# 1. Knihovny a moduly pro matematické výpočty

**Zadání:**

V tomto kurzu jste se učili s některými vybranými knihovnami. Některé sloužily pro rychlé vektorové operace, jako numpy, některé mají naprogramovány symbolické manipulace, které lze převést na numerické reprezentace (sympy), některé mají v sobě funkce pro numerickou integraci (scipy). Některé slouží i pro rychlé základní operace s čísly (numba).

Vaším úkolem je změřit potřebný čas pro vyřešení nějakého problému (např.: provést skalární součin, vypočítat určitý integrál) pomocí standardního pythonu a pomocí specializované knihovny. Toto měření proveďte alespoň pro 5 různých úloh (ne pouze jiná čísla, ale úplně jiné téma) a minimálně porovnejte rychlost jednoho modulu se standardním pythonem. Ideálně proveďte porovnání ještě s dalším modulem a snažte se, ať je kód ve standardním pythonu napsán efektivně. ​

**Řešení:**

**1. úloha: Výpočet faktoriálu**

Tato úloha se zabývá výpočtem faktoriálu. Nejdříve bez pomoci knihoven a poté pomocí

knihovny SymPy. Časová náročnost je následující:

Výpočet (Python): 1.8854195330059156

Výpočet (NumPy): 0.1536074520117836

**2. úloha: Výpočet sin(x)**

Tentokrát porovnáváme 3 knihovny - Math, NumPy a SymPy.

Čas výpočtu (Python): 0.00030469500052277

Čas výpočtu (NumPy): 0.0011716900044120848

Čas výpočtu (Sympy): 0.6211337630084017

Python s pomocí knihovny math je zde nejrychlejší. Po něm je NumPy a poté SymPy.

**3. úloha: Násobení matic**

Tato úloha ukazuje efektivitu násobení matic o velikosti 3x3 pomocí NumPy.

Čas výpočtu (Python): 0.00018488299974706024

Čas výpočtu (NumPy): 0.00012147700181230903

Výsledky ukazují, že je knihovna NumPy výrazně rychlejší. Python využívá tři vnořené smyčky k výpočtu, zatímco NumPy používá vektorové operace pro rychlejší výpočet.

**4. úloha: Výpočet faktoriálu**

Tato úloha se zabývá výpočtem faktoriálu. Nejdřív pomocí Pythonu a poté pomocí

knihovny Math.

Čas výpočtu (Python): 0.15173245201003738

Čas výpočtu (Math): 0.01574059799895622

**5. úloha: Aritmetický průměr**

Tato úloha ukazuje délku výpočtu aritmetického průměru listu čísel o délce 3 524 287 čísel.

Čas výpočtu (Python): 0.031385743990540504

Čas výpočtu (NumPy): 0.009323335994849913

Ve výsledcích můžeme vidět, že výpočet za pomici NumPy je výrazně rychlejší. Python musí projít celý seznam a získat sumu prvků, zatímco NumPy vypočítá průměr pomocí své vestavěné funkce mean().