Diferenciální rovnice jako modely

Matematické metody pro ITS (11MAMY)

Jan Přikryl

3. přednáška 11MAMY

středa 23. března 2022

verze: 2022-03-23 09:36

Ústav aplikované matematiky
ČVUT v Praze, Fakulta dopravní

Obsah přednášky

Obyčejné diferenciální rovnice

Existence, jedinečnost a podmíněnost

Numerické řešení ODR

Stabilita řešení ODR

Eulerova metoda

Přesnost a stabilita

Implicitní metody

Diferenciální rovnice

- Diferenciální rovnice zahrnují i derivace funkce, která je neznámou
- **Obyčejná diferenciální rovnice** (ODR): všechny derivace jsou vzhledem k jedné nezávislé proměnné, často představující čas
- Řešením diferenciální rovnice je spojitá *funkce* z nekonečně dimenzionálního prostoru funkcí
- Přibližné, numerické řešení diferenciálních rovnic je založeno na konečně-dimenzionální aproximaci
- Diferenciální rovnice je nahrazena algebraickou rovnicí, jejíž řešení aproximuje danou diferenciální rovnici

Řád ODR

- **Řád** rovnice je určen nejvyšším řádem derivace funkce řešení v ODR
- ODR s derivacemi vyššího řádu lze transformovat na ekvivalentní soustavu rovnic prvního řádu
- Budeme proto diskutovat pouze numerické metody řešení pro ODR prvního řádu
- Většina softwaru pro řešení ODR je navržena tak, aby řešila pouze rovnice prvního řádu

- Pro ODR k-tého řádu

$$y^{(k)}(t) = f(t, y, y', \ldots, y^{(k-1)})$$

definujme k nových neznámých funkcí

$$u_1(t) = y(t),\; u_2(t) = y'(t),\; \ldots,\; u_k(t) = y^{(k-1)}(t)$$

- Původní ODR k-tého řádu je pak ekvivalentní soustavě k ODR prvního řádu

$$\begin{bmatrix} u_1'(t) \\ u_2'(t) \\ \vdots \\ u_{k-1}'(t) \\ u_k'(t) \end{bmatrix} = \begin{bmatrix} u_2(t) \\ u_3(t) \\ \vdots \\ u_k(t) \\ f(t, u_1, u_2, \ldots, u_k) \end{bmatrix}$$

Příklad: Druhý Newtonův zákon

- Newtonův druhý pohybový zákon, $F = ma$, je ODR druhého řádu, protože zrychlení a je druhá derivace souřadnic polohy, které označujeme y
- ODR má tedy tvar

$$\ddot{y} = F/m,$$

  kde F je působící síla a m hmotnost tělesa
- Definováním $u_1 = y$ a $u_2 = \dot{y}$ vznikne ekvivalentní soustava dvou ODR prvního řádu

$$\begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \end{bmatrix} = \begin{bmatrix} u_2 \\ F/m \end{bmatrix}$$

- K řešení této soustavy lze použít metody pro řešení rovnic prvního řádu
- První složka řešení u_1 je řešení y původní rovnice druhého řádu
- Druhou složkou řešení u_2 je rychlost $\dot{y}$

- Obecný systém prvního řádu ODR má tvar

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)) \qquad \text{respektive} \qquad \dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$$

kde $\mathbf{y} : \mathbb{R} \to \mathbb{R}^n$, $\mathbf{f} : \mathbb{R}^{n+1} \to \mathbb{R}^n$, a $\mathbf{y}'(t) = \mathrm{d}\mathbf{y}(t)/\mathrm{d}t$ respektive $\dot{\mathbf{y}}$ značí derivaci vzhledem k t,

$$\mathbf{y}'(t) = \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = \begin{bmatrix} \mathrm{d}y_1(t)/\mathrm{d}t \\ \mathrm{d}y_2(t)/\mathrm{d}t \\ \vdots \\ \mathrm{d}y_n(t)/\mathrm{d}t \end{bmatrix} = \begin{bmatrix} f_1(t, \mathbf{y}(t)) \\ f_2(t, \mathbf{y}(t)) \\ \vdots \\ f_n(t, \mathbf{y}(t)) \end{bmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

- Je dána funkce $\mathbf{f}$ a my chceme určit neznámou funkci $\mathbf{y}$ splňující ODR
- Pro zjednodušení budeme často uvažovat speciální případ jednoduché skalární ODR, $n = 1$

Obsah přednášky

Obyčejné diferenciální rovnice

Existence, jedinečnost a podmíněnost

Numerické řešení ODR

Stabilita řešení ODR

Eulerova metoda

Přesnost a stabilita

Implicitní metody

Obsah přednášky

Obyčejné diferenciální rovnice

Existence, jedinečnost a podmíněnost

Numerické řešení ODR

- Eulerova metoda
- Přesnost a stabilita
- Implicitní metody
- Tuhost
- Metody Taylorovy řady
- Metody Runge-Kutta

