

Diferenciální rovnice jako modely

Matematické metody pro ITS (11MAMY)

Jan Příkryl

3. přednáška 11MAMY

středa 23. března 2022

verze: 2022-03-22 18:23

Obsah

<i>Obyčejné diferenciální rovnice</i>	1
<i>Existence, jedinečnost a podmíněnost</i>	6
<i>Numerické řešení ODR</i>	10
<i>Eulerova metoda</i>	10
<i>Přesnost a stabilita</i>	12
<i>Implicitní metody</i>	16
<i>Tuhost</i>	18
<i>Metody Taylorovy řady</i>	20
<i>Metody Runge-Kutta</i>	21
<i>Stabilita řešení ODR</i>	24
<i>Eulerova metoda</i>	26
<i>Přesnost a stabilita</i>	30
<i>Implicitní metody</i>	31

Tento text je do jisté míry experimentálním písčovištěm na odladění převodu textu prezentace vytvořené v \LaTeX ové třídě beamer do textu vysázeného pomocí *tuft*-handout. Obsah je oproti prezentaci mírně rozšířen o poznámky. Bude se ještě v průběhu semestru měnit, kontrolujte si prosím čas sestavení v záhlaví tohoto souboru.

Obyčejné diferenciální rovnice

Většina fyzických systémů se v průběhu času mění. Od družice na oběžné dráze po chladnoucí šálek kávy, od houpajícího se kyvadla k rozkládajícímu se radioizotopu, od reagujících chemických druhů ke konkurenčním biologickým druhům, normální stav je stavem neustálé změny. Matematický jazyk pro popis těchto spojitých změn

nám poskytují právě diferenciální rovnice. Jedním z motivačních problémů pro vynález diferenciálního počtu byla snaha charakterizovat pohyb nebeských těles a pozemských projektilů tak, aby šlo předvídat jejich budoucí umístění. Počínaje Newtonovými pohybovými zákony je většina základních zákonů vědy vyjádřena jako diferenciální rovnice. Dokonce i systém, který se nemění, je často nejlépe chápán jako systém, nacházející se v rovnovážném stavu příslušné diferenciální rovnice.

Předpokládejme, že stav systému v daném čase t je popsán nějakou vektorovou funkcí $\mathbf{y}(t)$, kde $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^n$. Složky $\mathbf{y}(t)$ mohou představovat například prostorové souřadnice výše zmíněného projektilu, rozložení teploty na plošném spoji elektrického zařízení nebo koncentraci různých chemických látek. Diferenciální rovnice předepisuje vztah mezi touto neznámou stavovou funkcí $\mathbf{y}(t)$ a jednou nebo více jejími derivacemi vzhledem k t , které musí platit v daném okamžiku. Při řešení diferenciální rovnice je cílem určit diferencovatelnou funkci $\mathbf{y}(t)$, která splňuje předepsaný vztah. Nalezení takového řešení diferenciální rovnice je důležité, protože nám umožní předpovědět budoucí vývoj systému v průběhu času.

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

Pokud existuje pouze jedna nezávislá proměnná, například čas, pak všechny derivace závislých proměnných jsou vzhledem k této nezávislé proměnné a máme obyčejnou diferenciální rovnici neboli ODR. V dalších přednáškách budeme uvažovat o systémech s více než jednou nezávislou proměnnou, u nichž je třeba počítat parciální derivace, což vede na parciální diferenciální rovnice neboli PDR. Aby bylo vyjádření ODR méně těžkopádné, použijeme zápis $\mathbf{y}'(t) = d\mathbf{y}(t)/dt$ k označení první derivace vzhledem k nezávislé proměnné t a v jasných případech nebudeme explicitně zapisovat závislost na t a použijeme notaci $\dot{\mathbf{y}} = \mathbf{y}' = d\mathbf{y}(t)/dt$. V této konvenci tak lze například Newtonův druhý zákon zapsat jako $F = m\dot{y}$.

Derivace nejvyššího řádu, která v ODR objevuje, určuje řád obyčejné diferenciální rovnice. Například již zmíněný Newtonův druhý zákon představuje ODR druhého řádu. Pokud má systém n závislých proměnných, pak pro vyjádření libovolného vztahu mezi

vektorem $\mathbf{y}(t) \in \mathbb{R}^n$ a jeho prvními k derivacemi můžeme použít nejobecnější implicitní tvar ODR k -tého řádu, jež má tvar

$$\mathbf{g}(t, \mathbf{y}, \mathbf{y}', \dots, \mathbf{y}^{(k)}) = \mathbf{0},$$

kde $\mathbf{g} : \mathbb{R}^{kn+n+1} \rightarrow \mathbb{R}^n$ známe a $\mathbf{y}(t)$ chceme určit. Tuto rovnici lze zapsat také ve tvaru explicitní ODR k -tého řádu ve formě

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

kde $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$. Mnoho ODR v této formě přirozeně vzniká a mnoho dalších do ní může být transformováno – například Newtonův druhý zákon je formulován jako implicitní ODR, může být ale převeden na explicitní formu vydělením obou stran rovnice hmotností m , čímž získáme tvar $\dot{\mathbf{y}} = F/m$. Pokud $\mathbf{y}^{(k)}$ nelze tímto způsobem izolovat na levé straně, pak řešení dané ODR vyžaduje kombinaci různých technik, ale takové problémy jsou nad rámec této přednášky (blíže viz Heath kapitoly 5 a 9). V dalším textu se budeme zabývat pouze explicitními 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

Budeme také uvažovat pouze ODR prvního řádu. To ale není skutečné omezení, protože ODR vyšších řádů můžeme vždy transformovat na soustavu ODR prvního řádu.

Lze to udělat následovně: Pro explicitní ODR k -tého řádu ODR ve výše uvedeném tvaru definujeme k nových neznámých $\mathbf{u}_1(t) = \mathbf{y}(t)$, $\mathbf{u}_2(t) = \mathbf{y}'(t), \dots, \mathbf{u}_k(t) = \mathbf{y}^{(k-1)}(t)$ tak, abychom původní rovnici k -tého řádu mohli přepsat na soustavu k rovnic prvního řádu ve tvaru

$$\begin{bmatrix} \mathbf{u}'_1(t) \\ \mathbf{u}'_2(t) \\ \vdots \\ \mathbf{u}'_{k-1}(t) \\ \mathbf{u}'_k(t) \end{bmatrix} = \begin{bmatrix} \mathbf{u}_2(t) \\ \mathbf{u}_3(t) \\ \vdots \\ \mathbf{u}_k(t) \\ \mathbf{f}(t, \mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k) \end{bmatrix} = \boldsymbol{\varphi}(t, \mathbf{u}).$$

Dobrym příkladem tohoto postupu je opět Newtonův druhý zákon, který je druhého řádu: Pokud definujeme $u_1(t) = y(t)$ a $u_2(t) = y'(t)$, vznikne pro Newtonův druhý zákon ekvivalentní soustava dvou ODR prvního řádu

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

Tato soustava shrnuje obecně známé fyzikální pojmy: rychlost $v(t) = u_2(t)$ je první derivací polohy $y(t) = u_1(t)$ a zrychlení je první derivací rychlosti.

V této přednášce se tedy z důvodů, zmíněných výše, zaměříme pouze explicitní ODR prvního řádu ve tvaru

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}),$$

kde $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$. Plný zápis takové ODR po složkách je přitom poměrně složitý a budeme jej používat zřídka:

$$\mathbf{y}'(t) = \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = \begin{bmatrix} dy_1(t)/dt \\ dy_2(t)/dt \\ \vdots \\ dy_n(t)/dt \end{bmatrix} = \begin{bmatrix} f_1(t, \mathbf{y}(t)) \\ f_2(t, \mathbf{y}(t)) \\ \vdots \\ f_n(t, \mathbf{y}(t)) \end{bmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

Pro $n > 1$ dostaneme tak zvanou *soustavu spřažených ODR*. Pro zjednodušení budeme v dalších ukázkách často uvažovat pouze $n = 1$, tj. jednu skalární ODR $\dot{y} = f(t, y)$.

V praxi se často objevuje řada zvláštních případů obecné explicitní ODR prvního řádu. Pokud \mathbf{f} nezávisí explicitně na t , pak jde o autonomní ODR, který může být zapsána ve tvaru $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$. Neautonomní ODR $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ pro n složek \mathbf{y} lze vždy převést na autonomní formu zavedením další závislé proměnné $y_{n+1}(t) = t$, čímž vznikne autonomní ODR

$$\begin{bmatrix} \dot{\mathbf{y}} \\ \dot{y}_{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}(y_{n+1}, \mathbf{y}) \end{bmatrix}$$

Má-li \mathbf{f} tvar $\mathbf{f}(t, \mathbf{y}) = \mathbf{A}(t)\mathbf{y} + \mathbf{b}(t)$, kde $\mathbf{A}(t)$ a $\mathbf{b}(t)$ jsou maticová a vektorová funkce t , pak ODR považujeme za lineární. O lineární ODR, kde maticová funkce $\mathbf{A}(t) = \mathbf{A}$ a nezávisí na t říkáme, že jde o ODR s konstantními koeficienty; pokud je $\mathbf{b}(t) \equiv \mathbf{0}$, pak jde o homogenní ODR. Lineární homogenní ODR prvního řádu s konstantními koeficienty má tedy jednoduchý tvar $\dot{\mathbf{y}} = \mathbf{A}\mathbf{y}$, kde $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Pro ODR k -tého řádu

