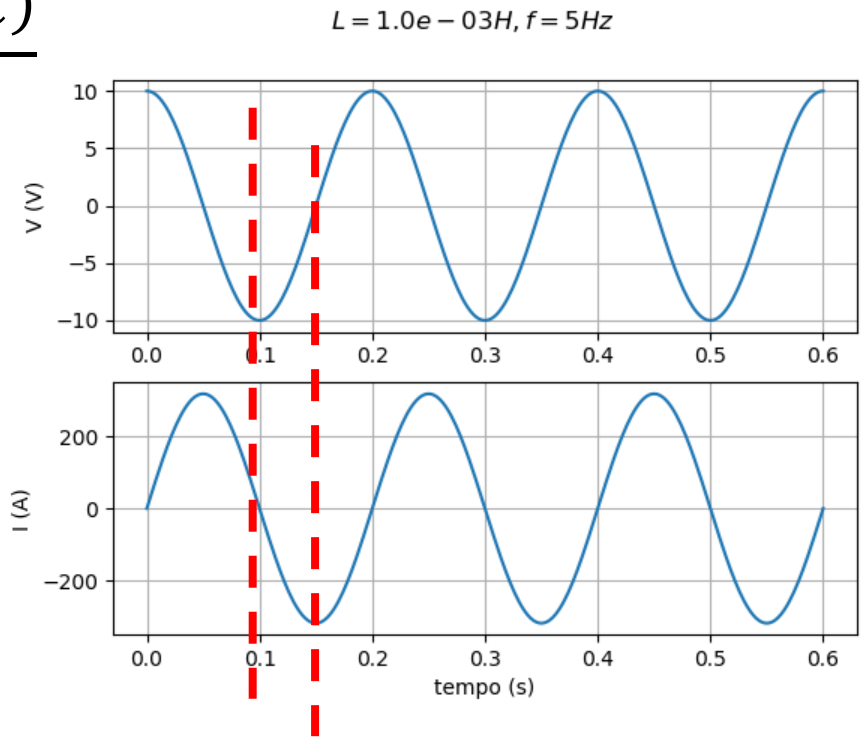



$$V_0 \cos(\omega t) = V_0 e^{i\omega t} = V_L = L\frac{dI}{dt}$$

$$I = I_0 \cos\left(\omega t - \frac{\pi}{2}\right) = \frac{V_0 \cos(\omega t)}{i\omega L}$$

$$V_L = Z_L I$$

$$Z_L = i\omega L$$



# Quando se definem impedâncias complexas…

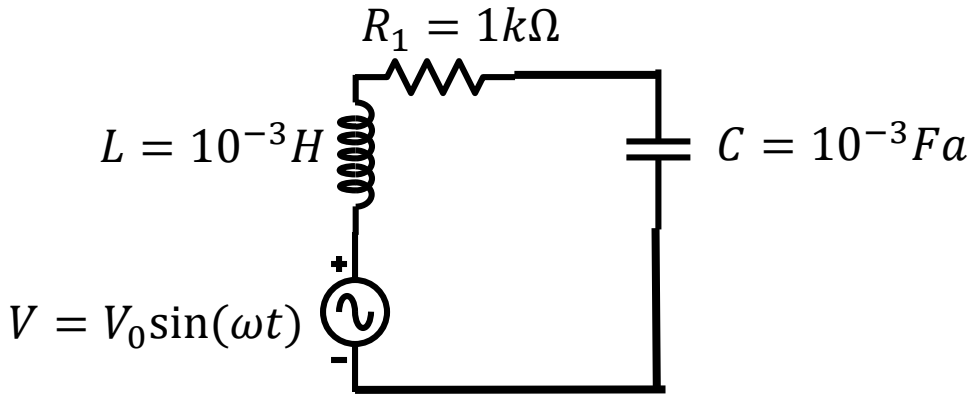
Os 3 componentes lineares satisfazem a mesma lei (de Ohm):

$$V = ZI$$

Onde $Z$ é real e constante no caso do resistor ($Z_R = R$), mas é imaginário nos outros dois componentes e varia com a frequência $\left(Z_C = -\frac{i}{\omega C}, Z_L = i\omega L\right)$.

Trata-se sempre de uma equação algébrica linear que resolve exatamente uma equação diferencial, para uma data frequência angular $\omega$.