Počáteční úlohy

- ODR $\dot{y} = f(t, y)$ samo o sobě neurčuje funkci $y(t)$, která by byla jedinečným řešením rovnice
- Důvodem je, že ODR pouze specifikuje *směrnici* $y'(t)$ řešení v každém bodě, ale ne skutečnou hodnotu $y(t)$ v tomto bodě
- Pokud je tedy $y(t)$ řešení a c nějaká konstanta, pak $y(t) + c$ je také řešení, protože $\mathrm{d}(y(t) + c)/\mathrm{d}t = y(t) + 0 = y(t)$
- Je-li f dostatečně hladká, ODR obecně splňuje nekonečně mnoho funkcí stejné rodiny
- Pro určení jednoho konkrétního řešení z této nekonečné množiny musí být ještě v určitém bodě t_0 specifikována hodnota $y_0 = y(t_0)$

Počáteční úlohy, pokračování

- Součástí zadání problému je proto požadavek, aby $y(t_0) = y_0$ – ten určuje jedinečné řešení ODR
- Z důvodu interpretace nezávislé proměnné t jako času uvažujeme často t_0 jako počáteční čas a y_0 jako počáteční hodnotu
- Proto se této úloze říká **počáteční úloha** respektive **problém počáteční hodnoty**
- Požadavku $y(0) = y_0$ říkáme **počáteční podmínka**
- ODR jako model dynamického systému popisuje vývoj tohoto systému v čase od jeho počátečního stavu y_0 v čase t_0 kupředu; hledáme funkci $y(t)$, která popisuje stav systému jako funkce času

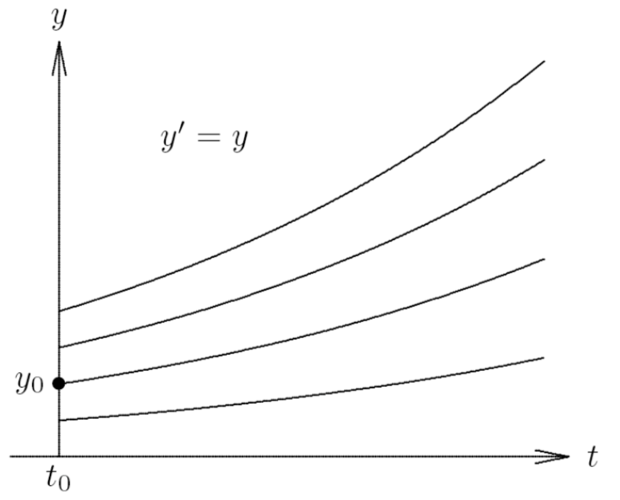
Příklad: Problém s počáteční hodnotou

- Uvažujme skalární ODR
$$\dot{y} = y$$
- Množina řešení je dána $y(t) = c\,\mathrm{e}^t$, kde $c \in \mathbb{R}$ je jakákoli konstanta
- Zavedení počáteční podmínky $y(t_0) = y_0$ vybere jedinečné řešení
- Pokud je pro výše uvedený příklad $t_0 = 0$, pak $y_0 = c\,\mathrm{e}^0$ neboli $c = y_0$, což znamená, že hledané řešení ODR je $y(t) = y_0\,\mathrm{e}^t$

Příklad: Problém s počáteční hodnotou

Množina řešení pro ODR

$\dot{y} = y$



Obsah přednášky

Stabilita řešení

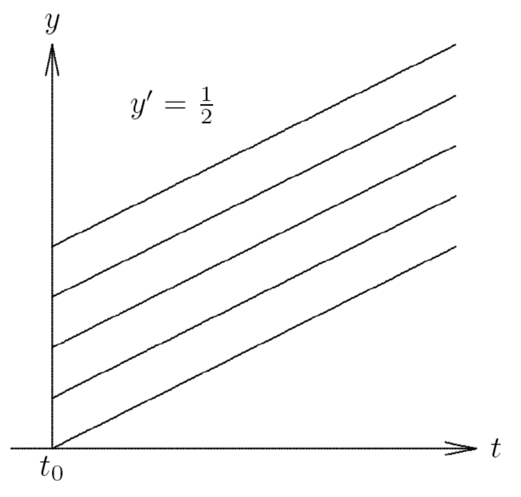
Řešení ODR je

- **Stabilní**, pokud řešení vzniklá perturbací počáteční hodnoty zůstávají blízko původního řešení
- **Asymptoticky stabilní**, pokud jsou řešení pro porušené počáteční podmínky , ale konvergují zpět k původnímu řešení
- **Nestabilní**, pokud řešení po změně počátečních podmínek divergují od originálního řešení nade všechny meze

