

# **Numerické řešení soustav diferenciálních rovnic pomocí softwaru Mathematica**

Numerical solution of differential equations using the software  
Mathematica

Tomáš Ondra

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2010/2011

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš ONDRA**  
Osobní číslo: **A07077**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Numerické řešení soustav diferenciálních rovnic pomocí software Mathematica**

Zásady pro vypracování:

1. Sestavte literární rešerši v oblasti numerického řešení soustav diferenciálních rovnic.
2. Najděte okruh metod řešení diferenciálních rovnic, které jsou vhodné pro řešení v software Mathematica především z hlediska programovatelnosti.
3. Navrhněte a vyřešte několik vzorových příkladů pro řešení.
4. Zpracujte postup, který by popisoval, jak v softwaru Mathematica řešit diferenciální rovnice nebo soustavy diferenciálních rovnic.
5. Dosažené výsledky řádně okomentujte a ohodnoťte výsledky z hlediska rychlosti, výpočetních nároků, složitosti apod.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Johnston, R. L. *Numerical Methods*. John Wiley & Sons. 1982.
2. Fausett, L. V. *Numerical Methods: Algorithms and Applications*. Prentice Hall; 2002.
3. Boháč, Z., Častová, N.: *Základní numerické metody*. Skriptum VŠB-TUO, Ostrava 1997.
4. Wolfram Research: *Mathematica, Technical and Scientific Software* [online]. Dostupné z URL: <http://www.wolfram.com/>
5. *Wolfram Mathematica 7 Documentation Center* [online]. Dostupné z URL: <http://reference.wolfram.com/mathematica/guide/Mathematica.html>

Vedoucí bakalářské práce: **Ing. Jiří Vojtěšek, Ph.D.**

Ústav řízení procesů


Datum zadání bakalářské práce: **25. února 2011**

Termín odevzdání bakalářské práce: **7. června 2011**

Ve Zlíně dne 25. února 2011

  
prof. Ing. Vladimír Vašek, CSc.  
*děkan*



  
prof. Ing. Vladimír Vašek, CSc.  
*ředitel ústavu*

## **ABSTRAKT**

Bakalářská práce se zabývá numerickými metodami řešení soustav diferenciálních rovnic v software Mathematica. Teoretická část práce vysvětluje možnost řešení diferenciální rovnice numerickými metodami, řešení soustavy diferenciálních rovnic numerickými metodami a některé numerické metody jsou popsány. V teoretické části jsou dále zahrnuty chyby numerických metod, zejména zdroje a typy chyb.

Praktická část se zabývá vytvořeným notebookem v softwaru Mathematica, který složí k numerickému řešení soustav diferenciálních rovnic. Vytvořený notebook je popsán jak s pohledu uživatele, tak s pohledu programátora.

Klíčová slova: numerické metody, diferenciální rovnice, Mathematica, notebook

## **ABSTRACT**

Bachelor thesis deals with the numerical solution methods of systems of differential equations in Mathematica software. The theoretical section explains the possibility of solving differential equations using numerical methods, solving systems of differential equations by numerical methods and some numerical methods are described. The theoretical parts are also included errors of numerical methods, including the sources and types of errors.

The practical part deals with established software Mathematica notebook in which to lodge a numerical solution of differential equation systems. Created notebook described how the user's perspective, so the programmer point of view.

Keywords: numerical methods, differential equations, Mathematica, notebook

## **PODĚKOVÁNÍ**

Děkuji vedoucímu bakalářské práce Ing. Jiřímu Vojtěškovi, Ph.D. za odbornou pomoc a veškerý čas, při tvorbě bakalářské práce

*Motto*

**„Škola jako dílna lidskosti“**

J.A.Komenský

Pašovice

25.5.2011

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 NUMERICKÉ METODY ŘEŠENÍ OBYČEJNÝCH DIFERENCIÁLNÍCH ROVNIC</b> .....	<b>11</b>
1.1 POČÁTEČNÍ ÚLOHA .....	12
1.2 RUNGE-KUTTOVY METODY .....	12
1.2.1 Eulerova metoda .....	12
1.2.2 Modifikace Eulerova metoda.....	15
1.2.3 Rungovy-Kuttovy metody .....	15
<b>2 NUMERICKÉ ŘEŠENÍ SOUSTAV DIFERENCIÁLNÍCH ROVNIC</b> .....	<b>18</b>
2.1 EULEROVA METODA .....	18
2.2 RUNGOVY-KUTTOVY METODY.....	19
<b>3 NUMERICKÉ METODY</b> .....	<b>21</b>
3.1 CHYBY PŘI NUMERICKÝCH VÝPOČTECH .....	21
3.1.1 Zdroje a typy chyb.....	21
3.1.2 Definice chyb.....	22
3.1.3 Zaokrouhlování. Šíření chyb .....	22
3.1.4 Podmíněnost numerických úloh a numerická stabilita algoritmu .....	23
<b>II PRAKTICKÁ ČÁST</b> .....	<b>24</b>
<b>4 POPIS PROGRAMU</b> .....	<b>25</b>
4.1 ZADÁNÍ VSTUPNÍCH PARAMETRŮ .....	25
4.2 METODY VÝPOČTU .....	25
4.3 ZOBRAZENÍ VÝSLEDKŮ.....	26
4.4 POROVNÁNÍ VÝSLEDKŮ JEDNOTLIVÝCH METOD .....	26
<b>5 ŘEŠENÍ SOUSTAVY DVOU DIFERENCIÁLNÍCH ROVNIC</b> .....	<b>27</b>
5.1 ZADÁNÍ VSTUPNÍCH PARAMETRŮ .....	27
5.1.1 Nulování jednotlivých proměnných .....	27
5.1.2 Definování konstant.....	27
5.1.3 Vstupní hodnoty .....	27
5.1.4 Vstupní diferenciální rovnice .....	28
5.1.5 Zadání kroku výpočtu.....	28
5.1.6 Zadání intervalu výpočtu .....	28
5.1.7 Zadání počátečních podmínek .....	28
5.1.1 Pomocné proměnné .....	29
5.2 ZOBRAZENÍ VÝSLEDKŮ.....	29
5.2.1 Zobrazení výsledků pomocí Eulerovy metody.....	29
5.2.2 Zobrazení výsledků pomocí Eulerovy metody-První modifikace.....	30
5.2.3 Zobrazení výsledků pomocí Eulerovy metody-Druhá modifikace.....	31
5.2.4 Zobrazení výsledků pomocí Runge-Kuttovy metody 4.řádu.....	31

5.3	POROVNÁNÍ METOD .....	32
<b>6</b>	<b>APLIKACE PROGRAMU NA NÁZORNÝ PŘÍKLAD .....</b>	<b>33</b>
6.1	ZADÁNÍ VSTUPNÍCH PARAMETRŮ .....	33
6.1.1	Definování konstant .....	33
6.1.2	Vstupní hodnoty .....	34
6.1.3	Vstupní diferenciální rovnice .....	34
6.1.4	Zadání kroku výpočtu .....	34
6.1.5	Zadání intervalu výpočtu .....	35
6.1.6	Zadání počátečních podmínek .....	35
6.1	ZOBRAZENÍ VÝSLEDKŮ .....	35
6.2.1	Zobrazení výsledků pomocí Eulerovy metody .....	35
6.2.2	Zobrazení výsledků pomocí Eulerovy metody-První modifikace .....	36
6.2.3	Zobrazení výsledků pomocí Eulerovy metody-Druhá modifikace .....	36
6.2.4	Zobrazení výsledků pomocí Runge-Kuttovy metody 4.řádu .....	37
5.3	POROVNÁNÍ METOD .....	37
<b>7</b>	<b>ROZDÍLY MEZI ŘEŠENÍ SOUSTAVY ČTYŘ A DVOU DIFERENCIÁLNÍCH ROVNIC .....</b>	<b>38</b>
7.1	ZADÁNÍ VSTUPNÍCH PARAMETRŮ .....	38
7.1.1	Vstupní diferenciální rovnice .....	38
7.1.2	Zadání počátečních podmínek .....	38
7.1.3	Pomocné proměnné .....	39
7.2	METODY VÝPOČTU .....	39
7.2.1	Eulerova metoda .....	39
7.2.2	Eulerova metoda-První modifikace .....	40
7.2.3	Eulerova metoda-Druhá modifikace .....	41
7.2.4	Runge-Kuttova metoda 4.řádu .....	42
7.3	ZOBRAZENÍ VÝSLEDKŮ .....	43
7.1	POROVNÁNÍ VÝSLEDKŮ .....	44
	<b>ZÁVĚR .....</b>	<b>46</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>48</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>49</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>50</b>
	<b>SEZNAM TABULEK .....</b>	<b>51</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>52</b>

## ÚVOD

Většina diferenciálních rovnic, jak z odvětví fyziky, matematiky atd. jsou analyticky řešitelné velmi obtížně. Ve většině případech přesné řešení ani nenalezneme. Numerické metody se tento nedostatek snaží odstranit a hledají pouze přibližná řešení.

Cílem této bakalářské práce bylo vytvořit notebook pro řešení soustav diferenciálních rovnic pomocí numerických metod v softwaru Mathematica, který nám usnadní výpočty a může se aplikovat na různou soustavu diferenciálních rovnic jak z odvětví fyziky tak matematiky.

Teoretická část je věnována teoretickému popisu jednotlivých numerických metod pro řešení diferenciální rovnice. Dále je zde popsán postup jak numerické metody pro řešení diferenciální rovnice upravit na numerické metody pro řešení soustav diferenciálních rovnic. V poslední části této kapitoly jsou popsány numerické metody, zejména chyby při numerických výpočtech.

Praktická část je věnována popisu vytvořeného notebooku. Tato kapitola obsahuje jak jednotlivé vstupní parametry zadávat správně do vytvořeného notebooku. Dále je zde ještě popsáno, jak můžeme vytvořený notebook rozšířit pro řešení soustavy čtyř diferenciálních rovnic oproti soustavě dvou diferenciálních rovnic.

## **I. TEORETICKÁ ČÁST**

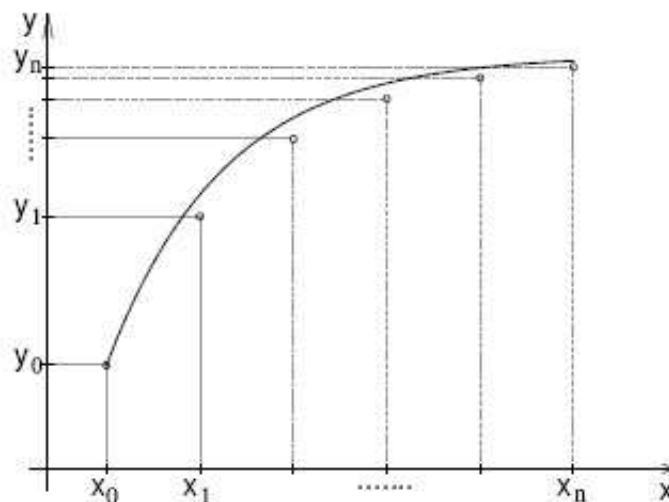
## 1 NUMERICKÉ METODY ŘEŠENÍ OBYČEJNÝCH DIFERENCIÁLNÍCH ROVNIC

Numerické metody se používají proto, že některé obyčejné diferenciální rovnice nejdou analyticky řešit vůbec a některé mají analytické řešení velmi obtížné.

Při řešení numerických metod počítáme pouze hodnoty přibližného řešení v konečném počtu bodů  $a = x_0 < x_1 < x_2 < \dots < x_n = b$ . Tyto body se nazývají uzlové body nebo uzly sítě a množině  $\{x_0, x_1, \dots, x_n\}$  říkáme síť. Krok sítě nazýváme rozdíl mezi uzlovým bodem  $x_{i+1}$  a uzlovým bodem  $x_i$ .