# Circuito RLC em corrente alterna

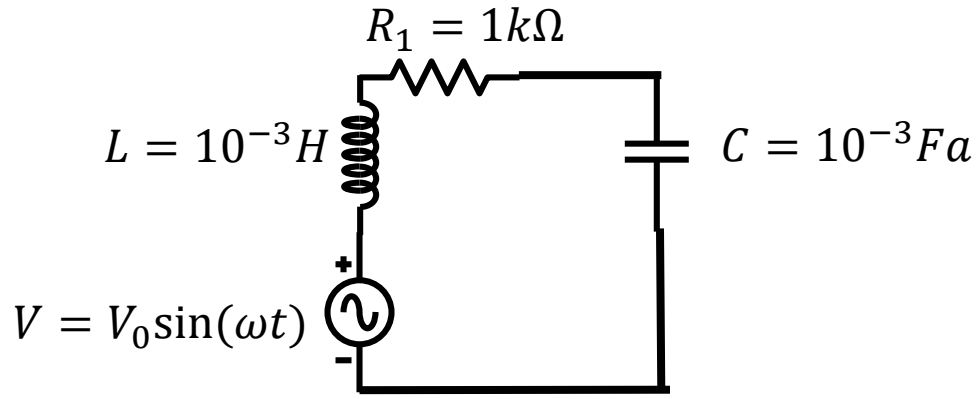



A vantagem das equações lineares é o princípio da sobreposição: as propriedades dos vários componentes podem ser adicionadas.

$$V = (Z_R + Z_L + Z_C)I = \left(R + i\omega L - \frac{i}{\omega C}\right)I$$

… notando que $Z_L, Z_C$ dependem de $\omega$

# Circuito RLC

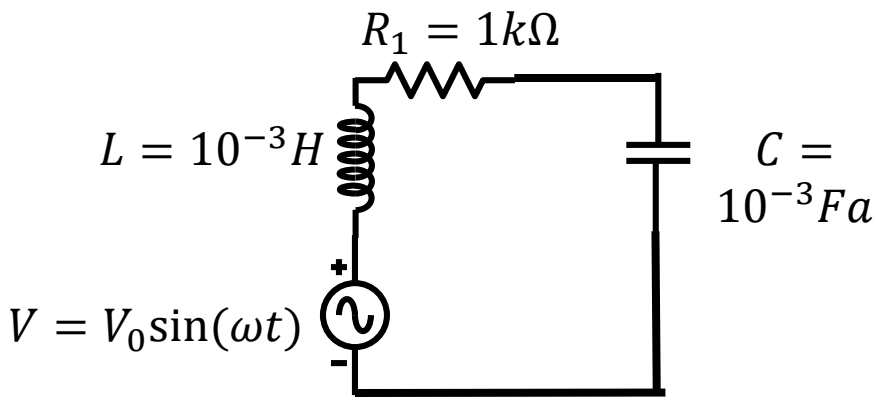


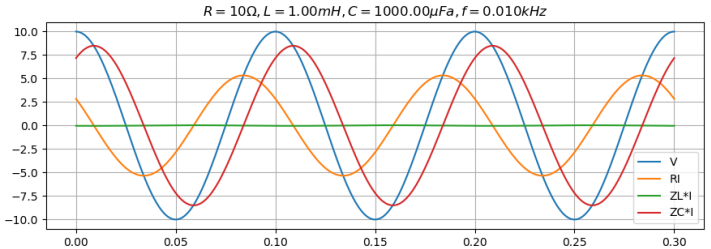

```
import numpy as np
import matplotlib.pyplot as plt
from math import pi
V0=10.;R=10.;C=1e-3;L=1e-3
i=complex(0,1.)
kp=0
for freq in[10,100,1000,10e3]:
    omega=2*pi*freq;   periodo=1./freq
    kp=kp+1
    t=np.linspace(0,3*periodo,301)
    V=V0*np.exp(i*omega*t)
    Z=R+i*L*omega-i/(C*omega)
    I=V/Z
    plt.subplot(4,1,kp)
    plt.plot(t,np.real(V),label='V');
    plt.plot(t,np.real(I*R),label='RI')
    plt.plot(t,np.real(I*i*omega*L),label='ZL*I')
    plt.plot(t,np.real(-I*i/(omega*C)),label='ZC*I')
    plt.legend();    plt.grid()
    plt.title(r'$R=%6.0f \Omega,L=%4.2f mH, C=%4.2f \mu Fa,f=%6.3f kHz$' \
         %(R,L*1000,C*1000000,freq/1000))
```

