Malujeme graf funkce

Budeme malovat graf funkce. Zvolíme součet Fourierovy řady

$$f(x) = \frac{a}{2} + \sum_{k=1}^{k_{\max}} \frac{\sin(ak)\ \cos(kx)}{k}$$

Tato funkce má jednak parametr a určující tvar výsledné funkce a pak také parametr $k_{\max}$, který říká, kolik členů řady sečíst.

V Pythonu jsou trigonometrické funkce v modulu `math`.

```
import math

def f(x, a = math.pi/2 , k_max = 30):
    s = a/2;
    for k in range(1,k_max+1):
        s += math.sin(a*k)*math.cos(k*x)/k
    return s
```

Malování gtrafu funkce představuje tabeleci hodnot x a f(x) v nějakém intervalu x. Mohli bychom tato data poslat do souboru a nechat vymalovat `gnuplotem` ale my použijeme "interní" malování dostupné přímo z použitého jazyka.

Proto místo výpisu na konsoli, naplníme příslušnými hodnotami dvě pole X a Y. Algoritmy pro práci s poli/seznamy budeme teprve probírat, ale zde potřebujeme pár elementárních operací:

`X = []` ... vytvoří prázdný seznam

`X.append(x)` ... přidá položku do seznamu

```
X = []
Y = []
x_min = -math.pi
x_max = math.pi
n_pts = 1200

for i in range(0,n_pts+1):
    x = x_min + i*(x_max-x_min)/n_pts
    X.append(x)
    Y.append(f(x,math.pi/2,300))
```

Postup pro vytvoření grafu je dán strukturou příslušné knihovny `matplotlib` a vyžaduje časté odvolávání se na dokumentaci

https://matplotlib.org/3.1.1/api/pyplot_summary.html

(Pokud vám přijde obrázek moc malý, použijte pod řádkem s `import` příkaz:

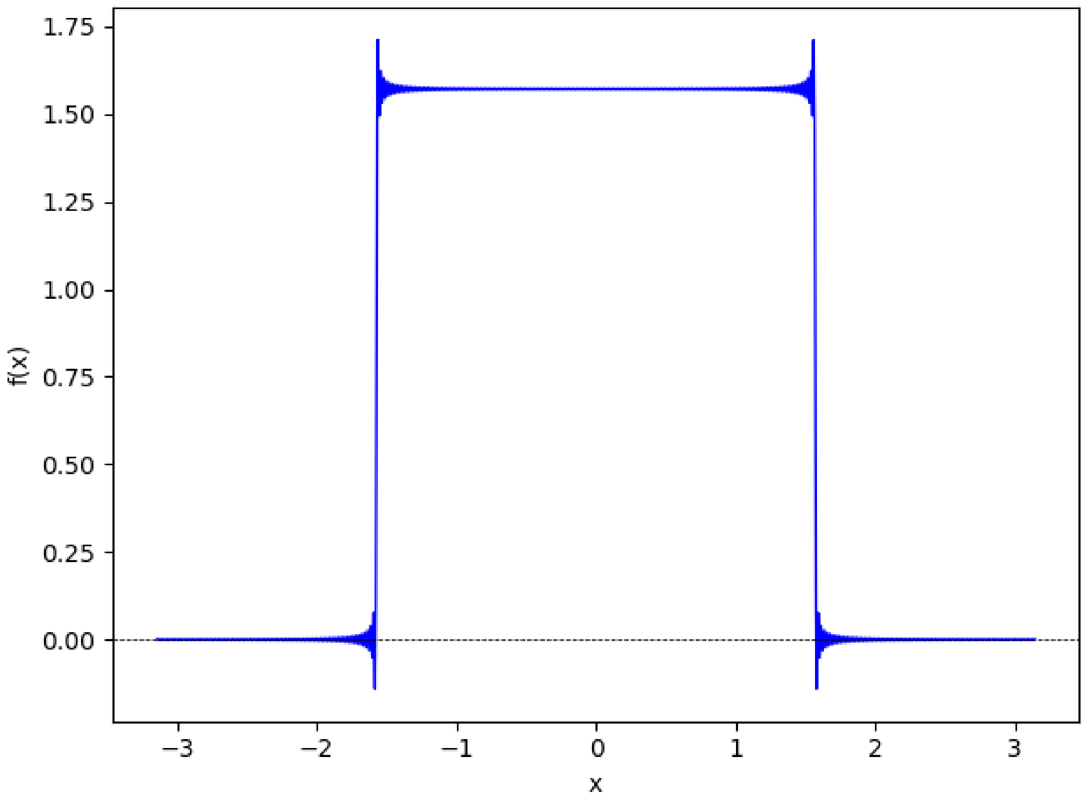
`plt.rcParams['figure.figsize'] = [12, 8]`

```
import matplotlib.pyplot as plt

plt.plot(X, Y, color='blue', linewidth=1)

plt.xlabel("x")
plt.ylabel("f(x)")
plt.axhline(y=0, color='k', linestyle='dashed', linewidth=0.5)

plt.show()
```