Přibližné hodnoty řešení v uzlových bodech jsou značeny  $y_0, y_1, \dots, y_n$  a hodnoty přesného řešení jsou značeny  $y(x_0), y(x_1), \dots, y(x_n)$ .

Na obrázku 1 je plnou čarou vyznačeno přesné řešení diferenciální rovnice a kroužky přibližné hodnoty řešení.



Obrázek 1 – Přesné a přibližné řešení diferenciální rovnice

Na obrázku 1 byl použit konstantní krok  $k$  mezi jednotlivými uzly. [1]

## 1.1 Počáteční úloha

V této práci se budeme zabývat řešením obyčejné diferenciální rovnice prvního řádu se zadanou počáteční podmínkou, kterou nalezneme v následném tvaru

$$y' = f(x, y) \quad (1.1)$$

$$y(x_0) = y_0 \quad (1.2)$$

kde  $x_0$  a  $y_0$  jsou zadané hodnoty.

Příkladem diferenciální rovnice s počáteční podmínkou je např.

$$y' = -2(y - x) + 1 \quad y(0) = 1 \quad (1.3)$$

[1] [2]

## 1.2 Rungovy-Kuttovy metody

### 1.2.1 Eulerova metoda

Eulerova metoda je jedna z nejjednodušších jednokrokových metod.

Pokud v počáteční úloze (1.1) nahradíme derivaci na levé straně diferenciálem podle vzorce, dostaneme:

$$\frac{y(x_{n+1}) - y(x_n)}{h} = f(x_n, y(x_n)) \quad (1.4)$$

Pokud hodnotu  $y(x_n)$  nahradíme přibližnou hodnotou  $y_n$ , vyjádřit přibližnou hodnotu  $y(x_{n+1})$  jako

$$y_{n+1} = y_n + h f(x_n, y_n) \quad (1.5)$$

[1]

**Algoritmus Eulerovy metody**

1.  $a, b, x_0 = a$ , zvolíme  $h, n = 0, c = (b - a) / h$
2. Vypočítáme  $f(x_0, y_0)$
3. Z (2.5) dostaneme  $y_1 = y_0 + h f(x_0, y_0), x_1 = x_0 + h$
4. Je-li  $n = c \Rightarrow$  KONEC
5.  $++n$  skok na 2 [1] [3]

**Příklad výpočtu**

**Zadání:** Eulerovou metodou s krokem  $h = 0,1$  řešte počáteční úlohu

$$y' = x + y, \quad y(0) = 1$$

na intervalu  $\langle 0; 0,6 \rangle$ .

**Řešení:**

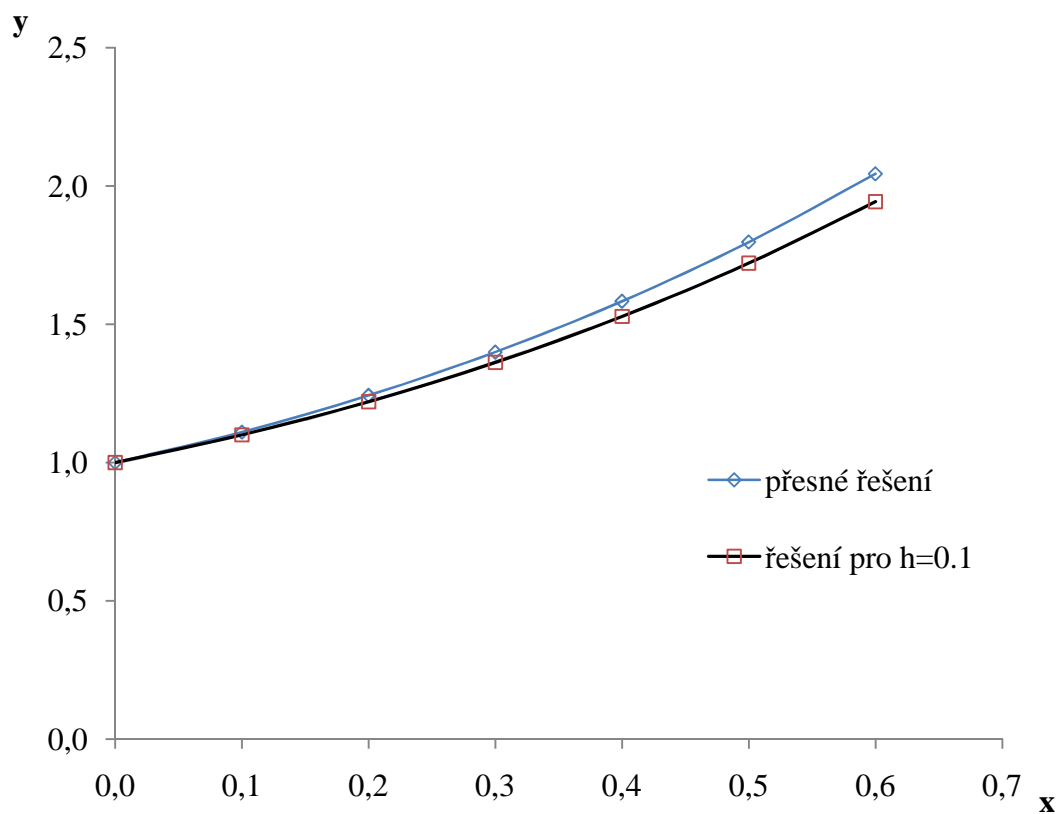
$x_0 = 0, y_0 = 1$  a  $f(x, y) = x + y$ . Podle vzorce (1.5) vypočítáme další přibližné body řešení.

$$y_{n+1} = y_n + 0,1(x_n + y_n), \quad n = 0, \dots, 4$$

Do tabulky zapíšeme vypočtené hodnoty společně s hodnotami přesného řešení  $y(x) = 2e^{-x} - x - 1$  v uzlových bodech. Z tabulky je zřejmé jak se tyto dvě řešení od sebe liší. Všechna čísla v tabulce jsou zaokrouhlena na 4 desetinná místa.

$n$	$x_n$	$y(x_n)$	$y_n$	$e_n$
0	0,0	1,00000	1,00000	0,0000
1	0,1	1,11034	1,10000	0,0103
2	0,2	1,24281	1,22000	0,0228
3	0,3	1,39972	1,36200	0,0377
4	0,4	1,58365	1,52820	0,0555
5	0,5	1,79744	1,72102	0,0764
6	0,6	2,04424	1,94312	0,1011

Tabulka 1 – Výsledky výpočtu EM



Obrázek 2 – Grafické znázornění výsledků výpočtu EM

Na obrázku 2 lze vypočítat následující vlastnosti chyby  $e$

- chyba  $e$  je úměrná kroku  $h$
- chyba  $e$  s rostoucím  $x$  vzrůstá

### 1.2.2 Modifikace Eulerovy metody

Modifikace Eulerovy metody vychází z Eulerovy metody. Při Eulerově metodě používáme při kroku z  $x_i$  do  $x_{i+1}$  konstantní hodnotu derivace  $f(x_i, y_i)$  na celém intervalu. Tato hodnota by se měla měnit. Tento fakt způsobuje, že Eulerova metoda je tak nepřesná.

První modifikace Eulerovy metody používá derivaci uprostřed intervalu na rozdíl od Eulerovy metody.

Výsledný vzorec je

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right) \\ y_{n+1} &= y_n + hk_2 \end{aligned} \quad (1.6)$$

Druhá derivace používá průměr na začátku a na konci intervalu

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + h, y_n + hk_1) \\ y_{n+1} &= y_n + \frac{1}{2}h(k_1 + k_2) \end{aligned} \quad (1.7)$$

[2]

### 1.2.3 Rungovy-Kuttovy metody

Rungovy – Kuttovy metody vycházejí z vylepšování dalších modifikací Eulerovy metody.

Při Rungově – Kuttově metodě získáváme několik odhadů derivace  $k_i$  v různých bodech.

Pro celý krok potom použijeme vážený průměr těchto derivací s váhami  $w_i$  tj.

$$y_{n+1} = y_n + h(\omega_1 k_1 + \dots + \omega_s k_s) \quad (1.8)$$

kde

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_i &= f\left(x_n + \alpha_i h, y_n + h \sum_{j=1}^{i-1} \beta_{ij} k_j\right), \quad i = 2, \dots, s \end{aligned} \quad (1.9)$$

a  $\omega_i, \alpha_i$  a  $\beta_{ij}$  jsou konstanty volené tak, aby metoda měla maximální řád.

U první modifikované Eulerovy metody bylo  $\omega_1 = 0, \omega_2 = 1, \alpha_2 = 1/2$  a  $\beta_{21} = 1/2$ , u druhé modifikace  $\omega_1 = \omega_2 = 1/2, \alpha_2 = 1$  a  $\beta_{21} = 1$ .

Nejproslulejší je následující metoda Runge-Kutta 4. řádu. Často, mluví-li se o Rungově-Kuttově metodě, myslí se tím právě tato konkrétní metoda.

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + 1/2h, y_n + 1/2hk_1) \\ k_3 &= f(x_n + 1/2h, y_n + 1/2hk_2) \\ k_4 &= f(x_n + h, y_n + hk_3) \\ y_{n+1} &= y_n + 1/6h(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (1.10)$$

[1] [2]

### Příklad výpočtu

**Zadání:** Rungovou-Kuttovou metodou řešte počáteční úlohu

$$y' = x^2 - y, \quad y(0) = 1$$

s krokem  $h = 0,1$  na intervalu  $\langle 0; 0,5 \rangle$ .

**Řešení:** První krok metody předvedeme podobně, výsledky dalších kroků pouze zapíšeme do tabulky.

Známe  $x_0 = 0, y_0 = 1$  budeme počítat  $y_1$ , tj. přibližnou hodnotu řešení v bodě  $x_1 = 0,1$ .

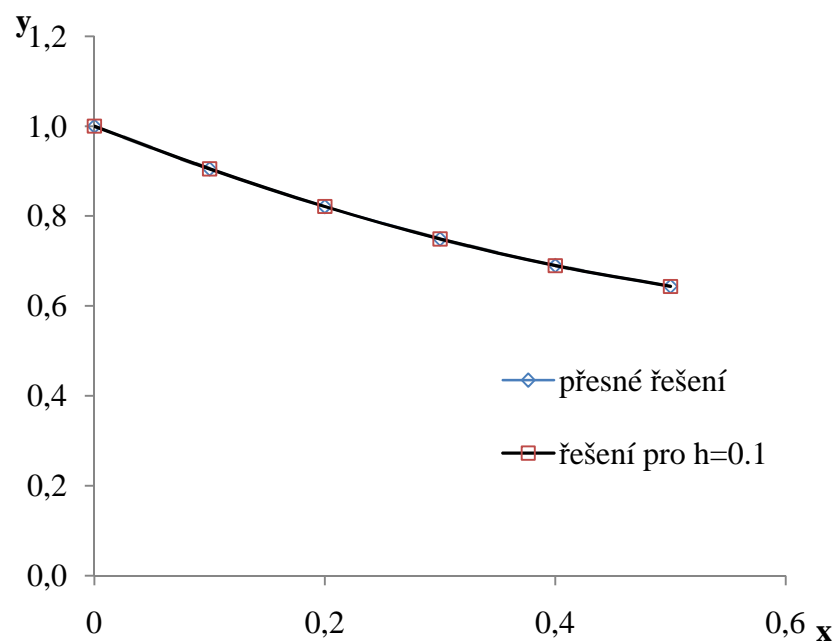
K tomu potřebujeme  $k_1, k_2, k_3, k_4$ . Ta vypočteme podle vzorců (2.10)

$$\begin{aligned} k_1 &= f(0;1) = 0^2 - 1 = -1 \\ k_2 &= f(0 + 1/2 \cdot 0,1; 1 + 1/2 \cdot 0,1 \cdot (-1)) = f(0,05; 0,95) = -0,9475 \\ k_3 &= f(0 + 1/2 \cdot 0,1; 1 + 1/2 \cdot 0,1 \cdot (-0,9475)) = f(0,1; 0,952625) = -0,950125 \\ k_4 &= f(0 + 0,1; 1 + 0,1 \cdot (-0,950125)) = f(0,1; 0,9049875) = -0,8949875 \\ y_1 &= y_0 + 1/6 \cdot 0,1 \cdot (k_1 + 2k_2 + 2k_3 + k_4) = 0,9051627 \end{aligned}$$

V každém dalším kroku budeme opět počítat čísla  $k_1, k_2, k_3$  a  $k_4$ . Pomocí nich pak přibližnou hodnotu řešení v dalším uzlovém bodě. V tabulce jsou pro srovnání vypsány hodnoty přibližného řešení i hodnoty přesného řešení  $y = -e^{-x} + x^2 - 2x + 2$ . Tentokrát jsou čísla zaokrouhlena na 7 desetinných míst.

