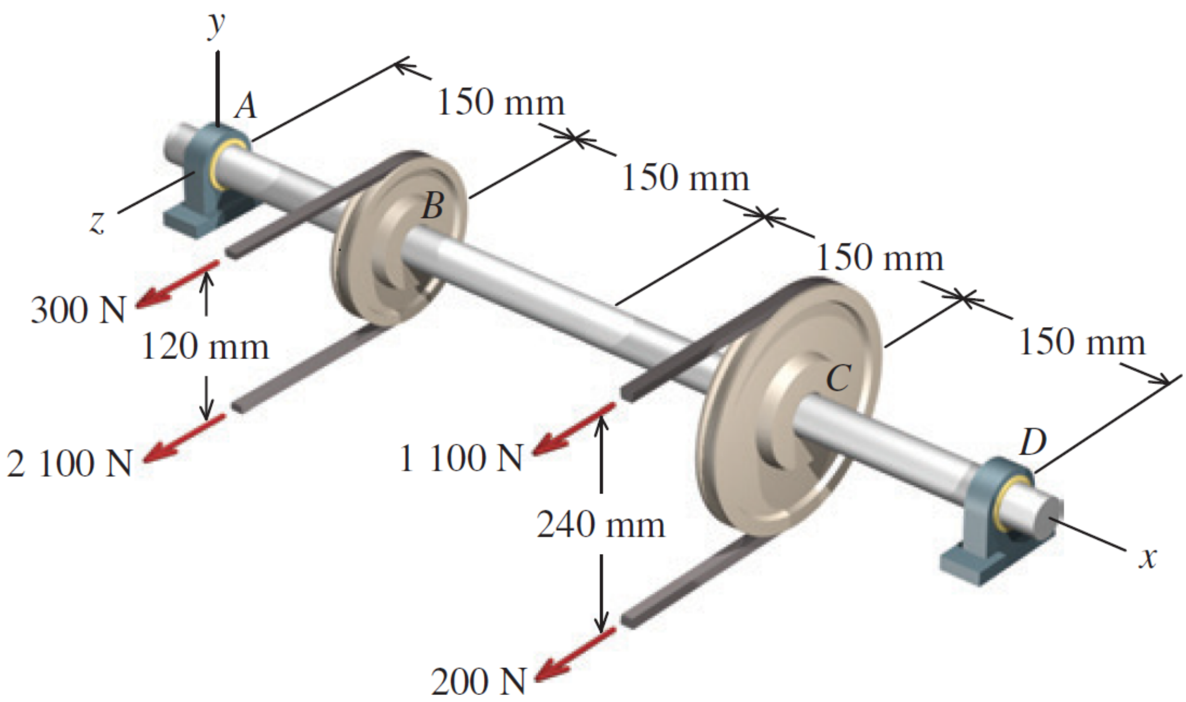
1 Példa 6.4

Egy $20\,\mathrm{mm}$ átmérőjű acél tengelyt az A és D csapágyak támasztják meg, melyek engedik a tengely kismértékű szögelfordulását. Az A csapágy gátolja a tengelyirányú elmozdulást, míg a D csapágy engedélyezi. A tengely terhelése a B és C szíjtárcsákról átadódó erők és nyomatékok. Határozzuk meg a tengelyben ébredő maximális Mohr és HMH-féle egyenértékű feszültségeket! Mekkora legyen a tengely átmérője ha $\sigma_{\mathrm{meg}} = 300\,\mathrm{MPa}$?



