

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

BAKALÁŘSKÁ PRÁCE

Metody Runge-Kutta



Vedoucí bakalářské práce:
RNDr. Jitka Machalová, Ph.D.
Rok odevzdání: 2011

Vypracoval:
Ondřej Balódy
MAP, III. ročník

Prohlášení

Prohlašuji, že jsem vytvořil tuto bakalářskou práci samostatně za vedení RNDr. Jitky Machalové, Ph.D. a že jsem v seznamu použité literatury uvedl všechny zdroje použité při zpracování práce.

V Olomouci dne 5. prosince 2011

Poděkování

Rád bych na tomto místě poděkoval vedoucí bakalářské práce RNDr. Jitce Machalové, Ph.D. za obětavou spolupráci i za čas, který mi věnovala při konzultacích. Dále si zaslouží poděkování můj počítač, že vydržel moje pracovní tempo, a typografický systém \TeX , kterým je práce vysázena.

Obsah

Použitá symbolika	4
Úvod	5
1 Diferenciální rovnice	7
1.1 Řešení diferenciálních rovnic	9
2 Eulerova metoda	13
3 Metody Runge-Kutta	50
3.1 Explicitní metody Runge-Kutta	51
3.2 Implicitní metody Runge-Kutta	73
Závěr	90
Literatura	91

Použitá symbolika

\mathbb{N}	...	množina přirozených čísel
\mathbb{R}	...	množina reálných čísel
\mathbb{R}^n	...	množina všech n -tic reálných čísel
$J \subseteq \mathbb{R}$...	interval z \mathbb{R}
$C^n(J)$...	množina funkcí, které mají spojité všechny derivace do řádu n , včetně, na intervalu $J \subseteq \mathbb{R}$
(x, y)	...	souřadnice bodu
$[a, b]$...	uzavřený interval
$y(x_i)$...	hodnota přesného řešení $y(x)$ v bodě x_i
y_i	...	hodnota přibližného řešení v bodě x_i
$\mathbf{x} = (x_1, x_2, \dots, x_n)^T$...	vektor z \mathbb{R}^n

Úvod

Tato bakalářská práce je zaměřena na metody Runge-Kutta, což jsou numerické metody, pomocí nichž hledáme hodnoty přibližného řešení Cauchyovy (počáteční) úlohy. Jelikož numerické řešení diferenciálních rovnic (NŘDR) nebylo v základním kurzu mého bakalářského studia, musel jsem danou problematiku nastudovat. Cílem této práce je seznámení čtenáře se základy NŘDR.

Hlavním těžištěm bakalářské práce je Eulerova metoda, jelikož ji můžeme považovat za nejjednodušší metodu Runge-Kutta. Na této metodě si v textu ukážeme základní princip fungování numerických metod. Dále si vysvětlíme, jak pomocí numerické metody najít hodnoty přibližného řešení diferenciální rovnice prvního řádu, včetně počáteční podmínky, tj. počáteční úlohy tvaru

$$\begin{aligned}y' &= f(x, y) \\ y(x_0) &= y_0\end{aligned}$$

ale také Cauchyovy úlohy, která je dána vyjádřením

$$\begin{aligned}\mathbf{y}' &= \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(x_0) &= \mathbf{y}_0.\end{aligned}$$

Tuto úlohu považujeme za systém diferenciálních rovnic prvního řádu s počátečními podmínkami.

V první kapitole nazvané Diferenciální rovnice se okrajově zabýváme teorií diferenciálních rovnic n -tého i prvního řádu. Dále uvádíme, jakým způsobem lze převést diferenciální rovnici n -tého řádu na systém diferenciálních rovnic prvního řádu a jaký je vztah mezi jejich řešeními.

Druhá kapitola nazvaná Eulerova metoda je těžištěm bakalářské práce. V této kapitole uvádíme základní princip Eulerovy metody, dva způsoby řešení počáteční úlohy pomocí Eulerovy metody (ekvidistantní a neekvidistantní síť uzlů) pro jedno i více rozměrný případ Cauchyovy úlohy.

Ve třetí kapitole nazvané Metody Runge-Kutta uvažujeme explicitní a implicitní numerické metody, přičemž se dále v jednotlivých podkapitolách zabý-

váme zvlášť explicitními metodami a zvlášť implicitními metodami Runge-Kutta. V každé podkapitole uvádíme různé metody Runge-Kutta, včetně příkladů na užití uvedených metod.

Dále jsou ke každému tématu ručně počítané příklady, k nimž jsem vytvořil i M-fily v matematickém softwaru MATLAB. Všechny tyto M-fily jsou také uloženy na CD, které je k dispozici na zadní straně přebalu.

Pro přehlednost je konec **Příkladu** označen symbolem ♣ a začátek a konec každého **M-filu** je oddělen od textu přerušovanou čarou.

1 Diferenciální rovnice

Nejprve definujeme diferenciální rovnici n -tého řádu, řešení diferenciální rovnice a Cauchyovu úlohu pro diferenciální rovnici n -tého řádu. Dále si ukážeme, jak diferenciální rovnici n -tého řádu převést na systém diferenciálních rovnic prvního řádu.

Definice 1.1. Nechť je dána funkce $F : J \times G \rightarrow \mathbb{R}$, kde $J \subseteq \mathbb{R}$ a $G \subset \mathbb{R}^{n+1}$.
Rovnici

$$F(x, y(x), y'(x), \dots, y^{(n)}(x)) = 0,$$

kde $x \in J, y(x) \in C^n(J)$, nazýváme *diferenciální rovnici n -tého řádu*.

Definice 1.2. Diferenciální rovnici n -tého řádu v *normálním tvaru* rozumíme diferenciální rovnici ve tvaru

$$y^{(n)}(x) = f(x, y(x), y'(x), \dots, y^{(n-1)}(x)), \quad (1.1)$$

kde $f : J \times G \rightarrow \mathbb{R}$, $J \subseteq \mathbb{R}, G \subset \mathbb{R}^n, x \in J$ a $y(x) \in C^n(J)$.

Poznámka 1.1. Řád diferenciální rovnice je nejvyšší derivace funkce $y(x)$, která se v diferenciální rovnici vyskytuje.

Definice 1.3. Funkci $y(x) : J \rightarrow \mathbb{R}$, kde $J \subseteq \mathbb{R}$, nazýváme *řešením diferenciální rovnice (1.1) na J* , jestliže

1. $y(x) \in C^n(J)$,
2. $(x, y(x), y'(x), \dots, y^{(n-1)}(x)) \in J \times G, J \times G \subset \mathbb{R}^{n+1}$, pro každé $x \in J$,
3. rovnost (1.1) platí pro každé $x \in J$.

Grafem řešení $y(x)$ diferenciální rovnice (1.1) je křivka, kterou nazýváme *křivkou integrální*.

Definice 1.4. Nechť $(x_0, \psi_0, \psi_1, \dots, \psi_{n-1}) \in J \times G$, kde $J \times G \subset \mathbb{R}^{n+1}$. Úlohu najít řešení $y(x)$ diferenciální rovnice (1.1) splňující podmínky

$$y^{(j)}(x_0) = \psi_j, \text{ pro } j = 0, 1, 2, \dots, n-1, \quad (1.2)$$

přičemž symbolem $y^{(0)}(x)$ rozumíme funkční hodnotu v bodě x , tj. platí vztah $y^{(0)}(x_0) = y(x_0) = \psi_0$, nazýváme *Cauchyho (počáteční) úlohou*. Podmínky dané vztahem (1.2) nazýváme *Cauchyho počáteční podmínky*.

Poznámka 1.2. Analogicky k výše uvedeným definicím můžeme uvažovat diferenciální rovnici prvního řádu tvaru

$$y'(x) = f(x, y(x)).$$

Dále se v textu budeme zabývat pouze diferenciálními rovnicemi prvního řádu, neboť jak si v následující poznámce ukážeme, každou diferenciální rovnici n -tého řádu lze převést na soustavu n diferenciálních rovnic prvního řádu. Pokud budeme uvažovat Cauchyho (počáteční) úlohu pro diferenciální rovnici prvního řádu, ta bude v souladu s předešlými formulacemi tvaru

$$y'(x) = f(x, y(x)) \tag{1.3}$$

$$y(x_0) = y_0, \tag{1.4}$$

kde $x_0 \in J$ a $y_0 \in \mathbb{R}$.

Jestliže uvažujeme $\boldsymbol{\psi} = (\psi_1, \psi_2, \dots, \psi_n)^T \in \mathbb{R}^n$, $\mathbf{y} = (y_1(x), y_2(x), \dots, y_n(x))^T$ a vektorovou funkci $\mathbf{f} = (f_1, f_2, \dots, f_n)^T$, kde $y_i(x) \in C^1(J)$, $f_i : J \times G \rightarrow \mathbb{R}$, $J \subseteq \mathbb{R}$, $G \subset \mathbb{R}$ pro $i = 1, 2, \dots, n$, pak počáteční úlohu můžeme vyjádřit vztahem

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)) \tag{1.5}$$

$$\mathbf{y}(x_0) = \boldsymbol{\psi}, \tag{1.6}$$

kde $x \in J$ a $x_0 \in J$. Vztah (1.5) nazýváme *system n diferenciálních rovnic prvního řádu s počátečními podmínkami* ve tvaru (1.6).

Poznámka 1.3. Uvažujme nyní diferenciální rovnici n -tého řádu tvaru (1.1), kde $n \geq 2$. Přiřaďme k ní system diferenciálních rovnic prvního řádu následujícím způsobem. Zavedeme substituci

$$\begin{aligned} y_1(x) &= y(x), \\ y_2(x) &= y'(x), \\ y_3(x) &= y''(x), \\ &\vdots \\ y_n(x) &= y^{(n-1)}(x), \end{aligned} \tag{1.7}$$

potom systém diferenciálních rovnic přiřazený k diferenciální rovnici (1.1) je tvaru

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= y_3(x), \\ &\vdots \\ y_{n-1}'(x) &= y_n(x), \\ y_n'(x) &= f(x, y_1(x), y_2(x), \dots, y_n(x)), \end{aligned} \tag{1.8}$$

kde $f : J \times G \rightarrow \mathbb{R}$, $J \times G \subset \mathbb{R}^{n+1}$. Tento systém diferenciálních rovnic je speciálním případem systému (1.5), přičemž systém (1.8) píšeme ve vektorové podobě následovně

$$\mathbf{y}' = g(x, \mathbf{y}),$$

kde $g(x, \mathbf{y}) = (y_2(x), y_3(x), \dots, y_n(x), f(x, \mathbf{y}))^T$, přičemž $\mathbf{y} = (y_1(x), \dots, y_n(x))^T$ a tedy $\mathbf{y}' = (y_1'(x), y_2'(x), \dots, y_n'(x))^T$.

Systém diferenciálních rovnic prvního řádu (1.8) je ekvivalentní s diferenciální rovnicí (1.1) ve smyslu následující věty publikované v [1], str. 40.

Věta 1.1. *Nechť $u : J \rightarrow \mathbb{R}$, kde $J \subseteq \mathbb{R}$, je řešení diferenciální rovnice (1.1), pak funkce*

$$y(x) = (u(x), u'(x), \dots, u^{(n-1)}(x))^T$$

je řešením systému (1.8) na intervalu J . Naopak je-li $y(x) = (y_1(x), \dots, y_n(x))^T$ řešením systému (1.8) na $J \subseteq \mathbb{R}$, pak funkce $u(x) = y_1(x)$ je řešením diferenciální rovnice (1.1) na J .

1.1 Řešení diferenciálních rovnic

K řešení diferenciální rovnice můžeme přistupovat ze dvou stran.

Prvním přístupem k řešení diferenciální rovnice je, že hledáme přesné řešení, tj. předpis funkce $y(x)$. Toto řešení se nazývá *analytické řešení*. Hledání analytického řešení diferenciální rovnice je často složité a velice pracné. K analytickým

metodám patří a nejčastěji užívané jsou např. užití Taylorovy řady, separace proměnných, atd. Tyto metody jsou rozebírány v základních kurzech Obyčejných diferenciálních rovnic, proto se zde jimi nebudeme zabývat. Pouze si pro názornost uvedeme příklad analytického řešení dané diferenciální rovnice.

Příklad 1.1. Analytickým řešením diferenciální rovnice

$$y'(x) = y(x)$$

je například funkce tvaru

$$y(x) = e^x,$$

ale také

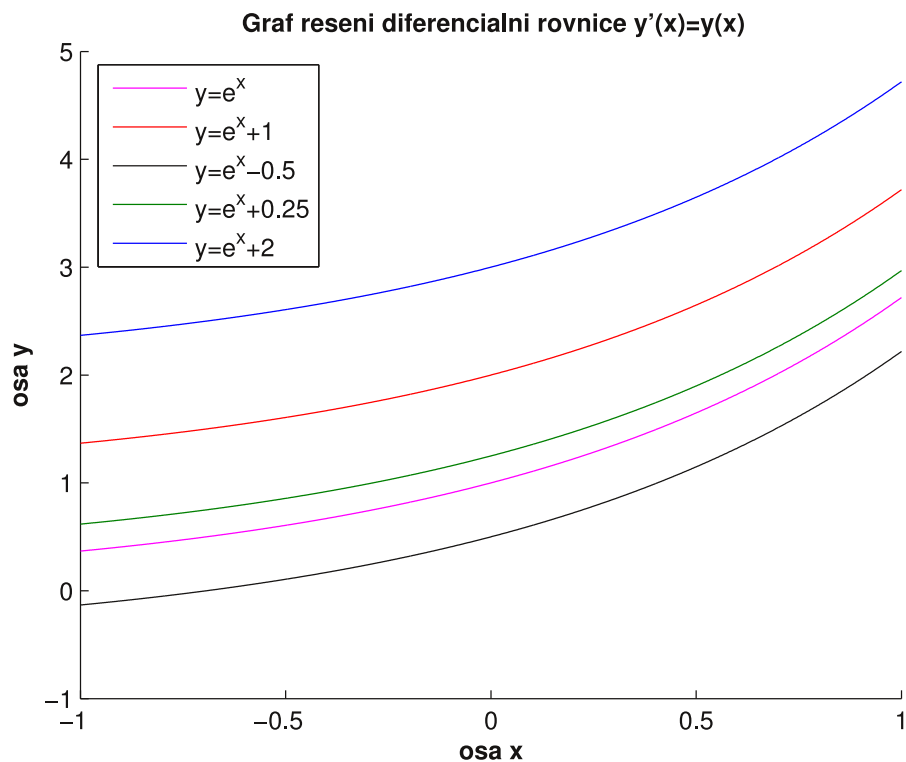
$$y(x) = e^x + C,$$

kde $C \in \mathbb{R}$ je libovolná konstanta, což lze jednoduše ověřit zderivováním funkce $y(x)$. Na závěr tohoto ilustračního příkladu si vykreslíme graf s některými řešeními uvažované diferenciální rovnice, viz. Obrázek 1. ♣

Druhým přístupem k řešení diferenciálních rovnic je hledání *numerického řešení*. Jedná se o hledání přibližných hodnot přesného řešení $y(x)$ diferenciální rovnice na diskrétní množině bodů

$$a = x_0 < x_1 < x_2 < \dots < x_N = b, \tag{1.9}$$

kde $x_i \in \mathbb{R}$ pro $i = 0, 1, 2, \dots, N$ a $N \in \mathbb{N}$ je předem zvolené. Numerické řešení je tedy znalost přibližných funkčních hodnot řešení v bodech x_i . Tyto hodnoty budeme značit y_i . Numerické řešení hledáme pomocí tzv. *numerické metody*, což je předpis (algoritmus), podle něhož počítáme hodnoty přibližného řešení y_i pro $i = 0, 1, 2, \dots, N$. Každý algoritmus numerické metody potřebuje nějaká vstupní data. Těmi jsou, kromě diferenciální rovnice, počáteční (startovací) bod (x_0, y_0) , který můžeme vyčíst z dané počáteční podmínky, krok h_i pro $i = 0, 1, 2, \dots, N - 1$, představující vzdálenost bodu x_i a x_{i+1} , a interval $[a, b]$, na kterém máme danou diferenciální rovnici řešit. Na základě těchto daných dat hledáme pomocí



Obrázek 1: Graf řešení diferenciální rovnice

$$y'(x) = y(x)$$

dané numerické metody posloupnost hodnot přibližného řešení $\{y_i\}_{i=1}^N$ v bodech x_1, x_2, \dots, x_N . Body x_1, x_2, \dots, x_N konstruujeme pomocí vztahu

$$x_{i+1} = x_i + h_i,$$

kde $i = 0, 1, 2, \dots, N - 1$.

Podle počtu hodnot přibližného řešení potřebných k výpočtu následující hodnoty přibližného řešení dělíme numerické metody na *jednokrokové* a *mnohokrokové*. K tomu, abychom mohli při hledání hodnoty přibližného řešení y_{i+1} diferenciální rovnice užít jedнокrokové metody, potřebujeme znát pouze předchozí hodnotu y_i . Obecně můžeme jedнокrokovou metodu psát ve tvaru

$$y_{i+1} = y_i + h_i \Phi(x_i, y_i, h_i),$$

kde funkce $\Phi(x_i, y_i, h_i)$ je funkcí tří proměnných x_i, y_i a h_i a závisí na pravé straně dané diferenciální rovnice.

Ovšem při hledání hodnoty y_{i+1} pomocí mnohokrokové metody potřebujeme znát několik předchozích hodnot řešení $y_i, y_{i-1}, y_{i-2}, \dots$, např. v případě *k-krokové metody* potřebujeme k získání hodnoty y_{i+1} znát $y_i, y_{i-1}, y_{i-2}, \dots, y_{i+1-k}$ předchozích hodnot.

Dále si uvedeme definice výše uvedených pojmů množiny $\{x_i\}_{i=0}^N$ a kroku h_i .

Definice 1.5. Nechť je dán interval $[a, b]$, kde $a, b \in \mathbb{R}$, a číslo $N \in \mathbb{N}$. Množinu bodů $\{x_i\}_{i=0}^N$ definovaných vztahem (1.9) nazýváme *síť* a body x_i , kde $i = 0, 1, 2, \dots, N$, nazýváme *uzly*, nebo *dělicí body*. Reálné číslo $h_i > 0$ definované vztahem

$$h_i = x_{i+1} - x_i,$$

kde $i = 0, 1, 2, \dots, N - 1$, nazýváme *krokem*, popř. *délkou kroku*. Pokud je krok h_i konstantní, tj. platí

$$h_0 = h_1 = \dots = h_{N-1} = h, \text{ kde } h \in \mathbb{R}, h > 0,$$

nazýváme množinu $\{x_i\}_{i=0}^N$ *ekvidistantní sítí*, přičemž platí

$$x_i = a + ih,$$

kde $i = 0, 1, 2, \dots, N$.

2 Eulerova metoda

Nejprve se budeme zabývat Eulerovou metodou pro diferenciální rovnici prvního řádu v normálním tvaru, uvažujme tedy

$$\left. \begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0, \end{aligned} \right\} \quad (2.1)$$

přičemž $x_0 \in J, J \subseteq \mathbb{R}$, a $y_0 \in \mathbb{R}$. Budeme tedy hledat hodnoty přibližného řešení y_i pro $i = 0, 1, 2, \dots, N$. Jednorozměrný případ uvažujeme z důvodu lepšího pochopení této metody a uvedení čtenáře do tématu, potom přejdeme k vektorovému zápisu, tj. k hledání přibližného řešení soustavy diferenciálních rovnic prvního řádu.

Základní myšlenka Eulerovy metody spočívá v tom, že průběh funkce $f(x, y)$ aproximujeme na intervalu $[x_i, x_{i+1}]$ přímkou procházející bodem (x_i, y_i) se směrnici $y'_i = f(x_i, y_i)$. Rovnice této přímky je potom tvaru

$$y = y_i + (x - x_i)f(x_i, y_i)$$

a položíme-li $x = x_{i+1}$ a $y = y_{i+1}$ dostaneme rekurentní vztah pro výpočet y_{i+1}

$$y_{i+1} = y_i + h_i f(x_i, y_i), \quad (2.2)$$

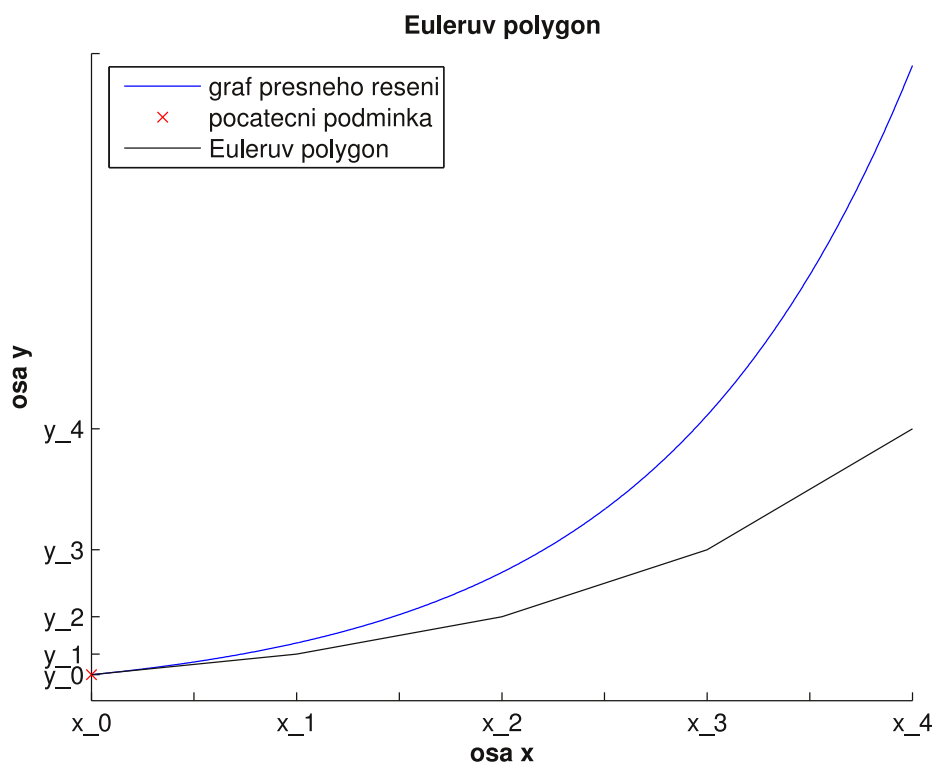
kde $i = 0, 1, 2, \dots, N - 1$. Vztah (2.2) nazýváme Eulerova metoda, což je jednokroková numerická metoda.