$n$	$x_n$	$y_n$	$y(x_n)$	$x$	$y$	
0	0	1	1	0	1	$k_1=-1$
				0,05	0,95	$k_2=-0,9475$
				0,05	0,95263	$k_3=-0,950125$
				0,1	0,9049875	$k_4=-0,8949875$
1	0,1	0,9051627	0,9051626	0,1	0,9051627	$k_1=-0,8951627$
				0,15	0,8604046	$k_2=-0,8379046$
				0,15	0,8632675	$k_3=-0,8407675$
				0,2	0,8210860	$k_4=-0,7810860$
2	0,2	0,8212695	0,8212693	0,2	0,8212695	$k_1=-0,7812695$
				0,25	0,7822060	$k_2=-0,7197060$
				0,25	0,7852842	$k_3=-0,7227842$
				0,3	0,7489911	$k_4=-0,6589911$
3	0,3	0,7491822	0,7491818	0,3	0,7491822	$k_1=-0,6591822$
				0,35	0,7162230	$k_2=-0,5937230$
				0,35	0,7194960	$k_3=-0,5969960$
				0,4	0,6894826	$k_4=-0,5294826$
4	0,4	0,6896804	0,6896800	0,4	0,6896804	$k_1=-0,5296804$
				0,45	0,6631964	$k_2=-0,4606964$
				0,45	0,6666456	$k_3=-0,4641456$
				0,5	0,6432659	$k_4=-0,3932659$
5	0,5	0,6434699	0,6434693			

Tabulka 2 – Výsledky výpočtu R-KM



Obrázek 3 – Grafické znázornění výsledků výpočtu R-KM

## 2 NUMERICKÉ ŘEŠENÍ SOUSTAV DIFERENCIÁLNÍCH ROVNIC

Všechny metody z předešlé kapitoly lze jednoduše zobecnit i na soustavu diferenciálních rovnic. V případě zadání soustavy dvou rovnic pro dvě neznáme funkce  $x(t)$ ,  $y(t)$

$$x' = f(t, x, y) \quad x(t_0) = x_0 \quad (2.1)$$

$$y' = g(t, x, y) \quad y(t_0) = y_0 \quad (2.2)$$

můžeme pomocí numerické metody získat alespoň přibližné hodnoty řešení  $x_i$ ,  $y_i$ .

Diferenciální rovnici druhého řádu lze zavedením další proměnné převést na soustavu rovnic prvního řádu.

Například pohybovou rovnicí

$$r'' = f(t, r, r') \quad (2.3)$$

lze zavedením rychlosti převést na soustavu

$$r' = v \quad (2.4)$$

$$r' = f(t, r, v) \quad (2.5)$$

a tuto soustavu již můžeme řešit nějakou z dále uvedených metod. [2] [4]

### 2.1 Eulerova metoda

Eulerova metoda je nejjednodušší numerickou jednokrokovou metodou pro řešení soustavy diferenciálních rovnic. Eulerova metoda pro soustavu diferenciálních rovnic vyplývá z Eulerovy metody pro řešení jedné diferenciální rovnice. Jeden krok Eulerovy metody pro soustavu diferenciálních rovnic.

$$t_{n+1} = t_0 + (n+1)h, \quad (2.6)$$

$$x_{n+1} = x_n + hf(t_n, x_n, y_n), \quad (2.7)$$

$$y_{n+1} = y_n + hg(t_n, x_n, y_n), \quad (2.8)$$

kde  $h = \frac{b-a}{N}$  značí krok metody,  $n = 0, 1, 2, \dots, N-1$ . [4]

**Příklad:** Pomocí Eulerovy metody řešte danou soustavu diferenciálních rovnic  $x' = 2 - e^y$ ,  $y' = x$  s počáteční podmínkou  $x(0) = 1$ ,  $y(0) = 0$  na  $\langle 0; 0, 4 \rangle$  s krokem  $h = 0, 1$ .

**Řešení:** Pro  $h = 0,1$  v čísle  $t_1 = 0,1$  máme  $t_0 = 0$ ,  $x_0 = 0$ ,  $y_0 = 0$ . Proto platí

$$\begin{aligned}x_1 &= x_0 + h f(t_0, x_0, y_0) = x_0 + h \cdot (2 - e^{y_0}) = 1 + 0,1 \cdot (2 - 1) = 1,1 \\y_1 &= y_0 + h g(t_0, x_0, y_0) = y_0 + h \cdot x_0 = 0 + 0,1 \cdot 1 = 0,1\end{aligned}$$

Opakováním tohoto postupu najdeme hodnotu v pravé části Tabulky 3.

Celkový výpočet je uspořádán do tabulky:

$n$	$t_n$	$x_n$	$y_n$
0	0,0	1,00000	0,000000
1	0,1	1,10000	0,100000
2	0,2	1,18948	0,210000
3	0,3	1,26612	0,328948
4	0,4	1,32716	0,455560

Tabulka 3 – Výsledky výpočtu EM pro soustavu

## 2.2 Rungovy-Kuttovy metody

Tato metoda je zobecněním Rungovy-Kuttovy metody pro jednu diferenciální rovnici.

$$\begin{aligned}k_1 &= f(x_n, y_n) \\k_2 &= f(x_n + h/2, y_n + h/2 \cdot k_1) \\k_3 &= f(x_n + h/2, y_n + h/2 \cdot k_2) \\k_4 &= f(x_n + h, y_n + h \cdot k_3)\end{aligned}\tag{2.9}$$

přičemž

$$k = 1/6 \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)\tag{2.10}$$

$$y_{n+1} = y_n + hk\tag{2.11}$$

Pro soustavu má jeden krok Rungovy-Kuttovy metody tvar

$$\begin{aligned}k_1 &= f(t_n, x_n, y_n) \\l_1 &= g(t_n, x_n, y_n) \\k_2 &= f(t_n + h/2, x_n + h/2 \cdot k_1, y_n + h/2 \cdot l_1) \\l_2 &= g(t_n + h/2, x_n + h/2 \cdot k_1, y_n + h/2 \cdot l_1) \\k_3 &= f(t_n + h/2, x_n + h/2 \cdot k_2, y_n + h/2 \cdot l_2) \\l_3 &= g(t_n + h/2, x_n + h/2 \cdot k_2, y_n + h/2 \cdot l_2) \\k_4 &= f(t_n + h, x_n + h \cdot k_3, y_n + h \cdot l_3) \\l_4 &= g(t_n + h, x_n + h \cdot k_3, y_n + h \cdot l_3)\end{aligned}\tag{2.12}$$

přičemž

$$k = 1/6 \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \quad (2.13)$$

$$l = 1/6 \cdot (l_1 + 2 \cdot l_2 + 2 \cdot l_3 + l_4) \quad (2.14)$$

$$x_{n+1} = x_n + hk \quad (2.15)$$

$$y_{n+1} = y_n + hl \quad (2.16)$$

[5]

**Příklad:** Řešte danou soustavu diferenciálních rovnic  $x' = 2 - e^y$ ,  $y' = x$  s počátečními podmínkami  $x(0) = 1$ ,  $y(0) = 0$  a krokem  $h = 0,1$  na intervalu  $\langle 0; 0,4 \rangle$

**Řešení:** Vypočítejme  $x_1, y_1$  pro  $t_1 = 0,2$

$$k_1 = f(t_0, x_0, y_0) = 1$$

$$l_1 = g(t_0, x_0, y_0) = 1$$

$$k_2 = f(t_0 + h/2, x_0 + h/2 \cdot k_1, y_0 + h/2 \cdot l_1) = 0,948729$$

$$l_2 = g(t_0 + h/2, x_0 + h/2 \cdot k_1, y_0 + h/2 \cdot l_1) = 1,05$$

$$k_3 = f(t_0 + h/2, x_0 + h/2 \cdot k_2, y_0 + h/2 \cdot l_2) = 0,953459$$

$$l_3 = g(t_0 + h/2, x_0 + h/2 \cdot k_2, y_0 + h/2 \cdot l_2) = 1,04744$$

$$k_4 = f(t_0 + h, x_0 + h \cdot k_3, y_0 + h \cdot l_3) = 0,889574$$

$$l_4 = g(t_0 + h, x_0 + h \cdot k_3, y_0 + h \cdot l_3) = 1,09435$$

$$k = 1/6 \cdot (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) = 1/6 \cdot (5,67395) = 0,945658$$

$$l = 1/6 \cdot (l_1 + 2 \cdot l_2 + 2 \cdot l_3 + l_4) = 1/6 \cdot (6,28923) = 1,0482$$

$$x_1 = x_0 + hk = 0 + 0,1 \cdot 0,945658 = 1,09457$$

$$y_1 = y_0 + hl = 0 + 0,1 \cdot 1,0482 = 0,10482$$

Celkový výpočet je uspořádán do tabulky:

$n$	$t_n$	$x_n$	$y_n$
0	0	1,00000	0,00000
1	0,1	1,09457	0,10482
2	0,2	1,17695	0,218507
3	0,3	1,24466	0,339721
4	0,4	1,29487	0,466857

Tabulka 4 – Výsledky výpočtu R-KM pro soustavy

## 3 NUMERICKÉ METODY

### 3.1 Chyby při numerických výpočtech

Při řešení numerických metod získáváme přibližné výsledky, které nám způsobí, že mezi přesným řešením úlohy a řešením získaným numerickou metodou nalezneme vždy nějaký rozdíl. [1]

#### 3.1.1 Zdroje a typy chyb

Základní druhy několika chyb:

- **chyby matematického modelu** – vzniká při přepsání daného procesu na matematický model. Např. přepsání fyzikálního děje na diferenciální rovnici nebo soustavu diferenciálních rovnic.
- **chyby vstupních dat** – vznikají při nepřesném měření a zadávání vstupních veličin.
- **chyby numerické metody** – nazýváme je zdroj chyb, které dostáváme při aproximaci matematické úlohy numerickou úlohou a následném řešení. Při řešení numerické metody by měl být odhad chyby součástí. V mnoha případech je odhad chyby složitější než samotné řešení numerické metody.
- **chyby zaokrouhlování** – vznikají zaokrouhlováním čísla během výpočtu. Číslo se vždy zaokrouhluje na jiný počet míst a tím jsou chyby zaokrouhlení odlišné. To způsobí, že se mohou kumulovat, nebo rušit. [1] [5]

### 3.1.2 Definice chyb

Absolutní chybu aproximace nazýváme rozdíl mezi přesnou hodnotou čísla  $\hat{x}$  a její aproximací  $x$ . Vyjadřuje se v procentech.

$$E(x) = \hat{x} - x$$

Relativní chybu aproximace užíváme, pokud je přesná hodnota velmi malá nebo velmi velká. Relativní chyba aproximace se taky vyjadřuje v procentech.

$$RE(x) = \frac{\hat{x} - x}{x},$$

Odhad absolutní chyby aproximace  $x$  nebo mezní absolutní chybu nazýváme každé nezáporné číslo  $ME(x)$ , pro které platí:

$$|\hat{x} - x| \leq ME(x), \text{ tj. } \hat{x} \in \langle x - ME(x), x + ME(x) \rangle$$

Odhad relativní chyby nazýváme každé nezáporné číslo  $MR(x)$ , pro které platí:

$$\frac{|\hat{x} - x|}{|x|} \leq MR(x), \quad x \neq 0$$

Většinou užívané zápisy:

$$\hat{x} = x \pm ME(x), \text{ resp. } \hat{x} = x(1 \pm MR(x)).$$

[1]

### 3.1.3 Zaokrouhlování. Šíření chyb při výpočtu

Je-li  $x$  reálné číslo, které má obecně nekonečné dekadické vyjádření, pak číslo  $x^{(d)}$ , které má  $d$  desetinných míst, je správně zaokrouhlenou hodnotou čísla  $x$ , platí-li

$$|x - x^{(d)}| \leq \frac{1}{2} 10^{-d}$$

Tady např. má-li být  $x^{(1)}$  správně zaokrouhlená hodnota čísla  $x$  na jedno desetinné místo, nesmí se od  $x$  lišit o více než o  $1/2 \cdot 10^{-1} = 0,05$ .

