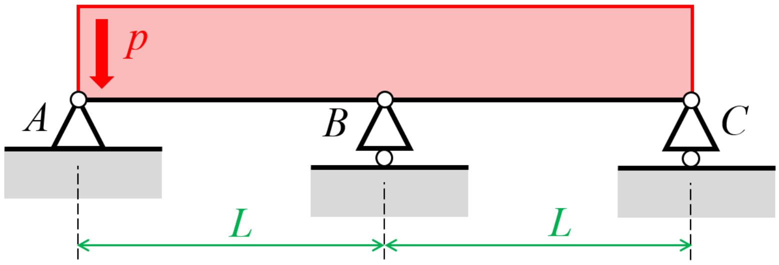
1 Példa 7.8

Hányszor nagyobb erő ébred az alábbi tartó B alátámasztásánál, mint az A vagy C helyen? Rajzoljuk fel a tartó igénybevételi függvényeit paraméteresen! A tartó anyaga és keresztmetszete végig állandó.



2 Megoldás

Betöltjük a `sympy` modult. Definiáljuk a szimbólumokat.

In [1]:

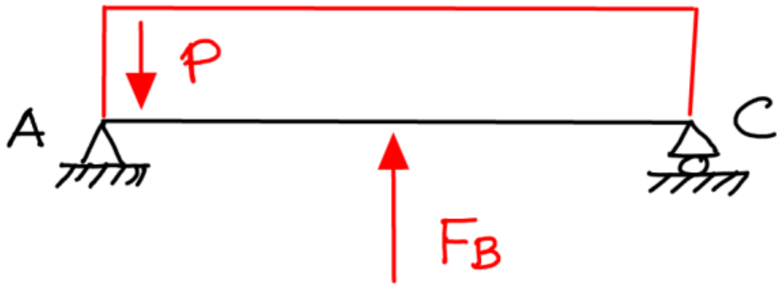
```
import sympy as sp
sp.init_printing()

L, p, x, F_A, F_B, F_C = sp.symbols('L, p, x, F_A, F_B, F_C')
```

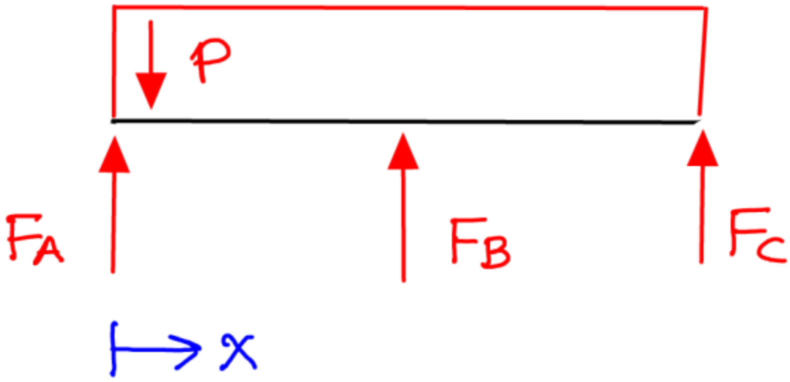
executed in 1.28s, finished 10:05:14 2020-05-01

2.1 $\frac{F_B}{F_A}$ hányados

A feladat statikailag határozatlan. Tegyük határozottá a B alátámasztás eltávolításával és vegyük fel a koordinátarendszert. Ebben az esetben az alakváltozási feltétel:

$$\frac{\partial U}{\partial F_B} = 0.$$





Ekkor $U = U_{Mh}$. A fenti parciális derivált a következő módon írható fel:

$$\frac{\partial U}{\partial F_B} = \frac{1}{IE}\left(\int_0^L M_{h1}\frac{\partial M_{h1}}{\partial F_B}\mathrm{d}x + \int_L^{2L} M_{h2}\frac{\partial M_{h2}}{\partial F_B}\mathrm{d}x\right) = 0.$$

Írjuk fel a reakcióerőket F_B függvényeként. Ehhez először írjuk fel a nyomatéki egyenletet az A pontra, majd fejezzük ki belőle F_C-t. Az egyenletet 0-ra rendezve írjuk fel.