2 Megoldás

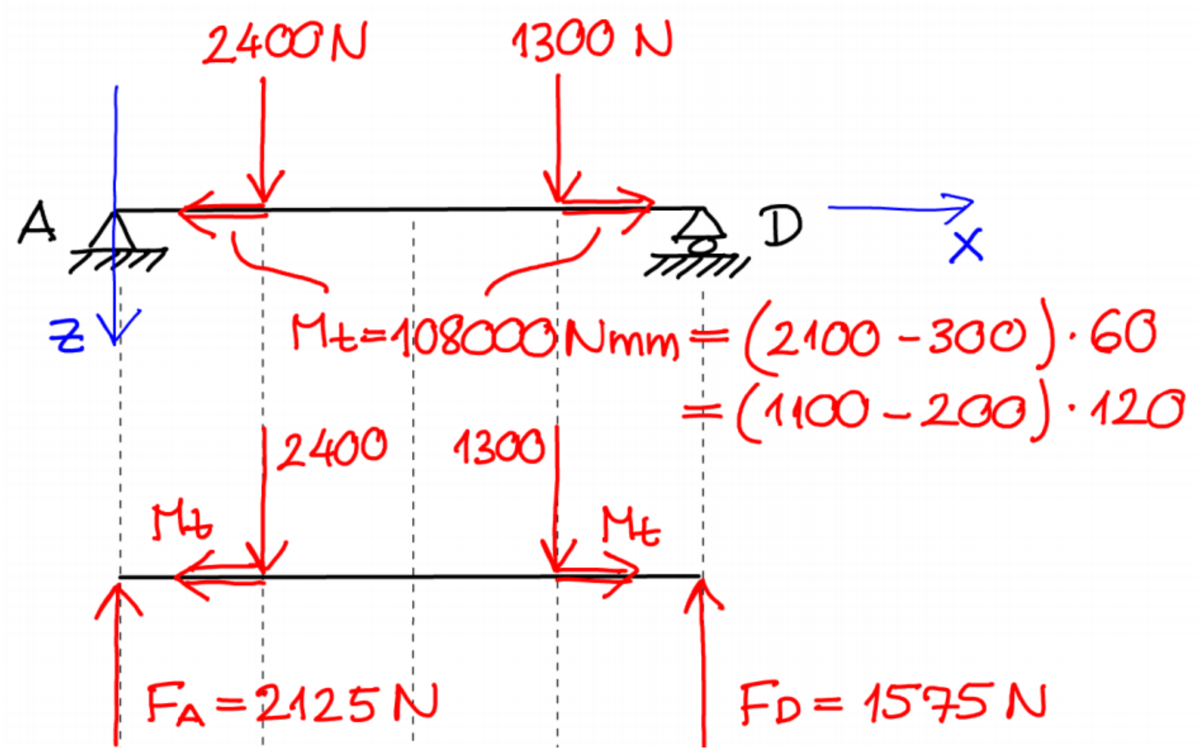
Szükségünk lesz a `sympy` modulra.

In [1]:

```
import sympy as sp
sp.init_printing()

F_A, F_D, x = sp.symbols("F_A, F_D, x")
```

executed in 859ms, finished 09:53:03 2020-04-02

Határozzuk meg a terheléseket!



Az F_{A} és F_{D} erők meghatározásának menete:

- Felírjuk a nyomatéki egyenletet az A pontra.
- Kiszámítjuk F_{D}-t.
- A z irányú erőegyensúly alapján kiszámítjuk F_{A}-t.

Természetesen ezek az egyenletek kézzel is könnyedén megoldhatóak lennének.

In [2]:

```
MA = -150*2400-450*1300+600*F_D #nyomatéki egyenlet az A pontra
F_D_megoldas = sp.solve(MA,F_D)
display(F_D_megoldas) #egy 1 elemű listát kapunk

F_D_megoldas = F_D_megoldas[0] #az egyszerűség kedvéért kiszedjük a megoldást a
F_D_megoldas #N
```

executed in 969ms, finished 09:53:04 2020-04-02

[1575]

Out[2]:

1575

In [3]:

```
Fz = 2400+1300-F_D_megoldas-F_A #z irányú erőegyensúly
F_A_megoldas = sp.solve(Fz,F_A)
F_A_megoldas = F_A_megoldas[0] #kiszedjük a megoldást a listából
F_A_megoldas #N
```

executed in 498ms, finished 09:53:05 2020-04-02

Out[3]:

2125

Az igénybevételi függvények szakaszonként adhatóak meg. Ehhez a `Piecewise()` függvényt használhatjuk.