Při numerických výpočtech pracujeme se zaokrouhlenými čísly. Výsledky početních operací s těmito čísly jsou opět zaokrouhlovány a dále se s nimi pracuje. Tím se zaokrouhlovací chyby šíří. [1]

### 3.1.4 Podmíněnost numerických úloh a numerická stabilita algoritmů

Zaokrouhlování během výpočtu a malé změny ve vstupních hodnotách mají vliv na výsledek při řešení numerických metod. Jedná se tak o přiřazování vstupních údajů výstupním datům. Korektní úlohu nazýváme, má-li toto přiřazení spojité zobrazení. V opačném případě se jedná o úlohu nekorektní.

Způsobí-li malé změny vstupních údajů malé změny na výstupních údajích, mluvíme o dobře podmíněné korektní úloze.

Způsobí-li malé změny vstupních údajů velké změny na výstupních údajích, mluvíme o špatně podmíněné úloze.

Číslem podmíněnosti úloh  $C_p$  charakterizuje podmíněnost korektních úloh a vypočítá se podílem relativní chyby výstupních údajů a relativní chyby vstupních údajů.

$$C_p = \frac{\text{relativní chyba výstupních údajů}}{\text{relativní chyba vstupních údajů}}$$

Pokud je číslo  $C_p$  blízké číslu 1 jedná se o dobře podmíněnou úlohu.

Je-li algoritmus málo citlivý na poruchy ve vstupních datech, mluvíme o dobře podmíněném algoritmu. Pokud je algoritmus numericky stabilní, neovlivňuje výsledek použitého algoritmu ani zaokrouhlování. [1]

## **II. PRAKTICKÁ ČÁST**

## 4 POPIS PROGRAMU

# Numerické řešení soustav diferenciálních rovnic

---

### Zadání vstupních parametrů

---

### Metody výpočtu

---

### Zobrazení výsledů

---

### Porovnání metod

Obrázek 4 – Úvodní obrazovka programu

Program se dělí na čtyři hlavní části, pro lepší orientaci a ovládání programu:

### 4.1 Zadání vstupních parametrů

Vstupní parametry jsou chápány jako diferenciální rovnice, počáteční podmínky, interval výpočtu nebo krok výpočtu.

---

### Zadání vstupních parametrů

- Nulování jednotlivých proměnných
- Definování konstant
- Vstupní hodnoty
- Vstupní diferenciální rovnice
- Zadání kroku výpočtu
- Zadání intervalu výpočtu
- Zadání počátečních podmínek
- Pomocné proměnné

Obrázek 5 – Část pro zadání vstupních parametrů

### 4.2 Metody výpočtu

Program zde nabízí výběr ze čtyř numerických metod.

---

### Metody výpočtu

- Eulerova metoda
- Eulerova metoda - První modifikace
- Eulerova metoda - Druhá modifikace
- Rungova – Kuttova metoda 4. řádu

Obrázek 6 – Část pro výběr metody výpočtu

### 4.3 Zobrazení výsledků

Program nabízí dva typy zobrazení. Prvním typ vypíše výsledek pomocí tabulky a druhý typ nám vykreslí řešení pomocí grafické formy.

---

#### Zobrazení výsledů

- Eulerova metoda
- Eulerova metoda - První modifikace
- Eulerova metoda - Druhá modifikace
- Rungova - Kuttova metoda 4. řádu

Obrázek 7 – Část pro výběr zobrazení výsledků

### 4.4 Porovnání výsledků jednotlivých metod.

Program jednotlivé výsledky řešení jednotlivých metod zobrazí v tabulce. Z tabulky je patrné, která z jednotlivých metod je přesnější a která méně přesná.

---

#### Porovnání metod

Obrázek 8 – Část pro porovnání výsledků jednotlivých metod

## 5 ŘEŠENÍ SOUSTAVY DVOU DIFERENCIÁLNÍCH ROVNIC

**Zadání:** Najděme řešení systému diferenciálních rovnic

$$\begin{aligned}m'_1 &= n_2 \\ n'_2 &= 1 + e^{m_1}\end{aligned}$$

s počátečními podmínkami  $m(0) = 0, n(0) = 0$  s krokem  $h = 0,1$  na intervalu  $\langle 0;0,4 \rangle$ .

### 5.1 Zadání vstupních parametrů

Program se v této části zabývá zadáváním vstupních parametrů. V této části programu můžeme zadat jak diferenciální rovnice, počáteční podmínky tak krok výpočtu. Více si popíšeme dále.

#### 5.1.1 Nulování jednotlivých proměnných:

Zde můžeme vynulovat veškeré proměnné, které se v programu vyskytují. Vynulování proměnných se provádí, aby při opakovaném výpočtu nedocházelo ke změně jednotlivých hodnot proměnných.

##### ■ Nulování jednotlivých proměnných

```
Clear[f, krok, DolniInterval, HorniInterval, x, y, y10, y20, k1, k2, k3, k4];
```

#### 5.1.2 Definování konstant

Můžeme si zde nadefinovat veškeré konstanty, které v programu budete používat.

##### ■ Definování konstant

```
konstanta1 = 0;  
konstanta2 = 0;
```

#### 5.1.3 Vstupní hodnoty

Zde si můžeme nadefinovat různé hodnoty vstupů, které dále v programu budeme používat.

##### ■ Vstupní hodnoty

```
Vstup1 = 0;  
Vstup2 = 0;
```

### 5.1.4 Vstupní diferenciální rovnice

Tato část programu nám slouží pro zadání diferenciálních rovnic. Program umí pracovat jen s proměnnými  $x, y_1, y_2$ . Veškerá jiná značení program nepodporuje. Diferenciální rovnice ze zadání  $m' = n, n' = 1 + e^m$  musíme přepsat, jinak by program nefungoval. Přepis provedeme následujícím způsobem: proměnné  $m$  odpovídá proměnná  $y_1$  a proměnné  $n$  odpovídá  $y_2$ . Přepsaný tvar diferenciálních rovnic  $y_1' = y_2, y_2' = 1 + e^{y_1}$ , který program podporuje.

#### ■ Vstupní diferenciální rovnice

```
y1' = y2;
y2' = 1 + Exp[y1];
```

### 5.1.5 Zadání kroku výpočtu:

Krok výpočtu se v programu definuje jednoduše. Do proměnné `krok` uložíme hodnotu, která udává vzdálenost mezi jednotlivými body řešení. Pokud hodnota kroku je desetinné číslo, zadává se do programu jako 0.1 ne jako 0,1.

#### ■ Zadání kroku výpočtu

```
krok = 0.1;
```

### 5.1.6 Zadání intervalu výpočtu

Vezněme interval výpočtu ze zadání  $\langle 0; 0,4 \rangle$ . Dolní interval se bude rovnat hodnotě 0. Horní interval se bude rovnat hodnotě 0,4.

#### ■ Zadání intervalu výpočtu

```
DolniInterval = 0;
HorniInterval = 0.4;
```

### 5.1.7 Zadání počátečních podmínek

V zadání jsme měli následné počáteční podmínky  $m(0) = 0, n(0) = 0$ . Nejprve musíme počáteční podmínky přepsat do tvaru, které program podporuje. To znamená, že proměnné  $m$  a  $n$  musíme přepřát do formy  $y_1$  a  $y_2$ . Po přepsání počáteční podmínky mají tvar  $y_1(0) = 0, y_2(0) = 0$ . Počáteční podmínce  $y_1(0) = 0$  v programu odpovídá proměnná `y10` a počáteční podmínce  $y_2(0) = 0$  odpovídá proměnná `y20`.

- Zadání počátečních podmínek

```
y10 = 0;
```

```
y20 = 0;
```

### 5.1.8 Pomocné proměnné

Tato část programu slouží pro pomocné proměnné, které budeme dále v programu používat. Např. Uložení jednotlivých diferenciálních rovnic do pole. Určení počáteční podmínky pro proměnnou  $x$ .

- Pomocné proměnné

```
f[0] = y1 ' ;
```

```
f[1] = y2 ' ;
```

```
PocetKroku = (HorniInterval - DolniInterval) / krok;
```

```
x[0] = DolniInterval;
```

## 5.2 Zobrazení výsledků

Výsledky můžeme zobrazovat dvěma způsoby.

První způsob je zobrazení výsledků řešení pomocí tabulky. Výsledná tabulka obsahuje počet kroků, hodnoty jednotlivých kroků a výsledky v jednotlivých krocích výpočtu. Hodnoty kroků jsou představovány proměnnou  $x$  a výsledky v jednotlivých krocích výpočtu jsou představovány hodnotou  $Y[1]$  a  $Y[2]$ .

Druhý způsob zobrazení výsledků řešení je pomocí grafu. Program nám vykreslí graf, ve kterém dostáváme průběh řešení dané numerické metody.

Zde si zobrazíme výsledky jednotlivých metod, které jsme aplikovali na náš příklad.

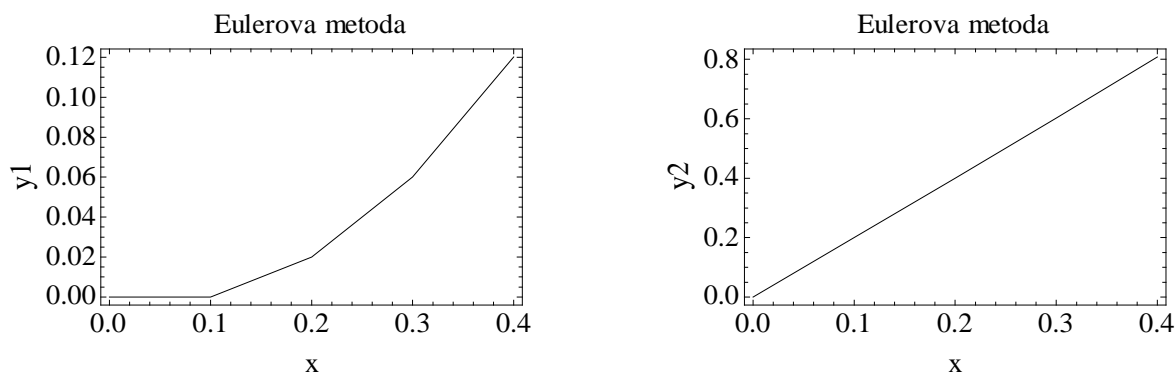
### 5.2.1 Zobrazení výsledků pomocí Eulerovy metody

- Pomocí tabulky

n	x	Y[1]	Y[2]
0	0	0	0
1	0.1	0	0.2
2	0.2	0.02	0.4
3	0.3	0.06	0.60202
4	0.4	0.120202	0.808204