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

definujeme k nových neznámých funkcí

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

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

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

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

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

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

$$\mathbf{y}'(t) = \begin{bmatrix} y'_1(t) \\ y'_2(t) \\ \vdots \\ y'_n(t) \end{bmatrix} = \begin{bmatrix} dy_1(t)/dt \\ dy_2(t)/dt \\ \vdots \\ dy_n(t)/dt \end{bmatrix} = \begin{bmatrix} f_1(t, \mathbf{y}(t)) \\ f_2(t, \mathbf{y}(t)) \\ \vdots \\ f_n(t, \mathbf{y}(t)) \end{bmatrix} = \mathbf{f}(t, \mathbf{y}(t))$$

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, $n = 1$

Rovnice $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ sama o sobě jednoznačně neurčuje funkci řešení, protože ODR pro libovolnou hodnotu t předepisuje pouze vektor směrnic $\mathbf{y}'(t)$ řešení, nikoliv přesnou hodnotu řešení $\mathbf{y}(t)$. Obvykle proto existuje nekonečná množina funkcí z jedné „rodiny“, splňujících danou ODR. Abychom vyčlenili konkrétní řešení, musíme znát ještě hodnotu funkce řešení \mathbf{y}_0 , pro nějakou hodnotu t , označenou t_0 . Součástí daných dat o problému je tedy kromě vlastní ODR také požadavek, aby

$$\mathbf{y}(t_0) = \mathbf{y}_0.$$

Za rozumných předpokladů (viz bod 9.2) tento dodatečný požadavek určuje jedinečné řešení dané ODR. Protože nezávislá proměnná t často představuje čas, můžeme t_0 považovat za počáteční čas a \mathbf{y}_0

za počáteční hodnotu **stavového asi ne** vektoru. Proto se požadavek, aby $\mathbf{y}(t_0) = \mathbf{y}_0$, nazývá **počáteční podmínka** a ODR spolu s počáteční podmínkou se nazývá **problém počáteční hodnoty** nebo IVP (z anglického *initial value problem*). Počínaje počátečním stavem \mathbf{y}_0 v čase t_0 popisuje ODR dynamický vývoj systému pro $t \geq t_0$ a naším úkolem je najít takovou funkci $\mathbf{y}(t)$, která splňuje počáteční podmínku a popisuje stav systému jako funkci času.

Příklad 1 (Problém s počáteční hodnotou.). Uvažujme skalární ODR $y' = y$. Nekonečná rodina řešení je dána $y(t) = ce^t$, kde c je libovolná reálná konstanta. Pokud uložíme počáteční podmínku $y(t_0) = y_0$, pak to vyčlení jedinečné řešení, které splňuje počáteční podmínku. Pokud například $t_0 = 0$, pak máme $y(t_0) = y(0) = ce^0 = c$, takže musíme mít $c = y_0$, což znamená, že jedinečné řešení splňující jak ODR, tak počáteční podmínku je $y(t) = y_0 e^t$. Některá řešení pro tuto ODR jsou načrtnuta na obr. 9.2, včetně konkrétního řešení, které splňuje danou počáteční podmínku.

Pokud integrujeme ODR $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ a použijeme počáteční podmínku $\mathbf{y}(t_0) = \mathbf{y}_0$, získáme **integrální rovnici**

$$\mathbf{y}(t) = \mathbf{y}_0 + \int_{t_0}^t \mathbf{f}(\tau, \mathbf{y}(\tau)) d\tau,$$

kteřou musí řešení $\mathbf{y}(t)$ IVP splnit. I když je to užitečné pro teoretické účely, v praxi obvykle neumíme tuto ekvivalentní integrální rovnici o nic snáze vyřešit, než onu původní ODR. Vysvětluje to však, proč je řešení ODR, ať už jakýmikoliv prostředky, často označováno jako **integrace ODR**. Ve zvláštním případě, kdy \mathbf{f} nezávisí na \mathbf{y} , tj. kdy ODR má tvar $\dot{\mathbf{y}} = \mathbf{f}(t)$, je řešení dáno integrálem

$$\mathbf{y}(t) = \mathbf{y}_0 + \int_{t_0}^t \mathbf{f}(\tau) d\tau,$$

kteřé lze vyhodnotit analyticky nebo numerickou kvadraturou (viz Heath kapitola 8.3).

Existence, jedinečnost a podmíněnost

Viděli jsme, že počáteční podmínka je nezbytná pro to, aby řešení ODR bylo jedinečné. Je však tato podmínka dostatečná? A navíc, opravdu existuje vždy řešení? Bez určitých omezení problému je odpověď na obě otázky „ne“, ale za rozumných předpokladů, které obvykle platí, lze na obě otázky odpovědět „ano“. Nejprve ale musíme upřesnit **doménu** problému, kterou předpokládáme jako uzavřenou a ohraničenou množinu $\mathcal{D} = \langle a, b \rangle \times \Omega \subseteq \mathbb{R}^{n+1}$. Nyní předpokládejme, že $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ je spojitá v t na $\langle a, b \rangle$ a Lipschitzovsky spojitá v \mathbf{y} na \mathcal{D} , tj. že existuje konstanta L taková, že pro libovolné $t \in \langle a, b \rangle$ a libovolné \mathbf{y} a $\hat{\mathbf{y}} \in \Omega$ **o normách se bavíme až u řešení soustav rovnic!**

$$\|\mathbf{f}(t, \hat{\mathbf{y}}) - \mathbf{f}(t, \mathbf{y})\| \leq L \|\hat{\mathbf{y}} - \mathbf{y}\|$$

Pokud platí tento předpoklad, lze ukázat, že pro libovolný vnitřní bod $(t_0, \mathbf{y}_0) \in \mathcal{D}$ existuje podinterval $\langle a, b \rangle$ obsahující t_0 , ve kterém existuje jedinečné řešení \mathbf{y} IVP $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ s počáteční podmínkou $\mathbf{y}(t_0) = \mathbf{y}_0$ a navíc toto řešení může jednoznačně pokračovat až k hranici \mathcal{D} . Lipschitzova spojitost \mathbf{f} je zajištěna, pokud je \mathbf{f} diferencovatelné, v takovém případě můžeme vzít

$$L = \max_{(t, \mathbf{y}) \in \mathcal{D}} \|\mathbf{J}_{\mathbf{f}}(t, \mathbf{y})\|,$$

kde $\mathbf{J}_{\mathbf{f}}$ je $n \times n$ Jacobiho matice funkce \mathbf{f} vzhledem k \mathbf{y} , tj. matice $\mathbf{J}_{\mathbf{f}}(t, \mathbf{y})_{ij} = \partial f_i(t, \mathbf{y}) / \partial y_j$.

Za předpokladu Lipschitzovy kontinuity na uzavřené a ohraničené doméně \mathcal{D} můžeme také posoudit citlivost řešení na poruchy v problémových datech, což pro IVP jsou počáteční hodnota \mathbf{y}_0 a funkce \mathbf{f} . Necht' $\hat{\mathbf{y}}(t)$ je řešením IVP $\dot{\hat{\mathbf{y}}} = \mathbf{f}(t, \hat{\mathbf{y}})$ s narušenou počáteční podmínkou $\hat{\mathbf{y}}(t_0) = \hat{\mathbf{y}}_0$. Pak lze ukázat, že pro libovolné $t \geq t_0$,

$$\|\hat{\mathbf{y}}(t) - \mathbf{y}(t)\| \leq e^{L(t-t_0)} \|\hat{\mathbf{y}}_0 - \mathbf{y}_0\|.$$

Pokud je funkce \mathbf{f} také narušena, máme IVP $\dot{\hat{\mathbf{y}}} = \hat{\mathbf{f}}(t, \hat{\mathbf{y}})$, pak pro libovolné $t \geq t_0$,

$$\|\hat{\mathbf{y}}(t) - \mathbf{y}(t)\| \leq e^{L(t-t_0)} \|\hat{\mathbf{y}}_0 - \mathbf{y}_0\| + \frac{e^{L(t-t_0)} - 1}{L} \|\hat{\mathbf{f}} - \mathbf{f}\|,$$

kde $\|\hat{\mathbf{f}} - \mathbf{f}\| = \max_{(t, \mathbf{y}) \in \mathcal{D}} \|\hat{\mathbf{f}}(t, \mathbf{y}) - \mathbf{f}(t, \mathbf{y})\|$. Tyto meze poruch ukazují, že jedinečným řešením IVP je spojitá funkce dat problému, a proto je problém **dobře podmíněný**. Ale exponenciální divergence narušených řešení v čase, která je povolena členem $e^{L(t-t_0)}$ v mezích a je realizována v nejhorším případě, znamená, že řešení může být stále vysoce citlivé na takové poruchy. Dále se touto otázkou zabýváme podrobněji.

Dříve jsme v této sérii přednášekpoužili termín podmiňování k označení citlivosti řešení problému na poruchy ve vstupu a stability k označení citlivosti algoritmu k poruchám, ke kterým dochází během výpočtu. Při studiu diferenciálních rovnic však existuje dlouholetá tradice používání termínu stabilita pro oba tyto pojmy, což je nejednoznačnost, která zjevně vyvolává zmatek. Při diskusi o numerickém řešení diferenciálních rovnic se budeme touto tradicí řídit, pokusíme se ale vždy objasnit, zda máme na mysli stabilitu vlastního problému nebo numerické metody pro jeho řešení. Jak však brzy uvidíme, tyto dva významy pro stabilitu pro diferenciální rovnice stejně značně propojené, takže možná je výše zmíněná nejednoznačnost pouze přirozeným jevem.