Na obrázku 2 je znázorněn tzv. Eulerův polygon, což je lomenná křivka spojující body (x_i, y_i) pro $i = 0, 1, 2, \dots, N$ získané užitím vztahu (2.2).

Užitím vztahu (2.2) vygenerujeme množinu hodnot přibližného řešení $\{y_i\}_{i=0}^N$, kde je hodnota y_0 dána počáteční podmínkou $y(x_0) = y_0$. Jednotlivé hodnoty y_i počítáme prostřednictvím vztahu (2.2), znalosti kroku h_i a posloupnosti $\{x_i\}_{i=0}^N$, která je dána vztahem

$$a = x_0 < x_1 < x_2 < \dots < x_N = b$$

a pro niž platí $x_{i+1} = x_i + h_i$ pro $i = 0, 1, 2, \dots, N - 1$. Uzly sítě tedy dělí interval $[a, b]$, na němž máme diferenciální rovnici $y' = f(x, y)$ řešit, na podintervaly,



Obrázek 2: Eulerův polygon

které budeme také nazývat dělicími intervaly. Počtem dělicích intervalů budeme rozumět číslo $N \in \mathbb{N}$.

Pokud budeme hledat numerické řešení diferenciální rovnice pomocí Eulerovy metody, budeme postupovat jedním z následujících způsobů.

I. První způsob řešení počáteční úlohy Eulerovou metodou.

Uvažujme ekvidistantní síť uzlů a počáteční úlohu (2.1). Naším úkolem je najít numerické řešení této úlohy pomocí Eulerovy metody na intervalu $[a, b]$, kde $a = x_0$ a $b = x_N$, s daným buďto

a) krokem h . Přičemž uvažujeme takový krok h , že rozdíl $b - a$ je celočíselným

násobkem tohoto kroku. Počet dělicích intervalů N určíme ze vztahu

$$N = \frac{b - a}{h}$$

nebo

b) počtem dělicích intervalů N . Pokud známe N , určíme krok h z vyjádření

$$h = \frac{b - a}{N}. \quad (2.3)$$

Dále pokračujeme tak, že ve vzorci (2.2) zvolíme $i = 0$. Předpis této metody bude tvaru

$$y_1 = y_0 + hf(x_0, y_0),$$

kde x_0 a y_0 známe z počáteční podmínky, přičemž $x_0 = a$. Nyní můžeme vypočítat hodnotu přibližného řešení y_1 v bodě $x_1 = x_0 + h$. Tímto jsme skončili s prvním krokem. Pokračujeme druhým krokem, volíme tedy $i = 1$ a vztah této metody pro y_2 bude

$$y_2 = y_1 + hf(x_1, y_1),$$

kde y_1 známe z předchozího kroku a $x_1 = x_0 + h$. Obdobně bychom pokračovali dále až do $i = N - 1$, v tomto případě bude vztah (2.2) tvaru

$$y_N = y_{N-1} + hf(x_{N-1}, y_{N-1}),$$

kde y_{N-1} známe a $x_{N-1} = x_{N-2} + h$.

II. Druhý způsob řešení počáteční úlohy Eulerovou metodou.

Nechť je nyní dána neekvidistantní síť uzlů a počáteční úloha (2.1). Naším úkolem je vypočítat přibližné řešení dané úlohy Eulerovou metodou, přičemž známe pouze bod a a vektor kroků $\mathbf{h} = (h_0, h_1, \dots, h_{N-1})^T$. Nejprve stanovíme počet dělicích intervalů N , což je počet složek vektoru \mathbf{h} , a síť $\{x_i\}_{i=0}^N$, kde $x_0 = a$ a $x_{i+1} = x_i + h_i$ pro $i = 0, 1, 2, \dots, N - 1$. Dále postupujeme stejně jako v případě I.

Nyní si uvedeme dva příklady, ve kterých budeme pomocí Eulerovy metody hledat hodnoty přibližného řešení y_i a užijeme dvou výše uvedených způsobů řešení.

Příklad 2.1. Pomocí Eulerovy metody najděte hodnoty přibližného řešení počáteční úlohy tvaru

$$\begin{aligned}y' &= y \\ y(0) &= 1\end{aligned}$$

na intervalu $[0, 1.4]$ s počtem dělicích intervalů $N = 7$. Uvažujme ekvidistantní síť.

Řešení: Jelikož uvažujeme ekvidistantní síť $\{x_i\}_{i=0}^7$, tak určíme krok h ze vztahu (2.3), tedy $h = 0.2$. Eulerova metoda je dána předpisem

$$y_{i+1} = y_i + hf(x_i, y_i) \text{ pro } i = 0, 1, 2, \dots, N - 1$$

a v tomto případě je tvaru

$$y_{i+1} = y_i + hy_i = (1 + h)y_i = 1.2y_i.$$

Nyní položíme $i = 0$ a počítáme hodnotu y_1 následujícím způsobem

$$y_1 = y_0 + hy_0 = 1.2y_0,$$

kde y_0 je dáno počáteční podmínkou. Po dosazení konkrétních hodnoty y_0 máme

$$y_1 = 1.2.$$

Pokračujeme dále volbou $i = 1$ a výpočtem hodnoty y_2 takto

$$y_2 = (1 + h)y_1 = 1.2 \cdot 1.2 = 1.44.$$

Obdobným dosazováním pokračujeme až do volby $i = 6$.

Na závěr tohoto příkladu si uvedeme tabulku vypočtených hodnot přibližného řešení y_i , hodnoty přesného řešení $y(x_i)$, chybu e_i , pro kterou platí

$$e_i = |y(x_i) - y_i|.$$

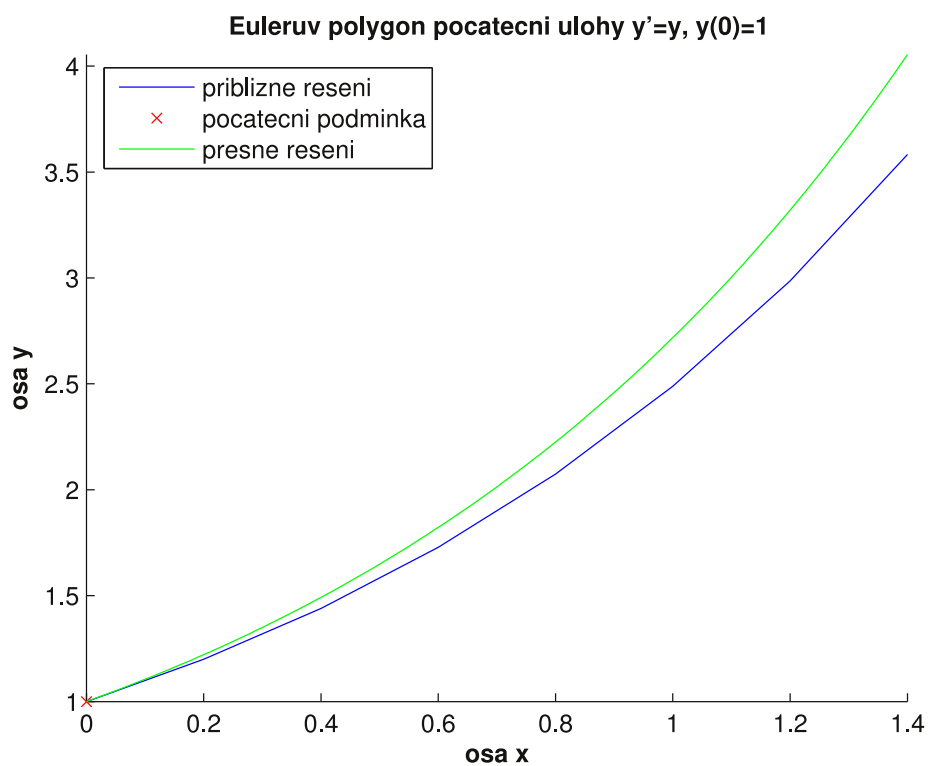
Dále si vykreslíme Eulerův polygon a graf přesného řešení dané počáteční úlohy, jež je tvaru

$$y = e^x,$$

viz. Obrázek 3.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i	1.0000	1.2000	1.4400	1.7280	2.0736	2.4883	2.9860	3.5832
$y(x_i)$	1.0000	1.2200	1.4900	1.8220	2.2255	2.7183	3.3201	4.0552
e_i	0.0000	0.0200	0.0500	0.0940	0.1519	0.2300	0.3341	0.4720

Tabulka 1: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení



Obrázek 3: Eulerův polygon a graf přesného řešení počáteční úlohy

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

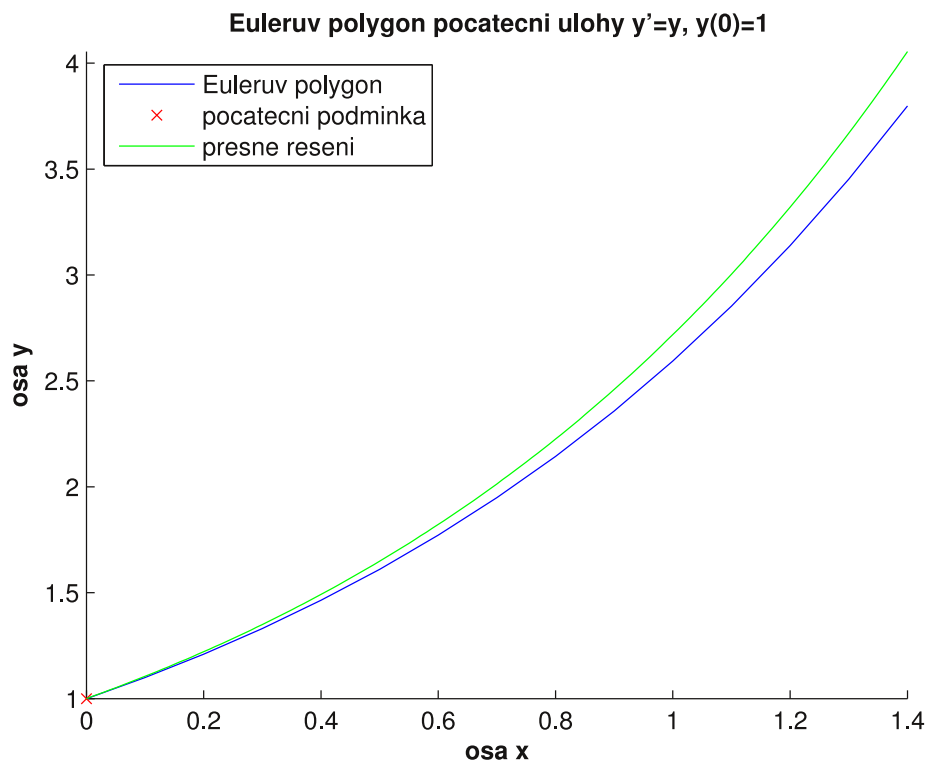
Nyní budeme uvažovat ten samý příklad s tím rozdílem, že zvolíme $N = 14$. Postupujeme stejným způsobem jako výše. Jelikož postup, kterým počítáme hodnoty přibližného řešení y_i známe, tak si pouze uvedeme tabulku s vypočtenými hodnotami přibližného a přesného řešení i s chybou e_i , včetně příslušného Eulera polygonu a přesného řešení dané počáteční úlohy, viz. Obrázek 4.

i	x_i	y_i	$y(x_i)$	e_i
0	0.0	1.0000	1.0000	0.0000
1	0.1	1.1000	1.1052	0.0052
2	0.2	1.2100	1.2214	0.0114
3	0.3	1.3310	1.3499	0.0189
4	0.4	1.4641	1.4918	0.0277
5	0.5	1.6105	1.6487	0.0382
6	0.6	1.7716	1.8221	0.0505
7	0.7	1.9488	2.0138	0.0650
8	0.8	2.1437	2.2255	0.0818
9	0.9	2.3581	2.4596	0.1015
10	1.0	2.5939	2.7183	0.1244
11	1.1	2.8533	3.0042	0.1509
12	1.2	3.1386	3.3201	0.1815
13	1.3	3.4525	3.6693	0.2168
14	1.4	3.7978	4.0552	0.2574

Tabulka 2: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení

Z právě uvedené tabulky 2 a tabulky 1 vyplývá, že pokud budeme hledat hodnoty přibližného řešení y_i na větším počtu uzlů x_i nebo s menším krokem h , bude chyba e_i menší. Pro srovnání se podívejme na chybu e_i v koncovém bodě daného intervalu, tedy v bodě 1.4, u obou tabulek, v případě tabulky 1 je $e_7 = 0.4720$ a v tabulce 2 je chyba $e_{14} = 0.2574$. Vidíme, že chyba je téměř poloviční, museli jsme ovšem vypočítat dvojnásobně více hodnot přibližného řešení. ♣

Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file Eulerovy metody pro ekvidistantní síť uzlů s kontrolou vstupních dat, který si zde uvedeme.



Obrázek 4: Eulerův polygon a graf přesného řešení počáteční úlohy

$$y' = y$$

$$y(0) = 1$$

M-file 2.1.

```

-----
% Eulerova metoda pro ekvidistantní síť uzlů
%
% VSTUP:
% funkce...předpis funkce na pravé straně diferenciální rovnice,
% musí být proměnných x a y, např. ve tvaru funkce = @(x,y) x+y
% PP...počáteční podmínka PP = [x_0 y_0], např. ve tvaru
% PP = [0 1]
% int...interval, na kterém hledáme řešení dané funkce ve tvaru

```

```

% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů,
% pokud chcete zadat krok h, pak zadejte N = 0 a jestliže
% chcete zadat N, potom zvolte h = 0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...vektor hodnot přibližného řešení v bodech x_i

function [x,y] = Euler_ekv(funkce,PP,int,h,N)

h = single(h);
x(1) = PP(1);
y(1) = PP(2);

if PP(1) == int(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),
    if h == 0,
        h = (int(2)-int(1))/N;
    else
        N = (int(2)-int(1))/h;
    end;
if N/fix(N) == 1,
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
    for i = 1:N,

```

```

        y(i+1) = y(i)+h*funkce(x(i),y(i));
    end;
    disp('Vektor uzlů')
    x
    disp('Vektor hodnot přibližného řešení')
    y
    plot(x,y,x(1),y(1),'rx')
    axis([int(1),int(2),min(y),max(y)]);
else
    disp('Rozdíl b-a není celočíselným násobkem kroku h!
    Změňte interval int = [a b] nebo krok h.')
end;
else
    disp('Špatně zadaný krok h nebo číslo N! Změňte krok h
    nebo číslo N.')
end;
else
    disp('Špatně zadaný interval [a b]! Změňte interval
    int = [a b].')
end;
else
    disp('Bod x_0 není shodný s bodem a! Změňte počáteční
    podmínku PP nebo interval int.')
end;

```

Data z příkladu 2.1 z tabulky 1 lze získat také prostřednictvím M-filu 2.1, což si ukážeme v následující poznámce.

Poznámka 2.1. Uvedené vypočtené hodnoty přibližného řešení získáme v programu Matlab vyvoláním M-filu 2.1 následujícím způsobem.

```
funkce = @(x,y) y;
```

```

PP = [0 1];
int = [0 1.4];
h = 0;
N = 7;

[x,y] = Euler_ekv(funkce,PP,int,h,N);

```

Toto zavolání nám kromě vykreslení obrázku příslušného Eulerova polygonu vypíše tyto hodnoty:

Vektor uzlů

```

x =
      0      0.2000      0.4000      0.6000      0.8000      1.0000
  1.2000      1.4000

```

Vektor hodnot přibližného řešení

```

y =
  1.0000      1.2000      1.4400      1.7280      2.0736      2.4883
  2.9860      3.5832

```

Nyní si pomocí M-filu [2.1](#) vypočítejme složitější počáteční úlohu.

Příklad 2.2. Pomocí M-filu [2.1](#) vypočítejte hodnoty přibližného řešení počáteční úlohy

$$y' = \frac{x}{y} + x + \cos y$$

$$y(-1) = 1$$

na intervalu $[-1, 1]$ s krokem $h = 0.1$. Uvažujme ekvidistantní síť uzlů. Volání daného M-filu bude

```

funkce = @(x,y) (x/y)+x+cos(y);

```

```

PP = [-1 1];
int = [-1 1];
h = 0.1;
N = 0;

```

```
[x,y] = Euler_ekv(funkce,PP,int,h,N);
```

Program Matlab vypíše vypočítané hodnoty:

Vektor uzlů

```

x =
-1.0000  -0.9000  -0.8000  -0.7000  -0.6000  -0.5000
-0.4000  -0.3000  -0.2000  -0.1000   0.0000   0.1000
 0.2000   0.3000   0.4000   0.5000   0.6000   0.7000
 0.8000   0.9000   1.0000


```

Vektor hodnot přibližného řešení

```

y =
 1.0000   0.8540   0.7243   0.6088   0.5058   0.4147
 0.3357   0.2709   0.2265   0.2157   0.2570   0.3537
 0.4858   0.6354   0.7931   0.9537   1.1140   1.2720
 1.4264   1.5769   1.7234

```

Na konec příkladu si uvedeme Eulerův polygon příslušný dané počáteční úloze, viz. Obrázek 5. 

Dále řešíme stejnou počáteční úlohu jako v příkladu 2.1 s tím rozdílem, že použijeme druhého způsobu řešení diferenciální rovnice.

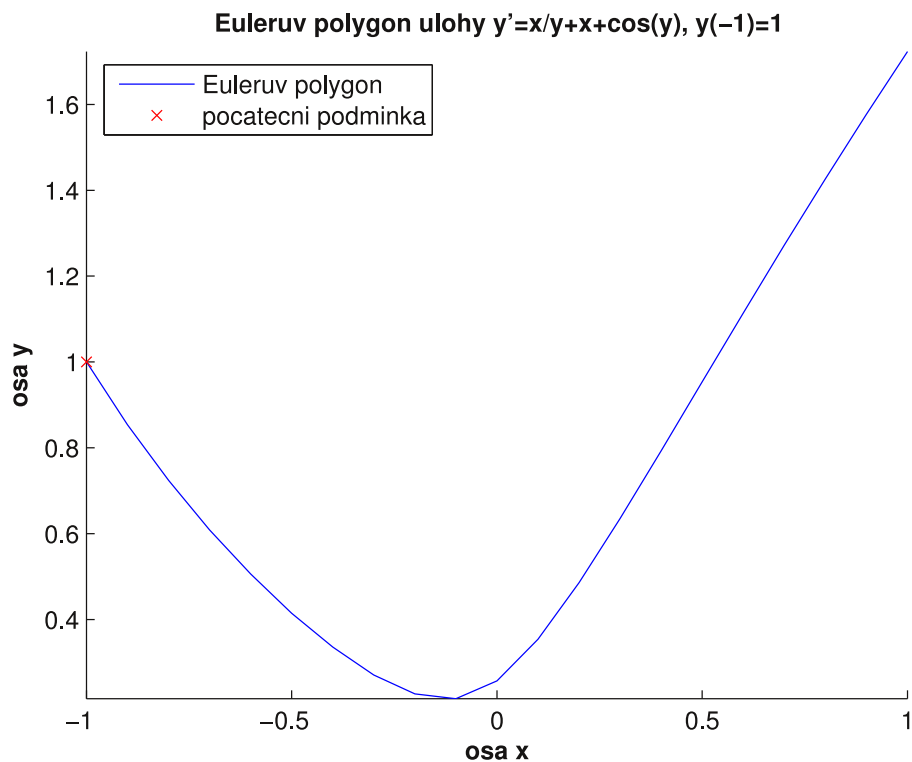
Příklad 2.3. Vypočítejte hodnoty přibližného řešení úlohy

$$y' = y$$

$$y(0) = 1$$

s krokem $\mathbf{h} = (h_0, h_1, h_2, h_3, h_4, h_5, h_6, h_7)^T$ tvaru

$$\mathbf{h} = (0.4, 0.4, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)^T$$



Obrázek 5: Eulerův polygon počáteční úlohy

$$y' = \frac{x}{y} + x + \cos y$$

$$y(-1) = 1$$

a počátečním bodem $a = 0$. Při hledání hodnot přibližného řešení užíjte Eulerovy metody.