Tabulka 5 – Výsledky výpočtu EM

- Pomocí grafu



Obrázek 9 – Grafické znázornění výsledků výpočtu EM

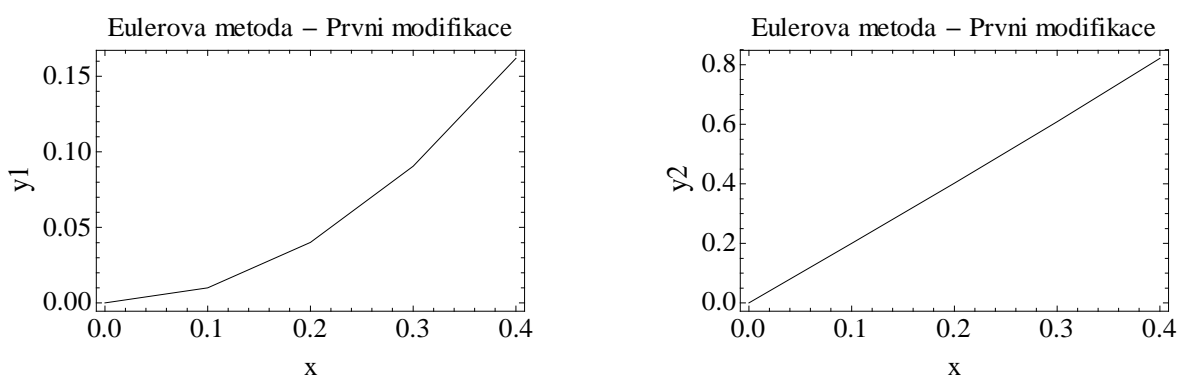
### 5.2.2 Zobrazení výsledků pomocí Eulerovy metody – První modifikace

- Pomocí tabulky

n	x	Y[1]	Y[2]
0	0	0	0
1	0.1	0.01	0.2
2	0.2	0.0400503	0.40202
3	0.3	0.0904571	0.60822
4	0.4	0.161753	0.821067

Tabulka 6 – Výsledky výpočtu EMP

- Pomocí grafu



Obrázek 10 – Grafické znázornění výsledků výpočtu EMP

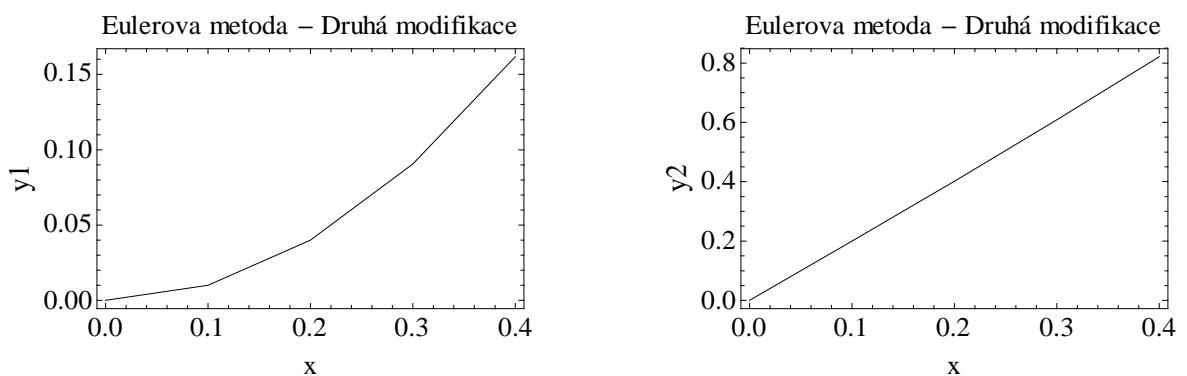
### 5.2.3 Zobrazení výsledků pomocí Eulerovy metody – Druhá modifikace

- Pomocí tabulky

n	x	Y[1]	Y[2]
0	0	0	0
1	0.1	0.01	0.2
2	0.2	0.0400503	0.402025
3	0.3	0.0904571	0.608246
4	0.4	0.161755	0.821146

Tabulka 7 – Výsledky výpočtu EMD

- Pomocí grafu



Obrázek 11 – Grafické znázornění výsledků výpočtu EMD

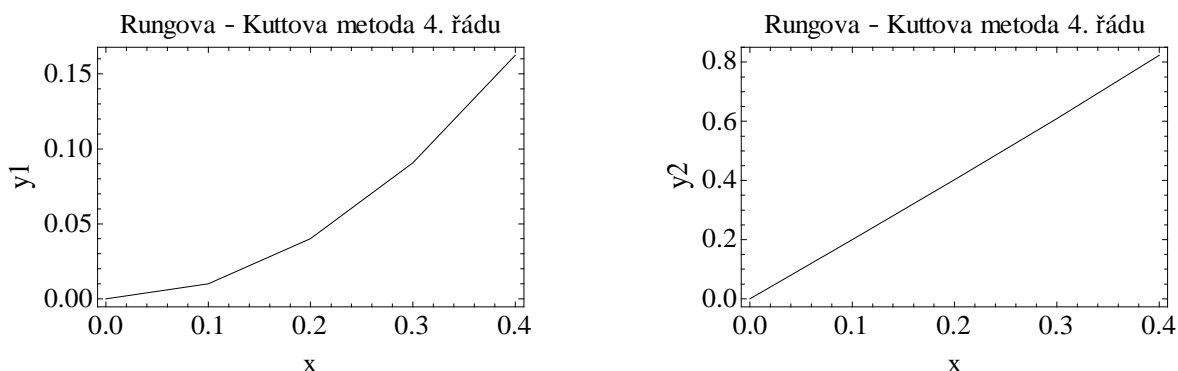
### 5.2.4 Zobrazení výsledků pomocí Runge – Kuttovy metody 4 řádu

- Pomocí tabulky

n	x	Y[1]	Y[2]
0	0	0	0
1	0.1	0.0100084	0.200335
2	0.2	0.0401516	0.402706
3	0.3	0.0907422	0.609297
4	0.4	0.162327	0.822609

Tabulka 8 – Výsledky výpočtu R-KM

- Pomocí grafu



Obrázek 12 – Grafické znázornění výsledků výpočtu R-KM

### 5.3 Porovnání metod

Tato část programu slouží pro porovnání výsledků jednotlivých numerických metod.

n	x	Y1 – E	Y1 – EP	Y1 – ED	Y1 – R	Y2 – E	Y2 – EP	Y2 – ED	Y2 – R
0	0	0	0	0	0	0	0	0	0
1	0.1	0	0.01	0.01	0.0100084	0.2	0.2	0.2	0.200335
2	0.2	0.02	0.0400503	0.0400503	0.0401516	0.4	0.40202	0.402025	0.402706
3	0.3	0.06	0.0904571	0.0904571	0.0907422	0.60202	0.60822	0.608246	0.609297
4	0.4	0.120202	0.161753	0.161755	0.162327	0.808204	0.821067	0.821146	0.822609

Tabulka 9 – Porovnání výsledky výpočtu jednotlivých metod

Již podle této tabulky můžeme usoudit, že výsledky jednotlivých metod se od sebe liší. Nejpresnější metodou je Runge – Kuttova metoda 4. řádu. Nejméně přesnou metodou s používaných metod je Eulerova metoda.

## 6 APLIKACE PROGRAMU NA NÁZORNÝ PŘÍKLAD

Zadání: Řešte soustavu dvou obyčejných diferenciálních rovnic, které popisují matematický model průtočného chemického reaktoru (CSTR).

Doba simulace bude 30 min a simulační krok 0,1 min.

### 6.1 Zadání vstupních parametrů

#### 6.1.1 Definování konstant

$$V = 100 \text{ l}$$

$$k_0 = 7,2 \cdot 10^{10} \text{ min}^{-1}$$

$$E/R = 1 \cdot 10^4 \text{ K}$$

$$T_0 = 350 \text{ K}$$

$$T_{c0} = 350 \text{ K}$$

$$\Delta H = -2 \cdot 10^5 \text{ cal} \cdot \text{mol}^{-1}$$

$$c_p = 1 \text{ cal} \cdot \text{g}^{-1} \cdot \text{K}^{-1}$$

$$c_{pc} = 1 \text{ cal} \cdot \text{g}^{-1} \cdot \text{K}^{-1}$$

$$\rho = 1 \cdot 10^3 \text{ g} \cdot \text{l}^{-1}$$

$$\rho_c = 1 \cdot 10^3 \text{ g} \cdot \text{l}^{-1}$$

$$c_{A0} = 1 \text{ mol} \cdot \text{l}^{-1}$$

$$h_a = 7 \cdot 10^5 \text{ cal} \cdot \text{min}^{-1} \cdot \text{K}^{-1}$$

Přepis zadaných hodnoty ze zadání do programu.

#### ■ Definování konstant

```
V = 100;
k0 = 7.2 * 10^10;
E1 = 1 * 10^4;
T0 = 350;
Tc0 = 350;
H = -2 * 10^5;
cp = 1;
cpc = 1;
ro = 1 * 10^3;
roc = 1 * 10^3;
cA0 = 1;
ha = 7 * 10^5;
```

### 6.1.2 Vstupní hodnoty

$$q = 100$$

$$q_c = 80$$

Kde  $q_1$  a  $q_{1c}$  se vypočítají ze vztahu:

$$q_1 = q + \frac{U_1 q}{100}$$

$$q_{1c} = q_c + \frac{U_2 q_c}{100}$$

kde  $U_1$  a  $U_2$  představují změny vstupní veličiny v procentech.

Přepsání vstupních hodnot ze zadání do programu.

#### ■ Vstupní hodnoty

$$q = 100;$$

$$q_c = 80;$$

$$U_1 = 0;$$

$$U_2 = 60;$$

$$q_1 = q + (q * U_1) / 100;$$

$$q_{1c} = q_c + (q_c * U_2) / 100;$$

### 6.1.3 Vstupní diferenciální rovnice

$$y_1' = \frac{q_1}{V} (T_0 - y_1) + \frac{-\Delta H}{\rho \cdot c_p} \cdot k_0 \cdot e^{\frac{-E}{RT}} \cdot y_2 + \frac{\rho_c \cdot c_{pc}}{\rho \cdot c_p \cdot V} \cdot q_{1c} \cdot \left(1 - e^{\frac{-h_a}{\rho_c \cdot c_{pc} \cdot q_{1c}}}\right) \cdot (T_{c0} - y_1)$$

$$y_2' = \frac{q_1}{V} (c_{A0} - y_2) - k_0 \cdot e^{\frac{-E}{RT}} \cdot y_2$$

Přepsání diferenciálních rovnic ze zadání do programu:

#### ■ Vstupní diferenciální rovnice

$$y1' = ((q1 / V) * (T0 - y1)) + ((-H / (ro * cp)) * k0 * \text{Exp}[-E1 / y1] * y2) +$$

$$(((roc * cpc) / (ro * cp * V)) * q1c * (1 - \text{Exp}[-ha / (roc * cpc * q1c)]) * (Tc0 - y1));$$

$$y2' = ((q1 / V) * (cA0 - y2)) - k0 * \text{Exp}[-E1 / y1] * y2;$$

### 6.1.4 Zadání kroku výpočtu

Simulační krok 0,1 min.

#### ■ Zadání kroku výpočtu

$$\text{krok} = 0.1;$$

### 6.1.5 Zadání intervalu výpočtu

Doba simulace 30 min. S toho vyplývá, že se budeme pohybovat na intervalu  $\langle 0, 30 \rangle$ .

