

# 1 Példa 4.3-4.5

Határozzuk meg a különböző feszültségi állapotok esetén a főfeszültségeket és a feszültségi főirányokhoz tartozó egységvektorokat.

## 2 Megoldás

A megoldás során szükségünk lesz a `sympy` modulra. A számolások során a feszültségi állapotot jellemző tenzort egy  $3 \times 3$ -as mátrixként adhatjuk meg.

In [1]:

```
1 import sympy as sp
2 sp.init_printing()
3 # A  $\sigma$  indexei a feladat sorszámait jelölik!
4
5  $\sigma_{43}$  = sp.Matrix([[ -20, 0, 30], [ 0, 15, 0], [ 30, 0, 40]])
6  $\sigma_{44}$  = sp.Matrix([[ -5, -10, 0], [-10, -20, 0], [ 0, 0, -30]])
7  $\sigma_{45}$  = sp.Matrix([[ 0, -10, 0], [-10, 0, 0], [ 0, 0, 10]])
```

executed in 903ms, finished 09:41:36 2020-03-19

### 2.1 Sajátérték, sajátvektor számítás

A feszültségi állapotot jellemző főfeszültségek és a hozzájuk tartozó főirányok megfeleltethetők a feszültségi állapotot jellemző tenzor sajátértékeivel és sajátvektoraival.

A `sympy` segítségével a sajátértékeket és sajátvektorokat egy utasítással ( `.eigenvects()` ) megkaphatjuk. Mivel a `sympy` speciális esetekre is fel van készítve, ezért a megoldást erősen "becsomagolva" kapjuk meg.

In [2]:

```
1 eig_system=  $\sigma_{43}$ .eigenvects()
2
3 # Az első sajátérték/multiplicitás/sajátvektor adatok
4 eig_system[0]
```

executed in 540ms, finished 09:41:36 2020-03-19

Out[2]:

$$\left( 15, 1, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right)$$

Eredményül a mátrixhoz tartozó sajátérték/sajátvektor párokat kapjuk. Az `.eigenvects()` egy listát ad vissza. Ennek a listának az elemei olyan `tuple`-ök (a mi szempontunkból a `tuple` gyakorlatilag egy lista), melyek:

- A sajátérték
- A sajátérték multiplicitása (pl: ha két sajátérték egybeesik, akkor annak a multiplicitása 2 lesz)
- A sajátértékhez tartozó sajátvektor(ok) listája melynek elemszáma a az adott sajátérték multiplicitása

Tartsuk észben, hogy egy  $3 \times 3$ -as mátrix esetén ha egy sajátérték multiplicitása nagyobb mint egy, akkor az `sp.eigenvecs` nem 3 elemű listát ad eredményül!

Ha csak egy mátrixot kell kiértékeljünk, akkor a fent említett mennyiségek 'kézzel' is kiolvashatóak.

In [3]:

```
1 # Példa:
2 display(sp.Matrix([[1,0,0],[0,2,0],[0,0,1]]).eigenvecs()) # Így néz ki, ha több
```

executed in 502ms, finished 09:41:37 2020-03-19

$$\left[ \left( 1, 2, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right), \left( 2, 1, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) \right]$$

In [4]:

```
1 sigma_43_sajat = sigma_43.eigenvecs()
2 display(sigma_43_sajat) # Ilyen, ha egyszeresek
```

executed in 519ms, finished 09:41:37 2020-03-19

$$\left[ \left( 15, 1, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right), \left( 10 - 30\sqrt{2}, 1, \begin{bmatrix} -\frac{30}{-30+30\sqrt{2}} \\ 0 \\ 1 \end{bmatrix} \right), \left( 10 + 30\sqrt{2}, 1, \begin{bmatrix} - \\ \\ \end{bmatrix} \right) \right]$$

Az `.eigenvecs()` függvény nem feltétlenül sorrendben adja vissza a sajátértékeket, ezért rendeznünk kell őket. Célszerű a sajátértékek alapján csökkenő sorrendbe tenni a fenti eredményt. Ezt a következő paranccsal tudjuk megtenni: `sigma_43_sajat.sort(key = lambda x : x[0], reverse=True)`.

Rövid magyarázat:

- `lista.sort()` : a listát helyben rendezi, azaz a rendezetlen listát felülírja a rendezett listával,
- `key =` : ami alapján rendez, egyszerű számoknál, stringeknél erre nincs szükség (a mi kódunk is lefut `key` nélkül, de nem tudjuk, hogy mi alapján rendezi az elemeket, így biztonságosabb megadni a `key`-t),
- `lambda x : x[0]` : egy úgynevezett `lambda` függvény, ami a bemenő `x`-nek az első elemét adja vissza, azaz az első elem alapján fogunk rendezni,
- `reverse = True` : növekvő helyett csökkenő sorrendbe teszi az eredményt.