Řešení: Počet složek vektoru \mathbf{h} je 8, počet dělicích intervalů N je tedy roven 8. Stanovíme síť $\{x_i\}_{i=0}^8$, kde $x_0 = a$ a $x_{i+1} = x_i + h_i$ pro $i = 0, 1, 2, \dots, 7$. Hledané dělicí body jsou

i	0	1	2	3	4	5	6	7	8
x_i	0.0	0.4	0.8	0.9	1.0	1.1	1.2	1.3	1.4

Eulerova metoda je pro tuto úlohu tvaru

$$y_{i+1} = y_i + h_i y_i = (1 + h_i) y_i,$$

kde $i = 0, 1, 2, \dots, N - 1$. Dále položíme $i = 0$ a vztah (2.2) Eulerovy metody pro danou diferenciální rovnici je tvaru

$$y_1 = y_0 + h_0 y_0 = (1 + h_0) y_0,$$

kde hodnota y_0 je dána počáteční podmínkou. Po dosazení konkrétních hodnot dostaneme

$$y_1 = (1 + 0.4) \cdot 1 = 1.4.$$

Nyní položíme $i = 1$. Eulerova metoda je dána

$$y_2 = (1 + h_1) y_1,$$

Dosazením dostáváme

$$y_2 = (1 + 0.4) \cdot 1.4 = 1.96.$$

Dále bychom pokračovali obdobně až do volby $i = 7$. V důsledku toho, že se jedná o příklad, v němž máme proměnlivý krok h_i , tak si uvedeme tabulku postupných výpočtů jednotlivých hodnot.

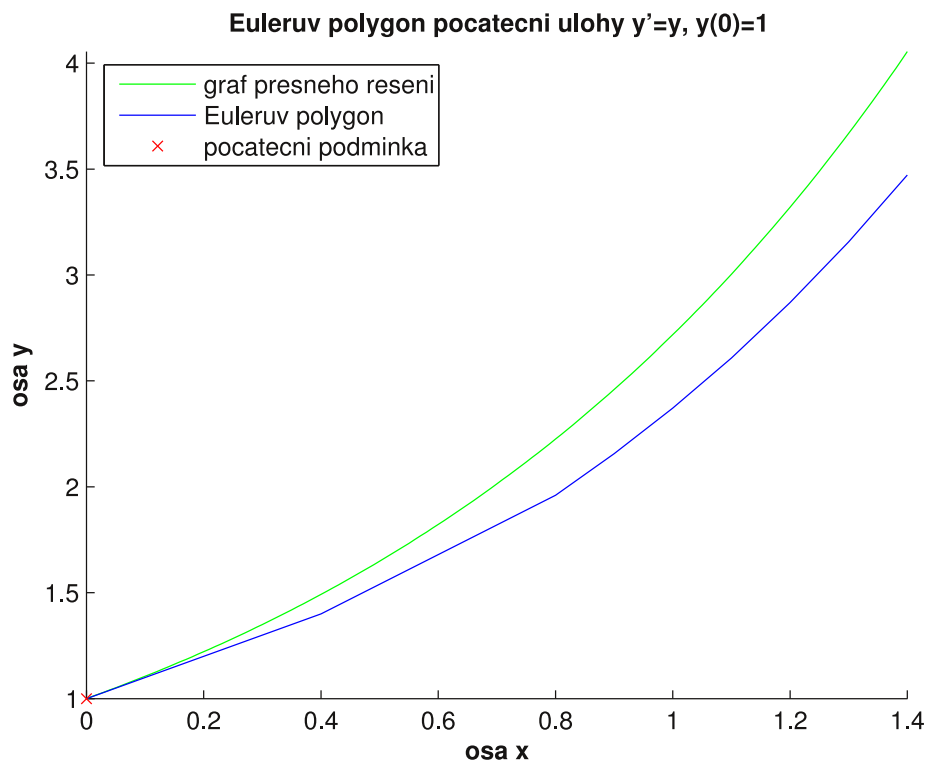
i	h_i	x_i	$y_{i+1} = y_i + h_i y_i$
0	0.4	0.0	$y_1 = y_0 + h_0 y_0 = 1.4000$
1	0.4	0.4	$y_2 = y_1 + h_1 y_1 = 1.9600$
2	0.1	0.8	$y_3 = y_2 + h_2 y_2 = 2.1560$
3	0.1	0.9	$y_4 = y_3 + h_3 y_3 = 2.3716$
4	0.1	1.0	$y_5 = y_4 + h_4 y_4 = 2.6088$
5	0.1	1.1	$y_6 = y_5 + h_5 y_5 = 2.8696$
6	0.1	1.2	$y_7 = y_6 + h_6 y_6 = 3.1566$
7	0.1	1.3	$y_8 = y_7 + h_7 y_7 = 3.4723$

Tabulka 3: Postupné výpočty jednotlivých hodnot přibližného řešení

Na závěr příkladu si opět uvedeme tabulku s vypočtenými hodnotami a příslušný Eulerův polygon s grafem přesného řešení, viz. Obrázek 6.

i	0	1	2	3	4	5	6	7	8
x_i	0.0	0.4	0.8	0.9	1.0	1.1	1.2	1.3	1.4
y_i	1.0000	1.4000	1.9600	2.1560	2.3716	2.6088	2.8696	3.1566	3.4723
$y(x_i)$	1.0000	1.4900	2.2300	2.4600	2.7183	3.0042	3.3201	3.6693	4.0552
e_i	0.0000	0.0900	0.2700	0.3040	0.3467	0.3954	0.4505	0.5127	0.5829

Tabulka 4: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení



Obrázek 6: Eulerův polygon počáteční úlohy

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

Zkusme si vypočítat tu samou úlohu, uvažujme ale následující krok

$$\mathbf{h} = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05)^T.$$

Uvedeme si pouze tabulku vypočtených hodnot a příslušný Eulerův polygon s přesným řešením, viz. Obrázek 7.

i	x_i	y_i	$y(x_i)$	e_i
0	0.00	1.0000	1.0000	0.0000
1	0.10	1.1000	1.1052	0.0052
2	0.20	1.2100	1.2214	0.0114
3	0.30	1.3310	1.3499	0.0189
4	0.40	1.4641	1.4918	0.0277
5	0.50	1.6105	1.6487	0.0382
6	0.60	1.7716	1.8221	0.0505
7	0.70	1.9488	2.0138	0.0650
8	0.80	2.1437	2.2255	0.0818
9	0.90	2.3581	2.4596	0.1015
10	1.00	2.5939	2.7183	0.1244
11	1.10	2.8533	3.0042	0.1509
12	1.15	2.9960	3.1582	0.1622
13	1.20	3.1458	3.3201	0.1743
14	1.25	3.3031	3.4903	0.1872
15	1.30	3.4683	3.6693	0.2010
16	1.35	3.6417	3.8574	0.2157
17	1.40	3.8238	4.0552	0.2314

Tabulka 5: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení



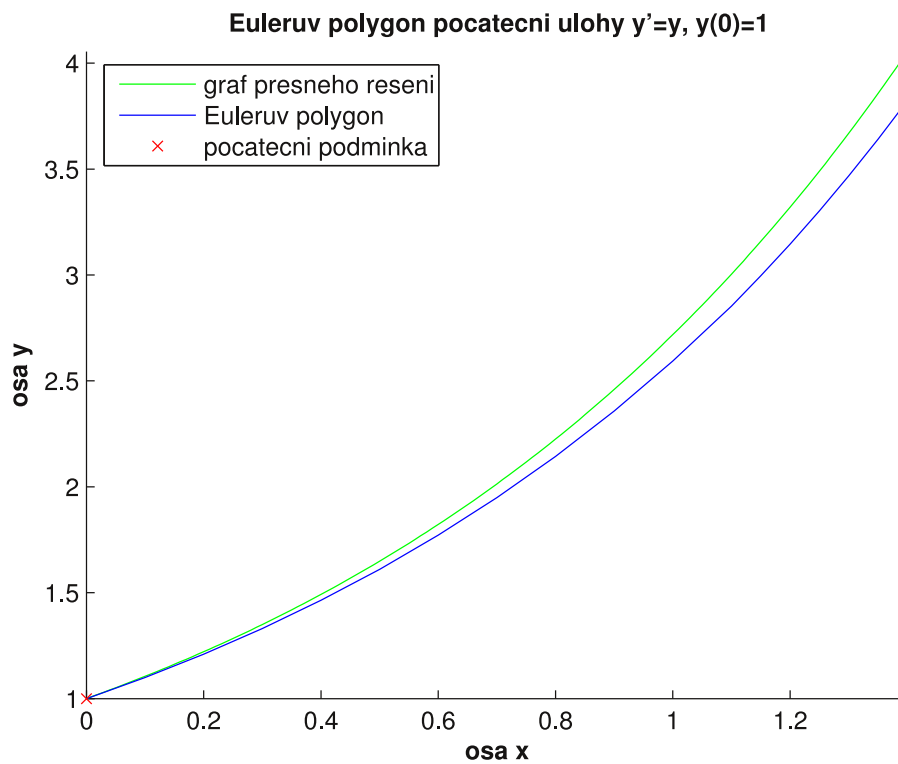
Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file Eulerovy metody pro neekvidistantní síť uzlů, včetně kontroly vstupních dat, který si zde uvedeme.

M-file 2.2.

```

-----
% Eulerova metoda pro neekvidistantní síť uzlů
%
% VSTUP:
% funkce...předpis funkce na pravé straně diferenciální rovnice,
% musí být proměnných x a y, např. ve tvaru funkce = @(x,y) x+y
% PP...počáteční podmínka PP = [x_0 y_0], např. ve tvaru
% PP = [0 1]
% int...bod, ve kterém chceme začít hledat hodnoty přibližného

```



Obrázek 7: Eulerův polygon počáteční úlohy

$$y' = y$$

$$y(0) = 1$$

```
% řešení dané funkce
% h...vektor kroků, např. ve tvaru h = [0.2 0.5 0.2 0.5]
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...vektor hodnot přibližného řešení v bodech x_i
```

```
function [x,y] = Euler_neekv(funkce,PP,int,h)
```

```
N = length(h);
```

```

x(1) = int;
y(1) = PP(2);
a = 0;

for i = 1:N,
    if h(i) <= 0,
        a = i;
    end;
end;
if PP(1) == int,
if a == 0,
    for i = 1:N,
        x(i+1) = x(i)+h(i);
    end;
    for i = 1:N,
        y(i+1) = y(i)+h(i)*funkce(x(i),y(i));
    end;
    disp('Vektor uzlů')
    x
    disp('Vektor hodnot přibližného řešení')
    y
    plot(x,y,x(1),y(1),'rx')
    axis([int,int+sum(h),min(y),max(y)]);
else
    disp('Chybně zadaný vektor kroků h! Změňte některý krok
h(i).')
end;
else
    disp('Bod x_0 není shodný s bodem int! Změňte počáteční
podmínku PP nebo bod int.')

```

```
end;
```

Opět můžeme data, která jsme vypočítali v příkladu 2.3 zkontrolovat pomocí M-filu 2.2, jakým způsobem to provést si ukážeme v následující poznámce.

Poznámka 2.2. Voláním

```
funkce = @(x,y) y;  
PP = [0 1];  
int = 0;  
h = [0.4 0.4 0.1 0.1 0.1 0.1 0.1 0.1];
```

```
[x,y] = Euler_nekv(funkce,PP,int,h);
```

získáme hodnoty

Vektor uzlů

```
x =  
      0      0.4000      0.8000      0.9000      1.0000      1.1000  
  1.2000      1.3000      1.4000
```

Vektor hodnot přibližného řešení

```
y =  
  1.0000      1.4000      1.9600      2.1560      2.3716      2.6088  
  2.8696      3.1566      3.4723,
```

což jsou vypočítané hodnoty přibližného řešení z příkladu 2.3 z Tabulky 4.

Nyní provedeme srovnání výše uvedených postupů hledání hodnot přibližného řešení.

Poznámka 2.3. Při srovnávání dvou výše uvedených postupů hledání hodnot přibližného řešení uvažujeme danou počáteční úlohu tvaru

$$y' = y$$
$$y(0) = 1$$

na intervalu $[0, 1.4]$. Z tohoto důvodu uvedeme tabulku chyb e_i , přičemž označíme e_i^{11} chybu v příkladu 2.1, dále e_i^{12} chybu uvedenou ve druhé části tohoto příkladu a obdobně e_i^{21} a e_i^{22} chyby v příkladu 2.3.

i	x_i	e_i^{11}	e_i^{12}	e_i^{21}	e_i^{22}
0	0.00	0.0000	0.0000	0.0000	0.0000
1	0.10		0.0052		0.0052
2	0.20	0.0200	0.0114		0.0114
3	0.30		0.0189		0.0189
4	0.40	0.0500	0.0277	0.0900	0.0277
5	0.50		0.0382		0.0382
6	0.60	0.0940	0.0505		0.0505
7	0.70		0.0650		0.0650
8	0.80	0.1519	0.0818	0.2700	0.0818
9	0.90		0.1015	0.3040	0.1015
10	1.00	0.2300	0.1244	0.3467	0.1244
11	1.10		0.1509	0.3954	0.1509
12	1.15				0.1622
13	1.20	0.3341	0.1815	0.4505	0.1743
14	1.25				0.1872
15	1.30		0.2168	0.5127	0.2010
16	1.35				0.2157
17	1.40	0.4720	0.2574	0.5829	0.2314

Tabulka 6: Chyby z příkladu 2.1 a 2.3

Nejprve budeme srovnávat chyby v rámci jednoho příkladu. Z uvedené tabulky vidíme, že čím menší krok h zvolíme, tím menší chybu dostaneme. Ovšem tato výhoda je pouze zdánlivá, neboť čím menší krok budeme volit, tím více hodnot přibližného řešení budeme počítat. Výpočet bude tedy časově náročnější i pokud budeme hodnoty zjišťovat pomocí počítače. Například si vezměme chyby e_{17}^{11} a e_{17}^{12} . V prvním případě jsme dostali chybu $e_{17}^{11} = 0.4720$ s krokem $h = 0.2$ a ve druhém chybu $e_{17}^{11} = 0.2574$ s krokem $h = 0.1$. Mohli bychom říci, že při rozdělení kroku napůl bude chyba poloviční, ovšem dostaneme také dvojnásobný počet dělicích bodů, tudíž by bylo třeba provést dvojnásobný počet výpočtů.

Dále se budeme zajímat o chyby mezi příklady. Proč je vhodné použít neekvidistantní sítě dělicích bodů? Pokud budeme uvažovat ekvidistantní síť bodů, viz. příklad 2.1, bude se kumulovat (hromadit) chyba e_i s rostoucím rozdílem

$e_{i+1} - e_i$. Pro ilustraci si vezměme chybu e_i^{11} a vypočtěme rozdíly chyb $e_2^{11} - e_0^{11} = 0.02$, $e_4^{11} - e_2^{11} = 0.03$, $e_6^{11} - e_4^{11} = 0.044$, atd. Jak vidíme chyba skutečně narůstá s rostoucím rozdílem.

Kdežto pokud budeme uvažovat neekvidistantní síť bodů, vezmeme si například chybu e_i^{21} z příkladu 2.3 a uděláme rozdíly mezi dvěma po sobě jdoucími chybami, tj. $e_4^{21} - e_0^{21} = 0.09$, $e_8^{21} - e_4^{21} = 0.18$, $e_9^{21} - e_8^{21} = 0.034$, $e_{10}^{21} - e_9^{21} = 0.0427$, atd. Vidíme, že v případě neekvidistantní sítě se nám chyba nekumuluje s rostoucím rozdílem $e_{i+1} - e_i$. Jestliže užijeme k hledání hodnot přibližného řešení neekvidistantní síť uzlů můžeme tím redukovat chybu e_i .

Pro názornost si vykreslíme Obrázek 8, na kterém budou všechny Eulerovy polygony příslušné úloze

$$\begin{aligned} y' &= y \\ y(0) &= 1, \end{aligned}$$

včetně přesného řešení $y = e^x$.

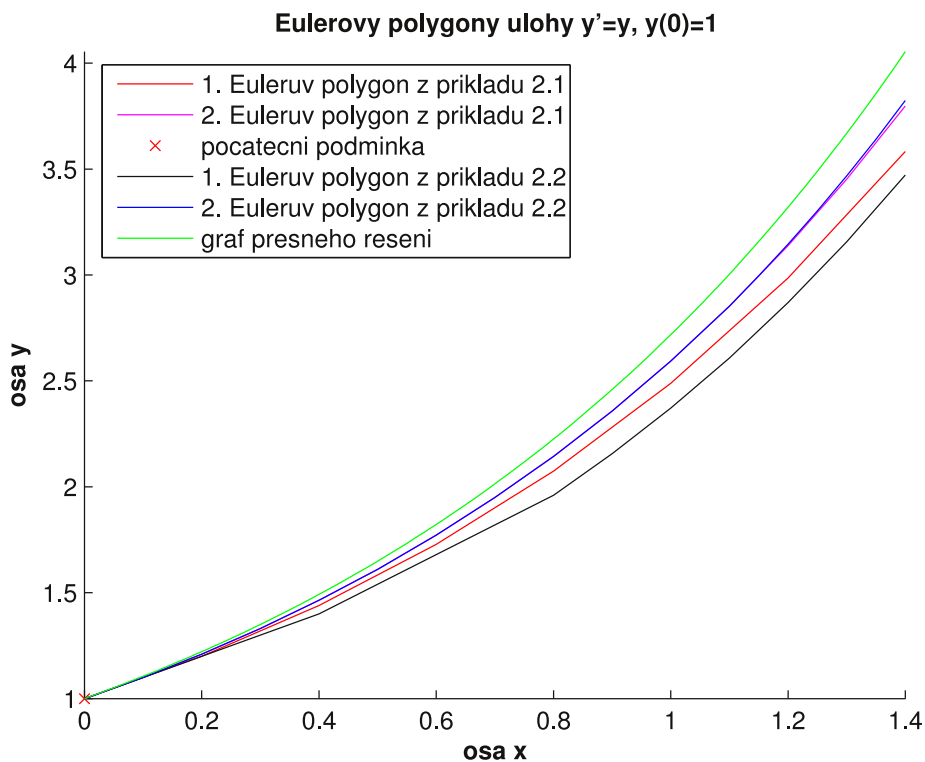
Nyní můžeme přejít k vektorovému zápisu Eulerovy metody. Vztah (2.2) pro úlohu

$$\left. \begin{aligned} \mathbf{y}' &= \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(x_0) &= \mathbf{y}_0, \end{aligned} \right\} \quad (2.4)$$

kde $x_0 \in \mathbb{R}$ a $\mathbf{y}_0 \in \mathbb{R}^n$, je tvaru

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \mathbf{f}(x_i, \mathbf{y}_i), \quad (2.5)$$

kde $i = 0, 1, 2, \dots, N-1$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$ a $\mathbf{f} = (f_1, f_2, \dots, f_n)^T$. Při hledání přibližného řešení úlohy (2.4) Eulerovou metodou postupujeme stejně jako v případě řešení úlohy (2.1) pomocí této metody s tím rozdílem, že namísto hodnot $y_i \in \mathbb{R}$ a hodnoty funkce $f(x_i, y_i)$ počítáme s vektory \mathbf{y}_i a vektorovou funkcí \mathbf{f} . Máme tedy dvě možnosti zadání příkladu pro hledání hodnot přibližného řešení počáteční úlohy (2.4).



Obrázek 8: Eulerovy polygony počáteční úlohy

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

Jelikož umíme každou diferenciální rovnici n -tého řádu v normálním tvaru vyjádřenou vztahem

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$$

s počátečními podmínkami

$$y^{(j)}(x_0) = \psi_j, \text{ pro } j = 0, 1, 2, \dots, n - 1,$$

převést na systém n diferenciálních rovnic prvního řádu, viz. poznámka 1.3, tj.

na systém tvaru

$$\begin{aligned}y_1' &= y_2, \\y_2' &= y_3, \\&\vdots \\y_{n-1}' &= y_n, \\y_n' &= f(x, y_1, y_2, \dots, y_n),\end{aligned}$$

jenž je speciálním případem systému

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}),$$

resp. počáteční úlohy (2.4), proto zde uvádíme vektorovou podobu Eulerovy metody tvaru (2.5).

Dále se také budeme zabývat hledáním hodnot přibližného řešení \mathbf{y}_i tohoto systému diferenciálních rovnic a proto se nám bude lépe pracovat s vektory a vektorovým zápisem této metody. Z tohoto důvodu je potřebné zavést označení složek vektoru $\mathbf{y}_i \in \mathbb{R}^n$, tj. $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{in})^T$, kde $i = 0, 1, 2, \dots, N$.

Nyní si uvedeme příklad na použití Eulerovy metody pro úlohu tvaru (2.4), jenž představuje fyzikální aplikaci vedoucí na soustavu diferenciálních rovnic, kterou budeme uvažovat i dále v této práci.