#### ■ Zadání intervalu výpočtu

```
DolniInterval = 0;  
HorniInterval = 30;
```

### 6.1.6 Zadání počátečních podmínek

$$y_1(0) = 354,26 \quad ; \quad y_2(0) = 0,9620$$

Přepsání počátečních podmínek do programu

#### ■ Zadání počátečních podmínek

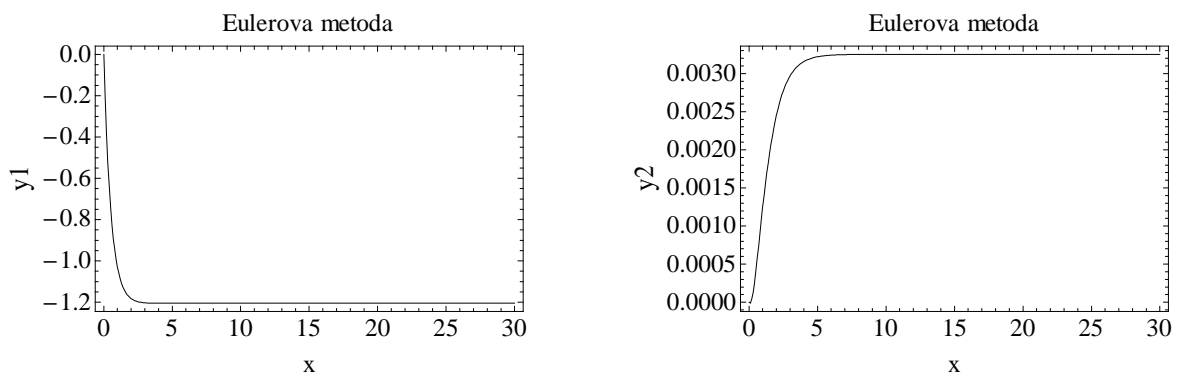
```
y10 = 354.26;  
y20 = 0.9620;
```

## 6.2 Zobrazení výsledků

Zobrazené výsledky odpovídají vstupním hodnotám  $U_1=0\%$  a  $U_2=60\%$

### 6.2.1 Zobrazení výsledků pomocí Eulerovy metody

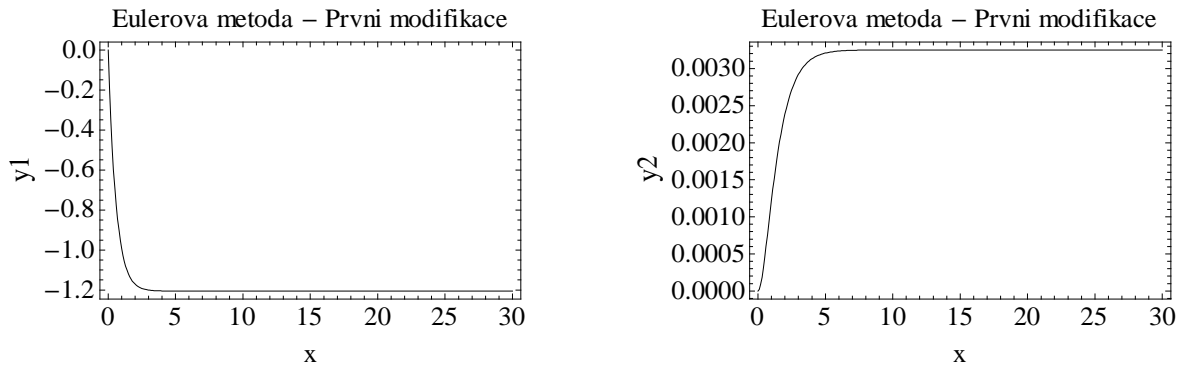
- Pomocí grafu



Obrázek 13 – Grafické znázornění výsledků výpočtu EM

### 6.2.2 Zobrazení výsledků pomocí Eulerovy metody – První modifikace

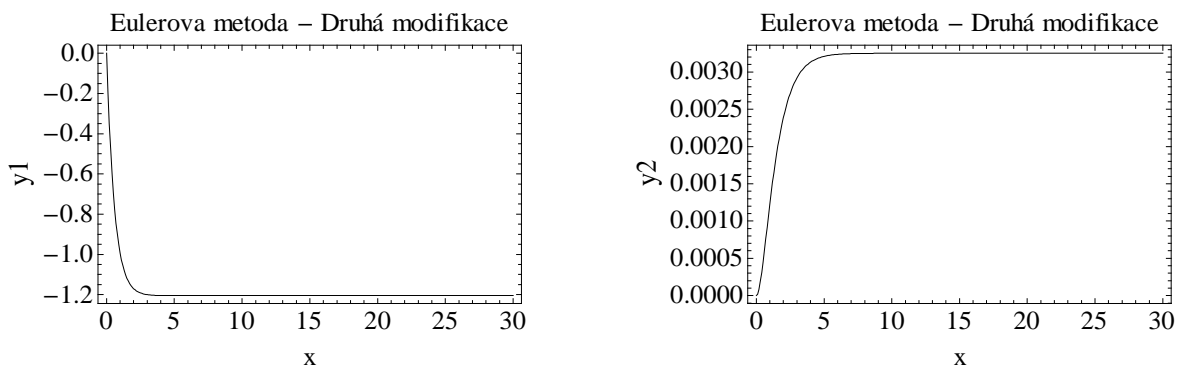
- Pomocí grafu



Obrázek 14 – Grafické znázornění výsledků výpočtu EMP

### 6.2.3 Zobrazení výsledků pomocí Eulerovy metody – Druhá modifikace

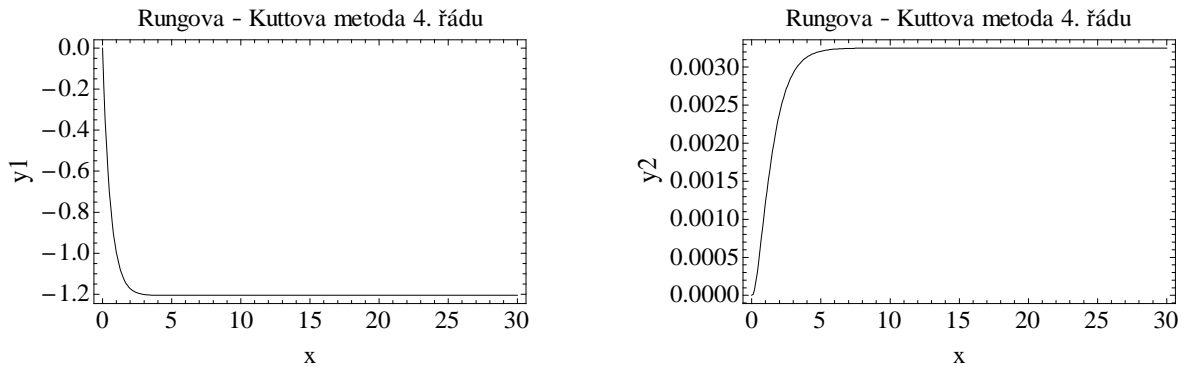
- Pomocí grafu



Obrázek 15 – Grafické znázornění výsledků výpočtu EMD

### 6.2.4 Zobrazení výsledků pomocí Rungove – Kuttově metody 4 řádu

- Pomocí grafu



Obrázek 16 – Grafické znázornění výsledků výpočtu R-KM

### 6.3 Porovnání metod

Zde je přehled prvních 20 korekčních jednotlivých numerických metod.

n	x	Y1 - E	Y1 - EP	Y1 - ED	Y1 - R	Y2 - E	Y2 - EP	Y2 - ED	Y2 - R
0	0	354.26	354.26	354.26	354.26	0.962	0.962	0.962	0.962
1	0.1	354.054	354.071	354.071	354.07	0.961987	0.962019	0.962018	0.962016
2	0.2	353.882	353.911	353.911	353.909	0.962037	0.962084	0.962084	0.96208
3	0.3	353.739	353.776	353.776	353.774	0.962133	0.962184	0.962184	0.962179
4	0.4	353.62	353.661	353.661	353.659	0.962261	0.962309	0.962309	0.962303
5	0.5	353.522	353.565	353.565	353.562	0.962411	0.96245	0.96245	0.962444
6	0.6	353.44	353.483	353.483	353.481	0.962573	0.962601	0.962601	0.962596
7	0.7	353.372	353.415	353.415	353.412	0.962742	0.962758	0.962757	0.962753
8	0.8	353.316	353.357	353.357	353.355	0.962913	0.962915	0.962915	0.962911
9	0.9	353.27	353.308	353.308	353.306	0.963082	0.963072	0.963072	0.963068
10	1.	353.232	353.267	353.267	353.265	0.963246	0.963225	0.963225	0.963222
11	1.1	353.2	353.233	353.233	353.231	0.963405	0.963373	0.963373	0.963371
12	1.2	353.174	353.204	353.204	353.202	0.963556	0.963515	0.963515	0.963513
13	1.3	353.152	353.179	353.179	353.178	0.963698	0.96365	0.96365	0.963648
14	1.4	353.134	353.159	353.159	353.157	0.963832	0.963777	0.963777	0.963777
15	1.5	353.12	353.142	353.142	353.14	0.963957	0.963897	0.963897	0.963897
16	1.6	353.108	353.127	353.127	353.126	0.964073	0.96401	0.96401	0.96401
17	1.7	353.098	353.115	353.115	353.114	0.96418	0.964114	0.964114	0.964115
18	1.8	353.09	353.105	353.105	353.104	0.964279	0.964212	0.964212	0.964212
19	1.9	353.083	353.097	353.097	353.096	0.96437	0.964302	0.964302	0.964303
20	2.	353.078	353.09	353.09	353.089	0.964454	0.964385	0.964385	0.964386

Tabulka 10 – Porovnání výsledky výpočtu jednotlivých metod

## 7 ROZDÍLY MEZI ŘEŠENÍ SOUSTAVY ČTYŘ A DVOU DIFERENCIÁLNÍCH ROVNIC

### 7.1 Zadání vstupních parametrů

#### 7.1.1 Vstupní diferenciální rovnice

Řešíme-li soustavu čtyř diferenciálních rovnic, musíme do části pro zadávání vstupní diferenciální rovnice doplnit k proměnným  $y'_1$  a  $y'_2$  proměnné  $y'_3$  a  $y'_4$  oproti řešení soustavy dvou diferenciálních rovnic. Ukázka na Obrázku.

##### ■ Vstupní diferenciální rovnice

$$\begin{aligned}y1' &= y2; \\ y2' &= 1 + \text{Exp}[y1];\end{aligned}$$

##### ■ Vstupní diferenciální rovnice

$$\begin{aligned}y1' &= (q1r / Vr) * (cA0 - y1) - k01 * \text{Exp}[-E1 / y3] * y1 - k03 * \text{Exp}[-E3 / y3] * y1 * y1; \\ y2' &= (-q1r / Vr) * y2 + k01 * \text{Exp}[-E1 * (1 / y3)] * y1 - k02 * \text{Exp}[-E2 * (1 / y3)] * y2; \\ y3' &= (q1r / Vr) * (Tr0 - y3) - ((h1 * k01 * \text{Exp}[-E1 * (1 / y3)] * y1) + (h2 * k02 * \text{Exp}[-E2 * (1 / y3)] * y2) + \\ &\quad (h3 * k03 * \text{Exp}[-E3 * (1 / y3)] * y1 * y1)) / (ror * cpr) + ((Ar * U) / (Vr * ror * cpr)) * (y4 - y3); \\ y4' &= 1 / (mc * cpc) * (Q1c + ((Ar * U) * (y3 - y4))); \end{aligned}$$

#### 7.1.2 Zadání počátečních podmínek

Při řešení soustavy dvou diferenciálních rovnic jsou potřeba dvě počáteční podmínky, které jsou v programu definovány proměnnými  $y10$  a  $y20$ . Chceme-li řešit soustavu čtyř diferenciálních rovnic, musíme program doplnit ještě o dvě počáteční podmínky. Tyto podmínky musíme v programu definovat jako proměnné  $y30$  a  $y40$ . Následné úpravy můžeme vidět na obrázcích.

##### ■ Zadání počátečních podmínek

$$\begin{aligned}y10 &= 0; \\ y20 &= 0;\end{aligned}$$

##### ■ Zadání počátečních podmínek

$$\begin{aligned}y10 &= 2.1403; \\ y20 &= 1.0903; \\ y30 &= 387.34; \\ y40 &= 386.06;\end{aligned}$$

### 7.1.3 Pomocné proměnné

V případě řešení soustavy čtyř diferenciálních rovnic musíme v programu upravit taky pomocné proměnné. Jedná se především o pole, do kterého se nám uloží vstupní diferenciální rovnice postupně od diferenciální rovnice s označením  $y'_1$  až po diferenciální rovnici s označením  $y'_4$ . V poli s indexem 0 (f[0]) budeme mít uloženou diferenciální rovnici  $y'_1$ . V případě řešení soustavy dvou diferenciálních rovnic máme tak nadefinované jen f[0] a f[1]. V případě řešení soustavy čtyř diferenciálních rovnic musíme k f[0] a f[1] doplnit f[2] a f[3] které odpovídají vstupní diferenciální rovnici  $y'_3$  a  $y'_4$ . Popsané úpravy jsou na následujících obrázcích.