In [2]:

```
MA_egy = -p*2*L*L+F_B*L+F_C*2*L
FC_kif = sp.solve(MA_egy,F_C) #kifejezzük F_C-t, a megoldást egy 1 elemű listáb
FC_kif = FC_kif[0] #kiszedjük a listából a megoldást
FC_kif
```

executed in 617ms, finished 10:05:15 2020-05-01

Out[2]:

$$-\frac{F_B}{2} + Lp$$

Írjuk fel a függőleges irányú erőegyensúlyt. Fejezzük ki F_A-t, behelyettesítve F_C-re a fent kifejezést.

In [3]:

```
Fy_egy = F_A+F_B+F_C-p*2*L
Fy_egy = Fy_egy.subs(F_C,FC_kif) #behelyettsítjük az F_C-re fentebb kiszámolt k
FA_kif = sp.solve(Fy_egy,F_A) #kifejezzük F_A-t
FA_kif = FA_kif[0]
FA_kif
```

executed in 496ms, finished 10:05:15 2020-05-01

Out[3]:

$$-\frac{F_B}{2} + Lp$$

Azonos eredményt kaptunk, mint F_C-re. Ez nem meglepő, hiszen a rendszer szimmetrikus.

Írjuk fel a hajlítónyomatéki függvényeket, szintén F_B-vel kifejezve!

In [4]:

```
Mh1 = -F_A*x+p*x**2/2 #a hajlítónyomatéki függvény F_A-val kifejezve
Mh1 = Mh1.subs(F_A,FA_kif)
display(Mh1.expand()) #.expand(): zárójelek kibontása az olvashatóbb eredményér
#parciális derivált:
parcMh1 = Mh1.diff(F_B)
parcMh1
```

executed in 1.04s, finished 10:05:16 2020-05-01

$$\frac{F_B x}{2} - Lpx + \frac{px^2}{2}$$

Out[4]:

$$\frac{x}{2}$$

In [5]:

```
Mh2 = Mh1-F_B*(x-L)
display(Mh2.expand()) #.expand(): zárójelek kibontása az olvashatóbb eredményér
#parciális derivált:
parcMh2 = Mh2.diff(F_B)
parcMh2
```

executed in 1.05s, finished 10:05:17 2020-05-01

$$F_B L - \frac{F_B x}{2} - Lpx + \frac{px^2}{2}$$

Out[5]:

$$L - \frac{x}{2}$$

Írjuk be a feladat elején szereplő integrálba a kiszámolt eredményeket. A `sympy` -ban integrálni az `integrate(kifejezes, valtozo)` utasítással tudunk. Határozott integrál számításához meg kell adnunk a határokat is a következő szintaktikával: `integrate(kifejezes, (valtozo, also_hatar, felso_hatar))` .

In [6]:

```
integr = sp.integrate(Mh1*parcMh1,(x,0,L)) + sp.integrate(Mh2*parcMh2,(x,L,2*L)
integr = integr.simplify() #.simplify(): egyszerűsítések elvégzése
integr
```

executed in 763ms, finished 10:05:18 2020-05-01

Out[6]:

$$\frac{L^3 (4F_B - 5Lp)}{24}$$

A fenti egyenletet kell megoldanuk F_B-re.

In [7]:

```
FB_sol = sp.solve(integr,F_B)
FB_sol = FB_sol[0]
FB_sol
```

executed in 543ms, finished 10:05:19 2020-05-01

Out[7]:

$$\frac{5Lp}{4}$$

Ezt írjuk vissza $F_A = F_C$-be.