# Circuito RLC

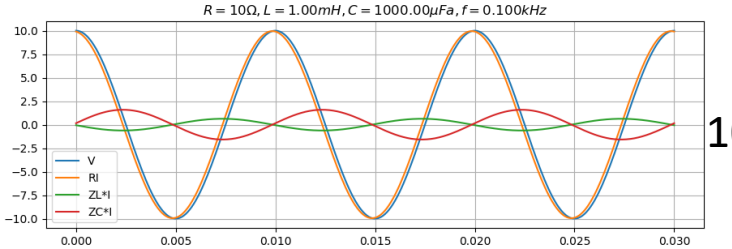
Equação linear (complexa):

$$V = (Z_R + Z_L + Z_C)I = \left(R + i\omega L - \frac{i}{\omega C}\right) I$$

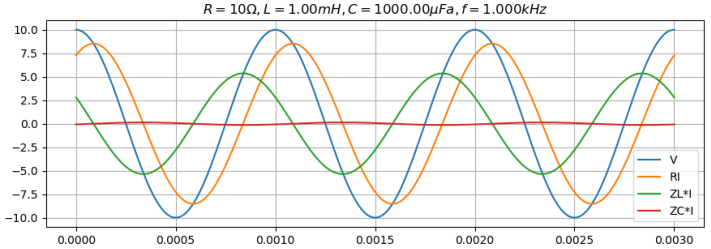


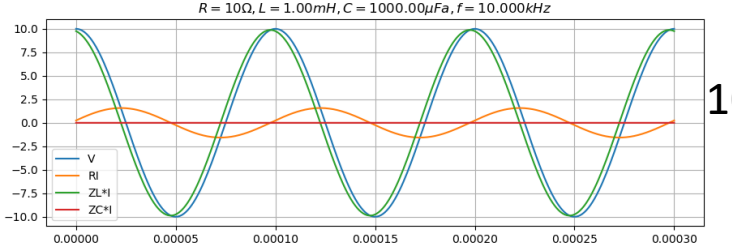

10Hz




100Hz

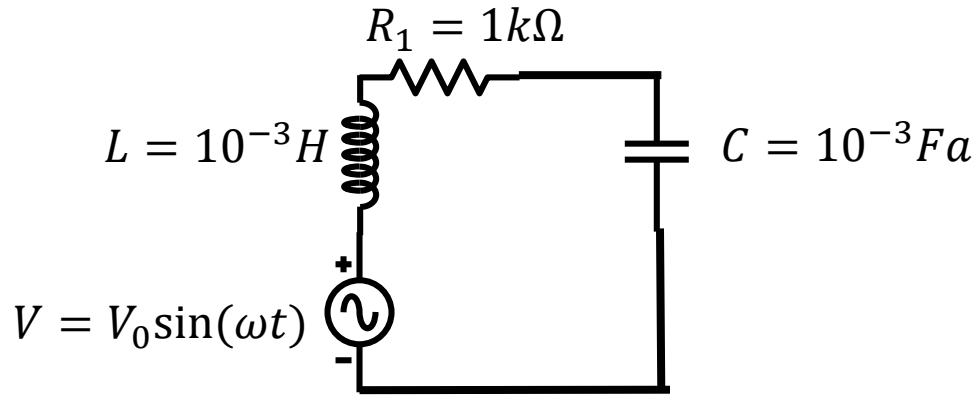



1kHz




10kHz

# Circuito RLC (v2)




```
import numpy as np
import matplotlib.pyplot as plt
from math import pi
V0=10.;R=10.;C=1e-3;L=1e-3
i=complex(0,1.)
kp=0
for freq in[10,100,1000]:
    omega=2*pi*freq;   periodo=1./freq
    kp=kp+1
    t=np.linspace(0,3*periodo,301)
    V=V0*np.exp(i*omega*t)
    Z=R+i*L*omega-i/(C*omega)
    I=V0/Z
    plt.subplot(3,1,kp)
    plt.plot(t,np.real(V),label='V');
    plt.plot(t,np.real(I*R*np.exp(i*omega*t)),label='RI')
    plt.plot(t,np.real(I*i*omega*L*np.exp(i*omega*t)),label='ZL*I')
    plt.plot(t,np.real(-I*i/(omega*C)*np.exp(i*omega*t)),label='ZC*I')
    plt.grid();plt.legend()
    plt.title(r'$R=%6.0f \Omega,L=%4.1f mH, C=%4.1f m Fa,f=%6.0f Hz$' \
         %(R,L*1000,C*1000,freq))
```