Řešení ODR $\mathbf{y}' = \mathbf{f}(\mathbf{y})$ je považováno za **stabilní**, pokud pro každé $\epsilon > 0$ existuje $\delta > 0$ takové, že pokud $\hat{\mathbf{y}}(t)$ splňuje ODR a $\|\hat{\mathbf{y}}(t_0) - \mathbf{y}(t_0)\| \leq \delta$, pak $\|\hat{\mathbf{y}}(t) - \mathbf{y}(t)\| \leq \epsilon$ pro všechna $t \geq t_0$. Pro stabilní řešení tedy platí, že pokud je počáteční hodnota řešení narušena, zůstává narušené řešení blízko původního řešení, což vylučuje exponenciální divergenci narušených řešení, kterou

umožňuje výše uvedené omezení. O stabilním řešení se říká, že je **asymptoticky stabilní**, pokud $\|\dot{\mathbf{y}}(t) - \mathbf{y}(t)\| \rightarrow 0$ když $t \rightarrow \infty$. Tato silnější forma stability znamená, že původní a narušená řešení nejen, že zůstávají blízko sebe, ale s rostoucím časem se k sobě opět sbíhají. Jak brzy uvidíme podrobně, význam těchto konceptů pro numerické řešení ODR spočívá v tom, že jakékoli chyby vzniklé během výpočtu mohou být v průběhu času buď zesíleny nebo sníženy v závislosti na stabilitě hledaného řešení. Nejprve však uvažme, jak určit, v jakých případech je řešení stabilní.

Příklad 2 (Stabilita řešení). Pro ODR z příkladu 9.2 s $n = 1$, tj. $\dot{y} = b$ pro danou konstantu b , jsou řešení rovnoběžné přímky se sklonem b , jak je znázorněno na obr. 9.1, takže řešení této ODR jsou stabilní, ale ne asymptoticky stabilní. Zajímavější je skalární ODR

$$\dot{y} = \lambda y,$$

kde λ je konstanta. Řešení je dáno

$$y(t) = y_0 e^{\lambda t},$$

kde $y(0) = y_0$ je počáteční hodnota. Viděli jsme konkrétní případ této ODR s $\lambda = 1$ v příkladu 9.5 a některá z jeho řešení jsou načrtnuta na obr. 9.2. Pokud $\lambda > 0$, pak všechna nenulová řešení rostou exponenciálně, takže jakákoli dvě řešení se od sebe odchyľují, jak je znázorněno na obr. 9.2, a proto každé řešení je nestabilní. Pokud $\lambda < 0$, pak se všechna nenulová řešení exponenciálně klesají, takže jakákoli dvě řešení konvergují k sobě, jak je znázorněno na obr. 9.3. V tomto případě je tedy každé řešení nejen stabilní, ale také asymptoticky stabilní. Pokud je λ komplexní, řekněme $\lambda = a + ib$, pak z předchozích kurzů matematické analýzy víme, že

$$e^{\lambda t} = e^{(a+ib)t} = e^{at} e^{ibt} = e^{at} (\cos bt + i \sin bt),$$

takže to, zda řešení rostou nebo klesají exponenciálně, je určeno znaménkem $\Re(\lambda)$. Zejména řešení jsou nestabilní, pokud $\Re(\lambda) > 0$ a asymptoticky stabilní, pokud $\Re(\lambda) < 0$. Pokud $\Re(\lambda) = 0$, pak řešení oscilují, ale jakákoliv dvě řešení zůstávají od sebe vzdálena, a proto jsou řešení stabilní, ale ne asymptoticky stabilní. **■ Viz póly přenosové funkce, pokud to měli. A nevím, jestli imaginární jednotku neznačit j. ■**

Obrázek 9.3: Některá řešení pro $y_0 = -y$.

Příklad 3 (Lineární soustava ODR). Lineární, homogenní soustava ODR s konstantními koeficienty má tvar

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

kde \mathbf{A} je matice $n \times n$ prvků. Předpokládejme, že máme počáteční podmínku $\mathbf{y}(0) = \mathbf{y}_0$. Necht' jsou vlastní čísla \mathbf{A} označena λ_i a odpovídající vlastní vektory \mathbf{v}_i , $i = 1, \dots, n$. Předpokládejme pro tuto chvíli, že \mathbf{A} je diagonalizovatelné, takže vlastní vektory jsou

lineárně nezávislé a můžeme vyjádřit y_0 jako lineární kombinaci

$$y_0 = \sum_{i=1}^n \alpha_i v_i$$

Pak lze snadno ukázat, že

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

je řešení ODR, které splňuje počáteční podmínku. Stejným uvažováním jako u skalárního případu v příkladu 9.6 vidíme, že vlastní čísla A s kladnými reálnými částmi poskytují exponenciálně rostoucí složky řešení, vlastní čísla se zápornými reálnými částmi poskytují exponenciálně se rozpadající složky řešení a vlastní čísla s nulovými reálnými díly poskytují komponenty oscilačního řešení. Řešení této ODR jsou tedy 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 existuje je libovolné vlastní číslo takové, že $\Re(\lambda_i) > 0$. Situace je komplikovanější, pokud A není diagonalizovatelná, v takovém případě stabilita vyžaduje nejen to, že $\Re(\lambda_i) \leq 0$ pro každé vlastní číslo, ale $\Re(\lambda_i) < 0$ pro jakékoli vlastní číslo, které není jednoduché.

Jak jsme právě viděli, pro lineární ODR s konstantními koeficienty lze analýzu vlastních čísel použít k určení stability nejen konkrétního řešení, ale všech řešení ODR. Pro lineární ODR s proměnnými koeficienty, $y_0(t) = A(t)y(t)$, analýza vlastních čísel matice $A(t)$ pro určitou hodnotu t může poskytnout určitý náznak chování řešení v krátkodobém horizontu, ale znaky vlastních čísel (nebo jejich reálných částí) se mohou měnit, jak se t mění, takže pokud $A(t)$ nemá nějakou zvláštní vlastnost, jako je bytí Periodické, málo lze zjistit pomocí analýzy vlastních čísel o dlouhodobé stabilitě řešení.

Pro obecnou nelineární ODR $y_0 = f(t; y)$ je stanovení stability řešení ještě komplikovanější. Pro dané řešení $y(t)$ může být ODR lokálně linearizována pomocí zkrácené expanze Taylorovy řady, čímž vznikne lineární ODR tvaru

$$z_0 = Jf(t; y(t))z;$$

kde Jf je $n \times n$ Jakobiánská matice f vzhledem k y , $Jf(t; y)_{ij} = \frac{\partial f_i(t; y)}{\partial y_j}$. Pokud je původní nelineární ODR autonomní, pak linearizovaná ODR má konstantní koeficienty, a proto její vlastní čísla určují stabilitu jejích řešení jako v příkladu 9.7. Pokud původní nelineární ODR není autonomní, pak linearizovaná ODR má variabilní koeficienty a platí stejné komentáře jako v předchozím odstavci. V obou případech, protože Jacobova matice je hodnocena pro konkrétní řešení, její vlastní čísla poskytují informace o stabilitě pouze o tomto řešení. Informace o stabilitě dané vlastními čísly Jacobovy matice tedy mohou mít pouze omezenou lokální platnost v y i v t .

Numerické řešení ODR

Existuje mnoho analytických technik pro řešení ODR, jako je oddělení proměnných, integrační faktory, sériová řešení a tak dále. Bohužel, většina ODR vznikajících v praxi není náchylná k řešení žádnou z těchto analytických metod a obvykle jedinou životaschopnou alternativou je výpočet číselné aproximace k požadovanému řešení. Ačkoli oba typy řešení obvykle postačují pro většinu účelů, existují mezi nimi základní rozdíly. Zatímco analytický řešení ODR je uzavřený vzorec pro funkci řešení kdykoliv, číselné řešení je tabulka přibližných hodnot funkce řešení v diskrétní množině bodů. Zatímco skutečným řešením je spojitá funkce v nekonečně dimenzionálním prostoru, numerické řešení je diskrétní vektor v konečném dimenzionálním prostoru.

Náš přístup k numerickému řešení diferenciálních rovnic bude založen na aproximacích konečných dimenzí, což je proces nazývaný diskretizace. Diferenciální rovnice nahradíme algebraickými rovnicemi, jejichž řešení se blíží řešením daných diferenciálních rovnic. Pro počáteční úlohu hodnoty jsou krok za krokem generovány přibližné hodnoty řešení v diskrétních krocích v intervalu, ve kterém je řešení hledáno. Z tohoto důvodu se numerické metody pro řešení ODR někdy nazývají metody diskrétních proměnných. Při přechodu z jednoho diskrétního bodu do druhého se obecně vyskytne nějaká chyba, což znamená, že naše nová přibližná hodnota řešení bude ležet na jiném řešení ODR z toho, na kterém jsme začali. Stabilita nebo nestabilita řešení částečně určuje, zda jsou tyto chyby časem zesíleny nebo sníženy.

Eulerova metoda

Numerické řešení IVP se získá tak, že se začne v čase t_0 s danou počáteční hodnotou y_0 a pokusí se sledovat trajektorii řešení diktovanou ODR. Počáteční sklon \dot{y} každé složky řešení můžeme určit vyhodnocením f na daných počátečních datech, tj. $\dot{y}_0 = f(t_0; y_0)$. Tyto informace používáme k předpovědi hodnoty y_1 řešení v nějakém budoucím čase $t_1 = t_0 + h_0$ pro nějaký vhodně zvolený přírůstek h_0 . Pak můžeme vyhodnotit $\dot{y}_1 = f(t_1; y_0)$ a opakovat proces, abychom udělali další krok vpřed a tak dále, dokud nedosáhneme konečného požadovaného času.

Nejjednodušším příkladem tohoto přístupu je Eulerova metoda, pro kterou je přibližné řešení v čase $t_{k+1} = t_k + h_k$ dáno vztahem

$$y_{k+1} = y_k + h_k f(t_k; y_k) :$$

Z důvodů, které brzy uvidíme, je Eulerova metoda obecně neúčinná, takže se v praxi používá jen zřídka, ale má zásadní význam pro pochopení základních pojmů. a principy numerického řešení diferenciálních rovnic. Eulerova metoda je navíc nejjednodušším

příkladem několika různých rodin metod ODR, které budeme zvažovat později. Eulerovu metodu proto budeme podrobně zkoumat, včetně několika různých způsobů jejího odvození.