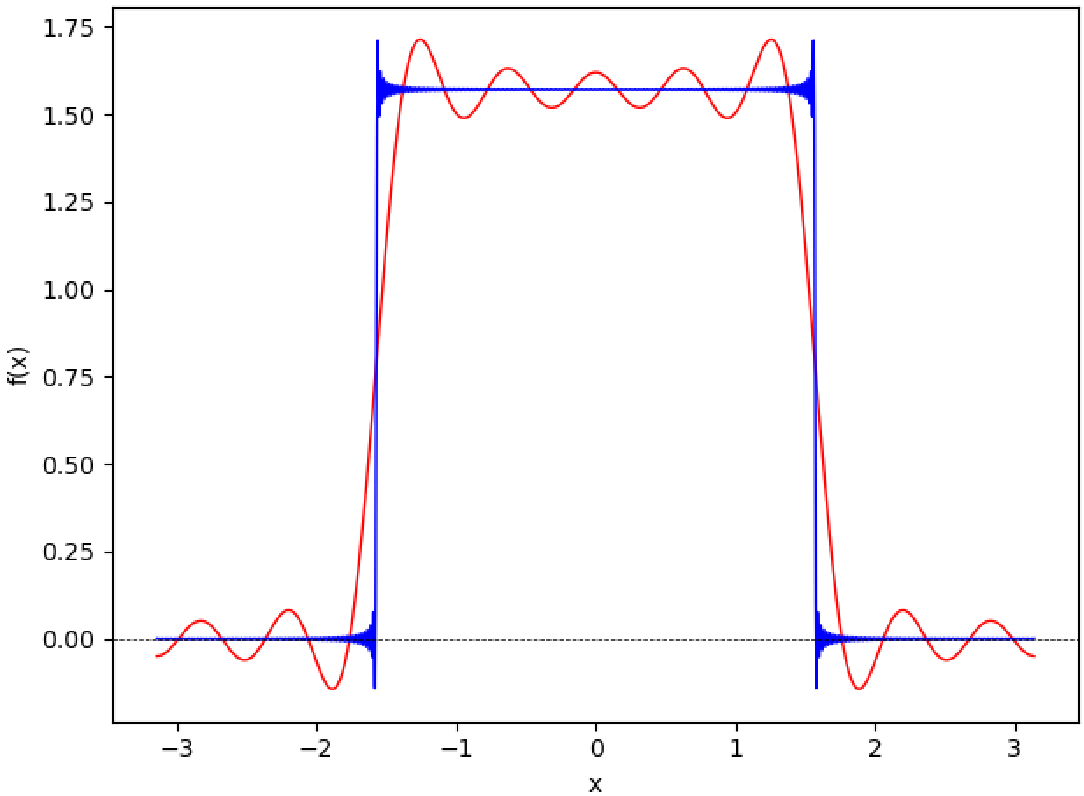
Moderní jazyky umožňují vytvořit pole i bez otravujícího `append`:

```
Z = [f(x,math.pi/2,10) for x in X]

plt.plot(X, Z, color='red', linewidth=1)
#následuje původní obrázek
plt.plot(X, Y, color='blue', linewidth=1)

plt.xlabel("x")
plt.ylabel("f(x)")
plt.axhline(y=0, color='k', linestyle='dashed', linewidth=0.5)


plt.show()
```



Problém

Na obrázku výše vidíme tzv. Gibbsův jev, tedy fakt, že způsob, jímž funkce konverguje k obdélníkovému tvaru zachová *překmity* pro libovolně velké `k_max`.

Vyzkoušejte, že pokud místo $\sum_{k=1}^{k_{max}} a_k f_k(x)$, kde v našem případě $f_k = \cos(kx)$, utneme sčítání řady kultivovaněji, např. tak, že vezmeme $\tilde{a}_k = (1 - k/k_{max})a_k$, budou překmity menší.