# Circuito RLC

Em geral (lei das malhas) temos uma equação diferencial:

$$V = V_R + V_L + V_C = RI + L\frac{dI}{dt} + \frac{1}{C}\int I dt$$

Mas, se

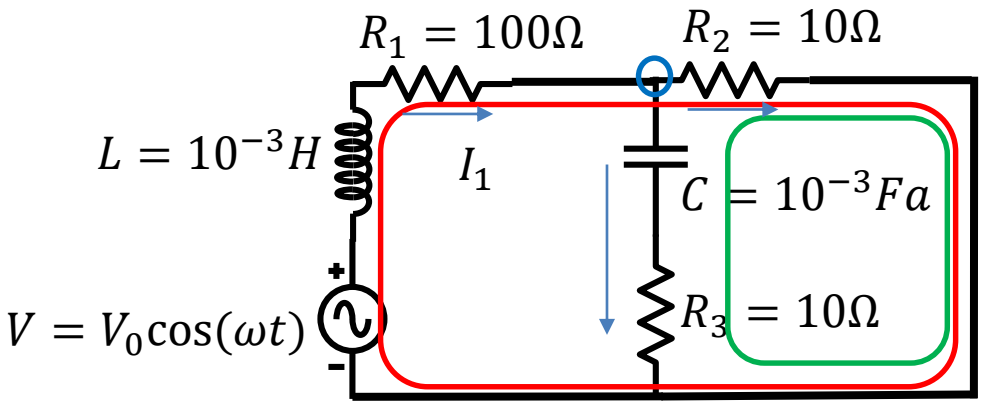
$$V = V_0 \cos(\omega t) = Re\{V_0 e^{i\omega t}\}$$

Ficamos com uma equação linear algébrica, com coeficientes complexos:

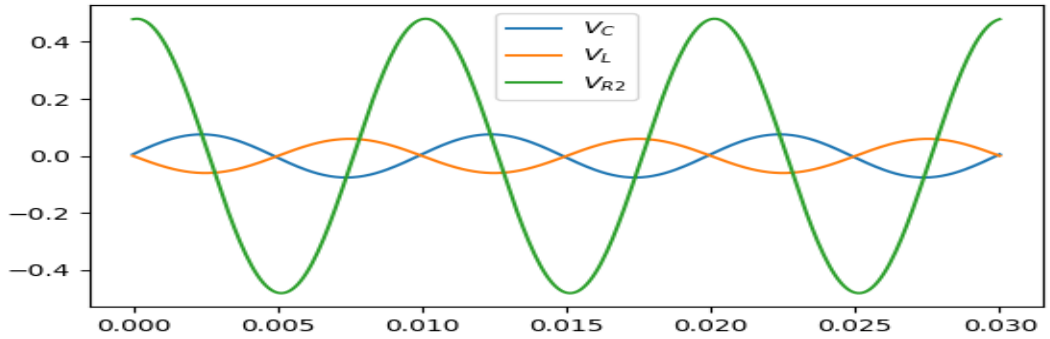
$$V = (Z_R + Z_L + Z_C)I = \left(R + i\omega L - \frac{i}{\omega C}\right) I$$

Esta transformação é um exemplo do método de Fourier de solução de equações diferenciais.

# Sistema de equações lineares complexas




$$\begin{cases} V = (i\omega L + R_1)I_1 + R_2 I_2 \\ R_2 I_2 - \left(R_3 - \frac{i}{\omega C}\right) I_3 = 0 \\ I_1 - I_2 - I_3 = 0 \end{cases} \Rightarrow \begin{bmatrix} (i\omega L + R_1) & R_2 & 0 \\ 0 & R_2 & -\left(R_3 - \frac{i}{\omega C}\right) \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} V_0 \\ 0 \\ 0 \end{bmatrix}$$




```
import numpy as np;import matplotlib.pyplot as plt
i=complex(0.,1.)
V0=10;R1=100;R2=10;R3=10;C=1e-3;L=1e-3;
freq=100;omega=2*np.pi*freq
t=np.linspace(0,3./freq,301)
V=V0*np.exp(i*omega*t)
M=np.array([[i*omega*L+R1,R2,0],\
            [0,R2,-(R3-i/(omega*C))],\
            [1,-1,-1]])
b=np.array([V0,0,0])
I=np.linalg.solve(M,b) #M é complexo
VC=-i/(omega*C)*I[2]*np.exp(i*omega*t)
VL=i*omega*L*I[0]*np.exp(i*omega*t)
VR2=R2*I[1]*np.exp(i*omega*t)
plt.plot(t,np.real(VC),label=r'$V_C$')
plt.plot(t,np.real(VL),label=r'$V_L$')
plt.plot(t,np.real(VR2),label=r'$V_{R2}$')
```