#### ■ Pomocné proměnné

```
f[0] = y1';
f[1] = y2';
PocetKroku = (HorniInterval - DolniInterval) / krok;
x[0] = DolniInterval;
```

#### ■ Pomocné proměnné

```
f[0] = y1';
f[1] = y2';
f[2] = y3';
f[3] = y4';
PocetKroku = (HorniInterval - DolniInterval) / krok;
x[0] = DolniInterval;
```

## 7.2 Metody výpočtu

### 7.2.1 Eulerova metoda:

Při řešení soustavy čtyř diferenciálních rovnic musíme Eulerovu metodu upravit oproti řešení soustavy pro dvě diferenciální rovnice následujícím způsobem. V prvním případě musíme Eulerovu metodu doplnit o proměnné  $yE[0,2]$  a  $yE[0,3]$  oproti soustavě dvou diferenciálních rovnic, kde jsme použili proměnné  $yE[0,0]$  a  $yE[0,1]$ . Proměnná  $yE[0,0]$  nám představuje počáteční podmínku pro diferenciální rovnici  $y'_1$ . Dále musíme upravit funkci f v Eulerově metodě. Pro dvě diferenciální rovnice má funkce jen tři argumenty  $f[x_,y1_,y2_]$  a  $f[x[n], yE[n,0], yE[n,1]]$ .

Pro čtyři diferenciální rovnice potřebujeme, aby funkce měla argumentů pět  $f[x\_ , y1\_ , y2\_ , y3\_ , y4\_ ]$  a  $f[x[n], yE[n,0], yE[n,1], yE[n,2], yE[n,3]]$ .

Poslední úpravou Eulerovi metody je změna hodnoty proměnné  $m$ . Při dvou diferenciálních rovnicích, máme proměnnou  $m$  následnou  $\{m,0,1\}$ . Při čtyřech diferenciálních rovnicích máme proměnnou  $m$  změněnou  $\{m,0,3\}$ , protože se používají čtyři diferenciální rovnice a musíme zajistit, aby se výpočty vykonaly ve všech rovnicích. Následné změny pro lepší orientaci jsou znázorněny na následujících obrázcích.

#### • Eulerova metoda

```
yE[0, 0] = y10;
yE[0, 1] = y20;
Do[{Do[{
  f[x_, y1_, y2_] = f[m],
  x[n + 1] = x[n] + krok,
  yE[n + 1, m] = yE[n, m] + krok * f[x[n], yE[n, 0], yE[n, 1]]}], {m, 0, 1}}],
{n, 0, PocetKroku}]
```

#### • Eulerova metoda

```
yE[0, 0] = y10;
yE[0, 1] = y20;
yE[0, 2] = y30;
yE[0, 3] = y40;
Do[{Do[{
  f[x_, y1_, y2_, y3_, y4_] = f[m],
  x[n + 1] = x[n] + krok,
  yE[n + 1, m] = yE[n, m] + krok * f[x[n], yE[n, 0], yE[n, 1], yE[n, 2], yE[n, 3]]}], {m, 0, 3}}],
{n, 0, PocetKroku}]
```

### 7.2.2 Eulerova metoda – První modifikace

Tuto metodu musíme taky doplnit o proměnné  $yEP[0,2]$  a  $yEP[0,3]$ , pokud chceme řešit soustavu čtyř diferenciálních rovnic oproti řešení soustavy dvou diferenciálních rovnic, kde máme pouze proměnné  $yEP[0,0]$  a  $yEP[0,1]$ . Dále musíme upravit následnou funkci  $f$ . Při řešení soustavy pro dvě diferenciální rovnice má funkce následující tvar  $f[x\_ , y1\_ , y2\_ ]$ , pro  $k1$  má následující tvar  $f[x[n], yEP[n,0], yEP[n,1]]$  a pro  $k2$  má funkce tvar  $f[x[n]+1/2krok, yEP[n,0]+1/2 krok k1[n,0], yEP[n,1]+1/2 krok k1[n,1]]$

Při řešení soustavy pro čtyři diferenciální rovnice musíme funkci  $f$  změnit následujícím způsobem  $f[x\_ , y1\_ , y2\_ , y3\_ , y4\_ ]$ , pro  $k1$   $f[x[n], yEP[n,0], yEP[n,1], yEP[n,2], yEP[n,3]]$  a pro  $k2$   $f[x[n]+1/2krok, yEP[n,0]+1/2 krok k1[n,0], yEP[n,1]+1/2 krok k1[n,1], yEP[n,2]+1/2 krok k1[n,2], yEP[n,3]+1/2 krok k1[n,3]]$ .

Posledním úpravou je úprava proměnné  $m$ . Popsanou v Eulerově metodě. Následné změny si můžete lépe prohlédnout na následujících obrázcích.

### • Eulerova metoda - První modifikace

```

yEP[0, 0] = y10;
yEP[0, 1] = y20;
Do [Do[{
  f[x_, y1_, y2_] = f[m],
  x[n + 1] = x[n] + krok,
  k1[n, m] = f[x[n], yEP[n, 0], yEP[n, 1]],
  k2[n, m] = f[x[n] + 1 / 2 krok, yEP[n, 0] + 1 / 2 * krok * k1[n, 0],
    yEP[n, 1] + 1 / 2 * krok * k1[n, 1]],
  yEP[n + 1, m] = yEP[n, m] + krok * k2[n, m]}, {m, 0, 1}], {n, 0, PocetKroku}]

```

### • Eulerova metoda - První modifikace

```

yEP[0, 0] = y10;
yEP[0, 1] = y20;
yEP[0, 2] = y30;
yEP[0, 3] = y40;
Do [Do[{
  f[x_, y1_, y2_, y3_, y4_] = f[m],
  x[n + 1] = x[n] + krok,
  k1[n, m] = f[x[n], yEP[n, 0], yEP[n, 1], yEP[n, 2], yEP[n, 3]],
  k2[n, m] = f[x[n] + 1 / 2 krok, yEP[n, 0] + 1 / 2 * krok * k1[n, 0], yEP[n, 1] + 1 / 2 * krok * k1[n, 1],
    yEP[n, 2] + 1 / 2 * krok * k1[n, 2], yEP[n, 3] + 1 / 2 * krok * k1[n, 3]],
  yEP[n + 1, m] = yEP[n, m] + krok * k2[n, m]}, {m, 0, 3}], {n, 0, PocetKroku}]

```

## 7.2.3 Eulerova metoda – Druhá modifikace

Řešení soustavy pro dvě diferenciální rovnice máme k dispozici proměnné yED [0,0] a yED[0,1]. Funkce má tvar f[x\_,y1\_,y2\_], pro k1 f[x[n], yED [n,0], yED [n,1]] a pro k2 f[x[n]+krok, yED [n,0]+krok k1[n,0], yED [n,1]+k1[n,1]]

Pokud chceme řešit soustavu pro čtyři diferenciální rovnice, musíme do programu doplnit proměnné yED [0,2] a yED [0,3]. Funkce musíme přepsat do následných tvarů f[x\_,y1\_,y2\_,y3\_,y4\_], pro k1 f[x[n], yED [n,0], yED [n,1], yED [n,2], yED [n,3]] a pro k2 f[x[n]+krok, yED [n,0]+krok k1[n,0], yED [n,1]+krok k1[n,1], yED [n,2] +krok k1[n,2], yED [n,3]+ krok k1[n,3]].

Dále nesmíme zapomenout změnit proměnnou m. Veškeré změny jsou znázorněny na obrázcích.

### • Eulerova metoda -Druhá modifikace

```

yED[0, 0] = y10;
yED[0, 1] = y20;
Do [Do[{
  f[x_, y1_, y2_] = f[m],
  x[n + 1] = x[n] + krok,
  k1[n, m] = f[x[n], yED[n, 0], yED[n, 1]],
  k2[n, m] = f[x[n] + krok, yED[n, 0] + krok * k1[n, 0], yED[n, 1] + krok * k1[n, 1]],
  yED[n + 1, m] = yED[n, m] + 1 / 2 * krok * (k1[n, m] + k2[n, m])}, {m, 0, 1}],
{n, 0, PocetKroku}]

```

### • Eulerova metoda -Druhá modifikace

```

yED[0, 0] = y10;
yED[0, 1] = y20;
yED[0, 2] = y30;
yED[0, 3] = y40;
Do [Do[{
  f[x_, y1_, y2_, y3_, y4_] = f[m],
  x[n+1] = x[n] + krok,
  k1[n, m] = f[x[n], yED[n, 0], yED[n, 1], yED[n, 2], yED[n, 3]],
  k2[n, m] = f[x[n] + krok, yED[n, 0] + krok * k1[n, 0], yED[n, 1] + krok * k1[n, 1],
    yED[n, 2] + krok * k1[n, 2], yED[n, 3] + krok * k1[n, 3]],
  yED[n+1, m] = yED[n, m] + 1/2 * krok * (k1[n, m] + k2[n, m])}], {m, 0, 3}], {n, 0, PocetKroku}]

```

### 7.2.4 Runge – Kuttova metoda 4. řádu

Řešení soustavy dvou diferenciálních rovnic:

- Proměnné:

$yR [0,0]$  a  $yR[0,1]$

- Funkce má tvar:

$f[x_,y1_,y2_]$

pro  $k1$   $f[x[n], yR [n,0], yR [n,1]]$

pro  $k2$   $f[x[n]+1/2krok, yR [n,0]+1/2 krok k1[n,0], yR [n,1]+1/2 k1[n,1]]$

pro  $k3$   $f[x[n]+1/2krok, yR [n,0]+1/2 krok k2[n,0], yR [n,1]+1/2 k2[n,1]]$

pro  $k4$   $f[x[n]+krok, yR [n,0]+krok k3[n,0], yR [n,1]+k3[n,1]]$

Rozdíl při řešení soustavy čtyř diferenciálních rovnic

- Proměnné:

$yR [0,0], yR[0,1], yR [0,2], yR [0,3]$

- Funkce má tvar:

$f[x_,y1_,y2_]$

pro  $k1$   $f[x[n], yR [n,0], yR [n,1], yR [n,2], yR [n,3]]$

pro  $k2$   $f[x[n]+1/2krok, yR [n,0]+1/2 krok k1 [n,0], yR [n,1]+1/2 k1[n,1], yR [n,2]+1/2 k1[n,2], yR [n,3]+1/2 k1[n,3]]$

pro  $k3$   $f[x[n]+1/2krok, yR [n,0]+1/2 krok k2[n,0], yR [n,1]+1/2 k2[n,1], yR [n,2]+1/2 k2[n,2], yR [n,3]+1/2 k2[n,3]]$

pro  $k4$   $f[x[n]+krok, yR [n,0]+krok k3[n,0], yR [n,1]+k3[n,1], yR [n,2]+ k3[n,2], yR [n,3]+1/2 k3[n,3]]$

Dále nesmíme zapomenout na proměnnou  $m$ . Všechny změny jsou zobrazeny na následujících obrázcích.