Taylorova řada. Vezměme si sérii Taylor

$$y(t+h) = y(t) + h y'(t) + \dots$$

Eulerova metoda vyplývá z přijetí $t = t_k, h = h_k, y_0(t_k) = f(t_k; y_k)$ a vynechání podmínek druhého a vyššího řádu.

Aproximace konečných rozdílů. Pokud nahradíme derivaci $y'(t)$ v ODR $\dot{y} = f(t, y)$ aproximací dopředného rozdílu prvního řádu (viz kapitola 8.6.1), získáme algebraickou rovnici

$$y_{k+1} - y_k = f(t_k; y_k);$$

což dává Eulerovu metodu při řešení pro y_{k+1} .

Polynomická interpolace. Pomocí Hermitovy interpolace můžeme přizpůsobit polynom stupně jedna funkci řešení tím, že porovnáme jak složky řešení y_k , tak derivační složky $f(t_k; y_k)$ ve společnosti t_k . Výsledný polynom v h je $p(h) = y_k + hf(t_k; y_k)$, který dává Eulerovu metodu při hodnocení na $h = h_k$. Eulerova metoda tedy posouvá řešení v každém kroku extrapolací podél tečny od $(t_k; y_k)$ se sklonem $f(t_k; y_k)$.

Numerická kvadratura. Z integrální charakterizace řešení ODR v kapitole 9.1 máme

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(s; y(s)) ds$$

Pokud tento integrál aproximujeme pomocí obdélníkového pravidla $\int_a^b g(s) ds \approx (b-a)g(a)$, pak získáme Eulerovu metodu.

Metoda neurčených koeficientů. V kroku k známe y_k a $f(t_k; y_k)$ a na základě těchto hodnot chceme v dalším kroku předpovědět hodnotu y_{k+1} . Pokud vezmeme lineární kombinaci známých dat, pak náš prediktor bude mít tvar

$$y_{k+1} = ff y_k + fi f(t_k; y_k);$$

kde ff a fi jsou koeficienty, které mají být stanoveny. Se dvěma parametry, které mají být určeny, můžeme přinutit prediktor, aby byl přesný pro první dva monomiály. Pokud $y(t) = e$, vektor, jehož složky jsou všechny jedničky (první monomiál), pak $y_0(t) = e$ a máme rovnici

$$e = ff e + fi f(t; e);$$

což znamená, že $ff = 1$. Pokud $y(t) = te$, pak $y_0(t) = e$ a máme rovnici

$$t_{k+1} e = ff t_k e + fi f(t_k; e);$$

což znamená, že $fi = t_{k+1} - t_k = h_k$. Náš prediktor je tedy

$$y_{k+1} = y_k + h_k f(t_k; y_k);$$

což je Eulerova metoda.

Příklad 4 (Eulerova metoda). V příkladu 9.5 jsme uvažovali IVP $y' = y$ s počáteční hodnotou y_0 v počátečním čase $t_0 = 0$. Tento jednoduchý problém lze snadno vyřešit analyticky, ale pro ilustraci použijme Eulerovu metodu k jeho numerickému řešení. Pro jednoduchost použijeme pevný krok velikosti h . Nejprve posuneme řešení z času $t_0 = 0$ na čas $t_1 = t_0 + h$,

$$y_1 = y_0 + h\dot{y} = y_0 + hy_0 = (1 + h)y_0 :$$

Všimněte si, že přibližná hodnota řešení y_1 , kterou získáme při t_1 , není přesná (tj. $y_1 \neq y(t_1)$). Pokud například $t_0 = 0$, $y_0 = 1$ a $h = 0,5$, pak $y_1 = 1,5$, zatímco přesné řešení této počáteční hodnoty je $y(0,5) = \exp(0,5) \approx 1,649$. Hodnota y_1 tedy leží na jiném řešení ODR, než na kterém jsme začali, jak je znázorněno na obr. 9.4. Čl.

Abychom pokračovali v procesu numerického řešení, uděláme další krok z t_1 na $t_2 = t_1 + h = 1,0$, získáme $y_2 = y_1 + h\dot{y}_1 = 1,5 + (0,5)(1,5) = 2,25$. Všimněte si, že y_2 se liší nejen od skutečného řešení původního problému při $t = 1$, jmenovitě $y(1) = \exp(1) \approx 2,718$, ale také se liší od řešení procházející předchozím bodem ($t_1; y_1$), který má přibližnou hodnotu 2,473 při $t = 1$. Přesunuli jsme se tedy k dalšímu řešení ODR. Můžeme pokračovat v dalších krocích a generovat tabulku diskretních hodnot přibližného řešení v libovolném intervalu, který si přejeme. Když tak učiníme, budeme v každém kroku přeskakovat z jednoho řešení do druhého. Řešení této ODR jsou nestabilní, takže chyby, které děláme v každém kroku, jsou časem zesíleny v důsledku divergence řešení, jak je vidět na obr. 9.4 Na druhé straně u rovnice s stabilními řešeními chyby v číselném řešení nerostou a u rovnice s asymptoticky stabilní řešení, například $y' = -y$, chyby se časem zmenšují, jak je znázorněno na obr. 9.5. Čl.

Eulerova metoda je příkladem jednokrokové nebo jednokrokové metody v tom, že další přibližná hodnota řešení závisí pouze na aktuálních hodnotách t_k , y_k a h . Tyto metody mají obecnou formu

$$y_{k+1} = y_k + h f(t_k; y_k; h)$$

pro některé funkce $f: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$. Uvidíme několik dalších jednokrokových metod, stejně jako více-krokové metody, pro které další přibližná hodnota řešení závisí na řešení a derivačních hodnotách v několika krocích.

Obrázek 9.5: Eulerova metoda pro $y' = -y$.

Přesnost a stabilita

Stejně jako jiné metody, které aproximují derivace konečnými rozdíly, číselný postup řešení ODR trpí dvěma odlišnými zdroji chyb:

Chyba zaokrouhlení, která je způsobena konečnou přesností aritmetiky v pohyblivé řádové čárce

Chyba zkrácení (nebo chyba diskretizace), která je způsobena použitou metodou a která by zůstala, i kdyby byla všechna aritmetika provedena přesně

Ačkoli pocházejí z různých zdrojů, tyto dva typy chyb nejsou na sobě nezávislé. Například chybu zkrácení lze obvykle zmenšit použitím menší velikosti kroku h , ale může dojít k větší chybě zaokrouhlení (viz příklad 1.3). Ve většině praktických situací je však chyba zkrácení dominantním faktorem při určování přesnosti numerických řešení ODR, takže budeme od nynějška ignorovat zaokrouhlování chyba v této souvislosti.

Chyba zkrácení v kroku k th přichází ve dvou odlišných, ale souvisejících příčutinách:

Globální chyba je kumulativní celková chyba $e_k = y_k - y(t_k)$, kde y_k je vypočtené řešení při t_k a $y(t)$ je skutečné řešení ODR procházející počátečním bodem (t_0, y_0) .

Lokální chyba je chyba provedená v jednom kroku numerické metody, $\ell_k = y_k - u_{k-1}(t_k)$, kde $u_{k-1}(t)$ je řešení ODR procházející předchozím bodem (t_{k-1}, y_{k-1}) .

Obrázek 9.6: Lokální a globální chyby v Eulerově metodě pro $y' = y$.

Jelikož

$$y_k = y_{k-1} + hk - 1(t_k - 1; y_{k-1}; hk - 1) = u_{k-1}(t_k - 1) + hk - 1(t_k - 1; u_{k-1}(t_k - 1); hk - 1);$$

vidíme, že lokální chyba v daném kroku je jednoduše částka, o kterou řešení ODR nespĺňuje rovnici konečného rozdílu. Globální chyba je samozřejmě prvořadým zájmem, ale pouze lokální chyba může být snadno odhadnuta a kontrolována, takže musíme pochopit vztah mezi dvě.

Na bankovním spořicímu účtu, který vydělává složený úrok, mají předčasné vklady více času na růst než pozdější vklady a tento růst znamená, že celková hodnota účtu není jednoduše součet jednotlivých vkladů. Podobně globální chyba přibližného řešení ODR v daném kroku odráží nejen lokální chybu v daném kroku, ale také složené účinky místních chyb ve všech předchozích krocích. Globální chyba tedy není pouhým součtem lokálních chyb. Pokud se řešení ODR liší, pak se lokální chyby v každém kroku v průběhu času zvětšují, takže globální chyba je větší než součet lokálních chyb, jak je znázorněno na obr. 9.6, kde jsou lokální chyby označeny malými svislými pruhy mezi řešeními a globální chyba je označena pruhem na konci. Pokud se řešení ODR sblíží, na druhé straně může být globální chyba menší než součet lokálních chyb, jako znázorněno na obr. 9.7 Abychom mohli posoudit účinnost numerické metody, musíme charakterizovat jak její lokální chybu (přesnost), tak složené účinky přes více kroků (stabilita).

O přesnosti numerické metody se říká, že je řádová p , jestliže

$$\tau_k = O(h^{p+1} k):$$

Obrázek 9.7: Lokální a globální chyby v Eulerově metodě pro $y_0 = -y$.

Motivací pro tuto definici, s pořadím přesnosti o jeden menším než exponent velikosti kroku v místní chybě, je to, že pokud je místní chyba $O(h^{p+1} k)$ pak lokální chyba na jednotku kroku, $\tau_k = h \tau_k$, je $O(h^p k)$ a lze prokázat, že za rozumných podmínek je globální chyba $e_k = O(h^p)$, kde h je průměrná velikost kroku.