Fontos, hogy a megfelelő sorrendben adjuk meg a szakaszokat, növekvő x szerint. A nyíró igénybevétel ($V(x)$):

In [4]:

```
Vx = sp.Piecewise((F_A_megoldas,x<=150),(F_A_megoldas-2400,x<=450),(F_A_megolda
Vx
```

executed in 517ms, finished 09:53:05 2020-04-02

Out[4]:

$$\begin{cases} 2125 & \text{for}\: x \leq 150 \\ -275 & \text{for}\: x \leq 450 \\ -1575 & \text{for}\: x \leq 600 \end{cases}$$

Hasonlóan írhatjuk fel a hajlító igénybevételi függvényt ($M_\mathrm{h}(x)$) is:

In [5]:

```
Mhx = sp.Piecewise((-F_A_megoldas*x,x<=150),(-F_A_megoldas*x+2400*(x-150),x<=45
Mhx
```

executed in 526ms, finished 09:53:06 2020-04-02

Out[5]:

$$\begin{cases} -2125x & \text{for}\: x \leq 150 \\ 275x - 360000 & \text{for}\: x \leq 450 \\ 1575x - 945000 & \text{for}\: x \leq 600 \end{cases}$$

Hasonlóan írhatjuk fel a csavaró igénybevételi függvényt ($M_\mathrm{t}(x)$) is:

In [6]:

```
Mtx = sp.Piecewise((0,x<=150),(108000,x<=450),(0,x<=600))
Mtx
```

executed in 512ms, finished 09:53:06 2020-04-02

Out[6]:

$$\begin{cases} 0 & \text{for}\: x \leq 150 \\ 108000 & \text{for}\: x \leq 450 \\ 0 & \text{for}\: x \leq 600 \end{cases}$$

Ezeket at 1.10-es feladathoz hasonló módon ábrázoljuk.

In [7]:

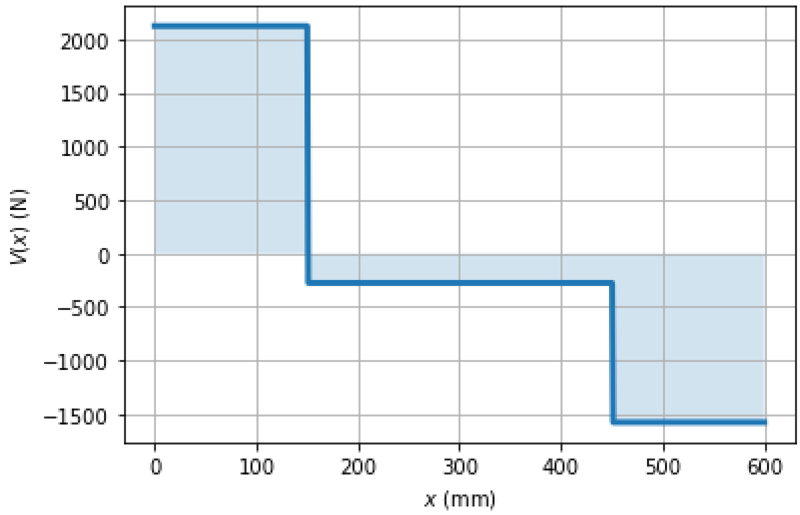
```
import matplotlib.pyplot as plt # Betöltjük a matplotlib modult, amivel plotolh
from numpy import linspace # Betöltjük csak linspace függvényt a numpy modulból
                               # összes függvényét), amivel majd a plotoláshoz kész

L = 600 # A rúd teljes hossza.
xs = linspace(0,L,601)  # Felveszünk 601 darab x értéket a rúd hossza mentén
                                    # (az L hosszat át kell alakítani float-tá).
Vxs = [Vx.subs(x,xi) for xi in xs] # Kiszámoljuk a nyíróerő függvény értékeit a
# Vx.subs(x,xi): a Vx függvényben lévő x változó helyére behelyettesítjük xi-t.
Mhxs = [Mhx.subs(x,xi) for xi in xs] # Kiszámoljuk a hajlítónyomatéki függvény
Mtxs = [Mtx.subs(x,xi) for xi in xs] # Kiszámoljuk a csavarónyomatéki függvény
```

executed in 1.38s, finished 09:53:08 2020-04-02

In [8]:

```
plt.plot(xs, Vxs, lw = 3) # A V(x) függvény képének létrehozása 3-as vonalvasta
Vx_PythonFloat=[float(Vx) for Vx in Vxs] #A fill_between nem tudja kezelni a sy
plt.fill_between(xs, Vx_PythonFloat, alpha=0.2) #Kiszínezi a függvény alatti te
plt.xlabel("$x$ (mm)") #x tengelyhez tartozó tengelyfelirat
plt.ylabel("$V(x)$ (N)") #y tengelyhez tartozó tengelyfelirat
plt.grid() #rács
plt.show() #kirajzolás
```