Příklad: Stabilní řešení

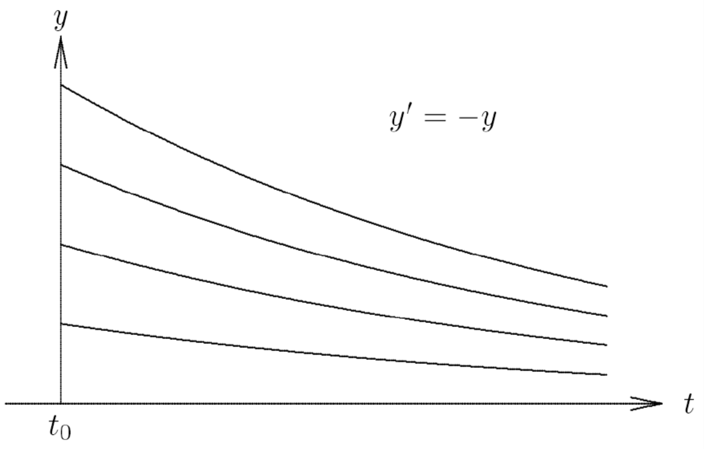
Množina řešení pro ODR

$\dot{y} = \frac{1}{2}$



Příklad: Asymptoticky stabilní řešení

Množina řešení pro ODR
$\dot{y} = -y$



Příklad: Stabilita skalární ODR

Uvažujme skalární ODR $\dot{y} = \lambda y$, kde λ je konstanta

- Řešení je dáno $y(t) = y_0 \, \mathrm{e}^{\lambda t}$, kde $t_0 = 0$ je počáteční čas a $y(0) = y_0$ je počáteční hodnota
- Pro reálné λ platí:
  - $\lambda > 0$: všechna nenulová řešení rostou exponenciálně, takže každé řešení je nestabilní
  - $\lambda < 0$: všechna nenulová řešení exponenciálně klesají, takže každé řešení je nejen stabilní, ale asymptoticky stabilní
- Pro komplexní λ:
  - $\Re(\lambda) > 0$: nestabilní
  - $\Re(\lambda) < 0$: asymptoticky stabilní
  - $\Re(\lambda) = 0$: stabilní, ale ne asymptoticky stabilní

Příklad: Lineární soustava ODR

- Lineární, homogenní soustava ODR s konstantními koeficienty má tvar

$$\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}$$

kde $\mathbf{A}$ je $n \times n$ matice a počáteční podmínka je $\mathbf{y}(0) = \mathbf{y}_0$

- Předpokládejme, že $\mathbf{A}$ je diagonalizovatelné, s vlastními čísly λ_i a odpovídajícími vlastními vektory $\mathbf{v}_i$, $i = 1, \ldots, n$
- Vyjádříme $\mathbf{y}_0$ jako lineární kombinaci vlastních vektorů $\mathbf{y}_0 = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i$
- Pak

$$\mathbf{y}(t) = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i e^{\lambda_i t}$$

je řešení ODR splňující počáteční podmínku $\mathbf{y}(0) = \mathbf{y}_0$

Příklad: Lineární soustava ODR, pokračování

- Vlastní čísla **A** s *kladnými* reálnými částmi reprezentují exponenciálně *rostoucí* komponenty řešení
- Vlastní čísla se *zápornými* reálnými částmi odpovídají exponenciálně *klesajícím* komponentám řešení
- Vlastní čísla s *nulovými* skutečnými částmi (tj. čistě imaginární) odpovídají *oscilujícím* komponentám řešení

Řešení soustavy je stabilní, pokud $\Re(\lambda_i) \leq 0$ pro každé vlastní číslo a asymptoticky stabilní, pokud $\Re(\lambda_i) < 0$ pro každé vlastní číslo, ale nestabilní pokud pro nějaké vlastní číslo $\Re(\lambda_i) > 0$.

Stabilita řešení, obecný případ

- Pro obecnou nelineární soustavu ODR $\dot{\mathbf{y}} = f(t, \mathbf{y})$ je určení stability řešení komplikovanější
- ODR lze lokálně linearizovat okolo řešení $y(t)$ částečným rozvojem Taylorovy řady, poskytujícím lineární ODR
$$\dot{\mathbf{z}} = \mathbf{J}_f(t, \mathbf{y})\mathbf{z}$$
kde $\mathbf{J}_f$ je matice Jakobiánu funkce f vzhledem k $\mathbf{y}$
- Vlastní čísla $\mathbf{J}_f$ určují lokální stabilitu, ale takto učiněné závěry nemusí být globálně platné

Obsah přednášky

Numerické řešení ODR

- Analytickým řešením ODR je vzorec, který může být vyhodnocen v libovolném bodě t
- Numerickým řešením ODR je tabulka přibližných hodnot řešení v diskrétní množině bodů
- Numerické řešení je generováno simulací chování systému popsaného danou ODR
- Počínaje t_0 s danou počáteční hodnotou y_0 sledujeme trajektorii řešení, předepsanou simulovanou diferenciální rovnicí
- Vyhodnocení $f(t_0, y_0)$ nám říká směrnici trajektorie v bodě $[t_0, y_0]$
- Tyto informace používáme k předpovědi hodnoty y_1 řešení v budoucím čase $t_1 = t_0 + h$ pro nějaký vhodně zvolený časový přírůstek h