In [8]:

```
FA_sol = FA_kif.subs(F_B,FB_sol)
display(FA_sol)
FC_sol = FC_kif.subs(F_B,FB_sol)
FC_sol
```

executed in 861ms, finished 10:05:20 2020-05-01

$$\frac{3Lp}{8}$$

Out[8]:

$$\frac{3Lp}{8}$$

Kiszámítva az $\frac{F_B}{F_A}$ hányados:

In [9]:

```
FB_sol/FA_sol
```

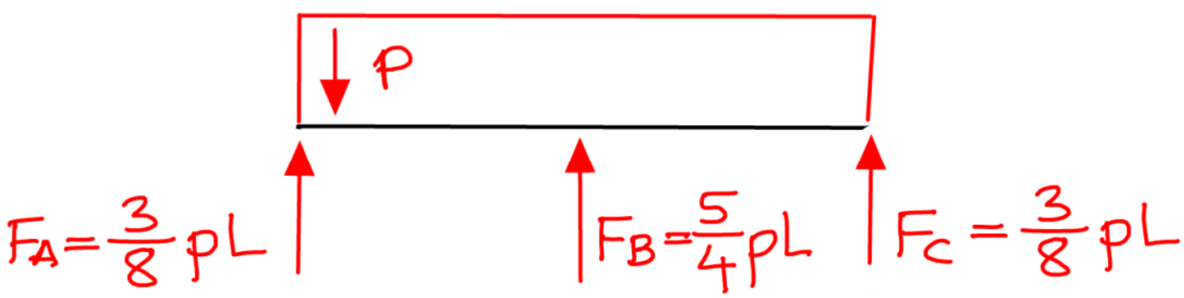
executed in 482ms, finished 10:05:20 2020-05-01

Out[9]:

$$\frac{10}{3}$$

2.2 Igénybevételi függvények

A korábbi ábrázolásokhoz hasonlóan szakaszosan folytonos függvényt az `sp.Piecewise` segítségével definiálhatunk. A szabadtest ábra, immár a kiszámolt értékekkel:



Nyíró igénybevétel: $V(x)$.

In [10]:

```
Vx = sp.Piecewise((FA_sol-x*p, x<L),(FA_sol-x*p+FB_sol, x<2*L))
Vx
```

executed in 440ms, finished 10:05:21 2020-05-01

Out[10]:

$$\begin{cases} \frac{3Lp}{8} - px & \text{for}\ L > x \\ \frac{13Lp}{8} - px & \text{for}\ x < 2L \end{cases}$$

Hajlító igénybevétel: $M_h(x)$. Felhasználjuk a korábbi eredményeket.

In [11]:

```
Mhx = sp.Piecewise((Mh1.simplify(),x<L),(Mh2.simplify(),x<2*L))
Mhx
```

executed in 499ms, finished 10:05:21 2020-05-01

Out[11]:

$$\begin{cases} \frac{x(F_B - 2Lp + px)}{2} & \text{for}\ L > x \\ F_B L - \frac{F_B x}{2} - Lpx + \frac{px^2}{2} & \text{for}\ x < 2L \end{cases}$$

2.2.1 Ábrázolás

Betöltjük a `matplotlib` -et és a `linspace` -t.