Koncept stability numerické metody pro ODR je analogický stabilitě řešení ODR. Připomeňme, že řešení ODR je stabilní, pokud se od něj v průběhu času neodchylují poruchy řešení. Podobně se o numerické metodě říká, že je stabilní, pokud malé poruchy nezpůsobí, že se výsledné numerické řešení rozchází bez vazby. Taková divergence numerických řešení by mohla být způsobena nestabilitou řešení ODR, ale jak uvidíme, může být také způsobena samotnou numerickou metodou, i když jsou řešení ODR stabilní. Abychom se zaměřili konkrétně na nestabilitu způsobenou numerickou metodou, alternativní definice stability vyžaduje, aby numerické řešení na libovolném, ale pevný čas t zůstává ohraničen jako $h \neq 0$. Obě definice jsou však ve skutečnosti rovnocenné, protože obě definice zakazují nadměrný růst, protože počet kroků se stává libovolně velkým.

Podívejme se nejprve na stabilitu a přesnost v jednoduchém kontextu Eulerovy metody aplikované na skalární ODR, kterou jsme zvažovali v příkladu 9.6, $\dot{y} = \lambda y$, kde λ je (možná komplexní) konstanta. Při počáteční podmínce $y(0) = y_0$ je přesné řešení IVP dáno $y(t) = y_0 e^{\lambda t}$. Použijeme-li na tuto ODR Eulerovu metodu s pevnou velikostí kroku h , dostaneme iterační předpis

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

což znamená, že

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

Člen $1 + h\lambda$ se nazývá růstový faktor. Pokud $\Re(\lambda) < 0$, pak přesné řešení ODR klesá s rostoucím t k nule, stejně jako po sobě jdoucí vypočtené hodnoty řešení, pokud $|1 + h\lambda| < 1$. Pokud $|1 + h\lambda| > 1$, na druhé straně, pak vypočtené hodnoty řešení rostou bez vazby bez ohledu na znaménko $\Re(\lambda)$, což znamená, že Eulerova metoda může být nestabilní, i když je přesné řešení stabilní. Aby byla Eulerova metoda stabilní, musí velikost kroku h splňovat nerovnost

$$|1 + h\lambda| \leq 1,$$

která nám říká, že $h\lambda$ musí v komplexní rovině ležet uvnitř kružnice o poloměru 1 se středem v -1 . Pokud je λ reálná, pak $h\lambda$ musí ležet v intervalu $(-2, 0)$, což znamená, že pro $\lambda < 0$ musíme mít $h - 2 = \lambda$, aby byla Eulerova metoda stabilní. Také si všimneme, že

růstový faktor $1 + h\lambda$ odpovídá ■ **expansion** ■ řady

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

až do členu prvního řádu v h , takže přesnost Eulerovy metody je prvního řádu.

Pro určení přesnosti a stability Eulerovy metody pro obecný systém ODR $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ zvažte Taylorovu řadu

$$y(t+h) = y(t) + h y'(t) + O(h^2) = y(t) + h f(t; y(t)) + O(h^2):$$

Vezmeme-li $t = t_k$ a $h = hk$, dostaneme $y(t_{k+1}) = y(t_k) + hk f(t_k; y(t_k)) + O(h^2k)$

Odečteme-li to nyní od výrazu pro y_{k+1} daného Eulerovou metodou, získáme

$$y_{k+1} - y(t_{k+1}) = [y_k - y(t_k)] + hk[f(t_k; y_k) - f(t_k; y(t_k))] - O(h^2k).$$

Rozdíl na levé straně je globální chyba e_{k+1} . Pokud by nebyly žádné předchozí chyby, pak bychom měli $y_k = y(t_k)$ a první dva rozdíly v závorkách na pravé straně by byly nulové, takže by zůstaly pouze termín $O(h^2k)$, což je místní chyba. Tento výsledek znamená, že Eulerova metoda je přesná prvního řádu.

Z předchozího odvození vidíme, že globální chyba v daném kroku je součtem místní chyby v tomto kroku a toho, co by se dalo nazvat rozšířenou chybou z předchozích kroků. Abychom charakterizovali druhou možnost, můžeme větou o střední hodnotě pro vektorovou funkci vektoru napsat

$$f(t_k; y_k) - f(t_k; y(t_k)) = \bar{\mathbf{J}}_f(y_k - y(t_k));$$

kde $\bar{\mathbf{J}}_f = R10Jf(t_k; ffy_k + (1 - ff)y(t_k))dff$ a Jf je Jakobiánská matice f vzhledem k y , takže globální chybu můžeme vyjádřit na krok $k+1$ jako

$$e_{k+1} = (I + hk\bar{\mathbf{J}}_f)e_k + ' k + 1 :$$

Globální chyba se tedy v každém kroku vynásobí faktorem $I + hk\bar{\mathbf{J}}_f$, který se nazývá růstový faktor nebo faktor zesílení. Chyby nerostou, pokud spektrální poloměr

$$\rho(I + hk\bar{\mathbf{J}}_f) \leq 1,$$

což je splněno, pokud všechna vlastní čísla $h_k\bar{\mathbf{J}}_f$ leží uvnitř kružnice v komplexní rovině o poloměru 1 se středem na -1 . Pokud tomu tak není, pak chyby rostou a metoda je nestabilní, bez ohledu na to, zda je řešení ODR stabilní. Dramatický příklad takové numerické nestability pro stabilní řešení uvidíme v příkladu 9.10.

Všimněte si, že tato obecnější analýza přinesla stejné výsledky stability a přesnosti, jaké jsme získali pomocí jednoduché rovnice skalárního testu $y' = \lambda y$. Zejména u složitějších numerických

metod se s tímto jednoduchým skalárním testem ODR pracuje mnohem snadněji než s obecným ODR a vytváří v podstatě stejnou stabilitu. výsledkem je, pokud srovnáme komplexní koeficient λ s vlastními čísly Jacobiho matice Jf v daném bodě. Důležitou námitkou však je, že λ je konstantní, zatímco Jacobian Jf se mění pro nelineární rovnici, a proto se stabilita může potenciálně změnit. Všimněte si také, že obecně faktor zesílení závisí na konkrétní řešené ODR (která určuje Jacobián Jf), konkrétní číselné použité metoda (která určuje tvar zesilovacího faktoru) a velikost kroku h .

Při výběru velikosti kroku h_k pro pokrok numerického řešení ODR v kroku k bychom chtěli minimalizovat výpočetní náklady tím, že uděláme co největší krok, ale musíme berou v úvahu jak stabilitu, tak přesnost. Aby bylo možné získat smysluplné řešení, musí velikost kroku odpovídat všem omezením stability uloženým použitou metodou. Kromě toho je nutný místní odhad chyb, aby bylo zajištěno dosažení požadované přesnosti. Například u Eulerovy metody rozšíření Taylorovy řady ukazuje, že lokální chyba je přibližně $(h^2k/2)y''_k$, takže velikost kroku by měla splňovat

$$h_k \leq \sqrt{\frac{2tol}{\|y''_k\|}},$$

kde tol je zadaná tolerance lokální chyby. Samozřejmě neznáme hodnotu y''_k , ale můžeme ji odhadnout rozdílovým kvocientem tvaru

$$y''_k \approx \frac{y'_k - y'_{k-1}}{t_k - t_{k-1}}.$$

Jiné metody získávání odhadů lokálních chyb jsou založeny na rozdílu mezi výsledky získanými metodami různých řádů přesnosti nebo různých velikostí kroků.

Implicitní metody

Eulerova metoda je explicitní metoda v tom, že používá pouze informace v čase t_k k posunu řešení do času t_{k+1} . To se může zdát jako ctnost, ale viděli jsme, že Eulerova metoda má poměrně omezenou oblast stability. Větší oblast stability lze získat pomocí informací v čase t_{k+1} , což činí metodu implicitní. Nejjednodušším příkladem je zpětná Eulerova metoda,

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h_k \mathbf{f}(t_{k+1}, \mathbf{y}_{k+1});$$

kterou lze snadno odvodit kteroukoli z metod uvedených v oddíle 9.3.1 pro odvození Eulerovy metody, například numerickou kvadraturou pomocí obdélníkového pravidla $\int_a^b g(s)ds \approx (b-a)g(b)$.

Zpětná Eulerova metoda je implicitní, protože musíme vyhodnotit \mathbf{f} s argumentem \mathbf{y}_{k+1} , aniž známe jeho hodnotu. Tento výrok jednoduše znamená, že musíme nalézt takové \mathbf{y}_{k+1} , jež splňuje výše uvedenou rovnici, a pokud \mathbf{f} je nelineární funkce \mathbf{y} , jak je tomu

často případ, pak musí být použita metoda iterativního řešení, jako je iterace s pevným bodem nebo Newtonova metoda. Dobrý počáteční odhad pro iteraci lze získat z explicitní metody, jako je Eulerova metoda, nebo z řešení v předchozím časovém kroku.

Příklad 9.9 Zpětná Eulerova metoda. Zvažte nelineární skalární ODR

$$y' = -y^3$$

s počáteční podmínkou $y(0) = 1$. Pomocí zpětné Eulerovy metody s velikostí kroku $h = 0,5$ získáme rovnici

$$y_1 = y_0 + h f(t_1; y_1) = 1 - 0,5 y_1^3$$

pro hodnotu řešení v dalším kroku. Tato nelineární rovnice pro y_1 je již nastavena tak, aby se řešila iterací s pevným bodem, opakovaně nahrazující po sobě jdoucí hodnoty pro y_1 na pravé straně, nebo bychom mohli použít jakoukoli jinou metodu z kapitoly 5, například Newtonovu metodu. V každém případě potřebujeme počáteční odhad pro y_1 , pro který bychom mohli jednoduše použít předchozí hodnotu řešení, $y_0 = 1$, nebo bychom mohli použít explicitní metodu k vytvoření počáteční odhad pro implicitní metodu. Například pomocí Eulerovy metody bychom získali $y_1 = y_0 - 0,5 y_0^3 = 0,5$ jako počáteční odhad pro iterativní řešení implicitní rovnice. Iterace nakonec konvergují ke konečné hodnotě $y_1 \approx 0,7709$.