Mějme těleso o hmotnosti 1 kg zavěšené na volném konci pružiny, která je pevně uchycená, a $z(x)$ je poloha tělesa v čase x . Pak síla, jež způsobuje kmitání tělesa je

$$g - [z(x) - l],$$

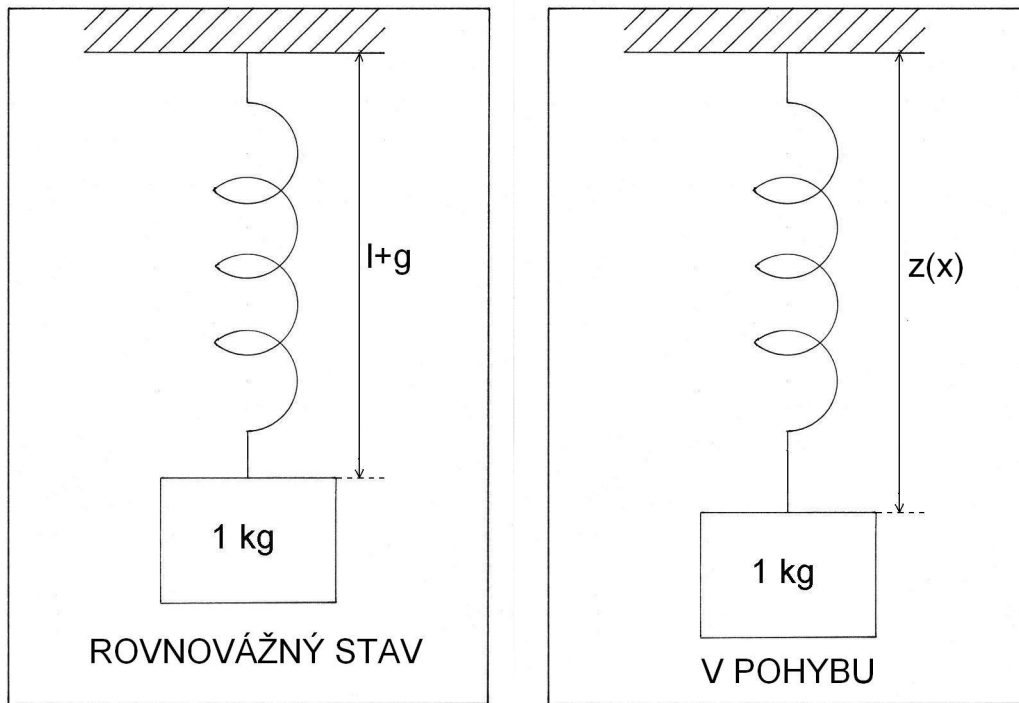
kde g je gravitační zrychlení a l je původní délka pružiny. Nechť je dále dána síla úměrná rychlosti tělesa $z'(x)$ působící proti pohybu. Pak má rovnice pohybu vyjádření

$$z''(x) = g - [z(x) - l] - z'(x),$$

pokud pro usnadnění uijeme substituce $y = z - (l + g)$ dostáváme diferenciální rovnici druhého řádu tvaru

$$y'' + y' + y = 0$$

s počátečními podmínkami $y(x_0) = \psi_0$ a $y'(x_0) = \psi_1$. Připojíme obrázek 9 ilustrující právě uvedené veličiny.



Obrázek 9: Pružina

Nyní si vypočítejme hodnoty přibližného řešení této diferenciální rovnice druhého řádu pomocí Eulerovy metody.

Příklad 2.4. Najděte hodnoty přibližného řešení diferenciální rovnice druhého řádu tvaru

$$y'' = -y' - y$$

s počátečními podmínkami $y(0) = 1, y'(0) = 1$. Tuto úlohu řešme na intervalu $[0, 1.4]$ s krokem $h = 0.2$, uvažujme ekvidistantní síť uzlů.

Řešení: Zadanou počáteční úlohu si nejprve převedeme na soustavu diferenciálních rovnic prvního řádu, ta je tvaru

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= -y_2 - y_1 \end{aligned}$$

s počátečními podmínkami $y_1(0) = 1, y_2(0) = 1$. Dále označme

$$\mathbf{f}(x, \mathbf{y}) = (y_2, f(x, y_1, y_2))^T = (y_2, -y_2 - y_1)^T$$

s počátečními podmínkami

$$\mathbf{y}(0) = (1, 1)^T.$$

Nyní již samotné řešení. Určíme počet dělicích intervalů N ze vztahu

$$N = \frac{b - a}{h},$$

po dosažení konkrétních hodnot dostaneme $N = 7$. Vzorec Eulerovy metody je tvaru

$$\mathbf{y}_{i+1} = \mathbf{y}_i + h\mathbf{f}(x_i, \mathbf{y}_i),$$

která je pro tuto úlohu dána vztahem

$$\mathbf{y}_{i+1} = \begin{pmatrix} y_{i+1,1} \\ y_{i+1,2} \end{pmatrix} = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} + h \begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix}$$

Dále budeme pokračovat prvním krokem a zvolíme $i = 0$, platí tedy

$$\mathbf{y}_1 = \mathbf{y}_0 + h\mathbf{f}(x_0, \mathbf{y}_0),$$

kde $\mathbf{y}_0 = (1, 1)^T$ a $\mathbf{f}(x_0, \mathbf{y}_0) = (y_{02}, -y_{02} - y_{01})^T$. Dosaďme do vzorce Eulerovy metody vektor \mathbf{y}_0 a $\mathbf{f}(x_0, \mathbf{y}_0)$ takto

$$\begin{aligned} \mathbf{y}_1 &= \begin{pmatrix} y_{01} \\ y_{02} \end{pmatrix} + h \begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix} = \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.2 \begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} = \begin{pmatrix} 1.2 \\ 0.6 \end{pmatrix} \end{aligned}$$

Tímto výpočtem jsme skončili s prvním krokem. Nyní položíme $i = 1$, hodnota uzlu x_1 dané sítě je 0.2 a pokračujeme výpočtem hodnot \mathbf{y}_2 , tedy

$$\begin{aligned} \mathbf{y}_2 &= \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix} + h \begin{pmatrix} y_{12} \\ -y_{12} - y_{11} \end{pmatrix} = \\ &= \begin{pmatrix} 1.2 \\ 0.6 \end{pmatrix} + 0.2 \begin{pmatrix} 0.6 \\ -0.6 - 1.2 \end{pmatrix} = \begin{pmatrix} 1.32 \\ 0.24 \end{pmatrix} \end{aligned}$$

Obdobně bychom postupovali dále až do $i = 6$.

Pro přehlednost si uvedeme tabulku získaných hodnot přibližného řešení, přesného řešení tvaru

$$y = \sqrt{3}e^{-\frac{x}{2}} \sin\left(\frac{\sqrt{3}}{2}x\right) + e^{-\frac{x}{2}} \cos\left(\frac{\sqrt{3}}{2}x\right),$$

a chyby e_i^k , pro něž platí

$$e_i^k = |y^{(k-1)}(x_i) - y_{ik}|,$$

kde $i = 0, 1, 2, \dots, N$ a $k = 1, 2$, a odpovídající Eulerův polygon, viz. Obrázek 10, kde je první složka řešení y_{i1} na prvním obrázku a druhá y_{i2} na druhém obrázku pro $i = 0, 1, \dots, 7$.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_{i1}	1.0000	1.2000	1.3200	1.3680	1.3536	1.2874	1.1802	1.0430
y_{i2}	1.0000	0.6000	0.2400	-0.0720	-0.3312	-0.5357	-0.6860	-0.7849
$y(x_i)$	1.0000	1.1614	1.2516	1.2802	1.2573	1.1932	1.0976	0.9796
$y'(x_i)$	1.0000	0.6212	0.2886	0.0059	-0.2258	-0.4073	-0.5412	-0.6313
e_i^1	0.0000	0.0386	0.0684	0.0878	0.0963	0.0942	0.0826	0.0634
e_i^2	0.0000	0.0212	0.0486	0.0779	0.1054	0.1284	0.1448	0.1536

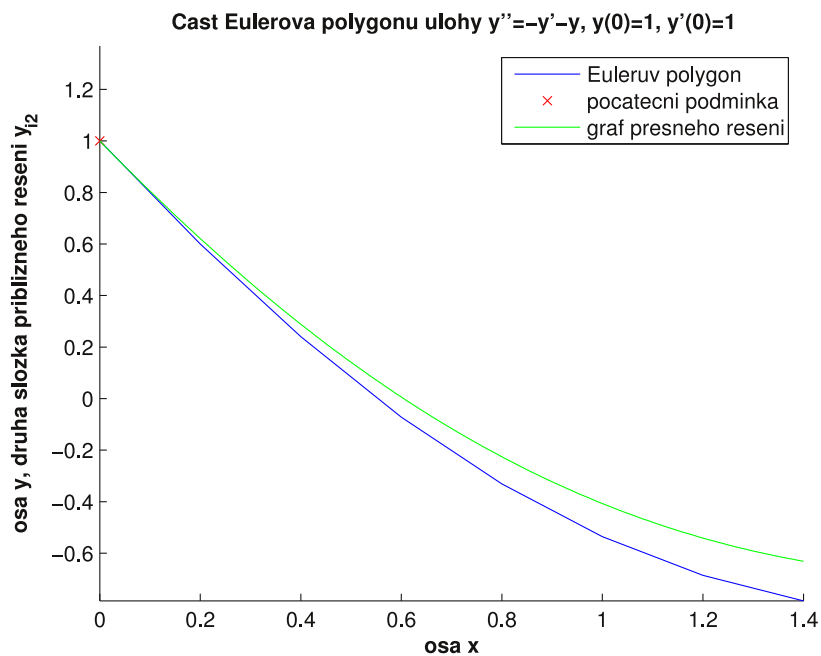
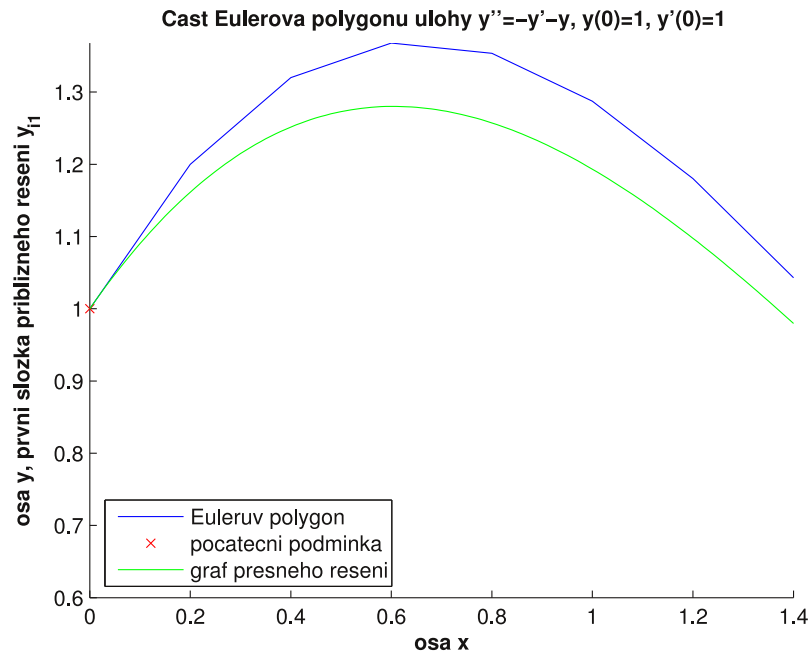
Tabulka 7: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení



Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení soustavy diferenciálních rovnic prvního řádu pomocí Eulerovy metody s ekvidistantní sítí uzlů s kontrolou vstupních dat, který si zde uvedeme.

M-file 2.3.

```
-----
% Eulerova metoda pro soustavu n diferenciálních rovnic prvního
% řádu a ekvidistantní sítí uzlů
%
```



Obrázek 10: Eulerův polygon příslušný počáteční úloze

$$y'' = -y' - y$$

s počátečními podmínkami $y'(0) = 1$, $y(0) = 1$

```

% VSTUP:
% Y...vektor funkcí pravých stran diferenciálních rovnic ve tvaru
% Y = @(x,y)([y(2);y(3); ...; f(x,y(1),y(2),...,y(N))]),
% tedy např.
% Y = @(x,y)([y(2);sin(x^2)-y(2)-cos(2*y(1))])
% y...počáteční podmínky y = [y(0) y(1) ... y(N)], např. ve tvaru
% y = [0 1 ... 3] nebo y = [0;1; ...; 3]
% x0...bod x_0 z počátečních podmínek
% int...interval, na kterém hledáme řešení dané funkce ve tvaru
% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů,
% pokud chcete zadat krok h, pak zadejte N=0 a jestliže chcete
% zadat N, potom zvolte h=0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...matice hodnot přibližného řešení v bodech x_i, v níž první
% řádek je první složkou y_i1 přibližného řešení, y_i2 druhou
% složkou přibližného řešení, atd.

function [x,y] = Euler_ekv_sous(Y,y,x0,int,h,N)

h = single(h);
x(1) = x0;

if x0 == int(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),

```



```

if h == 0,
    h = (int(2)-int(1))/N;
else
    N = (int(2)-int(1))/h;
end;
if N/fix(N) == 1,
    Q1 = length(y);
    for i = 1:Q1,
        y(i,1) = y(i);
    end;
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
    for i = 1:N,
        y(:,i+1) = y(:,i)+h*Y(x(i),y(:,i));
    end;
    k = y(1);
    kk = y(1);
    for i = 1:N+1,
        l = min(y(:,i));
        ll = max(y(:,i));
        if k > l,
            k = l;
        end;
        if kk < ll,
            kk = ll;
        end;
    end;
    disp('Vektor uzlů')
x

```

```

disp('Matice hodnot přibližného řešení')
y
hold on
for i = 1:Q1,
    plot(x,y(i,:),x(1),y(1,1),'rx')
end;
hold off
axis([int(1),int(2),k,kk]);
else
disp('Rozdíl b-a není celočíselným násobkem kroku h!
Změňte interval int = [a b] nebo krok h.')
end;
else
disp('Špatně zadaný krok h nebo číslo N! Změňte krok h nebo
číslo N.')
end;
else
disp('Špatně zadaný interval [a b]! Změňte interval int =
[a b].')
end;
else
disp('Bod x_0 není shodný s bodem a! Změňte počáteční bod x0
nebo interval int.')
end;

```

Data, která jsme vypočítali v příkladu 2.4 můžeme zkontrolovat pomocí M-flu 2.3, jakým způsobem to provést si ukážeme v následující poznámce.

Poznámka 2.4. Voláním

```

Y = @(x,y) ([y(2); -y(1)-y(2)]);
y = [1 1];

```

```
x0 = 0;
int = [0 1.4];
h = 0.2;
N = 0;
```

```
[x,y] = Euler_ekv_sous(Y,y,x0,int,h,N);
```

můžeme ověřit vypočítané hodnoty přibližného řešení z příkladu 2.4.

Dále budeme uvažovat složitější diferenciální rovnici vyššího řádu, přičemž hodnoty přibližného řešení najdeme pomocí M-filu 2.3.

Příklad 2.5. Pomocí M-filu 2.3 najděte hodnoty přibližného řešení diferenciální rovnice

$$y^{(IV)} = x^4 + \sin y + 2y' + y'' + y'''$$

s počátečními podmínkami $y(-1) = 1$, $y'(-1) = 1$, $y''(-1) = 1$ a $y'''(-1) = 1$ na intervalu $[-1, 1]$ s krokem $h = 0.5$. Uvažujeme ekvidistantní síť uzlů. Známým postupem převedeme danou diferenciální rovnici na soustavu diferenciálních rovnic prvního řádu tvaru

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= y_4 \\ y_4' &= x^4 + \sin y_1 + 2y_2 + y_3 + y_4 \end{aligned}$$

s počátečními podmínkami $y_1(-1) = 1$, $y_2(-1) = 1$, $y_3(-1) = 1$ a $y_4(-1) = 1$.

Voláním

```
Y = @(x,y) ([y(2);y(3);y(4);x^4+sin(y(1))+2*y(2)+y(3)+y(4)]);
y = [1 1 1 1];
x0 = -1;
int = [-1 1];
h = 0.5;
```

`N = 0;`

`[x,y] = Euler_ekv_sous(Y,y,x0,int,h,N);`

získáme tyto hodnoty přibližného řešení

Vektor uzlů


`x =`

`-1.0000 -0.5000 0 0.5000 1.0000`

Matice hodnot přibližného řešení

`y =`

```
1.0000  1.5000  2.2500  3.3750  5.3651
1.0000  1.5000  2.2500  3.9802  7.8756
1.0000  1.5000  3.4604  7.7909 16.4714
1.0000  3.9207  8.6611 17.3609 33.8326
```

Na závěr příkladu si uvedeme příslušné Eulerovy polygony uvedené počáteční úlohy, viz. Obrázek 11. 

Ještě si vyřešíme jeden příklad na nalezení hodnot přibližného řešení y_i úlohy (2.4). Tentokrát budeme předpokládat neekvidistantní síť uzlů.

Příklad 2.6. Najděte hodnoty přibližného řešení diferenciální rovnice druhého řádu tvaru

$$y'' = -y' - y$$

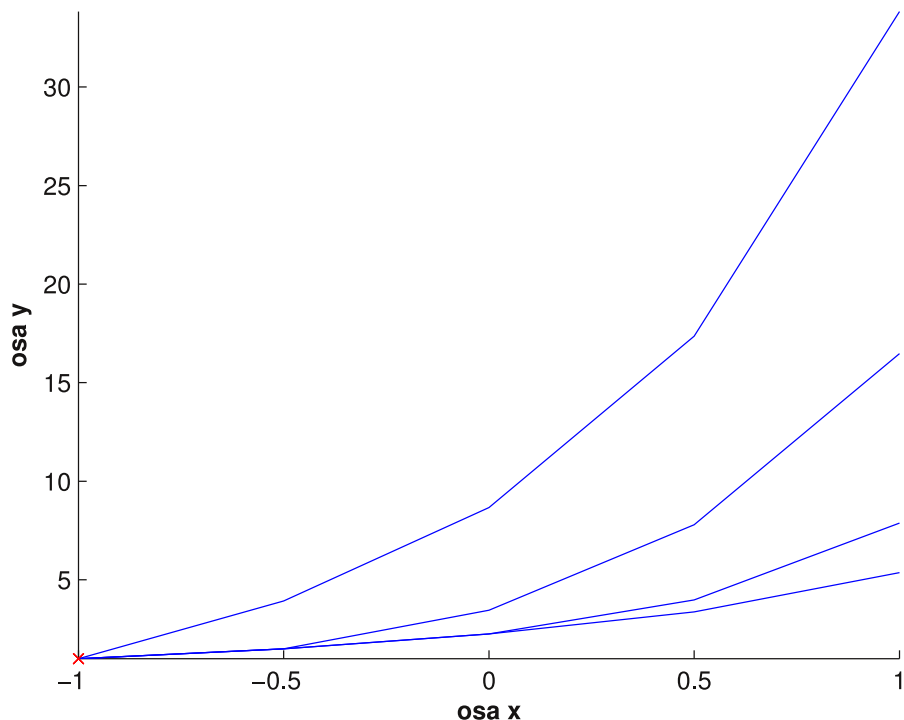
s počátečními podmínkami $y(0) = 1, y'(0) = 1$. Počáteční bod je $a = 0$ a vektor \mathbf{h} je dán takto

$$\mathbf{h} = (0.4, 0.4, 0.3, 0.15, 0.15)^T.$$

Řešení: Počet dělicích intervalů N je 5, což je počet složek vektoru \mathbf{h} . Nyní určíme síť $\{x_i\}_{i=0}^5$, kde $x_0 = a$ a $x_{i+1} = x_i + h_i$ pro $i = 0, 1, 2, \dots, 4$. Hledané dělicí body jsou

i	0	1	2	3	4	5
x_i	0.00	0.40	0.80	1.10	1.25	1.40

Eulerovy polygony príslušne úloze z príkladu 2.5



Obrázek 11: Eulerovy polygony príslušné počáteční úloze

$$y^{(IV)} = x^4 + \sin y + 2y' + y'' + y'''$$

s počátečními podmínkami $y(-1) = 1, y'(-1) = 1, y''(-1) = 1, y'''(-1) = 1$

Eulerova metoda je pro zadanou úlohu dána vyjádřením

$$\mathbf{y}_{i+1} = \begin{pmatrix} y_{i+1,1} \\ y_{i+1,2} \end{pmatrix} = \begin{pmatrix} y_{i1} \\ y_{i2} \end{pmatrix} + h \begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix}$$

Dále v tomto vztahu položíme $i = 0$ a dostaneme

$$\mathbf{y}_1 = \mathbf{y}_0 + h_0 \mathbf{f}(x_0, \mathbf{y}_0),$$

kde $x_0 = a$ a $h_0 = 0.4$. Po dosazení vektoru \mathbf{y}_0 a $\mathbf{f}(x_0, \mathbf{y}_0)$ dostaneme

$$\begin{aligned} \mathbf{y}_1 &= \begin{pmatrix} y_{01} \\ y_{02} \end{pmatrix} + h_0 \begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix} = \\ &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.4 \begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} = \begin{pmatrix} 1.4 \\ 0.2 \end{pmatrix} \end{aligned}$$

Nyní položíme $i = 1$. Eulerova metoda je

$$\mathbf{y}_2 = \mathbf{y}_1 + h_1 \mathbf{f}(x_1, \mathbf{y}_1),$$

kde $x_1 = 0.4$ a $h_1 = 0.4$. Dosazením dostáváme

$$\begin{aligned} \mathbf{y}_2 &= \begin{pmatrix} y_{11} \\ y_{12} \end{pmatrix} + h_1 \begin{pmatrix} y_{12} \\ -y_{12} - y_{11} \end{pmatrix} = \\ &= \begin{pmatrix} 1.4 \\ 0.2 \end{pmatrix} + 0.4 \begin{pmatrix} 0.2 \\ -0.2 - 1.4 \end{pmatrix} = \begin{pmatrix} 1.48 \\ -0.44 \end{pmatrix} \end{aligned}$$

Dále bychom pokračovali obdobně až do volby $i = 4$. V důsledku toho, že se jedná o příklad, v němž máme proměnlivý krok h_i , tak si uvedeme tabulku postupných výpočtu jednotlivých hodnot.