Numerické řešení ODR, pokračování

- Přibližné hodnoty řešení jsou generovány krok za krokem po těchto přírůstcích a pohybují se napříč intervalem, ve kterém je hledáno řešení
- Při přechodu z jednoho diskrétního bodu do dalšího nám vznikne nějaká chyba, což znamená, že další přibližná hodnota řešení leží na *odlišném řešení* od toho, na němž jsme začali
- Stabilita nebo nestabilita řešení ODR částečně také určuje, zda se takovéto chyby s časem zvětšují nebo zmenšují

Eulerova metoda

- Pro obecnou soustavu ODR $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ uvažujme Taylorův rozvoj

$$\begin{aligned} \mathbf{y}(t+h) &= \mathbf{y}(t) + h\,\mathbf{y}'(t) + \frac{h^2}{2}\mathbf{y}''(t) + \frac{h^3}{6}\mathbf{y}''(t) + \cdots \\ &= \mathbf{y}(t) + h\,\mathbf{f}(t, \mathbf{y}) + \cdots \end{aligned}$$

- **Eulerova metoda** vznikne zanedbáním členů druhého a vyšších řádů za účelem získání přibližného řešení

$$\mathbf{y}_{k+1}(t) = \mathbf{y}_k(t) + h\,\mathbf{f}(t_k, \mathbf{y}_k)$$

- Eulerova metoda postupuje řešením tak, že **extrapoluje** podél přímky, jejíž směrnice je dána hodnotou $\mathbf{f}(t_k, \mathbf{y}_k)$
- Eulerova metoda je **jednokroková**, protože výpočet postupu do dalšího bodu závisí pouze na informacích z jednoho předešlého kroku

Příklad: Eulerova metoda

- Použijeme-li Eulerovu metodu na skalární ODR $\dot{y} = y$ s velikostí kroku h, postup řešení od času $t_0 = 0$ do času $t_1 = t_0 + h$ je

$$y_1 = y_0 + hy'(t_0) = y_0 + hy_0 = (1 + h)y_0$$

- Hodnota řešení, kterou získáme pro t_1, není přesná, $y_1 \neq y(t_1)$
- Například pokud $t_0 = 0$, $y_0 = 1$ a $h = 0{,}5$, pak $y_1 = 1{,}5$, přičemž přesné řešení pro tuto počáteční hodnotu je $y(0{,}5) = \exp(0{,}5) \approx 1{,}649$

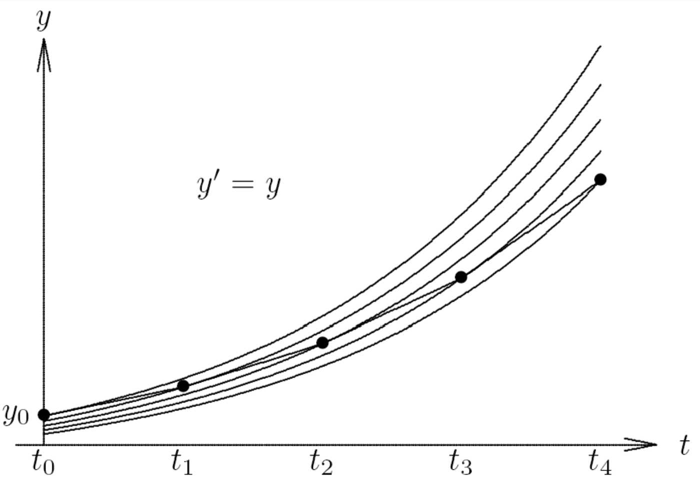
$$\Downarrow$$

- Hodnota y_1 tedy leží na jiné funkci řešení, než na které jsme začali!

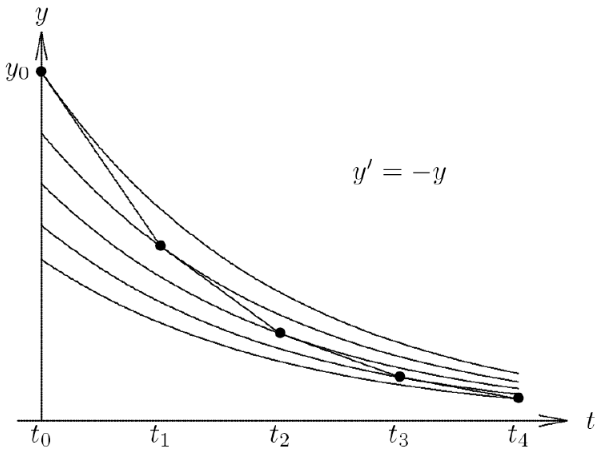
- Budeme-li pokračovat v procesu numerického řešení, uděláme další krok od t_1 do $t_2 = t_1 + h = 1{,}0$ a obdržíme $y_2 = y_1 + hy_1 = 1{,}5 + 0{,}5 \cdot 1{,}5 = 2{,}25$
- Nyní se y_2 liší
  - nejen od skutečného řešení původního problému v $t = 1{,}0$, $y(1{,}0) = \exp(1{,}0) \approx 2{,}718$, ale také
  - od řešení, které je přesné v předchozím bodě $[t_1, y_1]$; to má pro $t = 1{,}0$ hodnotu přibližně 2,473
- Tak jsme se opět přesunuli k další funkci řešení pro tuto ODR

