



Přednáška z předmětu: Algoritmizace inženýrských výpočtů

Téma č.8: Řešení diferenciálních rovnic

doc. Ing. Martin Krejsa, Ph.D.

Obsah

1. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Obsah

1	Řešení diferenciálních rovnic	3
1.1	Obyčejné diferenciální rovnice prvního řádu	4
1.1.1	Eulerova metoda	5
1.1.2	Metoda Runge-Kutta	16
1.1.3	Metoda skákající žaby	24
1.2	Obyčejné diferenciální rovnice druhého řádu	28



Obsah

2. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Kapitola 1

Řešení diferenciálních rovnic

Cíle

Kapitola má za cíl:

- seznámit studenty s numerickým řešením jednoduchých diferenciálních rovnic,
- ukázat jim jejich uplatnění v elementárních úlohách stavební mechaniky.

Obsah

3. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

V diferenciálních rovnicích se jako proměnné objevují derivace funkcí. Podle počtu proměnných a typu derivací funkcí lze diferenciální rovnice členit na:

- *obyčejné diferenciální rovnice*, jenž obsahují derivace hledané funkce jen podle jedné proměnné.
- *parciální diferenciální rovnice*, které obsahují derivace hledané funkce podle více proměnných, tedy parciální derivace.

Řád diferenciální rovnice je definován podle nejvyšší derivace, která je v dané diferenciální rovnice obsažena.

Řešením diferenciální rovnice je funkce, která má příslušné derivace a vyhovuje dané diferenciální rovnici - integrál diferenciální rovnice, kterých může být nekonečně mnoho. V praktických úlohách definují jednoznačné řešení počáteční podmínky.

1.1. Obyčejné diferenciální rovnice prvního řádu

Obyčejné diferenciální rovnice prvního řádu obsahují jednu derivaci funkce jedné závisle proměnné $y(x)$. Numericky lze např. řešit funkci $y = y(x)$, která v intervalu $\langle a, b \rangle$ vyhovuje rovnici:

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

kde $f(x, y(x))$ je pravá strana obyčejné diferenciální rovnice prvního řádu, pro jejíž jednoznačné určení musí být splněna počáteční podmínka ve tvaru:

$$y(a) = c . \quad (1.2)$$



Obsah

4. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

1.1.1. Eulerova metoda

Nejjednodušší výpočetní postup numerického řešení obyčejných diferenciálních rovnic s počáteční podmínkou publikoval v roce 1768 švýcarský matematik a fyzik Leonhard Euler. Řešení je založeno na přibližném výpočtu derivace funkce $y'(x)$ v rovnici (1.1) pomocí aproximace metodou konečných diferencí s využitím dvoubodové dopředné diferenční formule:

$$y'(x_i) = f(x_i, y(x_i)) \approx \frac{y(x_{i+1}) - y(x_i)}{h} = f(x_i, y_i), \quad (1.3)$$

kde h je krok, odpovídající hodnotě $(x_{i+1} - x_i)$. Ze vztahu (1.3) je možno jednoduchou úpravou získat rekurentní vzorec Eulerovy metody:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i), \quad (1.4)$$

pro $i = 0, 1, \dots, n - 1$, kde n je počet diferencí v řešeném intervalu $\langle a, b \rangle$. Hodnotu y_0 je nutno jednoznačně určit pomocí počáteční podmínky podle (1.2).

Příklad 1.1. Eulerovou metodou určete v intervalu $\langle -2; 3 \rangle$ přibližné řešení obyčejné diferenciální rovnice:

$$y'(x) = x^2 - 0,2 \cdot y(x), \quad (1.5)$$

s počáteční podmínkou $y(-2) = -1$. Výpočetní krok h postupně volte $h = 1$, $h = 0,5$, $h = 0,1$ příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením:

$$y(x) = 5 \cdot x^2 - 50 \cdot x + 250 - \frac{371}{e^{0,4}} \cdot e^{-0,2 \cdot x}, \quad (1.6)$$



Obsah

5. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Řešení. Řešení Eulerovou metodou vychází z rekurentního vzorce (1.4) a lze je naprogramovat např. následujícím způsobem:

```
f=inline('x^2-0.2*y');
a=-2;
b=3;
c=-1;
h=0.5;
n=(b-a)/h;
x(1)=a;
y(1)=c;
yp(1)=c;
for i=1:n
    x(i+1)=x(i)+h;
    y(i+1)=y(i)+h*f(x(i),y(i));
    yp(i+1)=5*x(i+1)^2-50*x(i+1)+250-(371/exp(0.4))*exp(-0.2*x(i+1));
end
[x' y' yp' (y-yp)']
plot(x,y,'r',x,yp,'b');
legend('přesné řešení','aproximace');
title('Eulerova aproximace');
xlabel('x'); ylabel('y(x)');
```

[Obsah](#)[6. strana ze 46](#)[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