i	h_i	x_i	$\mathbf{y}_{i+1} = \mathbf{y}_i + h_i \mathbf{f}(x_i, \mathbf{y}_i)$
0	0.40	0.00	$\mathbf{y}_1 = \mathbf{y}_0 + h_0 \mathbf{f}(x_0, \mathbf{y}_0) = (1.4000, 0.2000)^T$
1	0.40	0.40	$\mathbf{y}_2 = \mathbf{y}_1 + h_1 \mathbf{f}(x_1, \mathbf{y}_1) = (1.4800, -0.4400)^T$
2	0.30	0.80	$\mathbf{y}_3 = \mathbf{y}_2 + h_2 \mathbf{f}(x_2, \mathbf{y}_2) = (1.3480, -0.7520)^T$
3	0.15	1.10	$\mathbf{y}_4 = \mathbf{y}_3 + h_3 \mathbf{f}(x_3, \mathbf{y}_3) = (1.2352, -0.8414)^T$
4	0.15	1.25	$\mathbf{y}_5 = \mathbf{y}_4 + h_4 \mathbf{f}(x_4, \mathbf{y}_4) = (1.1090, -0.9005)^T$

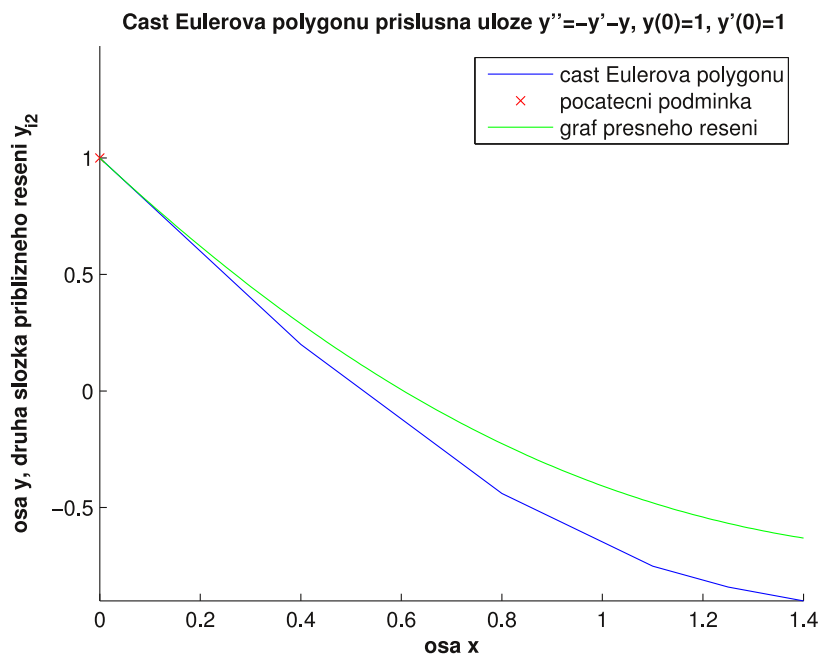
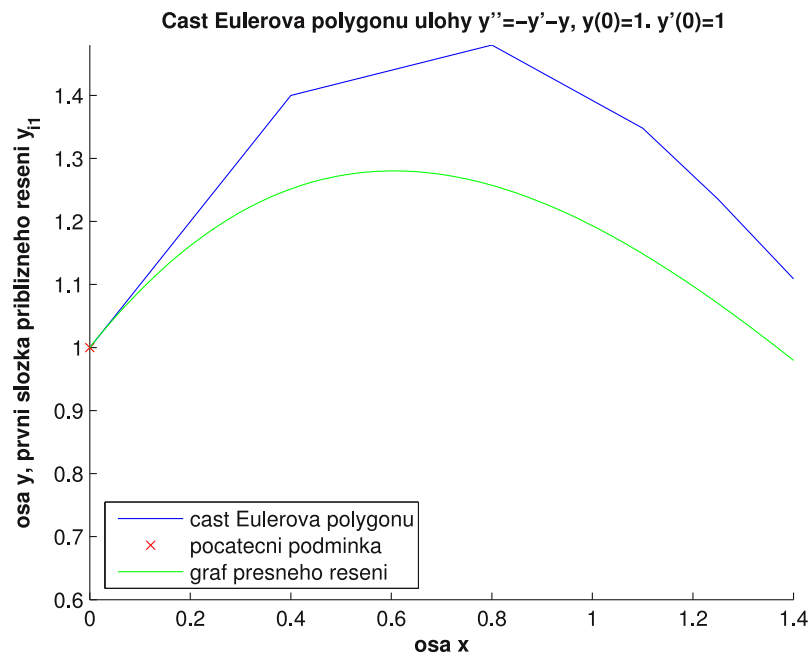
Na závěr příkladu si opět uvedeme tabulku s výsledky, včetně hodnot přesného řešení, a příslušný Eulerův polygon, viz. Obrázek 12, kde je první složka řešení y_{i1} na prvním obrázku a druhá y_{i2} na druhém obrázku pro $i = 0, 1, \dots, 5$.

i	0	1	2	3	4	5
x_i	0.00	0.40	0.80	1.10	1.25	1.40
y_{i1}	1.0000	1.4000	1.4800	1.3480	1.2352	1.1090
y_{i2}	1.0000	0.2000	-0.4400	-0.7520	-0.8414	-0.9005
$y(x_i)$	1.0000	1.2516	1.2573	1.1487	1.0699	0.9796
$y'(x_i)$	1.0000	0.2886	-0.2258	-0.4800	-0.5677	-0.6313
e_i^1	0.0000	0.1484	0.2227	0.1993	0.1653	0.1294
e_i^2	0.0000	0.0886	0.2142	0.2720	0.2737	0.2692

Tabulka 8: Vypočtené hodnoty přibližného a přesného řešení



Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení soustavy diferenciálních rovnic prvního řádu pomocí Eulerovy metody s neekvidistantní sítí uzlů, včetně kontroly vstupních dat, který si zde uvedeme.



Obrázek 12: Eulerovy polygony příslušné počáteční úloze

$$y'' = -y' - y$$

s počátečními podmínkami $y'(0) = 1$, $y(0) = 1$

M-file 2.4.

```
% Eulerova metoda pro soustavu n diferencíálních rovnic prvního
% řádu a neekvidistantní síť uzlů
%
% VSTUP:
% Y...vektor funkcí pravých stran diferencíálních rovnic ve tvaru
% Y = @(x,y)([y(2);y(3); ...; f(x,y(1),y(2),...,y(N))]),
% tedy např.
% Y = @(x,y)([y(2);sin(x^2)-y(2)-cos(2*y(1))])
% y...počáteční podmínky y = [y_0 y_1 ... y_N], např. ve tvaru
% y = [0 1 ... 3] nebo y = [0;1; ...; 3]
% x0...bod x_0, ve kterém chceme začít hledat hodnoty přibližného
% řešení dané funkce
% h...vektor kroků, např. ve tvaru h = [0.2 0.5 0.2 0.5]
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...matice hodnot přibližného řešení v bodech x_i, v níž první
% řádek je první složkou y_i1 přibližného řešení, y_i2 druhou
% složkou přibližného řešení, atd.

function [x,y] = Euler_neekv_sous(Y,y,x0,h)

N = length(h);
x(1) = x0;
a = 0;

for i = 1:N,
    if h(i) <= 0,
```



```

        a = i;
    end;
end;
if a == 0,
    Q1 = length(y);
    for i = 1:Q1,
        y(i,1) = y(i);
    end;
    for i = 1:N,
        x(i+1) = x(i)+h(i);
    end;
    for i = 1:N,
        y(:,i+1) = y(:,i)+h(i)*Y(x(i),y(:,i));
    end;
    k = y(1);
    kk = y(1);
    for i = 1:N+1,
        l = min(y(:,i));
        ll = max(y(:,i));
        if k > l,
            k = l;
        end;
        if kk < ll,
            kk = ll;
        end;
    end;
end;
disp('Vektor uzlů')
x
disp('Matice hodnot přibližného řešení')
y

```

```

hold on
for i = 1:Q1,
    plot(x,y(i,:),x(1),y(1,1),'rx')
end;
hold off
axis([x0,x0+sum(h),k,kk]);
else
    disp('Chybně zadaný vektor kroků h! Změňte některý krok
    h(i).')
end;

```

Data, která jsme vypočítali v příkladu 2.6 můžeme zkontrolovat prostřednictvím M-filu 2.4, jakým způsobem to provést si ukážeme v následující poznámce.

Poznámka 2.5. Voláním

```

Y = @(x,y) ([y(2);-y(1)-y(2)]);
y = [1 1];
x0 = 0;
h = [0.4 0.4 0.3 0.15 0.15];

[x,y] = Euler_neekv_sous(Y,y,x0,h);

```

můžeme ověřit vypočítané hodnoty přibližného řešení z příkladu 2.6.

3 Metody Runge-Kutta

V úvodu této kapitoly se budeme nejdříve, stejně jako v předchozí kapitole, zabývat hledáním hodnot přibližného řešení y_i pro počáteční úlohu tvaru

$$\left. \begin{aligned} y' &= f(x, y) \\ y(x_0) &= y_0, \end{aligned} \right\} \quad (3.1)$$

kde $x_0 \in J, J \subseteq \mathbb{R}$, a $y_0 \in \mathbb{R}$ prostřednictvím metod Runge-Kutta. Potom přejdeme k vektorovému zápisu úlohy (3.1), budeme tedy hledat hodnoty přibližného řešení \mathbf{y}_i pro soustavu diferenciálních rovnic prvního řádu, kterou můžeme s počátečními podmínkami vyjádřit vztahem

$$\left. \begin{aligned} \mathbf{y}' &= \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(x_0) &= \mathbf{y}_0, \end{aligned} \right\} \quad (3.2)$$

kde $x_0 \in J$ a $\mathbf{y}_0 \in \mathbb{R}^n$.

Metody Runge-Kutta spočívají v tom, že průběh funkce $f(x, y)$ aproximujeme na intervalu $[x_i, x_{i+1}]$ parabolou, resp. obloukem paraboly, která má osu rovnoběžnou s osou y , podrobněji je tato problematika rozebrána v knize J. Legrase [3].

Numerické metody pro řešení diferenciálních rovnic rozlišujeme na *explicitní* a *implicitní*.

Explicitní metodou přitom rozumíme takovou numerickou metodu, v jejímž předpise se přibližná hodnota y_{i+1} vyskytuje pouze na levé straně vzorce této metody. Příkladem explicitní numerické metody je Eulerova metoda, kterou jsme si uvedli v kapitole 2 a jež je tvaru

$$y_{i+1} = y_i + h_i f(x_i, y_i). \quad (3.3)$$

Implicitní metodou nazýváme takovou numerickou metodu, kde se hledaná hodnota přibližného řešení y_{i+1} objevuje na levé i pravé straně vzorce dané metody. Hledání hodnot přibližného řešení y_{i+1} je v případě implicitních metod složitější, neboť než uijeme vzorce implicitní metody k výpočtu, musíme nejprve nalézt vyjádření hodnoty y_{i+1} . Můžeme uvažovat i implicitní Eulerovu metodu, jejíž vzorec je

$$y_{i+1} = y_i + h_i f(x_{i+1}, y_{i+1}).$$

Je na místě uvést, že metody Runge-Kutta patří k jednokrokovým metodám. Dále budeme předpokládat ekvidistantní síť bodů, tedy krok h , to z důvodu zjednodušení zápisu těchto numerických metod, přičemž přechod k neekvidistantním případům není složitý, stačí položit $h = h_i$.

3.1 Explicitní metody Runge-Kutta

Základní princip explicitních metod Runge-Kutta je ve vyjádření hodnoty přibližného řešení y_{i+1} vztahem

$$y_{i+1} = y_i + \sum_{l=1}^m w_l k_l, \quad (3.4)$$

kde $m \in \mathbb{N}$, $w_l \in \mathbb{R}$ a $k_l = hf(x_i + \alpha_l h, y_i + \sum_{j=1}^{l-1} \beta_{lj} k_j)$ pro $\forall l = 1, 2, \dots, m$, přičemž $\alpha_1 = 0$, $\beta_{lj} \in \mathbb{R}$ a $\alpha_l \in \mathbb{R}$. Číslo $m \in \mathbb{N}$ nazýváme *řád metody Runge-Kutta*. Toto číslo souvisí s odhadem chyby metod Runge-Kutta, o chybě pojednává například A. Ralston v knize [4].

Koeficienty α_l , β_{lj} a w_l každé metody Runge-Kutta můžeme zapsat do tabulky, která slouží ke zpřehlednění těchto metod. Následující tabulka ilustruje zápis explicitních metod Runge-Kutta.

0					
α_2	β_{21}				
α_3	β_{31}	β_{32}			
\vdots	\vdots	\vdots			
α_m	β_{m1}	β_{m2}	\dots	$\beta_{m,m-1}$	
	w_1	w_2	\dots	w_{m-1}	w_m

K nejužívanějším explicitním metodám Runge-Kutta patří tzv. *klasická metoda Runge-Kutta*, jde o metodu čtvrtého řádu tvaru

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (3.5)$$

kde

$$\begin{aligned}k_1 &= hf(x_i, y_i), \\k_2 &= hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1), \\k_3 &= hf(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2), \\k_4 &= hf(x_i + h, y_i + k_3)\end{aligned}$$

pro $i = 0, 1, 2, \dots, N-1$, přičemž hodnoty x_0 a y_0 jsou dány počáteční podmínkou $y(x_0) = y_0$. Nyní si uvedeme tabulkový zápis této metody.

$$\begin{array}{c|ccc}0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & 0 & \frac{1}{2} & \\ 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}\end{array}$$

Dále si pomocí klasické metody Runge-Kutta vypočteme následující příklad.

Příklad 3.1. Pomocí klasické metody Runge-Kutta najděte hodnoty přibližného řešení počáteční úlohy

$$\begin{aligned}y' &= y \\ y(0) &= 1\end{aligned}$$

na intervalu $[0, 1.4]$ s krokem $h = 0.2$.

Řešení: Nejprve určíme počet dělicích intervalů $N = 7$. Klasická metoda Runge-Kutta je pro danou počáteční úlohu tvaru

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

kde

$$\begin{aligned}k_1 &= hy_i, & k_3 &= h(y_i + \frac{1}{2}k_2), \\k_2 &= h(y_i + \frac{1}{2}k_1), & k_4 &= h(y_i + k_3).\end{aligned}$$

V prvním kroku ve vztahu (3.5) zvolíme $i = 0$, tj.

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

kde

$$\begin{aligned} k_1 &= hy_0, & k_3 &= h(y_0 + \frac{1}{2}k_2), \\ k_2 &= h(y_0 + \frac{1}{2}k_1), & k_4 &= h(y_0 + k_3), \end{aligned}$$

přičemž hodnota y_0 je dána počáteční podmínkou. Po dosazení dostáváme

$$\begin{aligned} k_1 &= 0.2 \cdot 1 = 0.2, & k_3 &= 0.2(1 + \frac{1}{2} \cdot 0.22) = 0.222, \\ k_2 &= 0.2(1 + \frac{1}{2} \cdot 0.2) = 0.22, & k_4 &= 0.2(1 + 0.222) = 0.2444 \end{aligned}$$

a hodnota přibližného řešení y_1 je

$$y_1 = 1 + \frac{1}{6}(0.2 + 2 \cdot 0.22 + 2 \cdot 0.222 + 0.2444) = 1.2214.$$

Pokračujeme dále druhým krokem, volíme tedy $i = 1$ a ze vztahu (3.5) dostáváme

$$y_2 = y_1 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

přičemž

$$\begin{aligned} k_1 &= hy_1, & k_3 &= h(y_1 + \frac{1}{2}k_2), \\ k_2 &= h(y_1 + \frac{1}{2}k_1), & k_4 &= h(y_1 + k_3), \end{aligned}$$

kde hodnotu y_1 známe z předchozího kroku. Po dosazení konkrétních hodnot máme

$$\begin{aligned} k_1 &= 0.2 \cdot 1.2214 = 0.2443, & k_3 &= 0.2(1.2214 + \frac{1}{2} \cdot 0.2687) = 0.2712, \\ k_2 &= 0.2(1.2214 + \frac{1}{2} \cdot 0.2443) = 0.2687, & k_4 &= 0.2(1.2214 + 0.2712) = 0.2985 \end{aligned}$$

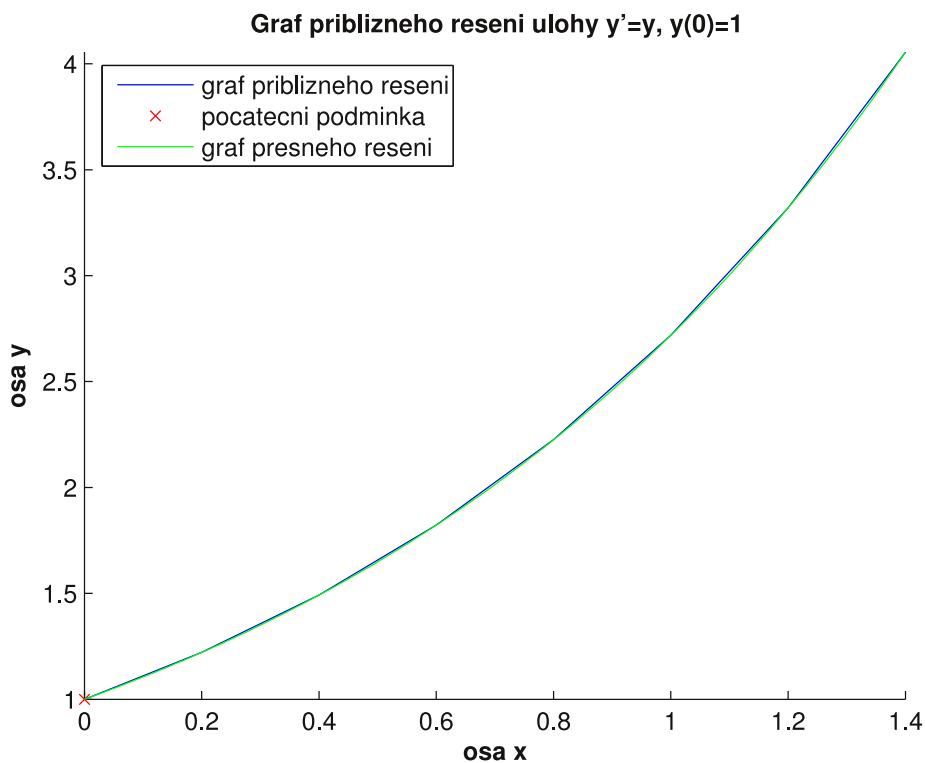
a hodnota přibližného řešení y_2 je

$$y_2 = 1.2214 + \frac{1}{6}(0.2443 + 2 \cdot 0.2687 + 2 \cdot 0.2712 + 0.2985) = 1.4918.$$

Obdobným způsobem bychom pokračovali až do volby $i = 6$. Dále si uvedeme tabulku vypočtených hodnot přibližného řešení a hodnoty přesného řešení. Na konec příkladu vykreslíme křivku spojující vypočtené hodnoty přibližného řešení a přesné řešení, viz. Obrázek 13.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i	1.0000	1.2214	1.4918	1.8221	2.2255	2.7182	3.3200	4.0550
$y(x_i)$	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183	3.3201	4.0552

Tabulka 9: Vypočtené hodnoty přibližného a přesného řešení



Obrázek 13: Graf přibližného a přesného řešení úlohy

$$y' = y$$

$$y(0) = 1$$

Jak vidíme z Obrázku 13, popř. z Tabulky 9, je klasická metoda Runge-Kutta přesnější numerickou metodou než Eulerova metoda, viz. příklad 2.1. Pro ilustraci si zde uvedeme tabulku, ve které si shrneme vypočtené hodnoty přibližného řešení jednotlivých metod a hodnoty přesného řešení $y(x_i)$. Označme y_i^1 hodnoty přibližného řešení z příkladu 2.1 a y_i^2 hodnoty řešení z příkladu 3.1.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i^1	1.0000	1.2000	1.4400	1.7280	2.0736	2.4883	2.9860	3.5832
y_i^2	1.0000	1.2214	1.4918	1.8221	2.2255	2.7182	3.3200	4.0550
$y(x_i)$	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183	3.3201	4.0552

Tabulka 10: Vypočtené hodnoty přibližného a přesného řešení



Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení počáteční úlohy tvaru (3.1) pomocí klasické metody Runge-Kutta s ekvidistantní sítí uzlů, včetně kontroly vstupních dat, který si zde uvedeme.

