

# Příklady II - používání funkcí

■ Napište program pro numerickou integraci dané funkce, předpokládejte přitom, že integrační metodu dostanete předepsanou vzorcem. Postupujte v těchto třech krocích:

- napište funkci  $f$ , která bude počítat hodnotu dané funkce (jako např.  $e^{\sin(2x+3)}$  nebo  $\text{odmocnina } z \ (x^2 + 7x)$ )
- napište funkci *Numint* pro numerické integrování funkce  $f$  použitím daného vzorce (jako např. obdélníkové nebo lichoběžníkové pravidlo):
  - vstup:
    - $a, b$  ... interval integrace
    - $n$  ... počet integračních bodů uvnitř intervalu
  - výstup:
    - ierr* ... indikátor chyby (pro  $n < 0$  nebo  $f(x)$  nedefinovanou v některém z integračních bodů)
    - sumf* ... určitý integrál  $f(x)$  od  $a$  do  $b$  (pouze pokud není indikována chyba)
- použijte funkci *Numint* v hlavním programu, který
  - požádá o hodnoty  $a, b$  a  $n$
  - vypíše hodnotu integrálu (nebo informaci o chybě)

■ Napište program pro řešení rovnice  $x = f(x)$  pro danou funkci  $f(x)$ , který použije metodu prosté iterace. Postupujte v těchto třech krocích:

- napište funkci  $f$ , která bude počítat hodnotu dané funkce (jako např.  $0.2 \sin(2x+3)$  nebo  $0.5 \cos(x)$ )
- napište funkci *Solve* pro přibližné řešení rovnice  $x = f(x)$  s použitím funkce  $f$  definované výše a metody prosté iterace:  $x_{i+1} = f(x_i)$ 
  - vstup:
    - $x_o$  ... výchozí aproximace řešení
    - eps* ... požadovaná přesnost:  $|\text{f}(x) - x| < \text{eps}$
    - maxit* ... maximální počet iterací
  - výstup:
    - $x$  ... přibližné řešení
    - nit* ... počet iterací (nebo 0, pokud by byl překročen počet *maxit*)
- použijte funkci *Solve* v hlavním programu, který
  - požádá o hodnoty  $x_o$  a *eps* (*maxit* definujte jako konstantu v hlavním programu)
  - vypíše hodnotu  $x$  a počet iterací potřebných pro výpočet (nebo oznámení, že řešení s přesností *eps* nebylo během *maxit* iterací nalezeno)

Položte *maxit* = 100 a vyzkoušejte program pro *eps* = 0.0001 a výchozí hodnotu  $x_o = 0$

■ Napište program pro určení množství a ceny hnojiva pro zahradu dané velikosti. Hnojivo se dodává ve dvou baleních: buď 1 kg za 50 Kč nebo 3 kg za 120 Kč. Na čtvereční metr zahrady je potřeba 0.1 kg hnojiva. Postupujte v těchto dvou krocích:

- napište funkci *fert* která určí nejlevnější kombinaci daných dvou typů balení pro danou zahradu:
  - vstup:
    - $w1, w2$  ... váhy 2 typů balení (kg)
    - $p1, p2$  ... ceny balení (Kč)
    - g\_size* ... velikost zahrady (m<sup>2</sup>)
    - d\_m* ... dávkování na m<sup>2</sup>
  - výstup:
    - nw1, nw2* ... počet balení obou typů
    - t\_am* ... celková cena
- použijte funkci *fert* v hlavním programu, který v cyklu
  - se zeptá na velikost zahrady (váhy a ceny dvou typů balení a dávkování hnojiva definujte jako konstanty v hlavním programu)
  - vypíše počet potřebných 1 kg a 3 kg balení a celkovou cenudokud není zadána jako velikost zahrady 0.

■ Napište program, který řeší vektorovou rovnici  $\mathbf{x} = \mathbf{U} \mathbf{x} + \mathbf{v}$  metodou prosté iterace, kde  $\mathbf{U}$  je  $n \times n$  matice a  $\mathbf{x}$  a  $\mathbf{v}$  jsou sloupcové vektory dimenze  $n$ . Postupujte v těchto dvou krocích:

- napište funkci *Iter* pro řešení rovnice  $\mathbf{x} = \mathbf{U} \mathbf{x} + \mathbf{v}$  použitím metody prosté iterace (tj.  $\mathbf{x_{i+1}} = \mathbf{U} \mathbf{x_i} + \mathbf{v}$  ) dokud není splněna podmínka  $|\mathbf{x_{i+1}} - \mathbf{x_i}| < \textit{eps}$  nebo dokud není překročen maximální povolený počet iterací *maxit*
  - vstup:
    - $\mathbf{x_o}$  ... výchozí aproximace řešení
    - $\mathbf{U}, \mathbf{v}$  ... daná matice a sloupcový vektor
    - eps* ... požadovaná přesnost:  $|\mathbf{x_{i+1}} - \mathbf{x_i}| < \textit{eps}$
    - maxit* ... max. povolený počet iterací
  - výstup:
    - $\mathbf{x}$  ... přibližné řešení
    - nit* ... počet iterací (nebo 0, pokud byl překročen počet *maxit* )
- použijte funkci *Iter* v hlavním programu, který
  - přečte matici  $\mathbf{U}$  a vektor  $\mathbf{v}$  z klávesnice nebo ze souboru (*maxit* a *eps* definujte jako konstanty v programu)
  - zvolí výchozí aproximaci  $\mathbf{x_o}$  jako nulový vektor a přibližně vyřeší rovnici  $\mathbf{x} = \mathbf{U} \mathbf{x} + \mathbf{v}$
  - vytiskne přibližné řešení  $\mathbf{x}$  a kolik iterací bylo potřeba pro jeho výpočet (nebo oznámí, že řešení s přesností *eps* nebylo během *maxit* iterací nalezeno)