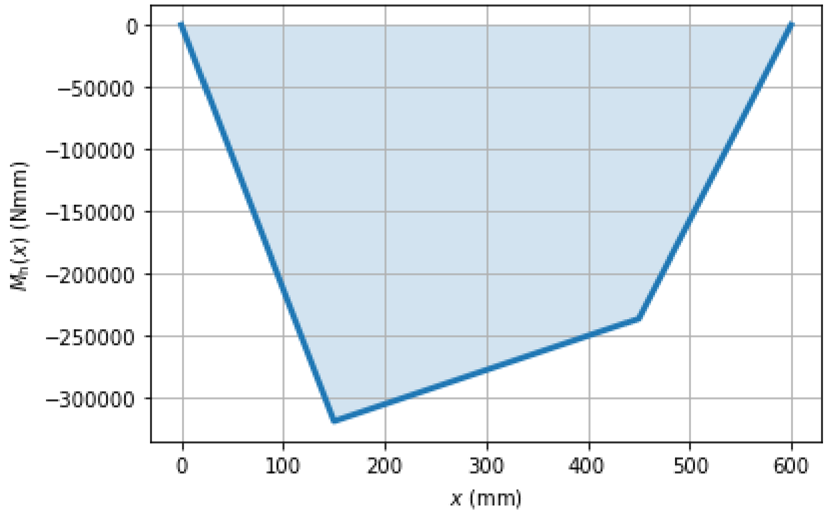
executed in 723ms, finished 09:53:08 2020-04-02



In [9]:

```python
plt.plot(xs, Mhxs, lw = 3) #az Mh(x) függvény képének létrehozása 3-as vonalvas
Mhx_PythonFloat=[float(Mhx) for Mhx in Mhxs] #A fill_between nem tudja kezelni
plt.fill_between(xs, Mhx_PythonFloat, alpha=0.2) #Kiszínezi a függvény alatti t
plt.ylabel(r"$M_{\rm h}(x)$ (Nmm)") # x tengelyhez tartozó tengelyfelirat. Az r
                                    # karaktert szeretnénk a stringbe tenni a L
plt.xlabel("$x$ (mm)") #y tengelyhez tartozó tengelyfelirat
plt.grid() #rács létrehozása
plt.show() #kirajzolás
```

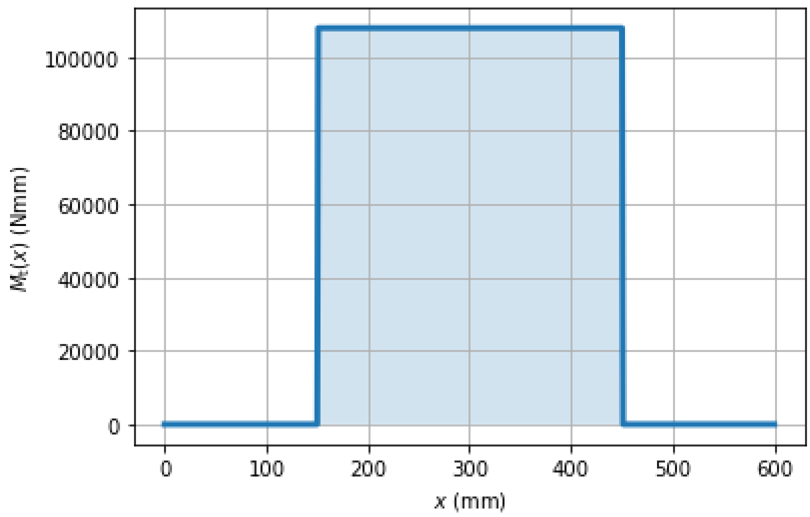
executed in 261ms, finished 09:53:09 2020-04-02