M-file 3.1.

```

-----
% Klasická metoda Runge-Kutta (4. řád) pro ekvidistantní síť uzlů
%
% VSTUP:
% funkce...předpis funkce na pravé straně diferenciální rovnice,
% musí být proměnných x a y, např. ve tvaru funkce = @(x,y) x+y
% PP...počáteční podmínka PP = [x_0 y_0], např. ve tvaru
% PP = [0 1]
% int...interval, na kterém hledáme řešení dané funkce ve tvaru
% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů
% pokud chcete zadat krok h, pak zadejte N = 0 a jestliže
% chcete zadat N, potom zvolte h = 0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...vektor hodnot přibližného řešení v bodech x_i

```



```

function [x,y] = Klasicka_RK_ekv(funkce,PP,int,h,N)

h = single(h);
x(1) = int(1);
y(1) = PP(2);

if int(1) == PP(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),
    if h == 0,
        h = (int(2)-int(1))/N;
    else
        N = (int(2)-int(1))/h;
    end;
if N/fix(N) == 1,
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
for i = 1:N,
    k1 = h*funkce(x(i),y(i));
    k2 = h*funkce(x(i)+(1/2)*h,y(i)+(1/2)*k1);
    k3 = h*funkce(x(i)+(1/2)*h,y(i)+(1/2)*k2);
    k4 = h*funkce(x(i)+h,y(i)+k3);
    y(i+1) = y(i)+(1/6)*(k1+2*k2+2*k3+k4);
end;
disp('Vektor uzlů')
x
disp('Vektor hodnot přibližného řešení')
y
plot(x,y,x(1),y(1),'rx')

```

```

        axis([int(1),int(2),min(y),max(y)]);
    else
        disp('Rozdíl b-a není celočíselným násobkem kroku h!
        Změňte interval int = [a b] nebo krok h.')
```

end;

```

else
    disp('Špatně zadaný krok h nebo číslo N! Změňte krok h nebo
    číslo N.')
```

end;

```

else
    disp('Špatně zadaný interval [a b]! Změňte interval int =
    [a b].')
```

end;

```

else
    disp('Bod x_0 není shodný s bodem a! Změňte počáteční podmínku
    PP nebo interval int.')
```

end;

Data, která jsme vypočítali v příkladu 3.1 můžeme zkontrolovat prostřednictvím M-filu 3.1, jakým způsobem tak provést si ukážeme v následující poznámce.

Poznámka 3.1. Voláním

```

funkce = @(x,y) y;
PP = [0 1];
int = [0 1.4];
h = 0.2;
N = 0;
```

```

[x,y] = Klasicka_RK_ekv(funkce,PP,int,h,N);
```

můžeme ověřit vypočítané hodnoty přibližného řešení z příkladu 3.1.

V dalším příkladu bude naším úkolem pomocí právě uvedeného M-filu najít hodnoty přibližného řešení počáteční úlohy tvaru

$$y' = \cos x \sin y + \frac{x}{y}$$

$$y(-1) = 1$$

Příklad 3.2. Najděte hodnoty přibližného řešení Cauchyovy úlohy, která je vyjádřena

$$y' = \cos x \sin y + \frac{x}{y}$$

$$y(-1) = 1.$$

na intervalu $[-1, 1]$ s krokem $h = 0.2$. Pro hledání hodnot použijte M-filu [3.1](#). Tento M-file v programu MATLAB vyvoláme příkazem

```
funkce = @(x,y) cos(x)*sin(y)+(x/y);
```

```
PP = [-1 1];
```

```
int = [-1 1];
```

```
h = 0.2;
```

```
N = 0;
```

```
[x,y] = Klasicka_RK_ekv(funkce,PP,int,h,N);
```

přičemž nám vyjdou hodnoty

Vektor uzlů

x =

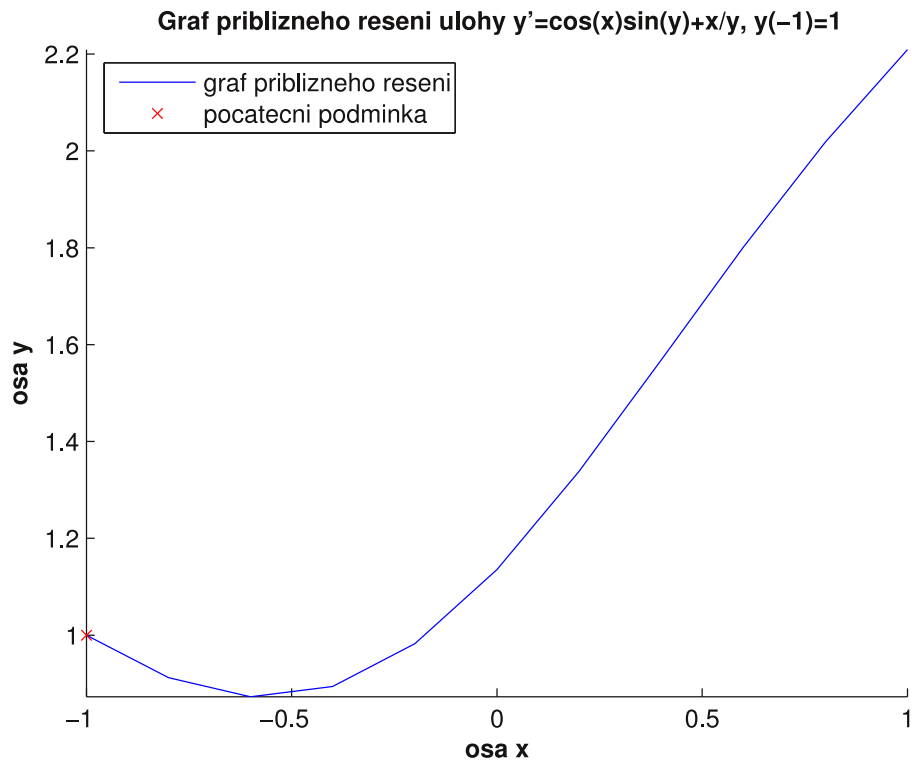
-1.0000	-0.8000	-0.6000	-0.4000	-0.2000	-0.0000
0.2000	0.4000	0.6000	0.8000	1.0000	

Vektor hodnot přibližného řešení

y =

1.0000	0.9122	0.8731	0.8939	0.9823	1.1354
1.3384	1.5684	1.8012	2.0182	2.2094	

Na závěr tohoto příkladu si vykreslíme graf přibližného řešení dané počáteční úlohy, viz. Obrázek 14. ♣



Obrázek 14: Graf přibližného a přesného řešení úlohy

$$y' = \cos x \sin y + \frac{x}{y}$$

$$y(-1) = 1$$

Dále si uvedeme metodu Runge-Kutta 2. řádu, též nazývanou Heunovou metodou. Tato metoda je tvaru

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$

kde

$$k_1 = hf(x_i, y_i),$$

$$k_2 = hf(x_i + h, y_i + k_1)$$

pro $i = 0, 1, 2, \dots, N - 1$. Tabulkový zápis koeficientů Heunovy metody.

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Nyní si uvedeme příklad, ve kterém k hledání hodnot přibližného řešení použijeme právě uvedené Heunovy metody.

Příklad 3.3. Pomocí Heunovy metody najděte hodnoty přibližného řešení úlohy

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

na intervalu $[0, 1.4]$ s krokem $h = 0.2$.

Řešení: Nejprve určíme počet dělicích intervalů $N = 7$. Heunova metoda je pro zadanou diferenciální rovnici tvaru

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2),$$

kde

$$\begin{aligned} k_1 &= hy_i, \\ k_2 &= h(y_i + k_1). \end{aligned}$$

Dále zvolíme ve vzorci Heunovy metody $i = 0$ a pokračujeme výpočtem

$$\begin{aligned} k_1 &= hy_0 = 0.2 \cdot 1 = 0.2, \\ k_2 &= h(y_0 + k_1) = 0.2(1 + 0.2) = 0.208 \end{aligned}$$

hodnota y_1 je dána

$$y_1 = y_0 + \frac{1}{2}(k_1 + k_2) = 1 + \frac{1}{2}(0.2 + 0.208) = 1.204.$$

Nyní zvolíme $i = 1$, tj.

$$\begin{aligned} k_1 &= hy_1 = 0.2 \cdot 1.22 = 0.244, \\ k_2 &= h(y_1 + k_1) = 0.2(1.22 + 0.244) = 0.2928 \end{aligned}$$

a hodnota y_2 je dána

$$y_2 = y_1 + \frac{1}{2}(k_1 + k_2) = 1.204 + \frac{1}{2}(0.244 + 0.2928) = 1.4884.$$

Obdobným způsobem bychom pokračovali až do volby $i = 6$. Na závěr příkladu si uvedeme tabulku s vypočtenými hodnotami přibližného řešení a hodnoty přesného řešení $y(x_i)$. Vykreslíme si také graf přibližného a přesného řešení dané počáteční úlohy, viz. Obrázek 15.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i	1.0000	1.2200	1.4884	1.8159	2.2154	2.7028	3.2975	4.0230
$y(x_i)$	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183	3.3201	4.0552

Tabulka 11: Vypočtené hodnoty přibližného a přesného řešení



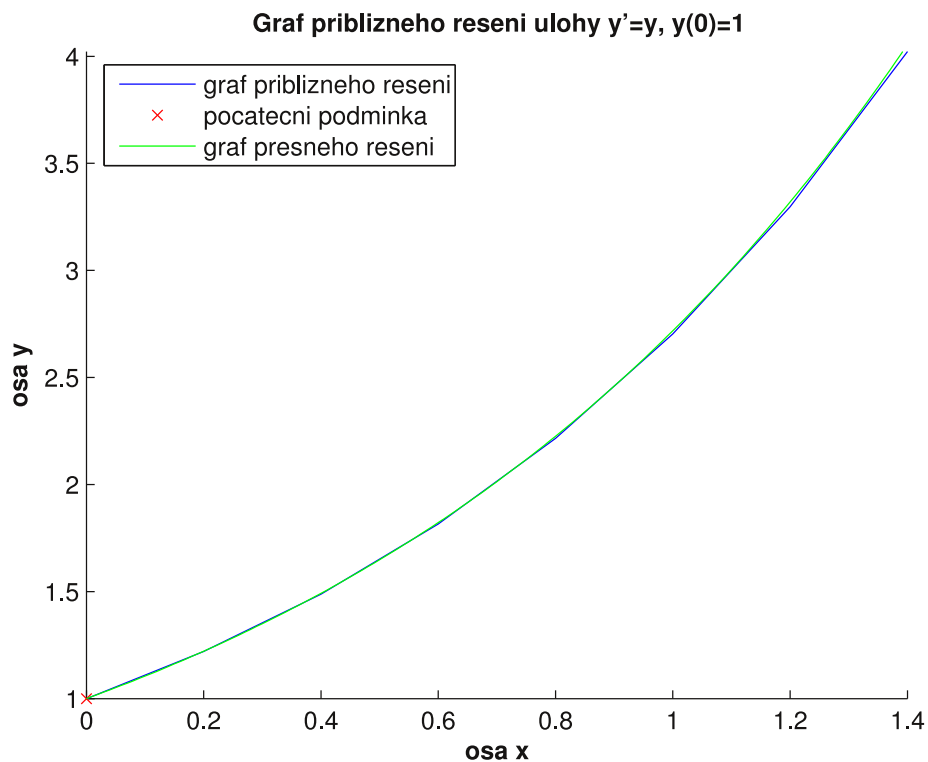
Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení počáteční úlohy tvaru (3.1) pomocí Heunovy metody s ekvidistantní sítí uzlů, včetně kontroly vstupních dat, který si zde uvedeme.

M-file 3.2.

```

-----
% Heunova metoda (2. řád) pro ekvidistantní síť uzlů
%
% VSTUP:
% funkce...předpis funkce na pravé straně diferenciální rovnice,
% musí být proměnných x a y, např. ve tvaru funkce = @(x,y) x+y
% PP...počáteční podmínka PP = [x_0 y_0], např. ve tvaru
% PP = [0 1]
% int...interval, na kterém hledáme řešení dané funkce ve tvaru
% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů

```



Obrázek 15: Graf přibližného a přesného řešení úlohy

$$y' = y$$

$$y(0) = 1$$

```
% pokud chcete zadat krok h, pak zadejte N=0 a jestliže chcete
% zadat N, potom zvolte h=0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...vektor hodnot přibližného řešení v bodech x_i

function [x,y] = Heun_ekv(funkce,PP,int,h,N)

h = single(h);
```

```

x(1) = int(1);
y(1) = PP(2);

if int(1) == PP(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),
    if h == 0,
        h = (int(2)-int(1))/N;
    else
        N = (int(2)-int(1))/h;
    end;
if N/fix(N) == 1,
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
    for i = 1:N,
        k1 = h*funkce(x(i),y(i));
        k2 = h*funkce(x(i)+h,y(i)+k1);
        y(i+1) = y(i)+(1/2)*(k1+k2);
    end;
    disp('Vektor uzlů')
    x
    disp('Vektor hodnot přibližného řešení')
    y
    plot(x,y,x(1),y(1),'rx')
    axis([int(1),int(2),min(y),max(y)]);
else
    disp('Rozdíl b-a není celočíselným násobkem kroku h!
    Změňte interval int = [a b] nebo krok h.')
end;

```



```

else
    disp('Špatně zadaný krok h nebo číslo N! Změňte krok h nebo
        číslo N.')
```

end;

```

else
    display('Špatně zadaný interval [a b]! Změňte interval int =
        [a b].')
```

end;

```

else
    display('Bod x_0 není shodný s bodem a! Změňte počáteční
        podmínku PP nebo interval int.')
```

end;

Data, která jsme vypočítali v příkladu 3.3 můžeme zkontrolovat prostřednictvím M-filu 3.2, jakým způsobem tak provést si ukážeme v následující poznámce.

Poznámka 3.2. Voláním

```

funkce = @(x,y) y;
PP = [0 1];
int = [0 1.4];
h = 0.2;
N = 0;

[x,y] = Heun_ekv(funkce,PP,int,h,N);
```

můžeme ověřit vypočítané hodnoty přibližného řešení z příkladu 3.3.

Nyní můžeme přejít k vektorovému zápisu metod Runge-Kutta. Vztah (3.4) explicitních metod Runge-Kutta je pro úlohu (3.2) tvaru

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \sum_{l=1}^m w_l \mathbf{k}_l,$$

kde $m \in \mathbb{N}$, $w_l \in \mathbb{R}$ a $\mathbf{k}_l = h\mathbf{f}(x_i + \alpha_l h, \mathbf{y}_i + \sum_{j=1}^{l-1} \beta_{lj} \mathbf{k}_j)$ pro $\forall l = 1, 2, \dots, m$, přičemž $\alpha_1 = 0$, $\beta_{lj} \in \mathbb{R}$ a $\alpha_l \in \mathbb{R}$.

Explicitní klasická metoda Runge-Kutta je pro úlohu (3.2) tvaru

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad (3.6)$$

kde

$$\begin{aligned} \mathbf{k}_1 &= h\mathbf{f}(x_i, \mathbf{y}_i), \\ \mathbf{k}_2 &= h\mathbf{f}(x_i + \frac{1}{2}h, \mathbf{y}_i + \frac{1}{2}\mathbf{k}_1), \\ \mathbf{k}_3 &= h\mathbf{f}(x_i + \frac{1}{2}h, \mathbf{y}_i + \frac{1}{2}\mathbf{k}_2), \\ \mathbf{k}_4 &= h\mathbf{f}(x_i + h, \mathbf{y}_i + \mathbf{k}_3) \end{aligned}$$

pro $i = 0, 1, 2, \dots, N - 1$.

Dále si uvedeme příklad na soustavu diferenciálních rovnic prvního řádu, ve kterém budeme hledat hodnoty přibližného řešení pomocí explicitní metody Runge-Kutta 4. řádu tvaru (3.6).

Příklad 3.4. Pomocí explicitní metody Runge-Kutta 4. řádu najděte hodnoty přibližného řešení diferenciální rovnice 2. řádu tvaru

$$y'' = -y' - y$$

s počátečními podmínkami $y(0) = 1, y'(0) = 1$ na intervalu $[0, 1.4]$ s krokem $h = 0.2$.

Řešení: Počet dělicích intervalů $N = 7$. Daná diferenciální rovnice je ekvivalentní se systémem diferenciálních rovnic prvního řádu

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -y_2 - y_1 \end{aligned}$$

s počátečními podmínkami $y_1(0) = 1, y_2(0) = 1$. Klasická metoda Runge-Kutta je v tomto případě dána vyjádřením

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4),$$

kde

$$\begin{aligned}\mathbf{k}_1 &= h \begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix}, \\ \mathbf{k}_2 &= h \left[\begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix} + \frac{1}{2}\mathbf{k}_1 \right], \\ \mathbf{k}_3 &= h \left[\begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix} + \frac{1}{2}\mathbf{k}_2 \right], \\ \mathbf{k}_4 &= h \left[\begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix} + \mathbf{k}_3 \right]\end{aligned}$$

pro $i = 0, 1, 2, \dots, 6$.

Dále v tomto vztahu položíme $i = 0$ a dostaneme

$$\mathbf{y}_1 = \mathbf{y}_0 + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4),$$

kde

$$\begin{aligned}\mathbf{k}_1 &= h \begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix}, \\ \mathbf{k}_2 &= h \left[\begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix} + \frac{1}{2}\mathbf{k}_1 \right], \\ \mathbf{k}_3 &= h \left[\begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix} + \frac{1}{2}\mathbf{k}_2 \right], \\ \mathbf{k}_4 &= h \left[\begin{pmatrix} y_{02} \\ -y_{02} - y_{01} \end{pmatrix} + \mathbf{k}_3 \right]\end{aligned}$$

kde $h = 0.2$ a $\mathbf{y}_0 = (1, 1)^T$. Po dosazení vektoru \mathbf{y}_0 a $\mathbf{f}(x_0, \mathbf{y}_0)$ dostaneme

$$\begin{aligned}\mathbf{k}_1 &= 0.2 \begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} = \begin{pmatrix} 0.2 \\ -0.4 \end{pmatrix}, \\ \mathbf{k}_2 &= 0.2 \left[\begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -0.4 \\ 0.2 \end{pmatrix} \right] = \begin{pmatrix} 0.16 \\ -0.38 \end{pmatrix}, \\ \mathbf{k}_3 &= 0.2 \left[\begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} -0.38 \\ 0.16 \end{pmatrix} \right] = \begin{pmatrix} 0.162 \\ -0.378 \end{pmatrix}, \\ \mathbf{k}_4 &= 0.2 \left[\begin{pmatrix} 1 \\ -1 - 1 \end{pmatrix} + \begin{pmatrix} -0.378 \\ 0.162 \end{pmatrix} \right] = \begin{pmatrix} 0.1244 \\ -0.3568 \end{pmatrix}.\end{aligned}$$

Hodnota \mathbf{y}_1 je

$$\begin{aligned} \mathbf{y}_1 &= \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{6} \left[\begin{pmatrix} 0.2 \\ -0.4 \end{pmatrix} + 2 \begin{pmatrix} 0.16 \\ -0.38 \end{pmatrix} + 2 \begin{pmatrix} 0.162 \\ -0.378 \end{pmatrix} + \begin{pmatrix} 0.1244 \\ -0.3568 \end{pmatrix} \right] = \\ &= \begin{pmatrix} 1.1614 \\ 0.6212 \end{pmatrix}. \end{aligned}$$

Dále bychom pokračovali obdobně až do volby $i = 6$. Na závěr příkladu si uvedeme tabulku s vypočtenými hodnotami přibližného řešení, včetně hodnot přesného řešení. Uvedeme si i graf přibližného a přesného řešení zadané počáteční úlohy, viz. Obrázek 16, na kterém je první složka přibližného řešení y_{i1} na prvním obrázku a druhá složka přibližného řešení y_{i2} na druhém obrázku.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_{i1}	1.0000	1.1614	1.2516	1.2802	1.2573	1.1932	1.0976	0.9797
y_{i2}	1.0000	0.6212	0.2886	0.0059	-0.2258	-0.4073	-0.5412	-0.6313
$y(x_i)$	1.0000	1.1614	1.2516	1.2802	1.2573	1.1932	1.0976	0.9796
$y'(x_i)$	1.0000	0.6212	0.2886	0.0059	-0.2258	-0.4073	-0.5412	-0.6313

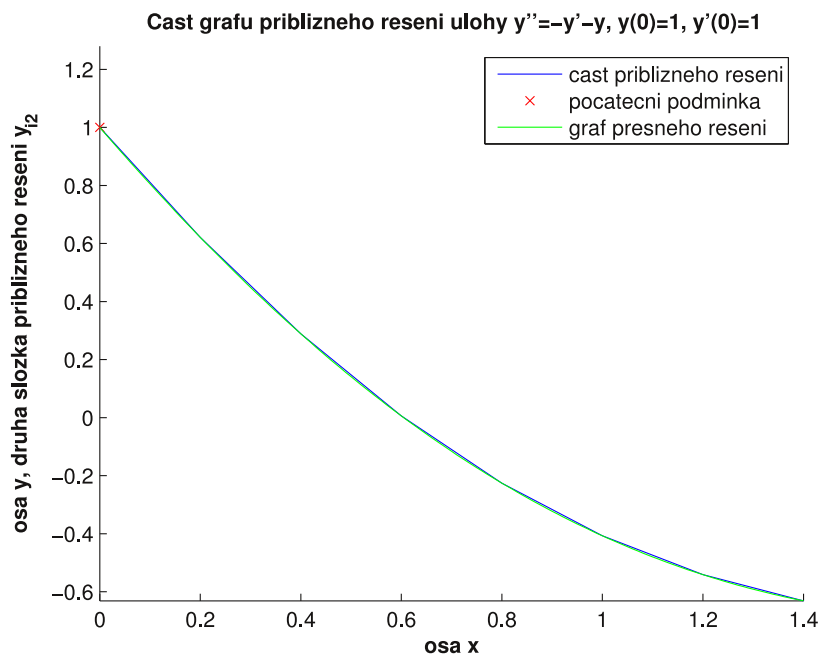
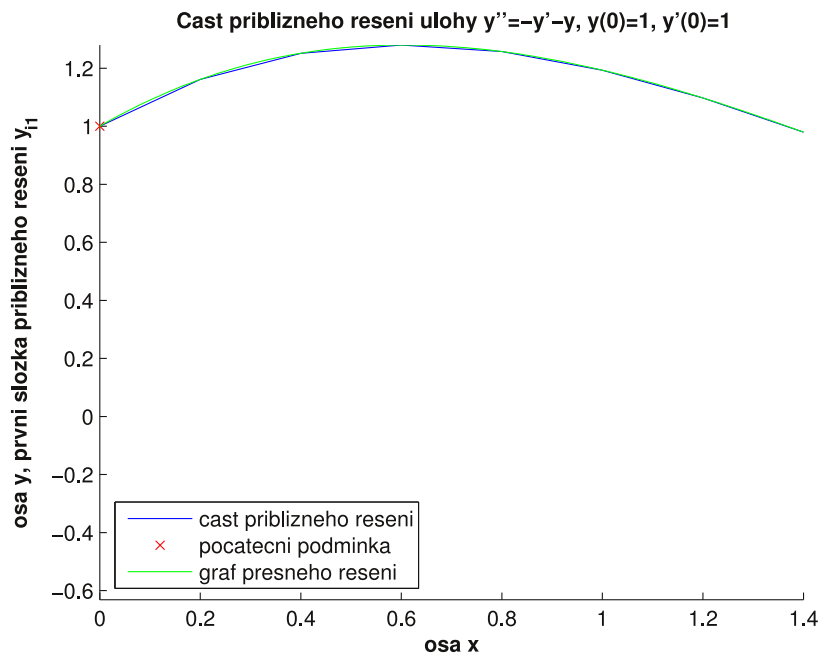
Tabulka 12: Vypočtené hodnoty přibližného a přesného řešení



Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení soustavy diferenciálních rovnic prvního řádu tvaru (3.2) pomocí klasické metody Runge-Kutta s ekvidistantní sítí uzlů, včetně kontroly vstupních dat, který si zde uvedeme.