Příklad: Nestabilní řešení

U nestabilních řešení chyby v numerickém řešení rostou s časem



U stabilních řešení se chyby v numerickém řešení mohou časem zmenšit



Chyby v numerickém řešení ODR

Numerické metody řešení ODR jsou zatíženy dvěma odlišnými typy chyb:

- **Zaokrouhlovací chyba** je způsobena konečnou přesností aritmetiky v pohyblivé řádové čárce
- **Chyba metody** je způsobena použitou aproximační metodou; přetrvává i v případě, kdy by všechny aritmetické výpočty byly přesné

V praxi je chyba metody dominantním faktorem určujícím přesnost numerických řešení ODR, zaokrouhlovací chyby proto pro jednoduchost zanedbáme.

Globální chyba a lokální chyba

Chybu metody v kterémkoli bodě t_k lze rozdělit na

- **Globální chybu**, rozdíl mezi vypočítaným řešením a skutečnou hodnotou funkce řešení $\mathbf{y}(t)$ procházející počátečním bodem $[t_0, \mathbf{y}_0]$,

$$\mathbf{e}_k = \mathbf{y}_k - \mathbf{y}(t_k)$$

- **Lokální chybu**, což je chyba jednoho kroku numerické metody

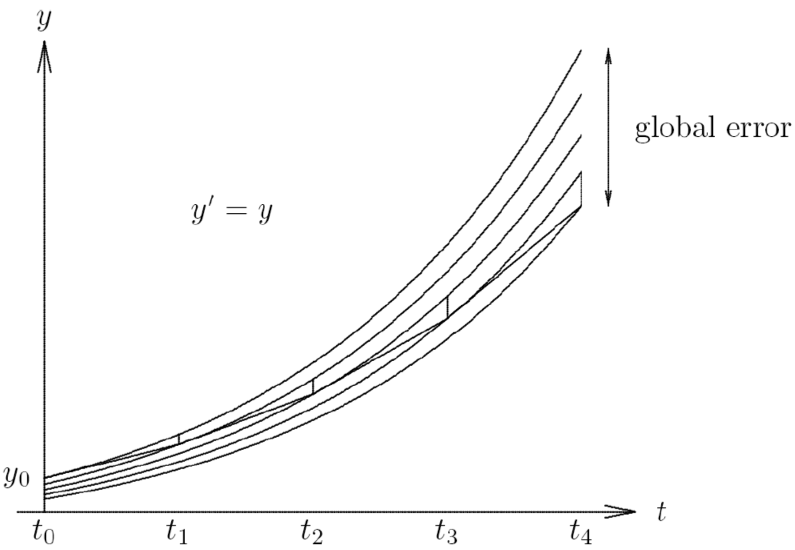
$$\ell_k = \mathbf{y}_k - \mathbf{u}_{k-1}(t_k)$$

kde $\mathbf{u}_{k-1}(t)$ je skutečné řešení procházející předchozím bodem $[t_{k-1}, \mathbf{y}_{k-1}]$

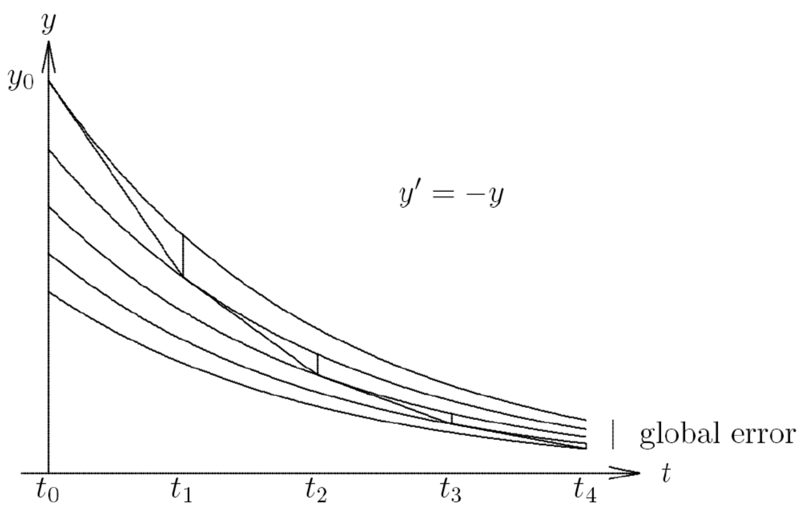
Globální chyba a lokální chyba, pokračování

- Globální chyba nemusí být nutně součtem lokálních chyb
- Globální chyba je obecně větší, než součet lokálních chyb, pokud se jedná o nestabilní řešení, pokud je ale řešení stabilní, může být i nižší, než tento součet
- Obecně chceme dosáhnout malé globální chyby, přímo ale můžeme ovládat pouze lokální chybu