Vzhledem k dalším potížím a výpočtům při použití implicitní metody by se člověk mohl divit, proč bychom se obtěžovali. Odpověď zní, že implicitní metody mají obecně výrazně větší oblast stability než srovnatelné explicitní metody. Pro stanovení stability a přesnosti zpětné Eulerovy metody ji aplikujeme na naši skalární testovací ODR $y' = \lambda y$ a získáme

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

neboli

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

z čehož

$$y_k = \left(\frac{1}{1 - h\lambda} \right)^k y_0.$$

Aby tedy byla zpětná Eulerova metoda stabilní, musí pro h a λ platit

$$\left| \frac{1}{1 - h\lambda} \right| \leq 1,$$

což je splněno pro libovolné $h > 0$, pokud $\Re(\lambda) < 0$. Oblast stability pro zpětnou Eulerovu metodu tedy zahrnuje celou levou polovinu komplexní roviny nebo interval $(-\infty, 0)$ pro $\lambda \in \mathbb{R}$. Odpadají tedy všechna omezení velikosti kroku při výpočtu stabilního řešení.

Růstový faktor

$$\frac{1}{1 - h\lambda} = 1 + h\lambda + (h\lambda)^2 + \dots$$

souhlasí s rozvojem nekonečné řady pro $e^{h\lambda}$ přes členy **řádu h** , takže i zpětná Eulerova metoda má přesnost pouze prvního řádu.

Nyní jsme viděli **dva** příklady implicitních metod, které jsou bezpodmínečně stabilní, ale ne všechny implicitní metody mají tuto vlastnost. Implicitní metody mají obecně větší oblasti stability než explicitní metody, ale povolená velikost kroku není vždy neomezená. Implicitnost sama o sobě nestačí k zajištění stability.

Tuhost

Asymptoticky stabilní řešení se k sobě v průběhu času sbíhají. Tato konvergence má příznivou vlastnost tlumit chyby v numerickém řešení, ale pokud je konvergence řešení příliš rychlá, pak obtíže může vzniknout jiný typ. Typický příklad je znázorněn na obr. 9.8, kde je pomalu se měnící řešení obklopeno jinými řešeními s rychle klesajícími přechodnými jevy. O takové ODR se říká, že je **tuhá**.

Obrázek 9.8: Řešení pro typickou tuhou ODR.

Tuhost lze definovat mnoha různými způsoby. Fyzicky tuhost odpovídá procesu, jehož složky mají velmi nesourodé časové stupnice, nebo procesu, jehož časové měřítko je velmi krátké ve srovnání s intervalem, během kterého je studována. Matematicky je stabilní ODR $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ tuhá, pokud její Jacobiho matice \mathbf{J}_f má vlastní čísla, která se značně liší velikostí. Mohou existovat vlastní čísla s relativně velkými zápornými reálnými částmi (odpovídajícími silně tlumeným složkám řešení) nebo relativně velkými imaginárními částmi (odpovídající rychle oscilujícím složkám řešení). Pragmaticky je ODR tuhá, pokud je explicitní metoda, jako je Eulerova metoda, vysoce neúčinná, protože velikost kroku potřebná k udržení stability je mnohem menší než velikost kroku potřebná k udržení požadované přesnosti. Tuhost je tedy relativní koncept, který závisí na oblasti stability použité metody, požadované přesnosti a délce intervalu integrace, stejně jako ODR.

Pro naši skalární testovací ODR $\dot{y} = \lambda y$ na intervalu $\langle a, b \rangle$ je IVP tuhá, pokud

$$(b - a)\Re(\lambda) \ll -1$$

protože omezení stability velikosti kroku h v Eulerově metodě je poměrně závažné. Obecněji řečeno, ODR v intervalu $\langle a, b \rangle$ je tuhá v blízkosti řešení $y(t)$, pokud

$$(b - a) \min_j (\Re(\lambda_j)) \ll -1,$$

kde λ_j jsou vlastní čísla Jacobiho matice $\mathbf{J}_f(t; y(t))$. Některé numerické metody jsou pro tuhé rovnice velmi neefektivní, protože rychle se měnící složka řešení nutí k udržení stability použít velmi malé velikosti kroků. Vzhledem k tomu, že omezení stability závisí na rychle se měnící složce řešení, zatímco omezení přesnosti závisí na pomalu se měnící složce, krok velikost může být mnohem silněji

omezena stabilitou než požadovanou přesností. Eulerova metoda je například extrémně neefektivní pro řešení tuhé rovnice kvůli její malé stabilitě. Bezpodmínečná stabilita implicitní zpětné Eulerovy metody ji na druhé straně činí vhodnou pro tuhé problémy. Tuhé ODR nemusí být obtížné řešit numericky za předpokladu, že je zvolena vhodná metoda, obecně implicitní.

Příklad 5 (Tuhost). Pro ilustraci numerického řešení tuhé ODR zvažte IVP

$$\dot{y} = -100y + 100t + 101$$

s počáteční podmínkou $y(0) = 1$. Obecné řešení této ODR je $y(t) = 1 + t + ce^{-100t}$ a konkrétní řešení, splňující počáteční podmínku, je $y(t) = 1 + t$ (tj. $c = 0$). Vzhledem k tomu, že řešení je lineární, poskytuje teoreticky Eulerova metoda pro tento problém přesné řešení. Pro ilustraci efektu zaokrouhlovacích a **chyb zkrácení** zkusíme mírně perturbovat počáteční hodnotu a nastavíme ji na 0,99 respektive 1,01. Při velikosti kroku $h = 0,1$ je prvních několik kroků pro dané počáteční hodnoty následujících:

t	0,0	0,1	0,2	0,3	0,4
Přesné řešení	1,00	1,10	1,20	1,30	1,40
Eulerův řešení	0,99	1,19	0,39	8,59	-64,2
Eulerův řešení	1,01	1,01	2,01	-5,99	67,0

Vypočítané řešení je neuvěřitelně citlivé na počáteční hodnotu, protože každá drobná porucha má za následek zcela odlišné řešení. Vysvětlení tohoto chování je znázorněno na obr. 9.9. Jakýkoli bod odchylovající se od požadovaného konkrétního řešení, byť jen o malé množství, leží na jiném řešení, pro který $c \neq 0$, a proto rychlý přechodný je přítomno obecné řešení. Eulerova metoda zakládá svou projekci na derivaci v aktuálním bodě a výsledná velká hodnota způsobí, že se numerické řešení radikálně odchýlí od požadovaného řešení. Toto chování by nás nemělo překvapit. Jacobián pro tuto rovnici je $Jf = -100$, takže podmínka stability pro Eulerovu metodu vyžaduje krokovou velikost $h < 0,02$, kterou porušujeme.

Naproti tomu zpětná Eulerova metoda nemá problém tento problém vyřešit. Ve skutečnosti je zpětné Eulerovo řešení extrémně necitlivé na počáteční hodnotu, jak vidíme v následující tabulce, kde jsme porušili počáteční podmínky podstatně silněji, než v případě, uvedeném výše:

t	0,0	0,1	0,2	0,3	0,4
Přesné řešení	1,00	1,10	1,20	1,30	1,40
BE řešení	0,00	1,01	1,19	1,30	1,40
BE řešení	2,00	1,19	1,21	1,30	1,40

a znázorněno na obr. 9.10. I při velmi velké perturbaci počáteční hodnoty je přechodná hodnota rychle tlumena použitím derivace v dalším bodě namísto aktuálního bodu a zpětné Eulerovo řešení

konverguje k požadovanému řešení již po několika krocích. Toto chování je v souladu s bezpodmínečnou stabilitou zpětné Eulerovy metody pro stabilní rovnici.

Obrázek 9.10: Stabilní řešení tuhé ODR pomocí zpětné Eulerovy metody.

Viděli jsme, že pro tuhé ODR jsou vyžadovány implicitní metody, u nichž musí být v každém iteračním kroku navíc řešena (obecně nelineární) rovnice pro \mathbf{y}_{k+1} . Ačkoli přirozenou volbou pro řešení této rovnice v jiných kontextech je **iterace v pevném bodě**, není to v případě tuhé ODR vhodná volba. Problém spočívá v tom, že aby iterace v pevném bodě konvergovala, musí být velikost kroku h velmi malá, což zmaří primární účel použití implicitní metody, jímž je dosažení snesitelné velikosti integračního kroku h . Výsledkem je, že standardním přístupem pro řešení rovnic v rámci implicitní metody určené pro iterativní numerické řešení tuhé ODR bude Newtonova metoda nebo nějaká její varianta. To na jedné straně přináší značné režijní náklady pro výpočet nebo aproximaci Jacobiho matice a řešení výsledné soustavy lineárních rovnic v každém iteračním kroku, na druhé straně to ale dovoluje mnohem větší velikost integračního kroku a obecně se tato dodatečná režie stále vyplatí. Další námitkou týkající se tuhosti je, že naše diskuse se zaměřila na tuhost způsobenou rychle mizejícími přechodovými jevy. Tuhost způsobená rychle oscilujícími součástmi vyžaduje poněkud odlišný přístup, který zde nebudeme rozebírat.

