

# Kelvin\_Maxwell\_zadani

October 11, 2023

## 1 Výpočet odezvy Kelvina a Maxwellova článku na zatížení

---

Vypracované řešení zašlete včetně scriptu nebo pomocných výpočtů na adresu petr.havlassek@cvut.cz

Řešení je možné odevzdat do příštího cvičení.

Jako předmět zprávy uveďte **PRPM: BONUS**

---

### 1.1 Zadání:

Je zadán Kelvinův a Maxwellův článek, oba jsou charakterizované tuhostí pružiny  $E = 1$  MPa a viskozitou tlumiče  $\eta = 1$  MPa den. Články byly podrobeny zatěžovací zkoušce, při které byla měněna hladina napětí  $\sigma$ :

- $t \in \langle 0, \tau \rangle \dots \sigma(t) = \hat{\sigma} = 20$  MPa
- $t \in \langle \tau, 2\tau \rangle \dots \sigma(t) = 0$  MPa
- $t \in \langle 2\tau, 3\tau \rangle \dots \sigma(t) = -\hat{\sigma} = -20$  MPa
- $t \in \langle 3\tau, 4\tau \rangle \dots \sigma(t) = 0$  MPa

kde  $\tau$  je retardační, příp. relaxační čas definovaný jako  $\tau = \eta/E$ .

Vykreslete průběh deformace obou článků a vypočítejte její hodnotu v násobcích charakteristického času ( $\tau \dots 4\tau$ ).

Pokud v těchto časech dochází ke skoku, uveďte, prosím, hodnotu zleva i zprava.

Pro řešení můžete využít následující skript, který ukazuje výpočet funkce poddajnosti obou článků.

## 2 Example: Compliance functions of Kelvin and Maxwell units

```
[8]: %matplotlib inline
```

```
import math
import numpy as np
```

```
import matplotlib.pyplot as plt
```

## 2.1 Compliance function of a Kelvin unit

Notation:

- $E$  ... spring stiffness [MPa]
- $\eta$  ... dashpot viscosity [MPa day]
- $t$  ... time of interest [day]
- $t_t$  ... time of loading [day]
- $\tau$  ... retardation time [day]

$$J(t, t_t) = \frac{1}{E} [1 - \exp(-(t - t_t)/\tau)]$$

```
[9]: def Kelvin_J_func(E, eta, t, tt):  
  
    tau = eta / E # retardation time  
  
    if t > tt:  
        dt = t-tt  
  
        if tau > 0.:  
            return 1./ E * ( 1. - math.exp(-(dt)/ tau) )  
        else:  
            return 1./ E  
  
    else:  
        return 0.
```

## 2.2 Compliance function of a Maxwell unit

Notation:

- $E$  ... spring stiffness [MPa]
- $\eta$  ... dashpot viscosity [MPa day]
- $t$  ... time of interest [day]
- $t_t$  ... time of loading [day]
- $\tau$  ... relaxation time [day],  $\tau = \eta/E$

$$J(t, t_t) = \frac{1}{E} + \frac{t-t_t}{\eta}$$

```
[10]: def Maxwell_J_func(E, eta, t, tt):  
  
    if t >= tt:  
        dt = t-tt
```

```

        return 1./ E  + dt / eta

    else:
        return 0.

```

## 2.3 Solution

```

[11]: # Definition of parameters
E = 1. # MPa
eta = 1. # MPa * day

tau = eta/E # day

times = np.linspace(0., 4, num = 100 )

# Evaluation of the compliance function
J_Kelvin = []
for t in times:
    J_Kelvin.append( Kelvin_J_func(E, eta, t, 0.) )

J_Maxwell = []
for t in times:
    J_Maxwell.append( Maxwell_J_func(E, eta, t, 0.) )

```

## 2.4 Plotting

```

[7]: plt.figure(figsize=(10,6))
plt.rcParams.update({'font.size': 14})

plt.plot(times, J_Kelvin, lw=2., color="blue", label=r'$J_{Kelvin}(t)$')
plt.plot(times, J_Maxwell, lw=2., color="red", label=r'$J_{Maxwell}(t)$')

# vertical line corresponding to characteristic (retardation) time
plt.axvline(x = tau, lw=1.5, color="green", linestyle = "--")
plt.axvline(x = 2.*tau, lw=1.5, color="green", linestyle = "--")
plt.axvline(x = 3.*tau, lw=1.5, color="green", linestyle = "--")
plt.axvline(x = 4.*tau, lw=1.5, color="green", linestyle = "--")

plt.grid(True)
plt.legend()
plt.xlabel('Time, t [day]')
plt.ylabel(r'Compliance, $J$ [1/MPa]')

```

```
plt.savefig('Compliance_function_Maxwell_Kelvin.pdf')  
  
plt.show()
```