In [10]:

```python
plt.plot(xs, Mtxs, lw = 3) #az Mt(x) függvény képének létrehozása 3-as vonalvas
Mtx_PythonFloat=[float(Mtx) for Mtx in Mtxs] #A fill_between nem tudja kezelni
plt.fill_between(xs, Mtx_PythonFloat, alpha=0.2) #Kiszínezi a függvény alatti t
plt.ylabel(r"$M_{\rm t}(x)$ (Nmm)") # x tengelyhez tartozó tengelyfelirat. Az r
                                    # karaktert szeretnénk a stringbe tenni a L
plt.xlabel("$x$ (mm)") #y tengelyhez tartozó tengelyfelirat
plt.grid() #rács létrehozása
plt.show() #kirajzolás
```

executed in 247ms, finished 09:53:09 2020-04-02



Láthatjuk, hogy mindegyik igénybevétel abszolut értéke a B keresztmetszetben a legnagyobb, ezért ez lesz a veszélyes keresztmetszet.

$$M_\mathrm{h}(B) = -318750\,\mathrm{Nmm},$$
$$M_\mathrm{t}(B) = 108000\,\mathrm{Nmm}.$$

In [11]:

```
Mhb = Mhx.subs(x,150)
display(Mhb) #Nmm

Mtb = Mtx.subs(x,150)
display(Mtb) #Nmm
```

executed in 1.37s, finished 09:53:10 2020-04-02

-318750

0

A B pontban szakadása van az M_t függvénynek. A függvény definiálási módja miatt itt 0-t kapunk. Egy lehetséges mód a helyes érték kinyerésére, ha a plothoz létrehozott adatsorból kiválasztjuk a legnagyobb értéket. Ez a megoldás ebben a konkrét esetben működik, viszont általánosságban nem.

Megbízhatóbb módszer, ha felveszünk egy kis δ távolságot, és B helye előtt és után δ-nal vizsgáljuk a függvény értékét, és válasszuk ki azt, hogy az abszolut érték nagyobb. Ezzel a módszerrel csak nagyon kis hibát követünk el, akkor is, ha a vizsgált függvény nem konstans szakaszokból áll.

In [12]:

```
δ = 1e-10 #\delta = 10^-10 mm
MtB_elott = Mtx.subs(x,150-δ)
MtB_utan = Mtx.subs(x,150+δ)
nagyobb_abs = max(abs(MtB_elott),abs(MtB_utan))
Mtb = nagyobb_abs
Mtb
```

executed in 748ms, finished 09:53:11 2020-04-02

Out[12]:

108000

A Mohr és HMH elmélet szerinti redukált nyomatékok:

In [13]:

```
MredMohr = sp.sqrt(Mhb**2+Mtb**2)
print('Mohr alapján:')
display(MredMohr.evalf(5)) #Nmm

MredHMH = sp.sqrt(Mhb**2+3*Mtb**2/4)
print('HMH alapján:')
MredHMH.evalf(5) #Nmm
```

executed in 1.37s, finished 09:53:12 2020-04-02

```
Mohr alapján:
```

$$3.3655 \cdot 10^5$$

```
HMH alapján:
```

Out[13]:

$$3.3219 \cdot 10^5$$

A keresztmetszeti tényező $d = 20\,\mathrm{mm}$ esetén:

In [14]:

```
d = 20
Ky = d**3*sp.pi/32
```

executed in 9ms, finished 09:53:12 2020-04-02

2.0.0.1 A Mohr és HMH elmélet szerinti egyenértékű feszültségek:

In [15]:

```
σegyMohr = MredMohr/Ky
print('Mohr alapján:')
display(σegyMohr.evalf(5)) #MPa

σegyHMH = MredHMH/Ky
print('HMH alapján:')
σegyHMH.evalf(5) #MPa
```

executed in 1.40s, finished 09:53:14 2020-04-02

```
Mohr alapján:
```

$$428.51$$

```
HMH alapján:
```

Out[15]:

$$422.96$$

2.0.0.2 Méretezés:

Az előbbi összefüggésekből az átmérőt kifejezve a következő kifejezésre jutunk:

$$d_{min} = \sqrt[3]{\frac{32 M_{red}}{\pi \sigma_{meg}}}.$$

In [16]:

```
σmeg = 300 #MPa
dminMohr = sp.root(32*MredMohr/(sp.pi*σmeg),3)
print('Mohr alapján:')
display(dminMohr.evalf(5)) #mm

dminHMH = sp.root(32*MredHMH/(sp.pi*σmeg),3)
print('HMH alapján:')
dminHMH.evalf(5) #mm
```

executed in 1.34s, finished 09:53:15 2020-04-02

```
Mohr alapján:
```

22.524

```
HMH alapján:
```

Out[16]:

22.426