In [5]:

```
1 sigma_43_sajat.sort(key = lambda x : x[0], reverse=True)
2 display(sigma_43_sajat)
```

executed in 534ms, finished 09:41:38 2020-03-19

$$\left[ \left( 10 + 30\sqrt{2}, 1, \begin{bmatrix} -\frac{30}{-30\sqrt{2}-30} \\ 0 \\ 1 \end{bmatrix} \right), \left( 15, 1, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right), \left( 10 - 30\sqrt{2}, 1, \begin{bmatrix} - \\ \\ \end{bmatrix} \right) \right]$$

## 2.2 Automatizálás

Tegyük fel, hogy a lustaságunk által vezérelve nem szeretnénk minden mátrix eredményét kézzel kiolvasni/kiíratni a listákból. Írjunk egy függvényt ami automatizálja ezt a feladatot!

In [6]:

```
1 def print_eigensystem(matrix):
2     # Legyen a függvény bemenete a vizsgálandó mátrix
3     eig_system= matrix.eigenvecs()
4
5     # Az eredményt a program nem rendezi alaphól a sajátértékek nagysága szerint
6     # A rendezést a 'tuple'-ök első elemei szerint végezzük
7     # Utánanézési lehetőség: lambda függvények, sorted() függvény
8     eig_system.sort(key=lambda x: x[0], reverse=True)
9
10    # Végig iterálva az 'eig_system' elemein, kiíratjuk a főfeszültségeket és f
11    n = 1
12    for elem in eig_system:
13        # Ha egy főfeszültség többszörös multiplicitású, akkor többször kell ki
14        for i in range(elem[1]): # Az 'elem[1]' értéke a multiplicitást mutatja
15            # 'range(elem[1])' : csinál egy 'range' objekt
16            # tud futni. Pontosan annyiszor fut le így a c
17            # 'range' függvény argumentumaként, jelen eset
18            sajáttertek = elem[0].evalf(5)
19            # Normáljuk a sajátvektorokat, hogy egység hosszúságúak legyenek
20            sajátvektor = (elem[2][i].normalized()).evalf(5) # '.normalize()':
21            # Az 'n' változóval sorszámozzuk az egyes értékeket
22            print(str(n)+' . Főfeszültség: '+str(sajáttertek.evalf(5))+' MPa')
23            print(str(n)+' . Főirány: ')
24            display(sajátvektor)
25            n += 1 # sorszám léptetése
```

executed in 10ms, finished 09:41:38 2020-03-19

A függvényünket használva a feladatok megoldásai:

In [7]:

```
1 print_eigensystem(σ_43)
```

executed in 1.66s, finished 09:41:39 2020-03-19

1. Főfeszültség: 52.426 MPa

1. Főirány:

$$\begin{bmatrix} 0.38268 \\ 0 \\ 0.92388 \end{bmatrix}$$

2. Főfeszültség: 15.000 MPa

2. Főirány:

$$\begin{bmatrix} 0 \\ 1.0 \\ 0 \end{bmatrix}$$

3. Főfeszültség: -32.426 MPa

3. Főirány:

$$\begin{bmatrix} -0.92388 \\ 0 \\ 0.38268 \end{bmatrix}$$

In [8]:

```
1 print_eigensystem(σ_44)
```

executed in 1.96s, finished 09:41:41 2020-03-19

1. Főfeszültség: 0 MPa

1. Főirány:

$$\begin{bmatrix} -0.89443 \\ 0.44721 \\ 0 \end{bmatrix}$$

2. Főfeszültség: -25.000 MPa

2. Főirány:

$$\begin{bmatrix} 0.44721 \\ 0.89443 \\ 0 \end{bmatrix}$$

3. Főfeszültség: -30.000 MPa

3. Főirány:

$$\begin{bmatrix} 0 \\ 0 \\ 1.0 \end{bmatrix}$$

In [9]:

```
1 print_eigensystem(σ_45)
```

executed in 1.74s, finished 09:41:43 2020-03-19

1. Főfeszültség: 10.000 MPa

1. Főirány:

$$\begin{bmatrix} -0.70711 \\ 0.70711 \\ 0 \end{bmatrix}$$

2. Főfeszültség: 10.000 MPa

2. Főirány:

$$\begin{bmatrix} 0 \\ 0 \\ 1.0 \end{bmatrix}$$

3. Főfeszültség: -10.000 MPa

3. Főirány:

$$\begin{bmatrix} 0.70711 \\ 0.70711 \\ 0 \end{bmatrix}$$