Příklad: Nestabilní řešení



Příklad: Stabilní řešení



Obsah přednášky

- **Řád přesnosti numerické metody** je p, pokud

$$\ell_k = \mathcal{O}(h_k^{p+1})$$

- Lokální chyba na jednotku kroku je potom řádu $\ell_k = \mathcal{O}(h_k^p)$
- Za přiměřených podmínek $\mathbf{e}_k = \mathcal{O}(h^p)$, kde h je *průměrná velikost* kroku

Stabilita numerické metody

- Numerická metoda je **stabilní**, pokud malé poruchy nezpůsobí divergenci výsledných numerických řešení nade všechny meze
- Taková divergence numerických řešení může být způsobena nestabilitou řešení ODR, ale může být také způsobena samotnou numerickou metodou, jež se takto chová i když jsou řešení ODR stabilní

Stanovení stability a přesnosti

- Jednoduchý postup určení stability a přesnosti numerické metody je použít ji na skalární ODR $\dot{y} = \lambda y$, kde λ je (případně komplexní) konstanta
- Přesné analytické řešení je $y(t) = y_0 \, \mathrm{e}^{\lambda t}$, kde $y(0) = y_0$ je počáteční stav
- *Stabilitu numerické metody* určíme pomocí charakteristiky růstu hodnot numerického řešení
- *Přesnost numerické metody* určíme porovnáním přesného a numerického řešení

- Použijeme-li na rovnici $\dot{y} = \lambda y$ Eulerovu metodu s pevnou velikostí kroku h, máme

$$y_{k+1} = y_k + h\lambda y_k = (1 + h\lambda) y_k$$

  což znamená, že

$$y_k = (1 + h\lambda)^k y_0$$

- Je-li $\Re(\lambda) < 0$, přesné řešení se s rostoucím t blíží k nule a to samé platí i pro numerické řešení, pokud ovšem

$$|1 + h\lambda| < 1.$$

  Tato podmínka je splněna v případech, kdy $h\lambda$ leží v komplexní rovině uvnitř kruhu o poloměru 1 se středem v -1.

Eulerova metoda, pokračovala

- Pokud je λ reálné, pak $h\lambda$ musí ležet v intervalu $(2, 0)$, takže pro $\lambda < 0$ musíme mít

$$h \leq -\frac{2}{\lambda}$$

  aby byla Eulerova metoda stabilní
- **Růstový faktor** $1 + h\lambda$ odpovídá částečnému rozvoji řady

$$\mathrm{e}^{h\lambda} = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \cdots$$

  pomocí členu prvního řádu v h – Eulerova metoda je tedy prvního řádu přesnosti

Obecně platí, že růstový faktor závisí na

- Numerické metodě, která určuje tvar růstového faktoru
- Jakobiánu $\mathbf{J}_f$, jenž je dán konkrétní ODR
- Velikostí kroku h

Výběr velikosti kroku

- Při výběru velikosti kroku pro postupné numerické řešení ODR chceme použít co největší krok, abychom snížili výpočetních nároky; musíme ale také vzít v úvahu stabilitu i přesnost použité metody
- Smysluplné řešení získáme pouze v případě, kdy velikost kroku vyhovuje všem omezením stability
- K zajištění požadované přesnosti výsledku je navíc zapotřebí použít odhad lokální chyby

Výběr velikosti kroku, pokračování

- Například u Eulerovy metody je lokální chyba přibližně $(h_k^2/2)\ddot{y}$, takže volíme velikost kroku h tak, abychom pro toleranci ε splnili

$$h_k \leq \sqrt{2\varepsilon/\|\ddot{y}\|}$$

- Hodnotu $\ddot{y}$ přitom neznáme, můžeme ji ale odhadnout poměrem rozdílů

$$\ddot{y} \approx \frac{\dot{y}_k - \dot{y}_{k-1}}{t_k - t_{k-1}}$$

- Další metody získávání odhadů chyb jsou založeny na rozdílech mezi výsledky získanými metodami různých řádů nebo s různou velikostí kroku

Obsah přednášky

- Eulerova metoda je **explicitní** v tom, že používá pouze informace v čase t_k k posunu řešení do času t_{k+1}
- Může se to zdát vhodné, Eulerova metoda má ale poměrně omezenou oblast stability
- Větší oblasti stability lze získat zužitkováním informací v čase t_{k+1}, což činí metodu **implicitní**
- Nejjednodušším příkladem je **zpětná Eulerova metoda**

$$y_{k+1} = y_k + h_k f(t_{k+1}, y_{k+1})$$

- Metoda je implicitní proto, že v ní musíme vyhodnotit f s argumentem y_{k+1}, který ovšem neznáme