M-file 3.3.

```
-----
% Klasická metoda Runge-Kutta pro soustavu n diferenciálních
% rovnic prvního řádu a ekvidistantní síť uzlů
%
% VSTUP:
% Y...vektor funkcí pravých stran diferenciálních rovnic ve tvaru
```



Obrázek 16: Graf přibližného a přesného řešení úlohy

$$y'' = -y' - y$$

s počátečními podmínkami $y(0) = 1, y'(0) = 1$

```

% Y(1) = y(2)
% Y(2) = y(3)
% ...
% Y(N) = f(x,y(1),y(2),...,y(N)),
% nebo také tvaru
% Y = [y(2) y(3) ... f(x,y(1),y(2),...,y(N))], který zadáme do
% samostatného m-filu nazvaného zadani.m, tedy např.
% function [Y] = zadani(x,y)
% zadání soustavy
% Y(1) = y(2)
% Y(2) = y(3)
% Y(3) = x+y(1)+y(2)
% y...počáteční podmínky y = [y(0) y(1) ... y(N)], např. ve tvaru
% y = [0 1 ... 3] nebo y = [0;1; ...; 3]
% x0...bod x_0 z počátečních podmínek
% int...interval, na kterém hledáme řešení dané funkce ve tvaru
% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů
% pokud chcete zadat krok h, pak zadejte N=0 a jestliže chcete
% zadat N, potom zvolte h=0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...matice hodnot přibližného řešení v bodech x_i, v níž první
% řádek je první složkou y_i1 přibližného řešení, y_i2 druhou
% složkou přibližného řešení, atd.

function [x,y] = Klasicka_RK_ekv_sous(y,x0,int,h,N)

```

```

h = single(h);
x(1) = x0;

if x0 == int(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),
    if h == 0,
        h = (int(2)-int(1))/N;
    else
        N = (int(2)-int(1))/h;
    end;
if N/fix(N) == 1,
    Q1 = length(y);
    for i = 1:Q1,
        y(i,1) = y(i);
    end;
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
    for i = 1:N,
        k1 = h*zadani(x(i),y(:,i));
        k1 = k1';
        k2 = h*zadani(x(i)+(1/2)*h,y(:,i)+(1/2)*k1);
        k2 = k2';
        k3 = h*zadani(x(i)+(1/2)*h,y(:,i)+(1/2)*k2);
        k3 = k3';
        k4 = h*zadani(x(i)+h,y(:,i)+k3);
        k4 = k4';
        y(:,i+1) = y(:,i)+(1/6)*(k1+2*k2+2*k3+k4);

```

```

end;
k = y(1);
kk = y(1);
for i = 1:N+1,
    l = min(y(:,i));
    ll = max(y(:,i));
    if k > l,
        k = l;
    end;
    if kk < ll,
        kk = ll;
    end;
end;
disp('Vektor hodnot uzlů')
x
disp('Matice vypočítaných hodnot přibližného řešení')
y
hold on
for i = 1:Q1,
    plot(x,y(i,:),x(1),y(1,1),'rx')
end;
hold off
axis([int(1),int(2),k,kk]);
else
    disp('Rozdíl b-a není celočíselným násobkem kroku h!
    Změňte interval int = [a b] nebo krok h.')
end;
else
    disp('Špatně zadaný krok h nebo číslo N! Změňte krok h nebo
    číslo N.')

```



```

end;
else
    disp('Špatně zadaný interval [a b]! Změňte interval int =
    [a b].')
end;
else
    disp('Bod x_0 není shodný s bodem a! Změňte bod x0 nebo
    interval int.')
end;

```

Nyní si uvedeme M-file zadani.m, ve kterém bude zadání soustavy diferen-
ciálních rovnic.

```

function [Y] = zadani(x,y)

% zadání soustavy
Y(1) = y(2);
Y(2) = -y(2)-y(1);

```

Data, která jsme vypočítali předešlém příkladu můžeme zkontrolovat pro-
střednictvím právě uvedeného M-filu, jakým způsobem tak provést si ukážeme v
následující poznámce.

Poznámka 3.3. Voláním

```

y = [1;1];
x0 = 0;
int = [0 1.4];
h = 0.2;
N = 0;

```

```
[x,y] = Klasicka_RK_ekv_sous(y,x0,int,h,N);
```

můžeme ověřit vypočítané hodnoty přibližného řešení.

Výše uvedené explicitní metody a přehled dalších explicitních metod Runge-Kutta, včetně odhadu jejich chyb, je podrobně uveden např. v knize A. Ralstona [4].

3.2 Implicitní metody Runge-Kutta

Nejprve se budeme kvůli jednoduchosti a názornosti zabývat implicitní Eulerovou metodou, ta je tvaru

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1}), \quad (3.7)$$

kde $i = 0, 1, 2, \dots, N - 1$. Jak jsme již na začátku této kapitoly uvedli, je hledání hodnot přibližného řešení pomocí implicitních metod složitější. Jak hledat hodnoty přibližného řešení také už víme, uveďme si tedy příklad, ve kterém budeme hledat hodnoty přibližného řešení pomocí implicitní Eulerovy metody.

Příklad 3.5. Najděte hodnoty přibližného řešení úlohy

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

pomocí implicitní Eulerovy metody na intervalu $[0, 1.4]$ s krokem $h = 0.2$.

Řešení: Počet dělicích intervalů je $N = 7$. Implicitní Eulerova metoda je pro danou diferenciální rovnici tvaru

$$y_{i+1} = y_i + hy_{i+1},$$

předpis podle něhož budeme počítat hodnoty y_{i+1} je

$$y_{i+1} = \frac{y_i}{1 - h} = \frac{y_i}{0.8}.$$

Ve vztahu (3.7) zvolme $i = 0$ a pokračujeme výpočtem

$$y_1 = \frac{y_0}{0.8} = \frac{1}{0.8} = 1.25.$$

Dále položíme $i = 1$, dostaneme

$$y_2 = \frac{y_1}{0.8} = \frac{1.25}{0.8} = 1.5625.$$

Takto bychom pokračovali až do volby $i = 6$. Na závěr příkadu si uvedeme tabulku s vypočítanými hodnotami přibližného řešení, včetně přesného řešení a chyby.

Také si vykreslíme příslušný Eulerův polygon, viz. Obrázek 17.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i	1.0000	1.2500	1.5625	1.9531	2.4414	3.0518	3.8148	4.7685
$y(x_i)$	1.0000	1.2200	1.4900	1.8220	2.2255	2.7183	3.3201	4.0552
e_i	0.0000	0.0300	0.0729	0.1311	0.2159	0.3335	0.4947	0.7133

Tabulka 13: Vypočtené hodnoty přibližného řešení a hodnoty přesného řešení



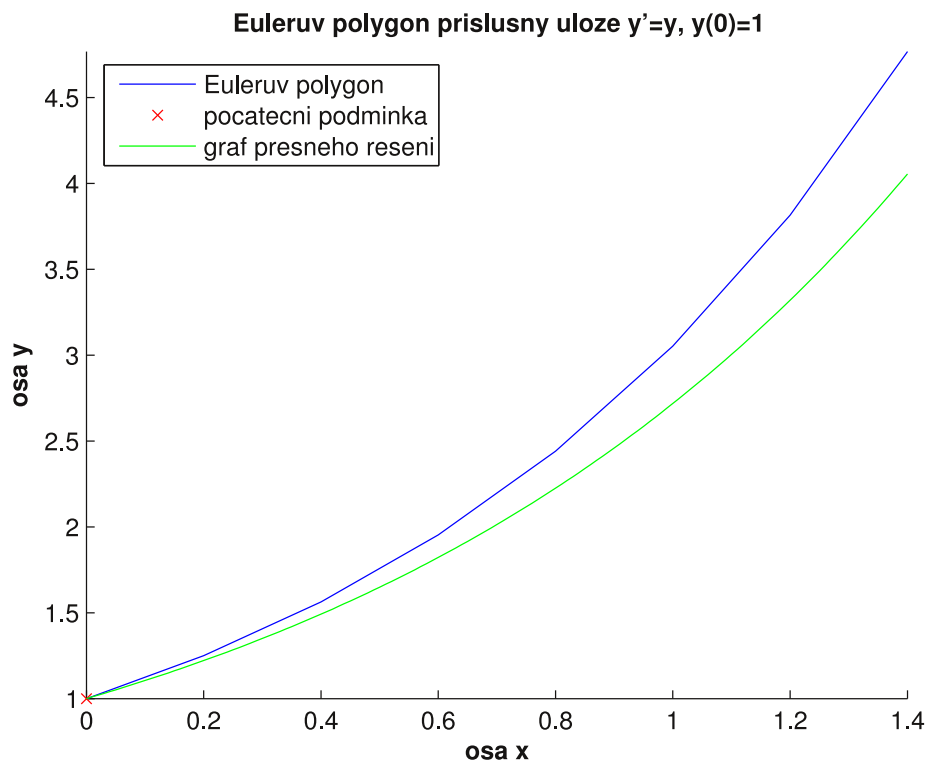
Prostřednictvím matematického softwaru MATLAB jsem vytvořil M-file pro řešení počáteční úlohy tvaru (3.1) pomocí implicitní Eulerovy metody s ekvidistantní sítí uzlů, včetně kontroly vstupních dat, který si zde uvedeme.

M-file 3.4.

```

-----
% Implicitní Eulerova metoda pro ekvidistantní sít' uzlů
%
% VSTUP:
% funkce...předpis funkce na pravé straně diferenciální rovnice,
% musí být proměnných x a y, např. ve tvaru funkce = @(x,y) x+y
% PP...počáteční podmínka PP = [x_0 y_0], např. ve tvaru
% PP = [0 1]
% int...interval, na kterém hledáme řešení dané funkce ve tvaru

```



Obrázek 17: Eulerův polygon příslušný počáteční úloze

$$y' = y$$

$$y(0) = 1$$

```

% int = [a b], např. int = [0 1]
% h...krok
% nebo
% N...počet dělicích intervalů
% pokud chcete zadat krok h, pak zadejte N = 0 a jestliže
% chcete zadat N, potom zvolte h = 0
%
% VÝSTUP:
% x...vektor hodnot, v nichž hledáme přibližné řešení (uzly)
% y...vektor hodnot přibližného řešení v bodech x_i

```

```

function [x,y] = Euler_imp_ekv(funkce,PP,int,h,N)

h = single(h);
x(1) = int(1);
y(1) = PP(2);

if PP(1) == int(1),
if int(1) < int(2),
if (h > 0 && N == 0) || (N >= 1 && N/floor(N) == 1 && h == 0),
    if h == 0,
        h = (int(2)-int(1))/N;
    else
        N = (int(2)-int(1))/h;
    end;
if N/fix(N) == 1,
    for i = 1:N,
        x(i+1) = x(i)+h;
    end;
    for i = 1:N,
        EU = fzero(@(EU) (y(i)-EU+h.*funkce(x(i),EU)),y(i));
        y(i+1) = EU;
    end;
    disp('Vektor uzlů')
    x
    disp('Vektor hodnot přibližného řešení')
    y
    plot(x,y,x(1),y(1),'rx')
    axis([int(1),int(2),min(y),max(y)]);
else

```

```

        disp('Rozdíl b-a není celočíselným násobkem kroku h!
        Změňte interval int = [a b] nebo krok h.')
    end;
else
    disp('Špatně zadaný krok h nebo číslo N! Změňte krok h nebo
    číslo N.')
end;
else
    disp('Špatně zadaný interval [a b]! Změňte interval int =
    [a b].')
end;
else
    disp('Bod x_0 není shodný s bodem a! Změňte počáteční
    podmínku PP nebo interval int.')
end;

```

Data, která jsme vypočítali předešlém příkladu můžeme zkontrolovat prostřednictvím právě uvedeného M-filu, jakým způsobem tak provést si ukážeme v následující poznámce.

Poznámka 3.4. Voláním

```

funkce = @(x,y) y;
PP = [0 1];
int = [0 1.4];
h = 0.2;
N = 0;

[x,y] = Euler_imp_ekv(funkce,PP,int,h,N);

```

můžeme ověřit vypočítané hodnoty přibližného řešení.

Je na místě uvést, z jakého důvodu můžeme Eulerovu metodu, ať explicitní nebo implicitní, pokládat za metodu Runge-Kutta 1. řádu. Vezměme si proto vztah explicitní Eulerovy metody tvaru (3.3) a vztah pro explicitní metody Runge-Kutta (3.4) a zvolme $m = 1$. Vyjádření metody Runge-Kutta bude nyní následující

$$y_{i+1} = y_i + w_1 k_1,$$

kde $w_1 \in \mathbb{R}$, $k_1 = hf(x_i, y_i)$ pro $i = 0, 1, 2, \dots, N - 1$. Pokud položíme $w_1 = 1$ a dosadíme do výše uvedeného vztahu, dostaneme

$$y_{i+1} = y_i + hf(x_i, y_i),$$

což je Eulerova metoda (3.3) pro ekvidistantní síť bodů.

Nyní přejdeme k implicitním metodám Runge-Kutta pro úlohu (3.1). Tyto numerické metody jsou dány vztahem

$$y_{i+1} = y_i + \sum_{l=1}^m w_l k_l, \quad (3.8)$$

kde $m \in \mathbb{N}$, $w_l \in \mathbb{R}$ a $k_l = hf(x_i + \alpha_l h, y_i + \sum_{j=1}^m \beta_{lj} k_j)$ pro $l = 1, 2, \dots, m$, přičemž $\alpha_1 = 0$, $\beta_{lj} \in \mathbb{R}$ a $\alpha_l \in \mathbb{R}$ pro $l = 2, 3, \dots, m$ a $j = 1, 2, \dots, m$.

Obdobně jako koeficienty explicitních metod Runge-Kutta, můžeme psát i koeficienty implicitních metod Runge-Kutta do tabulky následujícím způsobem.

α_1	β_{11}	β_{12}	\dots	$\beta_{1,m-1}$	β_{1m}
α_2	β_{21}	β_{22}	\dots	$\beta_{2,m-1}$	β_{2m}
α_3	β_{31}	β_{32}	\dots	$\beta_{3,m-1}$	β_{3m}
\vdots	\vdots	\vdots		\vdots	\vdots
α_m	β_{m1}	β_{m2}	\dots	$\beta_{m,m-1}$	$\beta_{m,m}$
	w_1	w_2	\dots	w_{m-1}	w_m

Příkladem implicitní metody Runge-Kutta 4. řádu je numerická metoda tvaru

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2), \quad (3.9)$$

kde

$$k_1 = hf\left(x_i + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, y_i + \frac{1}{4}hk_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)hk_2\right),$$

$$k_2 = hf\left(x_i + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, y_i + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_1 + \frac{1}{4}hk_2\right)$$

pro $i = 0, 1, 2, \dots, N-1$, přičemž x_0 a y_0 jsou dány počáteční podmínkou. Uvedme si uspořádání koeficientů metody (3.9) do tabulky.

$$\begin{array}{c|cc} \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right) & \frac{1}{4} & \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right) \\ \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right) & \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right) & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Vypočtěme si příklad, ve kterém budeme hledat hodnoty přibližného řešení pomocí implicitní metody Runge-Kutta dané vztahem (3.9).

Příklad 3.6. Implicitní metodou Runge-Kutta 4. řádu najděte hodnoty přibližného řešení počáteční úlohy tvaru

$$y' = y$$

$$y(0) = 1$$

na intervalu $[0, 1.4]$ s počtem dělicích intervalů $N = 7$. Uvažujme ekvidistantní síť uzlů.

Řešení: Nejprve určíme s jakým krokem budeme počítat hodnoty přibližného řešení. Ze vztahu

$$h = \frac{(b - a)}{N}$$

po dosazení konkrétních hodnot dostaneme $h = 0.2$. Implicitní metoda Runge-Kutta (3.9) je pro danou úlohu tvaru

$$y_{i+1} = y_i + \frac{1}{2}h(k_1 + k_2),$$

kde

$$k_1 = y_i + \frac{1}{4}hk_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)hk_2, \quad (3.10)$$

$$k_2 = y_i + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_1 + \frac{1}{4}hk_2. \quad (3.11)$$

Z první rovnice (3.10) si vyjádříme k_2 , tj.

$$k_2 = \frac{k_1 - \frac{1}{4}hk_1 - y_i}{\left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)h},$$

jež dosadíme do vztahu (3.11). Dále si po tomto dosazení z rovnice (3.11) vyjádříme k_1 , po vhodných úpravách máme

$$k_1 = 1.1709y_i,$$

kterým nahradíme k_1 v rovnici (3.10) a vypočteme k_2 . Hledané k_2 je tvaru

$$k_2 = \frac{0.95k_1 - y_i}{0.1077}.$$

Nyní již samotný výpočet. Ve vztahu (3.9) zvolme $i = 0$, tedy

$$y_1 = y_0 + \frac{1}{2}h(k_1 + k_2),$$

kde

$$k_1 = 1.1709y_0,$$

$$k_2 = \frac{0.95k_1 - y_0}{0.1077}.$$

Po dosazení konkrétních hodnot dostaneme

$$k_1 = 1.1709 \cdot 1 = 1.1709,$$

$$k_2 = \frac{0.95 \cdot 1.1709 - 1}{0.1077} = 1.0432.$$

Hodnota y_1 je tedy

$$y_1 = 1 + \frac{0.2}{2}(1.1709 + 1.0432) = 1.2214.$$

Pokračujeme druhým krokem, volíme $i = 1$, tj.

$$y_2 = y_1 + \frac{1}{2}h(k_1 + k_2),$$

kde

$$k_1 = 1.1709y_1,$$
$$k_2 = \frac{0.95k_1 - y_1}{0.1077}.$$

Po dosazení konkrétních hodnot dostaneme

$$k_1 = 1.1709 \cdot 1.2214 = 1.4301,$$
$$k_2 = \frac{0.95 \cdot 1.4301 - 1.2214}{0.1077} = 1.2739.$$

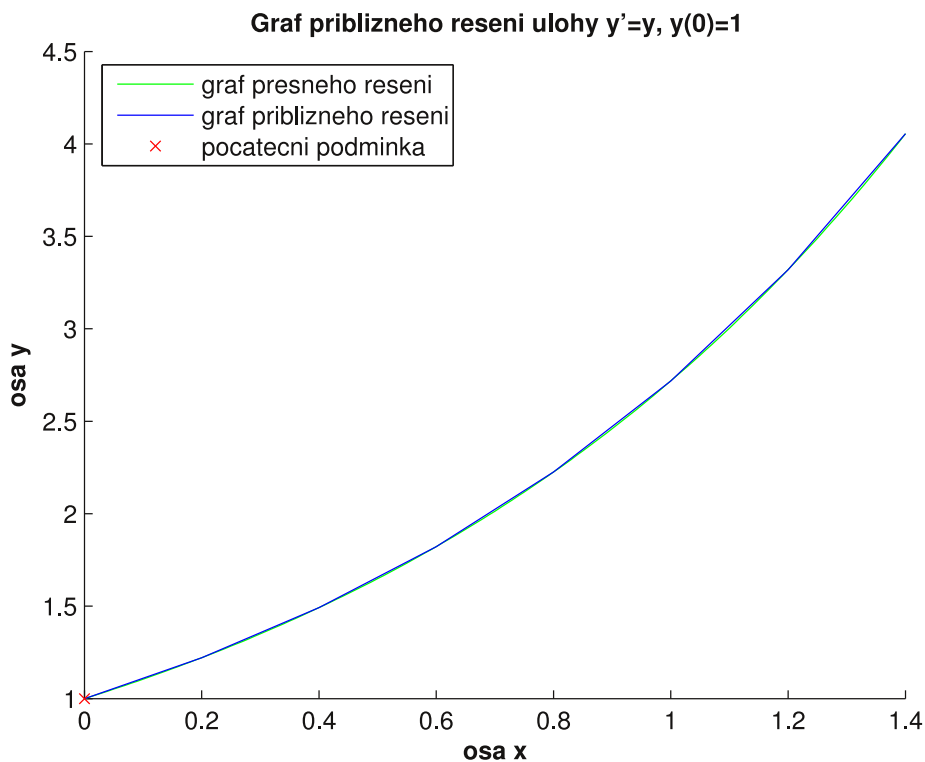
Hodnota y_2 je tedy

$$y_2 = 1.2214 + \frac{0.2}{2}(1.4301 + 1.2739) = 1.4918.$$

Stejným způsobem bychom pokračovali až do volby $i = 6$. Na závěr příkladu si uvedeme tabulku s vypočtenými hodnotami přibližného řešení, včetně hodnot přesného řešení. Vykreslíme i graf přesného a přibližného řešení zadané počáteční úlohy, viz. Obrázek 18.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_i	1.0000	1.2214	1.4918	1.8221	2.2255	2.7182	3.3200	4.0551
$y(x_i)$	1.0000	1.2214	1.4918	1.8221	2.2255	2.7183	3.3201	4.0552





Obrázek 18: Graf přibližného řešení příslušný úloze

$$\begin{aligned} y' &= y \\ y(0) &= 1 \end{aligned}$$

Zabývejme se nyní implicitními metodami Runge-Kutta pro úlohu (3.2). Pro tuto úlohu jsou implicitní metody Runge-Kutta (3.8) dány vztahem

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \sum_{l=1}^m w_l \mathbf{k}_l,$$

kde $m \in \mathbb{N}$, $w_l \in \mathbb{R}$ a $\mathbf{k}_l = h\mathbf{f}(x_i + \alpha_l h, \mathbf{y}_i + \sum_{j=1}^m \beta_{lj} \mathbf{k}_j)$ pro $l = 1, 2, \dots, m$, přičemž $\alpha_1 = 0$, $\beta_{lj} \in \mathbb{R}$ a $\alpha_l \in \mathbb{R}$ pro $l = 2, 3, \dots, m$ a $j = 1, 2, \dots, m$.