Metody Taylorovy řady

Viděli jsme, že Eulerova metoda může být odvozena z rozvoje Taylorovy řady. Zachováním více členů v Taylorově řadě můžeme generovat jednokrokové metody vyššího řádu. Například zachováním jednoho dalšího členu v Taylorově řadě

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

dostaneme metodu druhého řádu

$$y_{k+1} = y_k + hky_0' + \frac{h^2}{2}ky_0''.$$

Všimněte si však, že tento přístup vyžaduje výpočet vyšších derivací funkce $\mathbf{y}(t)$. Ty lze získat diferenciací $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$ pomocí řetězového pravidla, například

$$\ddot{\mathbf{y}} = \mathbf{f}_t(t, \mathbf{y}) + \mathbf{f}_y(t, \mathbf{y})\dot{\mathbf{y}} = \mathbf{f}_t(t, \mathbf{y}) + \mathbf{f}_y(t, \mathbf{y})\mathbf{f}(t, \mathbf{y}),$$

kde dolní indexy označují parciální derivace vzhledem k dané proměnné. Jak se řád derivace zvyšuje, tyto výrazy pro derivace se rychle stávají příliš komplikovanými na to, aby byly praktické pro výpočet, takže metody Taylorových řad vyššího řádu nejsou moc často v praxi používány. V poslední době však díky dostupnosti

symbolických manipulačních (Maple, Mathematica, SymPy) a automatických diferenčních systémů (například v programovacím jazyce Julia) se o tyto metody opět výpočtáři začínají zajímat.

Příklad 6 (Metoda Taylorovy řady). Pro ilustraci metody Taylorovy řady druhého řádu ji použijeme k řešení nelineární skalární ODR

$$y'(t) = f(t, y) = -2ty^2$$

s počáteční hodnotou $y(0) = 1$. Diferencujeme f , abychom získali druhou derivaci

$$y''(t) = f_t(t, y) + f_y(t, y)f(t, y) = -2y^2 + (-4ty)(-2ty^2) = 2y^2(4t^2y - 1).$$

Uděláme-li jeden iterační krok z $t_0 = 0$ do $t_1 = 0,25$ (tedy s velikostí kroku $h = 0,25$, získáme

$$y_1 = y_0 + hy'_0 + \frac{h^2}{2}y''_0 = 1 + 0 - 0,0625 = 0,9375.$$

Pokračujeme dalším krokem z $t_1 = 0,25$ na $t_2 = 0,5$ a dostaneme

$$y_2 = y_1 + hy'_1 + \frac{h^2}{2}y''_1 = 0,9375 - 0,1099 - 0,0421 = 0,7856.$$

Pro srovnání, přesné řešení tohoto problému je $y(t) = 1/(1 + t^2)$, a proto skutečné řešení v integračních bodech je $y(0,25) = 0,9412$ a $y(0,5) = 0,8$.

Metody Runge-Kutta

Runge-Kuttovy metody jsou jednokrokové metody, které jsou v motivaci podobné metodám Taylorovy řady, ale nezahrnují explicitní výpočet vyšších derivací. Místo toho Metody Runge-Kutta nahrazují vyšší derivace aproximacemi konečných rozdílů založenými na hodnotách \mathbf{f} v bodech mezi t_k a t_{k+1} . Alternativně lze metody Runge-Kutta odvodit použitím numerických kvadraturních pravidel pro vyhodnocení integrálu

$$y_{k+1} - y_k = \int_{t_k}^{t_{k+1}} f(t; y(t)) dt :$$

V obou případech bude k získání potřebných hodnot \mathbf{f} vyžadován nějaký bootstrapping, protože neznáme druhý argument \mathbf{f} , jmenovitě řešení $y(t)$, pro t mezi t_k a t_{k+1} .

Chcete-li demonstrovat odvození metody Runge-Kutta, vzpomeňte si z oddílu 9.3.5 že druhá derivace y je dána vztahem

$$\ddot{\mathbf{y}} = \mathbf{f}_t + \mathbf{f}_y \mathbf{f},$$

kde každá funkce je vyhodnocena na $(t; y)$. Termín vpravo můžeme aproximovat rozšířením \mathbf{f} v Taylorově řadě ve dvou proměnných

$$f(t + h; y + hf) = f + hft + hfyf + O(h^2);$$

ze kterých získáváme

$$f(t+h; y+hf) = f(t; y)h + O(h) :$$

S touto aproximací k druhé derivaci, Taylorově řadě druhého řádu metoda uvedená v bodě 9.3.5 se stává

$$y_{k+1} = y_k + h_k f(t_k; y_k) + \frac{h_k^2}{2} (2f(t_k + h_k; y_k + h_k f(t_k; y_k)) - f(t_k; y_k)) = y_k + h_k^2 (f(t_k; y_k) + f(t_k + h_k; y_k + h_k f(t_k; y_k)));$$

který je znám jako **Heunova metoda** a může být implementována ve formě

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

kde $k_1 = f(t_k, y_k)$ a $k_2 = f(t_k + h_k, y_k + h_k k_1)$. Alternativně lze Heunovu metodu odvodit použitím lichoběžníkového kvadraturního pravidla pro integraci funkce na intervalu $\langle t_k, t_{k+1} \rangle$, ale s neznámou hodnotou $y(t_{k+1})$ nahrazenou Eulerovou předpovědí $y_k + h_k k_1$ při vyhodnocení f v bodě t_{k+1} .

Příklad 7 (Heunova metoda). Pro ilustraci použití Heunovy metody ji použijeme k řešení nelineární skalární ODR z příkladu 9.11,

$$y'(t) = -2ty^2(t),$$

s počáteční hodnotou $y(0) = 1$. Uděláme-li krok od $t_0 = 0$ do $t_1 = 0,25$ s integračním krokem velikosti $h = 0,25$, získáme

$$k_1 = f(t_0, y_0) = 0, \quad k_2 = f(t_0 + h, y_0 + h k_1) = -0,5,$$

takže

$$y_1 = y_0 + \frac{h}{2} (k_1 + k_2) = 1 + (0,125)(-0,5) = 0,9375.$$

Pokračováním dalšího kroku z $t_1 = 0,25$ na $t_2 = 0,5$ získáme

$$k_1 = f(t_1, y_1) = -0,4395, \quad k_2 = f(t_1 + h, y_1 + h k_1) = -0,6850,$$

a z toho

$$y_2 = y_1 + \frac{h}{2} (k_1 + k_2) = 0,9375 + (0,125)(-1,1245) = 0,7969.$$

Pro srovnání, přesné řešení tohoto problému je $y(t) = 1/(1+t^2)$, a proto skutečné řešení v integračních bodech je $y(0,25) = 0,9412$ a $y(0,5) = 0,8$.

Nejznámější metodou Runge-Kutta je klasické schéma čtvrtého řádu, označované v literatuře často zkratkou RK4:

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

kde

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

Tato metoda je analogická Simpsonovu pravidlu pro numerický výpočet určitých integrálů; ve skutečnosti je Simpsonovým pravidlem, pokud f závisí pouze na t . Pro ilustraci použití klasické metody Runge-Kutta čtvrtého řádu pro řešení systému ODR viz Příklad 10.6.

Metody Runge-Kutta mají řadu ctností. Chcete-li přejít na čas t_{k+1} , nevyžadují žádnou historii řešení před časem t_k , což je činí tak zvané **■ samostartujícími ■** na začátku integrace, a také umožňuje snadnou změnu integračního kroku během integrace. Díky těmto vlastnostem se také metody Runge-Kutta poměrně snadno programují, což částečně odpovídá jejich popularitě.

Na druhou stranu, klasické metody Runge-Kutta neposkytují žádný odhad chyb, na kterém by bylo možné založit volbu velikosti kroku. Naštěstí byly následně vyvinuty **■ vestavěné metody ■** Runge-Kutta, které poskytují odhad chyb založený na rozdílu mezi dvojicí metod různého řádu, sdílejících funkční hodnoty. Například Fehlbergova metoda používá šest vyhodnocení funkcí na krok k vytvoření přesných aproximací řešení čtvrtého i pátého řádu. Další účinný 4(5) **■ vestavěný pár ■** byl vyvinut Dormandem a Princem a je dostupný jako metoda ode45 v Matlabu.

Tento přístup vedl k automatickým řešičům Runge-Kutta, které jsou účinné pro mnoho problémů, ale relativně neefektivní pro tuhé problémy nebo když je vyžadována velmi vysoká přesnost. Je však možné definovat i implicitní Runge-Kuttovy metody s vynikajícími vlastnostmi stability, jež jsou vhodné pro řešení tuhých obyčejných diferenciálních rovnic.