Implicitní metody, pokračování

- Fakt, že y_{k+1} neznáme, není ale významnou překážkou
- K určení y_{k+1} musíme „pouze“ vyřešit (nelineární) algebraickou rovnici

$$y_{k+1} - h_k f(t_{k+1}, y_{k+1}) - y_k = 0$$

  kde známe h_k, y_k, f i t_{k+1}.
- K nalezení y_{k+1} typicky používáme nějakou iterační metodu pro hledání kořenů funkce, jako je Newtonova metoda nebo iterace v pevném bodě
- Dobrý počáteční odhad y_{k+1} pro iterační výpočet lze získat z explicitní metody (jako je přímá Eulerova metoda), nebo z řešení v minulém kroku

Příklad: Zpětná Eulerova metoda

- Uvažujme nelineární skalární ODR $\dot{y} = -y^3$ s počáteční podmínkou $y(0) = 1$
- Pomocí zpětné Eulerovy metody s velikostí kroku $h = 0{,}5$ získáme implicitní rovnici $y_1 = y_0 + hf(t_1, y_1) = 1 - 0{,}5y_1^3$ pro hodnotu řešení v dalším kroku
- Tuto nelineární rovnici pro y_1 lze vyřešit iterací v pevném bodě nebo Newtonovou metodou
- Pro získání počátečního odhadu pro y_1 bychom mohli použít předchozí hodnotu řešení, $y_0 = 1$, nebo bychom mohli použít explicitní metodu, jako je Eulerova metoda, což dává $y_1 = y_0 - 0{,}5y_0^3 = 0{,}5$
- Iterace nakonec konvergují na konečnou hodnotu $y_1 \approx 0{,}7709$

Implicitní metody, pokračování

- Vzhledem k dodatečným těžkostem a nárůstu složitosti výpočtu při použití implicitních metod se nejspíše budete divit, proč se s nimi vůbec obtěžujeme . . .
- Odpověď je, že implicitní metody mají obecně podstatně větší oblasti stability, než srovnatelné explicitní metody

Bezpodmínečně stabilní metody

- Pro výpočet stabilního řešení je zpětný Euler stabilní pro *všechny pozitivní velikosti* kroku h, což znamená, že je **bezpodmínečně stabilní**
- Velká ctnost bezpodmínečně stabilních metod je, že jediným omezením při volbě velikosti kroku je požadovaná přesnost
- Můžeme tedy být schopni dělat mnohem hrubější kroky, než pro explicitní metodu srovnatelného řádu, a dosáhnout celkově mnohem vyšší efektivity výpočtu, i když metoda vyžaduje více výpočtů na jeden krok
- Ačkoli zpětná Eulerova metoda je bezpodmínečně stabilní, je přesnost je pouze prvního řádu; to výrazně omezuje její užitečnost

Implicitní metody, pokračování

- Ne všechny implicitní metody to jsou bezpodmíněčně stabilní
- Implicitní metody mají obecně větší oblasti stability, než explicitní metody, ale přípustná velikost kroku není vždy neomezená
- Implicitnost sama o sobě nestačí k zajištění stability

Tuhost (angl. stiffness) vynecháváme.

V češtině tento typ úloh často označujeme také jako **úlohy se silným tlumením**.

Existuje mnoho různých metod řešení ODR, z nichž většina spadá do někeré z následujících kategorií:

- Jednokrokové metody
    - Taylorovy řady
    - Runge-Kutta
    - Extrapolační
- Vícekrokové metody
- Vícehodnotové metody

Soustředíme se pouze na jednokrokové metody.

ČVUT ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

- Eulerovu metodu lze odvodit z rozvoje Taylorovy řady
- Ponecháním více výrazů v Taylorově rozvoji můžeme generovat jednostupňové metody vyššího řádu
- Například zachování jednoho dalšího členu v Taylorově řadě

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{6}y'''(t) + \cdots$$

dává metodu druhého řádu

$$y_{k+1} = y_k + h_k y'_k + \frac{h_k^2}{2}y''_k$$

Metody Taylorovy řady, pokračování

- Tento přístup vyžaduje výpočet vyšších derivací $y(t)$. Ty lze získat diferenciací $\dot{y} = f(t, y)$ pomocí řetězového pravidla, například

$$\ddot{y} = f_t(t, y) + f_y(t, y)\dot{y} = f_t(t, y) + f_y(t, y)f(t, y)$$

  kde dolní indexy označují parciální derivace vzhledem k dané proměnné

- S rostoucím řádem metody rychle narůstá výpočetní složitost výrazů pro vyšší derivace, takže metody Taylorovy řady vyšších řádů se v praxi příliš často nepoužívají

- **Metody Runge-Kutta** jsou jednokrokové metody s podobnou motivací jako metody Taylorovy řady, nevyžadují ale výpočet vyšších derivací
- Místo toho metody Runge-Kutta simulují účinek vyšších derivací vyhodnocením f několikrát mezi body t_k a t_{k+1}
- De facto používají lokální aproximaci pomocí konečných diferencí

Heunova metoda

- Nejjednodušším příkladem je Heunova metoda druhého řádu

$$y_{k+1} = y_k + \frac{h_k}{2}(k_1 + k_2)$$

  kde

$$k_1 = f(t_k, y_k)$$
$$k_2 = f(t_k + h_k, y_k + h_k k_1)$$

- Heunova metoda je analogie implicitní lichoběžníkové metody, použitím Eulerovy predikce $y_k + h_k k_1$ místo $y(t_{k+1})$ pro vyhodnocení f v t_{k+1} ale zůstává explicitní