In [12]:

```
import matplotlib.pyplot as plt
from numpy import linspace
```

executed in 568ms, finished 10:05:22 2020-05-01

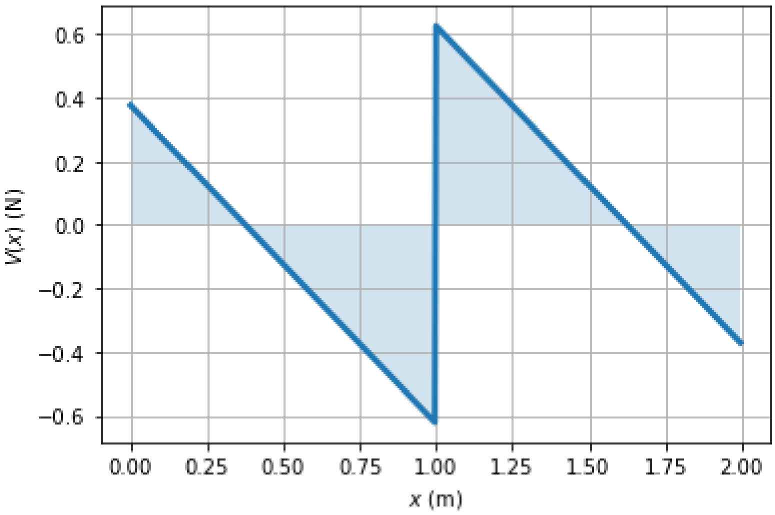
Láthatjuk, hogy az x koordinátán kívül az L és p is benne marad az igénybevételi függvények kifejezéseiben. Mivel plotolni csak konkrét számokkal tudunk, ezért ezeknek szükséges valamilyen önkényes (de fizikailag értelmes) értéket adnunk. Ez nem befolyásolja az ábrák jelleghelyességét. Itt most $L = 1\ \mathrm{m}$-t és $p = 1\ \frac{\mathrm{N}}{\mathrm{m}}$-t választunk.

In [13]:

```
L_val = 1
p_val = 1
xs = linspace(0,2*L_val,501) #felveszünk 501 pontot a 2L távolságon
Vx_konkret = Vx.subs([(L,L_val),(p,p_val)])
display(Vx_konkret)

Vxs = [float(Vx_konkret.subs(x,xi)) for xi in xs] #kiszámoljuk V(x) értékeit az
                                                   #és egyből 'float'-tá alakítj
plt.plot(xs, Vxs, lw = 3) # A V(x) függvény képének létrehozása 3-as vonalvastag
plt.fill_between(xs, Vxs, alpha=0.2) #Kiszínezi a függvény alatti területet
plt.xlabel("$x$ (m)") #x tengelyhez tartozó tengelyfelirat
plt.ylabel("$V(x)$ (N)") #y tengelyhez tartozó tengelyfelirat
plt.grid() #rács
plt.show() #kirajzolás
```

executed in 1.79s, finished 10:05:23 2020-05-01

$$\begin{cases} \frac{3}{8} - x & \text{for}\ x < 1 \\ \frac{13}{8} - x & \text{for}\ x < 2 \end{cases}$$



A hajlítónyomatéki függvényben F_B is szerepel. F_B-t már kifejeztük L és p függvényeként, viszont a behelyettesítéseknél figyelnünk kell arra, hogy először F_B helyére írjuk be az L-t és p-t tartalmazó kifejezést, és csak utána ezek konkrét értékeit. Ellenkező esetben L-t és p-t tartalmazó eredményt kapunk, és újra be kell helyettesítenünk a konkrét értékeket.

In [14]:

```
Mhx_konkret = Mhx.subs([(F_B,FB_sol),(L,L_val),(p,p_val)])
display(Mhx_konkret)
```

executed in 663ms, finished 10:05:24 2020-05-01

$$\begin{cases} \frac{x\left(x-\frac{3}{4}\right)}{2} & \text{for}\ x < 1 \\ \frac{x^2}{2} - \frac{13x}{8} + \frac{5}{4} & \text{for}\ x < 2 \end{cases}$$

In [15]:

```
Mhxs = [float(Mhx_konkret.subs(x,xi)) for xi in xs] #kiszámoljuk V(x) értékeit 
                                                    #és egyből 'float'-tá alakí
plt.plot(xs, Mhxs, lw = 3) # A V(x) függvény képének létrehozása 3-as vonalvast
plt.fill_between(xs, Mhxs, alpha=0.2) #Kiszínezi a függvény alatti területet
plt.xlabel("$x$ (m)") #x tengelyhez tartozó tengelyfelirat
plt.ylabel("$M_h(x)$ (N)") #y tengelyhez tartozó tengelyfelirat
plt.grid() #rács
plt.show() #kirajzolás
```

executed in 1.25s, finished 10:05:25 2020-05-01