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 $d(y(t) + c)/dt = y'(t) + 0 = y'(t)$

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 $y_0 = y(t_0)$

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

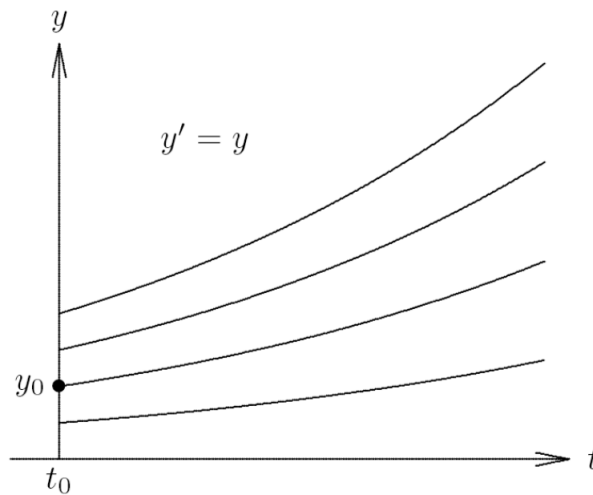
Uvažujme skalární ODR

$$\dot{y} = y$$

Množina řešení je dána $y(t) = c e^t$, 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 $y_0 = c e^0$ neboli $c = y_0$, což znamená, že hledané řešení ODR je $y(t) = y_0 e^t$



Množina řešení pro ODR $\dot{y} = y$

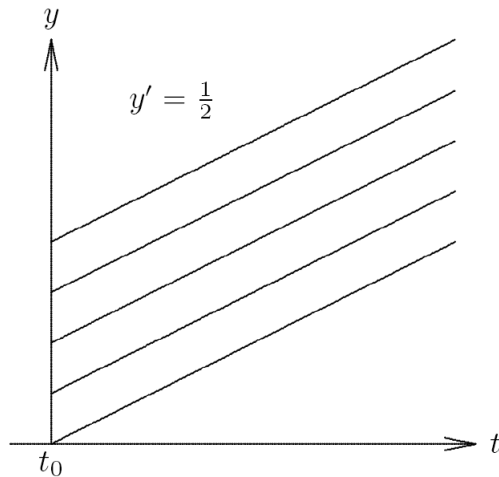
Stabilita řešení ODR

Řešení ODR je

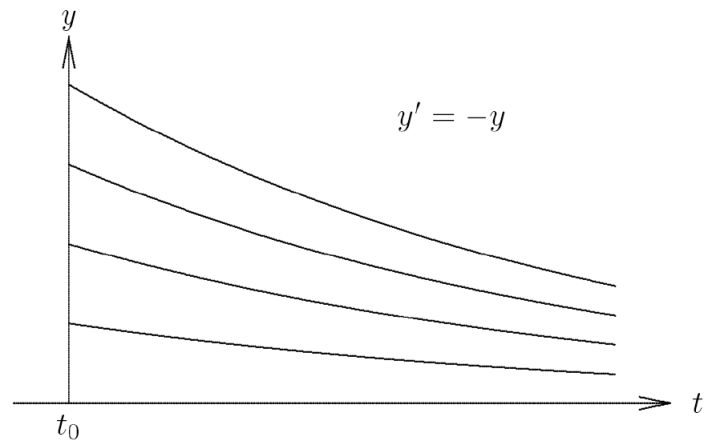
Stabilní, pokud řešení vzniklá perturbací počáteční hodnoty zůstávají blízko původního řešení

Asymptoticky stabilní, pokud jsou řešení pro porušené počáteční podmínky **■ rozdílná? ■**, ale konvergují zpět k původnímu řešení

Nestabilní, pokud řešení po změně počátečních podmínek divergují od originálního řešení nade všechny meze



Množina řešení pro ODR $\dot{y} = \frac{1}{2}$



Množina řešení pro ODR $\dot{y} = -y$

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

Řešení je dáno $y(t) = y_0 e^{\lambda t}$, 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

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 $\mathbf{v}_i, i = 1, \dots, n$

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 $\mathbf{y}(0) = \mathbf{y}_0$

Vlastní čísla \mathbf{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 $\Re(\lambda_i) > 0$.

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

Eulerova metoda

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

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

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

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

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

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 $y(0,5) = \exp(0,5) \approx 1,649$

↓

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

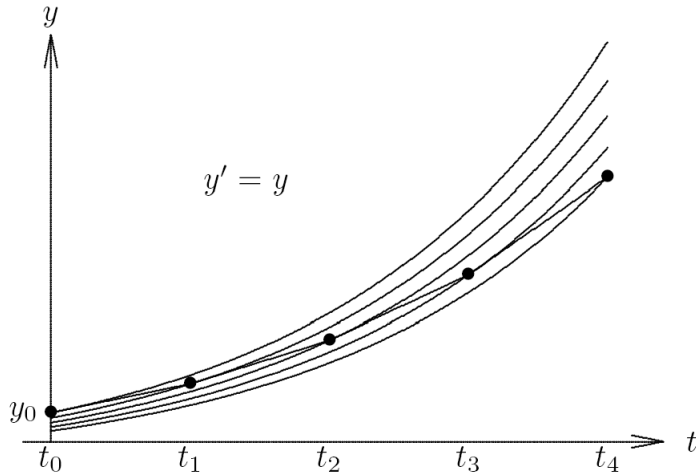
Budeme-li pokračovat v procesu numerického řešení, uděláme další krok od t_1 do $t_2 = t_1 + h = 1,0$ a obdržíme $y_2 = y_1 + hy_1 = 1,5 + 0,5 \cdot 1,5 = 2,25$

Nyní se y_2 liší

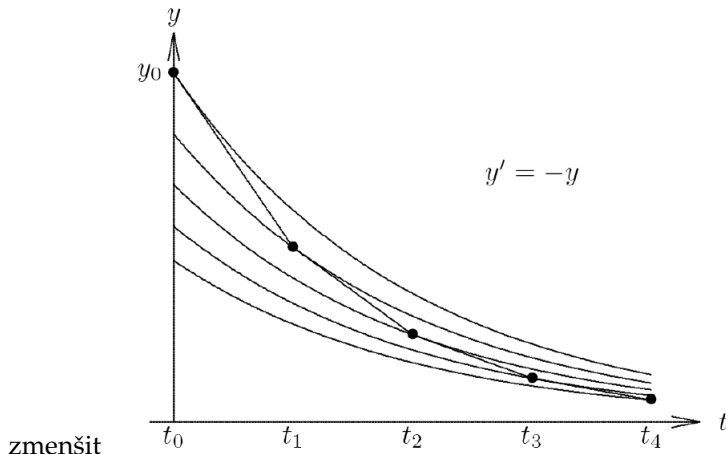
nejen od skutečného řešení původního problému v $t = 1,0$,
 $y(1,0) = \exp(1,0) \approx 2,718$, ale také
 od řešení, které je přesné v předchozím bodě $[t_1, y_1]$; to má
 pro $t = 1,0$ hodnotu přibližně 2,473

Tak jsme se opět přesunuli k další funkci řešení pro tuto ODR

U nestabilních řešení chyby v numerickém řešení rostou s časem



U stabilních řešení se chyby v numerickém řešení mohou časem



zmenšit

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.

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

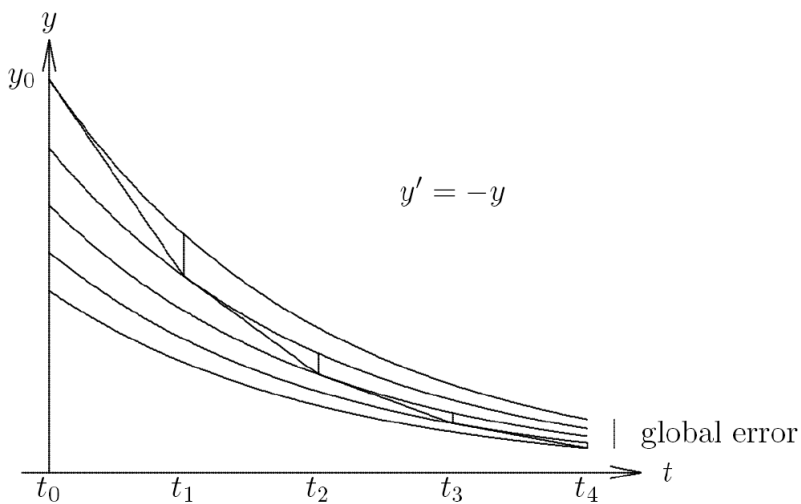
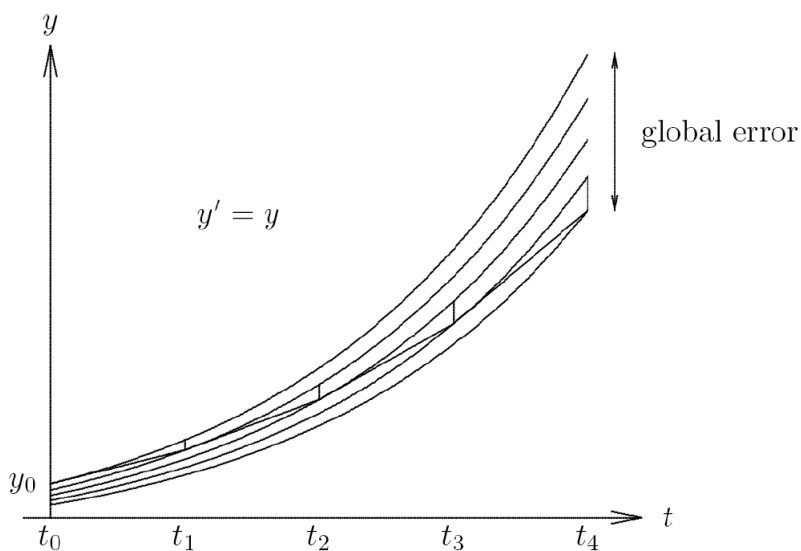
$$\ell_k = \mathbf{y}_k - \mathbf{u}_{k-1}(t_k)$$

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

Globální chyba 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



Přesnost a stabilita

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*

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

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 $y(t) = y_0 e^{\lambda t}$, 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 .

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

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

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

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

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

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

Hodnotu \dot{y} přitom neznáme, můžeme ji ale odhadnout poměrem rozdílů

$$\dot{y} \approx \frac{y_k - y_{k-1}}{t_k - t_{k-1}}$$

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

Implicitní metody

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

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

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 $y_1 \approx 0,7709$

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

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

Ne všechny implicitní metody to jsou bezpodmínečně 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 některé z následujících kategorií:

Jednokrokové metody

Taylorovy řady

Runge-Kutta

Extrapoláčnické

Více krokové metody

Vícehodnotové metody

Soustředíme se pouze na jednokrokové metody.

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

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

dává metodu druhého řádu

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

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

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

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

Nejjednodušším příkladem je Heunova metoda druhého řádu

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

kde

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

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

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

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

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

Extrapoláčnické 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 $Y(0)$

Extrapoláčnické 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

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

Vícehodnotové metody vynecháváme

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 $\lambda < 0$

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