#### • Rungova – Kuttova metoda 4. řádu

```

yR[0, 0] = y10;
yR[0, 1] = y20;
Do [Do[{
  f[x_, y1_, y2_] = f[m],
  x[n + 1] = x[n] + krok,
  k1[n, m] = f[x[n], yR[n, 0], yR[n, 1]],
  k2[n, m] = f[x[n] + 1/2 krok, yR[n, 0] + 1/2 * krok * k1[n, 0], yR[n, 1] + 1/2 * krok * k1[n, 1]],
  k3[n, m] = f[x[n] + 1/2 krok, yR[n, 0] + 1/2 * krok * k2[n, 0], yR[n, 1] + 1/2 * krok * k2[n, 1]],
  k4[n, m] = f[x[n] + krok, yR[n, 0] + krok * k3[n, 0], yR[n, 1] + krok * k3[n, 1]],
  yR[n + 1, m] = yR[n, m] + 1/6 * krok * (k1[n, m] + 2 * k2[n, m] + 2 * k3[n, m] + k4[n, m])}, {m, 0, 1}], {n, 0, PocetKroku}]

```

#### • Rungova – Kuttova metoda 4. řádu

```

yR[0, 0] = y10;
yR[0, 1] = y20;
yR[0, 2] = y30;
yR[0, 3] = y40;
Do [Do[{
  f[x_, y1_, y2_, y3_, y4_] = f[m],
  x[n + 1] = x[n] + krok,
  k1[n, m] = f[x[n], yR[n, 0], yR[n, 1], yR[n, 2], yR[n, 3]],
  k2[n, m] = f[x[n] + 1/2 krok, yR[n, 0] + 1/2 * krok * k1[n, 0], yR[n, 1] + 1/2 * krok * k1[n, 1],
  yR[n, 2] + 1/2 * krok * k1[n, 2], yR[n, 3] + 1/2 * krok * k1[n, 3]],
  k3[n, m] = f[x[n] + 1/2 krok, yR[n, 0] + 1/2 * krok * k2[n, 0], yR[n, 1] + 1/2 * krok * k2[n, 1],
  yR[n, 2] + 1/2 * krok * k2[n, 2], yR[n, 3] + 1/2 * krok * k2[n, 3]],
  k4[n, m] = f[x[n] + krok, yR[n, 0] + krok * k3[n, 0], yR[n, 1] + krok * k3[n, 1],
  yR[n, 2] + krok * k3[n, 2], yR[n, 3] + krok * k3[n, 3]],
  yR[n + 1, m] = yR[n, m] + 1/6 * krok * (k1[n, m] + 2 * k2[n, m] + 2 * k3[n, m] + k4[n, m])}, {m, 0, 3}], {n, 0, PocetKroku}]

```

## 7.3 Zobrazení výsledků

- Zobrazení tabulkou:

V této části programu měníme jen proměnnou  $j$  z  $\{j,0,1\}$  pokud jde o řešení soustavy pro dvě diferenciální rovnice na  $\{j,0,3\}$  pokud jde o řešení soustavy pro čtyři diferenciální rovnice. Změnou proměnné  $j$  se nám vypíšou výsledky Y1, Y2, Y3 a Y4 oproti Y1 a Y2.

- Zobrazení grafem:

Pokud provádíme řešení soustavy pro dvě diferenciální rovnice, vystačíme si se dvěma grafy. Pokud chceme provádět řešení soustavy čtyř diferenciálních rovnic, budeme potřebovat k zobrazení řešení čtyři grafy. Dále nesmíme zapomenout, že v každém ListPlot musí být odlišná proměnná  $yE$ . Zajistíme tím vykreslení pokaždé jiného grafu a ne například dva stejné. Např v prvním ListPlot bude proměnná  $EP[i,0]$ , odpovídající výsledkům pro první diferenciální rovnici. V druhém ListPlot bude  $EP[i,1]$  atd.

```
ListPlot[Table[{x[i], yE[i, 0]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y1", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]
ListPlot[Table[{x[i], yE[i, 1]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y2", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]

ListPlot[Table[{x[i], yE[i, 0]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y1", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]
ListPlot[Table[{x[i], yE[i, 1]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y2", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]
ListPlot[Table[{x[i], yE[i, 2]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y3", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]
ListPlot[Table[{x[i], yE[i, 3]}, {i, 0, PocetKroku}], Joined → True,
  PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"x", "y4", "Eulerova metoda"},
  TextStyle → {FontFamily → "Times New Roman", FontSize → 12, FontColor → RGBColor[0, 0, 0]},
  Frame → True]
```

## 7.4 Porovnání výsledků

Zde na obrázku je nastíněno, jak vytvořit tabulku ve které si můžeme nejlépe porovnat jednotlivé metody. Je zde nastíněno jen vypsání hodnot  $y_1$  Eulerovy metody. Pokud chceme vypsát hodnotu  $y_2$  Eulerovy metody musíme změnit hodnotu  $j$  na  $\{j,1,1\}$ . Pro vypsání  $y_1$  Runge – Kuttovu metodu 4. Řádu, musíme změnit proměnnou  $yE$  na  $yR$ . Pro vypsání všechny výsledky jednotlivých metod, musíme si GridBox upravit. Pomocí kláves Ctrl 1 , (čárka) přidáme další sloupec. Podle toho kolik chceme zobrazovat výsledků, si přidáme sloupce.

```
GridBox[
  "n"                                "x"                                "Y1-E"
  Table[{i}, {i, 0, PocetKroku}] GridBox[Table[{x[i]}, {i, 0, PocetKroku}], GridBox[Table[yE[i, j],
  // TableForm                      ColumnWidths → 3] // DisplayForm          {i, 0, PocetKroku},
  {j, 0, 0}],
  ColumnWidths → 6] // DisplayForm
, GridFrame → True, RowLines → True, ColumnLines → True, RowMinHeight → 3] // DisplayForm
```

## ZÁVĚR

Cílem bakalářské práce bylo vytvořit notebook v softwaru Mathematica, který se bude zabývat numerickým řešením soustav diferenciálních rovnic. Veškeré podmínky, které jsem si stanovil, byly splněny. Do notebooku můžeme zadávat konstanty, vstupní hodnoty, vlastní diferenciální rovnice, počáteční podmínky, dobu řešení a krok výpočtu. Dále si můžeme vybrat ze čtyř metod řešení. Výsledky řešení si můžeme zobrazit ve formě tabulky nebo formě grafické. Notebook ještě nabízí srovnání jednotlivých metod, kde jednotlivé výsledky řešení jsou pro srovnání zapsány do jedné tabulky. Z tabulky nám vyplývá, že nejpřesnější s uvedených numerických je Runge – Kuttova metoda 4. řádu. Méně přesnější jsou Eulerova metoda – První modifikace a Eulerova metoda – Druhá modifika, které se Runge – Kuttovi metody liší jen minimálně. Nejméně přesnou metodou je Eulerova metoda. Ta se od dříve zmiňovaných metod liší více. Z hlediska programovatelnosti je Eulerova metoda nejméně náročná. Nejnáročnější na programovatelnost je Runge – Kuttova metoda 4. Řádu.

Vytvořený notebook by měl sloužit jako pomůcka pro složitější výpočty diferenciálních rovnic, které by člověku trvaly několik hodin.

## ZÁVĚR V ANGLIČTINĚ

The goal of my work was to create a notebook of Mathematica software, which will deal with the numerical solution of differential equation systems. Any conditions that I set were met. You can enter your notebook constants, input values, their differential equations, initial conditions for solving step and calculating. In addition, you can choose from four methods. The results we can see in tabular form or graphical form. Notebook yet offers a comparison of different methods, where individual results are compared to solutions entered in one table. The table shows us that the best is to set numerical Runge - Kutta method 4 Procedure. Less accurate the Euler method - Euler and modified the first method - the second of the modified to Runge - Kutta methods differ only minimally. Least accurate method is the Euler method. This is the previously mentioned methods differ more. In terms of programmability Euler method is at least challenging. Programmability is the most difficult to Runge - Kutta method 4 Procedure.

Created notebook should serve as an aid for complicated calculations of differential equations, which lasted several hours man.

**SEZNAM POUŽITÉ LITERATURY**

- [1] FAJMON, Břetislav; RŮŽIČKOVÁ, Irena. *Matematika 3* [online]. [s.l.] : [s.n.], [cit. 2010-05-11]. Dostupné z WWW:  
<<http://www.umat.feec.vutbr.cz/~hlavicka/skripta/matematika3.pdf>>.
- [2] VICHER, Miroslav. *NUMERICKÁ MATEMATIKA* [online]. [s.l.] : [s.n.], 7.1.2003 [cit. 2010-05-10]. Dostupné z  
WWW: <[http://physics.ujep.cz/~mlisal/nm\\_1/vicher\\_nm1.pdf](http://physics.ujep.cz/~mlisal/nm_1/vicher_nm1.pdf)>
- [3] PŘÍKRYL, Petr. *Numerické metody matematické analýzy*. Praha : SNTL, 1985. 192 s
- [4] DUBCOVÁ, Miroslav; PURMOVÁ, Lucie; SIMERSKÁ, Carmen. *Sbírka příkladů z Matematiky II ve strukturovaném studiu*. Praha. 1st ed. Praha : VŠCHT, 2009. 198 s. ISBN 978-80-7080-706-4.
- [5] RIEČANOVÁ, Zdena a kol. *Numerické metody a matematická statistika*. 1. vyd. Bratislava : Alfa, 1987. 492 s. ISBN 063-559-87.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

Y – E Výsledky Eulerova metoda.

Y – EP Výsledky Eulerova metoda – První modifikace.

Y – ED Výsledky pomocí Eulerova metoda – Druhá modifikace.

Y – R Výsledky pomocí Runge – Kuttova metoda 4. řádu

EM Eulerova metoda

EMP Eulerova metoda – První modifikace

EMD Eulerova metoda – Druhá modifikace

R-KM Runge – Kuttova metoda 4. řádu

**SEZNAM OBRÁZKŮ**

Obrázek 1 – Přesné a přibližné řešení diferenciálních rovnic.....	11
Obrázek 2 – Grafické znázornění výsledků výpočtu EM.....	14
Obrázek 3 – Grafické znázornění výsledků výpočtu R-KM .....	17
Obrázek 4 – Úvodní obrazovka programu.....	25
Obrázek 5 – Část pro zadání vstupních parametrů .....	25
Obrázek 6 – Část pro výběr metody výpočtu .....	25
Obrázek 7 – Část pro výběr zobrazení výsledků.....	26
Obrázek 8 – Část pro porovnání výsledků jednotlivých metod.....	26
Obrázek 9 – Grafické znázornění výpočtu EM.....	30
Obrázek 10 – Grafické znázornění výpočtu EMP .....	30
Obrázek 11 – Grafické znázornění výpočtu EMD .....	31
Obrázek 12 – Grafické znázornění výpočtu R-KM.....	32
Obrázek 13 – Grafické znázornění výpočtu EM .....	35
Obrázek 14 – Grafické znázornění výpočtu EMP .....	36
Obrázek 15 – Grafické znázornění výpočtu EMD .....	36
Obrázek 16 – Grafické znázornění výpočtu R-KM.....	37

**SEZNAM TABULEK**

Tabulka 1 – Výsledky výpočtu EM.....	13
Tabulka 2 – Výsledky výpočtu R-KM .....	17
Tabulka 3 – Výsledky výpočtu EM pro soustavu .....	19
Tabulka 4 – Výsledky výpočtu R-KM pro soustavy .....	20
Tabulka 5 – Výsledky výpočtu EM.....	30
Tabulka 6 – Výsledky výpočtu EMP.....	30
Tabulka 7 – Výsledky výpočtu EMD.....	31
Tabulka 8 – Výsledky výpočtu R-KM.....	31
Tabulka 9 – Porovnání výsledků jednotlivých metod .....	32
Tabulka 10 – Porovnání výsledků jednotlivých metod .....	37

## SEZNAM PŘÍLOH

Příloha 1 – Prázdný notebook pro soustavu dvou diferenciálních rovnic

Příloha 2 – Prázdný notebook pro soustavu čtyř diferenciálních rovnic

Příloha 3 – Notebook s vyřešenou soustavou dvou diferenciálních rovnic.

Příloha 4 – Notebook s aplikačním příkladem na soustavu dvou diferenciálních rovnic.

Zdrojové kódy na CD

**PŘÍLOHA P I: NÁZEV PŘÍLOHY**