Implicitní metoda Runge-Kutta (3.9) je pro úlohu (3.2) tvaru

$$\mathbf{y}_{i+1} = \mathbf{y}_i + \frac{1}{2}(\mathbf{k}_1 + \mathbf{k}_2), \tag{3.12}$$

kde

$$\mathbf{k}_1 = hf\left(x_i + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, \mathbf{y}_i + \frac{1}{4}h\mathbf{k}_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)h\mathbf{k}_2\right),$$
$$\mathbf{k}_2 = hf\left(x_i + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, \mathbf{y}_i + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)h\mathbf{k}_1 + \frac{1}{4}h\mathbf{k}_2\right)$$

pro $i = 0, 1, 2, \dots, N - 1$,

Dále si uvedeme příklad pro hledání hodnot přibližného řešení úlohy (3.2) pomocí právě uvedené implicitní metody Runge-Kutta.

Příklad 3.7. Najděte hodnoty přibližného řešení úlohy

$$y'' = -y' - y$$

s počátečními podmínkami tvaru $y(0) = 1, y'(0) = 1$ pomocí implicitní metody Runge-Kutta. Tuto úlohu uvažujme na intervalu $[0, 1.4]$ s krokem $h = 0.2$.

Řešení: Počet dělicích intervalů N je roven 7. Soustava diferenciálních rovnic prvního řádu příslušná dané diferenciální rovnici druhého řádu je

$$y_1' = y_2$$
$$y_2' = -y_2 - y_1$$

s počátečními podmínkami $y_1(0) = 1, y_2(0) = 1$. Z této soustavy dosadíme vektorovou funkci

$$\mathbf{f}(x_i, \mathbf{y}_i) = \begin{pmatrix} y_{i2} \\ -y_{i2} - y_{i1} \end{pmatrix}$$

do vztahu (3.12) a zvolíme-li v tomto vzorci $i = 0$, dostaneme

$$\mathbf{k}_1 = \begin{pmatrix} 0.1874 \\ -0.3939 \end{pmatrix},$$
$$\mathbf{k}_2 = \begin{pmatrix} 0.1966 \\ -0.3981 \end{pmatrix}.$$

Hodnota \mathbf{y}_1 je

$$\mathbf{y}_1 = \mathbf{y}_0 + \frac{1}{2}h(\mathbf{k}_1 + \mathbf{k}_2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 0.1 \left[\begin{pmatrix} 0.1874 \\ -0.3939 \end{pmatrix} + \begin{pmatrix} 0.1966 \\ -0.3981 \end{pmatrix} \right] = \begin{pmatrix} 1.192 \\ 0.604 \end{pmatrix}.$$

Dále bychom pokračovali obdobně až do volby $i = 6$. Na závěr příkladu si uvedeme tabulku s vypočtenými hodnotami přibližného řešení, včetně hodnot přesného řešení, a také graf ilustrující přibližné a přesné řešení dané úlohy, viz. Obrázek 19, přičemž první složka přibližného řešení y_{i1} bude na prvním obrázku a druhá složka y_{i2} na druhém obrázku.

i	0	1	2	3	4	5	6	7
x_i	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
y_{i1}	1.0000	1.1920	1.3056	1.3493	1.3329	1.2670	1.1623	1.0293
y_{i2}	1.0000	0.6040	0.2496	-0.0562	-0.3094	-0.5088	-0.6554	-0.7522
$y(x_i)$	1.0000	1.1614	1.2516	1.2802	1.2573	1.1932	1.0976	0.9796
$y'(x_i)$	1.0000	0.6212	0.2886	0.0059	-0.2258	-0.4073	-0.5412	-0.6313

Tabulka 14: Vypočtené hodnoty přibližného a přesného řešení



Výše uvedené implicitní metody a podrobný přehled dalších implicitních metod Runge-Kutta lze najít např. v knize E. Hairera a G. Wannera [2].

Na závěr této kapitoly se budeme věnovat programu MATLAB, v němž jsem vytvořil všechny M-fily a obrázky (grafy). Konkrétně se zajímáme o možnost numerického řešení diferenciálních rovnic v tomto matematickém software.

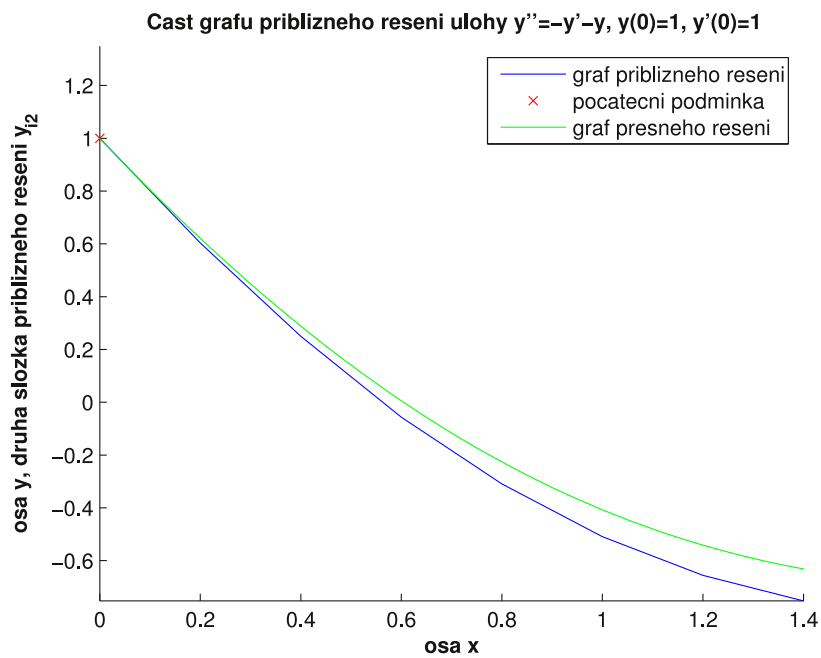
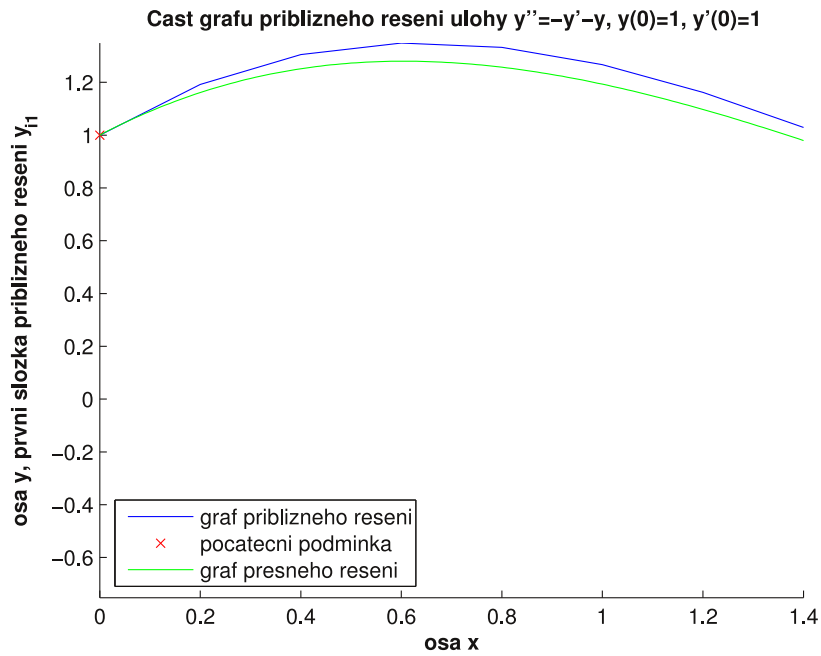
V MATLABu jsou implementovány funkce *ode45*, *ode23*, atd., které slouží právě k numerickému řešení diferenciálních rovnic. Přičemž funkce *ode45* kombinuje metody Runge-Kutta 4. a 5. řádu a funkce *ode23* používá kombinaci metod Runge-Kutta 2. a 3. řádu. Uvedené funkce mění velikost kroku na základě potřeby dosažení požadované přesnosti, krok je tedy adaptabilní.

Nyní si uvedeme vyvolání uvedených funkcí, vykreslíme příslušné grafy včetně grafu přesného řešení a ukážeme si na nich rozdíl jednotlivých funkcí.

Předpokládejme, že máme numericky řešit úlohu

$$y' = y - x^2 + x$$

$$y(0) = 0$$



Obrázek 19: Graf přibližného řešení příslušný počáteční úloze

$$y'' = -y' - y$$

s počátečními podmínkami $y(0) = 1, y'(0) = 1$

na intervalu $[0, 1]$. Přesné řešení je tvaru

$$y = -e^x + x^2 + x + 1.$$

Nejprve tedy graf dané počáteční úlohy pomocí funkce *ode45* a graf přesného řešení vyvoláme příkazem

```
x1 = [0:0.01:1];  
f1 = -exp(x1)+x1.^2+x1+1  
hold on  
plot(x1(1),f1(1),'rx',x1,f1,'g')
```

```
funkce = @(x,y) y-x^2+x;  
[x,y]=ode45(funkce,[0 1],0);  
plot(x,y)  
hold off
```

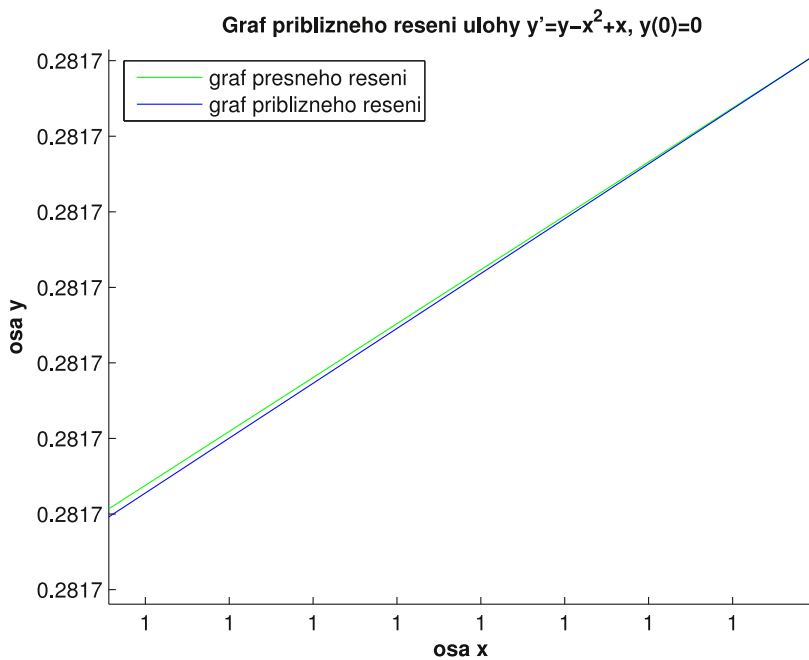
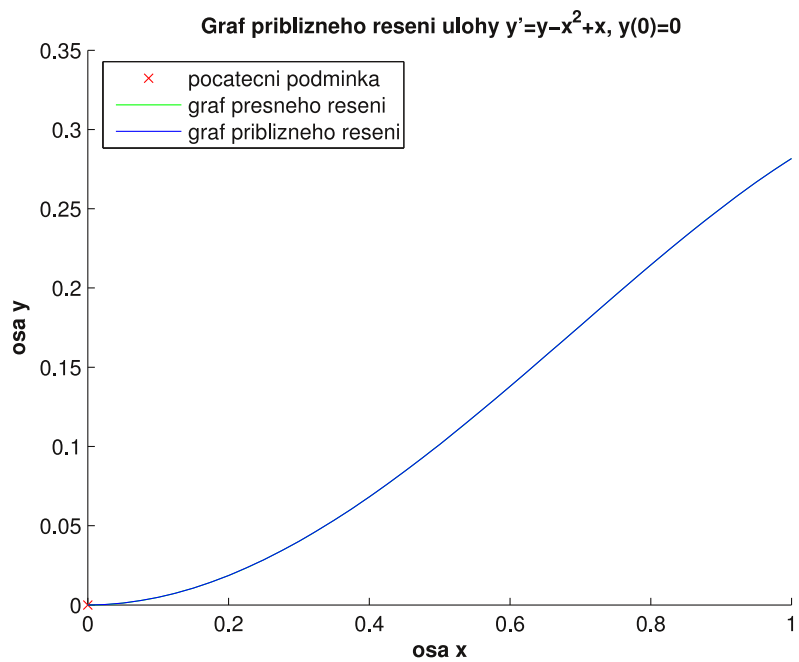
a dostaneme obrázek [20](#), kde je na horním obrázku celkový graf přibližného a přesného řešení a na dolním obrázku je výřez z horního obrázku v pravém koncovém bodě intervalu $[0, 1]$.

Nyní podobným způsobem vyvoláme funkci *ode23*, tedy

```
x1 = [0:0.01:1];  
f1 = -exp(x1)+x1.^2+x1+1  
hold on  
plot(x1(1),f1(1),'rx',x1,f1,'g')
```

```
funkce = @(x,y) y-x^2+x;  
[x,y]=ode23(funkce,[0 1],0);  
plot(x,y)  
hold off
```

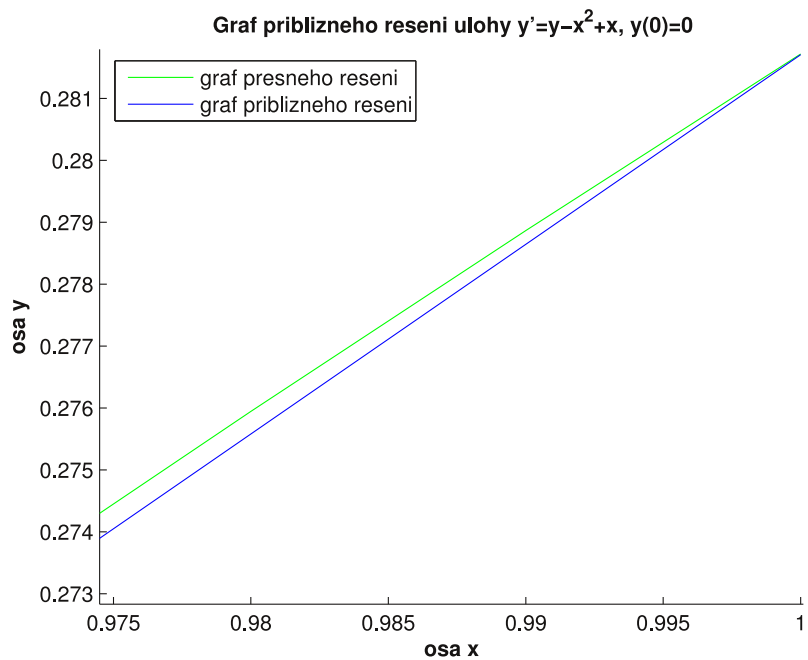
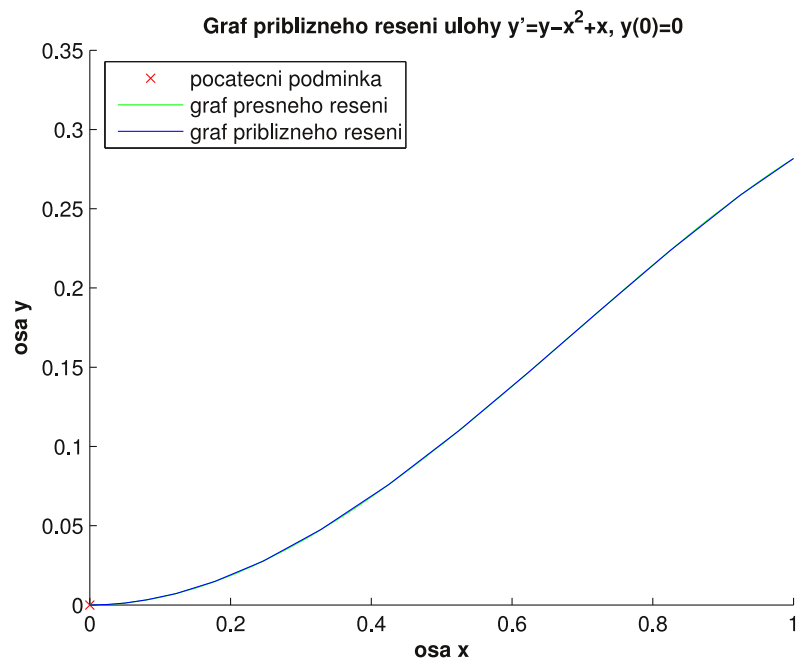
dostáváme obrázek [21](#), kde je na horním obrázku celkový graf přibližného a přesného řešení a na dolním obrázku je výřez z horního obrázku v pravém koncovém bodě intervalu $[0, 1]$.



Obrázek 20: Graf přibližného řešení počáteční úlohy

$$y' = y - x^2 + x$$

$$y(0) = 0$$



Obrázek 21: Graf přibližného řešení počáteční úlohy

$$y' = y - x^2 + x$$

$$y(0) = 0$$

Z obrázků [20](#) a [21](#) vidíme, že funkce *ode45* je přesnější numerickou metodou. Tento fakt jsme mohli vyvodit také z toho, že funkce *ode45* kombinuje metody Runge-Kutta 4. a 5. řádu, kdežto *ode23* je kombinací metod 2. a 3. řádu.

Více o těchto a dalších metodách numerického řešení diferenciálních rovnic se můžete dovědět v nápovědě programu MATLAB.

Závěr

V této bakalářské práci jsou uvedeny pouze základní metody Runge-Kutta. Další typy metod Runge-Kutta můžeme najít zejména v knize E. Hairera a G. Wannerera [2] a v knize A. Ralstona [4], kde je také uveden způsob odvození koeficientů w_l, α_l a β_{lj} těchto numerických metod, včetně odhadu chyb.

Odvozováním koeficientů a odhadem chyb metod Runge-Kutta jsme se nezabývali z časových důvodů a důvodu přesahu mé bakalářské práce. Prvořadým cílem bylo základní seznámení čtenáře s principy fungování numerických metod pro řešení počáteční úlohy, jak pro jednu diferenciální rovnici prvního řádu s počáteční podmínkou, tak pro soustavu diferenciálních rovnic prvního řádu s počátečními podmínkami.

Výsledky, které dostaneme vyvoláním uvedených M-filů se mohou lišit od výsledků uvedených v příkladech, protože příklady jsou ručně počítané a také proto, že jsem používal při výpočtu hodnot přibližného řešení zaokrouhlování na čtyři desetinná místa.

Věřím, že se stanoveného cíle této bakalářské práce podařilo dosáhnout.

Literatura

- [1] GREGUŠ, Michal; ŠVEC, Marko; ŠEDA, Valter. Obyčajné diferenciálne rovnice. 1. vydanie. Bratislava : Alfa, 1985. 376 s.
- [2] HAIRER, Ernst; WANNER, Gerhard. Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems. Second Revised Edition. Berlin : Springer, 1996. 614 s
- [3] LEGRAS, Jean. Metódy a použitie numerickej matematiky. Ing. Kamil Hrubina. 1. vydanie. Bratislava : ALFA, 1978. 383 s. ISBN 63-116-78
- [4] RALSTON, Anthony. Základy numerickej matematiky. Praha : Academia, 1973. 636 s. ISBN 21-116-73
- [5] VITÁSEK, Emil. Základy teorie numerických metod pro řešení diferenciálních rovnic. Vydání 1. Praha : Academia, 1994. 409 s. ISBN 80-200-0281-2