Klasická metoda Runge-Kutta čtvrtého řádu

- Nejznámější Runge-Kuttova metoda je klasické schéma čtvrtého řádu

$$y_{k+1} = y_k + \frac{h_k}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

kde

$$\begin{aligned}
k_1 &= f(t_k, y_k) \\
k_2 &= f(t_k + h_k/2, y_k + (h_k/2)k_1) \\
k_3 &= f(t_k + h_k/2, y_k + (h_k/2)k_2) \\
k_4 &= f(t_k + h_k, y_k + h_k k_3)
\end{aligned}$$

- Analogie k Simpsonovu pravidlu při numerické integraci

Metody Runge-Kutta – klady a zápory

Výhody klasických metod Runge-Kutta

- Pro posun k času t_{k+1} není vyžadována žádná historie řešení před časem t_k
  - Metody jsou **samostartovací** na začátku integrace
  - Snadno lze během integrace ODR měnit velikost kroku
- Relativně snadno se programují; to částečně způsobilo jejich popularitu

Nevýhody klasických metod Runge-Kutta

- Neuvádí žádný odhad chyby, na němž by mohla být založena volba velikosti kroku
- Neúčinné pro „tuhé“ (stiff) počáteční úlohy na ODR

Vestavěné metody Runge-Kutta

- Runge-Kutta-Fehlberg využívá šest vyhodnocení f na jeden krok k vytvoření odhadu řešení pátého i čtvrtého řádu; rozdíl odhadů poskytuje odhad lokální chyby
- Další vestavěná metoda Runge-Kutta je Dormand-Prince (tu používá `ode45` v Matlabu)
- Tento přístup vedl k automatickým řešičům ODR pomocí metod Runge-Kutta, které jsou použitelné pro mnoho typických úloh, ale které jsou stále relativně neúčinné pro „tuhé“ problémy nebo když je vyžadována velmi vysoká přesnost
- Je však možné odvodit i *implicitní Runge-Kuttovy metody* s vynikajícími vlastnostmi stability, jež jsou vhodné pro řešení „tuhých“ ODR

Extrapolační metody

- Extrapolační metody jsou založeny na použití jednokrokové metody pro integraci ODR v intervalu $t_k \leq t \leq t_{k+1}$ pomocí několika různě velikých kroků h_i, generujících vzorky $Y(h_i)$
- Tím získáme diskrétní aproximaci funkce $Y(h)$, kde $Y(0) = y(t_{k+1})$
- Na data $\{Y(h_i)\}_i$ použijeme interpolační polynom nebo racionální funkci $\hat{Y}(h)$ a $\hat{Y}(0)$ je poté bráno jako aproximace $Y(0)$
- Extrapolační metody jsou schopné dosáhnout velmi vysoké přesnosti, jsou ale mnohem méně efektivní a méně flexibilní než jiné metody pro ODR
- Nejčastěji se používají, pokud potřebujeme dosáhnout extrémně vysoké přesnosti a výpočetní nároky nejsou podstatným faktorem

Vícekrokové metody (viz např. *prediktor-korektor*) vynecháváme

Vícehodnotové metody vynecháváme

Shrnutí – Počáteční úlohy pro ODR (1/2)

- Numerickým řešením počátečních úloh je tabulka přibližných hodnot funkce řešení v diskrétních bodech, generovaná simulací chování systému řízeného ODR krok za krokem
- Každý krok způsobí *lokální* diskretizační chybu a také šíření předchozích chyb, které se kombinují a určují celkovou *globální* chybu
- O tom, zda je skutečné řešení stabilní nebo nestabilní, rozhoduje vývoj chyby v průběhu času – zda roste nebo klesá
- Eulerova metoda postupuje při řešení extrapolací podél přímky, jejíž sklon je dán pravou stranou ODR
- Skalární test pomocí ODR $\dot{y} = \lambda y$ ukazuje, že přesnost Eulerovy metody je prvního řádu a že hranice stability je dána podmínkou velikosti kroku $h \leq -2/\lambda$ pro $\lambda < 0$

Shrnutí – Počáteční úlohy pro ODR (2/2)

- Přesnost lze zlepšit použitím metod vyššího řádu a oblast stability lze rozšířit pomocí *implicitních* metod
- Například implicitní *lichoběžníková metoda* (o níž jsme se zde nebavili) má přesnost druhého řádu a je bezpodmínečně stabilní
- Implicitní metody jsou zvláště důležité pro řešení „tuhých“ ODR, které mají velmi rozdílná časové měřítka, takže velikost kroku je ve výrazně větší míře omezena požadavkem na stabilitu výpočtu než požadovanou přesností
- Mezi důležité rodiny metod ODR patří *Runge-Kutta* a vícekrokové / vícehodnotové metody