Zápisky k předmětu Využití počítačů ve fyzice

Pavel Stránský

9. března 2021

Obsah

1 Obyčejné diferenciální rovnice **1**
- 1.1 Diferenciální rovnice prvního řádu . . . 1
  - 1.1.1 Pár důležitých pojmů . . . 1
  - 1.1.2 Eulerova metoda 1. řádu . . . 2
  - 1.1.3 Eulerova metoda 2. řádu . . . 2
- 1.2 Runge-Kuttova metoda 4. řádu . . . 3

1 Obyčejné diferenciální rovnice

1.1 Diferenciální rovnice prvního řádu

Nejprve se budeme věnovat řešení jedné diferenciální rovnice prvního řádu,

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y, t) \tag{1}$$

s počáteční podmínkou

$$y(t_0) = y_0. \tag{2}$$

Zde $y = y(t)$ je hledaná funkce a t je nezávisle proměnná.

Numerické řešení diferenciální rovnice spočívá v nahrazení infinitezimálních přírůstků přírůstky konečnými:

$$\frac{\Delta y}{\Delta t} = \phi(y, t) \tag{3}$$

kde ϕ je funkce, která udává směr, podél kterého se při numerickém řešení vydáme. Volbá této funkce je klíčová a záleží na ní, jak přesné řešení dostaneme a jak rychle ho dostaneme.

1.1.1 Pár důležitých pojmů

- **Explicitní algoritmy:** K výpočtu hodnoty funkce y_{i+1} se vyžadují pouze hodnoty z aktuálních a minulých kroků, tj. y_i, y_{i-1}, atd.

- **Jednokrokové algoritmy:** K výpočtu hodnoty funkce y_{i+1} v následujícím kroku vyžadují pouze znalost hodnoty funkce v aktuálním kroku y_i. Rozepsáním (3) dostaneme

$$\boxed{y_{i+1} = y_i + \underbrace{\phi(y_i, t)}_{\phi_i} \Delta t}, \tag{4}$$

  přičemž počáteční hodnota y_0 je dána počáteční podmínkou. My se omezíme pouze na tyto algoritmy.

- **Lokální diskretizační chyba:**
$$\mathcal{L} = y(t + \Delta t) - y(t) - \phi(y(t), t)\Delta t, \tag{5}$$
kde $y(t)$ udává přesné řešení v čase t.

- **Akumulovaná diskretizační chyba:**
$$\epsilon_i = y_i - y(t_i) \tag{6}$$

- **Řád metody:** Metoda je p-tého řádu, pokud
$$L(\Delta t) = \mathcal{O}(\Delta t^{p+1}). \tag{7}$$

- **Kontrola chyby řešení:** Chybu numerického řešení diferenciální rovnice lze zmenšit 1) menším krokem, 2) lepší metodou (metodou vyššího řádu). Menší krok však znamená vyšší výpočetní čas. Sofistikované metody proto průběžně mění velikost kroku: když se funkce mění pomalu, krok prodlouží, když se mění rychle, krok zkrátí (tzv. **metody s adaptivním krokem**). Tím se docílí vysoké přesnosti při co nejmenším výpočetním čase.

1.1.2 Eulerova metoda 1. řádu

$$\phi_i = f(y_i, t_i), \tag{8}$$

tj. krok do y_{i+1} děláme vždy ve směru tečny v bodě y_i.

- Nejjednodušší metoda integrace diferenciálních rovnic.
- Chyba je obrovská, k dosažení přesných hodnot je potřeba velmi malého kroku, což znamená dlouhý výpočetní čas.

1.1.3 Eulerova metoda 2. řádu

$$\begin{aligned} k_1 &= f(y_i, t_i) \\ k_2 &= f\left(y_i + k_1 \Delta t, t + \Delta t\right) \\ \phi_i &= \frac{1}{2}\left(k_1 + k_2\right), \end{aligned} \tag{9}$$

tj. uděláme jednoduchý Eulerův krok ve směru k_1, spočítáme derivaci k_2 po tomto kroku a vyrazíme z bodu y_i ve směru, který je průměrem obou směrů (doporučuji si nakreslit obrázek).

Ekvivalentní je udělat „Eulerův půlkrok“ a vyrazit z bodu y_i ve směru derivace spočtené po tomto půlkroku:

$$\begin{aligned} k_1' &= f(y_i, t_i) \\ k_2' &= f\left(y_i + k_1' \frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\ \phi_i &= k_2' \end{aligned} \tag{10}$$

1.2 Runge-Kuttova metoda 4. řádu

$$\begin{aligned}
k_1 &= f(y_i, t_i) \\
k_2 &= f\left(y_i + k_1\frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\
k_3 &= f\left(y_i + k_2\frac{\Delta t}{2}, t + \frac{\Delta t}{2}\right) \\
k_4 &= f\left(y_i + k_3\Delta t, t + \Delta t\right) \\
\phi_i &= \frac{1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)
\end{aligned} \tag{11}$$

- Jedna z nejčastěji používaných metod.
- Vysoká rychlost a přesnost při relativní jednoduchosti.
- Existují i Runge-Kuttovy metody vyššího řádu p, avšak vyžadují výpočet více než p dílčích derivací k_j. Obecně platí, že metoda řádu $p \leq 4$ vyžaduje p derivací, metoda řádu $5 \leq p \leq 7$ vyžaduje $p+1$ derivací a metoda řádu $p = 8, 9$ vyřaduje $p+2$ derivací.

Úkol 1.1: *Naprogramujte Eulerovu metodu 1. a 2. řádu a Runge-Kuttovu metodu. Vyřešte diferenciální relaxační rovnici*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -y \tag{12}$$

s počátečními podmínkami $y_0 = 1$ (analytickým řešením je funkce e^{-t}). Integrační krok Δt ponechte jako volný parametr. Nakreslete grafy řešení $y(t)$ pro rozdílné hodnoty integračních kroků, například $\Delta t = 0.01$ a $\Delta t = 0.1$ pro čas $t \in \langle 0; 10\rangle$.

Úkol 1.2: *Rozšiřte kód tak, aby počítal průměrnou kumulovanou chybu*

$$\mathcal{E} = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}\left(y_i - \mathrm{e}^{-t_i}\right)^2} \tag{13}$$

a nakreslete závislost $\mathcal{E}(\Delta t)$ pro $\Delta t \in \langle 0.002; 0.1\rangle$ a pro různé metody. Jelikož očekáváme mocninnou závislost dle (7), kde exponent je tím větší, čím větší je řád metody, je výhodné graf $\mathcal{E}(\Delta t)$ kreslit v log-log měřítku. V Pythonu použijete místo `plot(...)` *funkci* `loglog(...)` *z knihovny* `matplotlib.pyplot`*. Ověřte, že získané křivky jsou v souladu s řády použitých metod.*

Úkol 1.3: *Pomocí naprogramovaných metod vyřešte nelineární diferenciální rovnici*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \sin\left(ty\right) \tag{14}$$

s počáteční podmínkou $y_0 = 1$ a vykreslete graf jejího řešení.