Výsledné řešení např. pro výpočetní krok $h = 0,5$ je následující:

x	y	yp	y-yp
-2.0000	-1.0000	-1.0000	0.0000
-1.5000	1.1000	0.5553	0.5447
-1.0000	2.1150	1.2509	0.8641
-0.5000	2.4035	1.4064	0.9971
0.0000	2.2882	1.3113	0.9769
0.5000	2.0593	1.2271	0.8322
1.0000	1.9784	1.3909	0.5875
1.5000	2.2806	2.0169	0.2637
2.0000	3.1775	3.2990	-0.1214
2.5000	4.8598	5.4127	-0.5529
3.0000	7.4988	8.5167	-1.0179

Srovnání dosažené přesnosti řešení pro jednotlivé hodnoty výpočetních kroků $h = 1$, $h = 0,5$, $h = 0,1$ a $h = 0,01$ je zobrazeno na obrázku 1.1.



Obsah

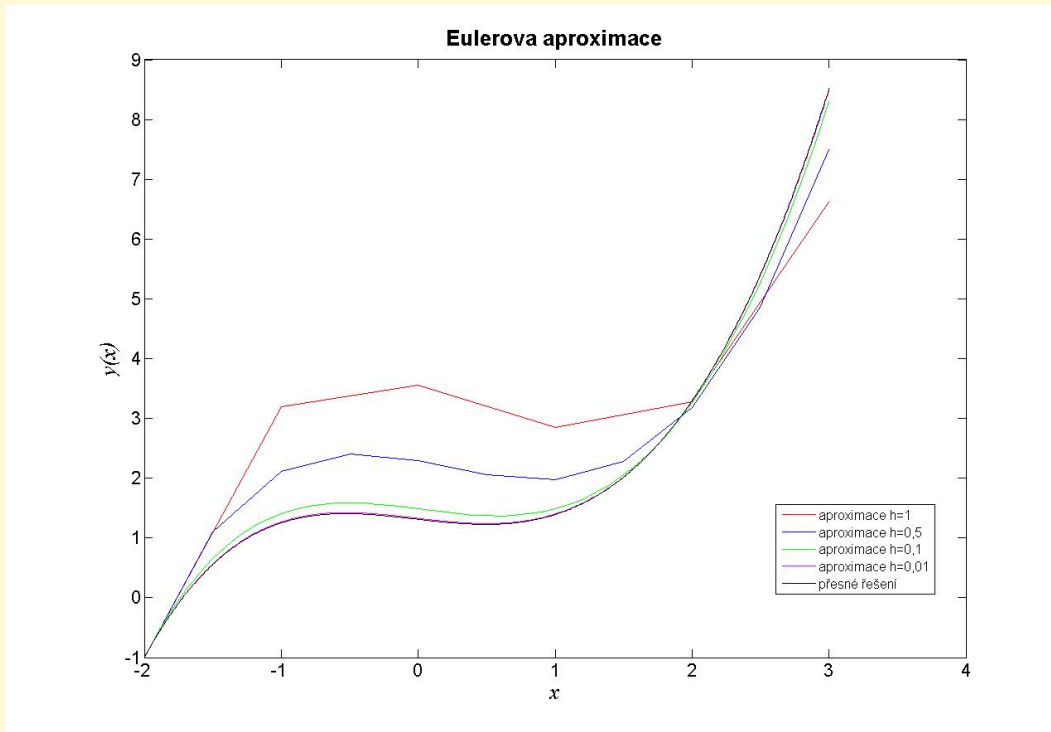
7. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.1 Výsledná aproximace funkce $y(x)$ z příkladu 1.1 pro kroky $h = 1$, $h = 0,5$, $h = 0,1$ a $h = 0,01$



Obsah

8. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Poznámka 1.2. Řešení úlohy z příkladu 1.1 je možno zkontrolovat rovněž matematickými prostředky programového systému MATLAB. Jednou z možností je využití funkce `ode45`, např. s požadovanou tolerancí nepřesnosti řešení $1 \cdot 10^{-9}$, následujícími příkazy:

Nejprve je potřeba zadat řešenou diferenciální rovnici s využitím samostatné `m`-funkce, např.:

```
function y=fce(x,y);  
y=x^2-0.2*y;
```

na kterou se pak lze odkázat:

```
options=odeset('AbsTol',1e-9);  
[x,y]=ode45(@fce,[-2 3],-1,options)
```

Poslední příkaz vypíše hodnoty výsledné funkce $y(x)$ v bodech x_i . Pokud se příkaz upraví na tvar:

```
ode45(@fce,[-2 3],-1)
```

zobrazí se graf vyřešené funkce $y(x)$ (viz obrázek 1.2).

Druhou možností, jak úlohu z příkladu 1.1 vyřešit příkazy programu MATLAB, je použití funkce `dsolve` pro symbolické řešení obyčejných diferenciálních rovnic s následnou vektorizací stanovené funkce $y(x)$ pomocí sekvence příkazů:

```
y=dsolve('Dy=x^2-0.2*y','y(-2)=-1','x')  
x=linspace(-2,3,1000);  
z=eval(vectorize(y));  
plot(x,z)
```



Obsah

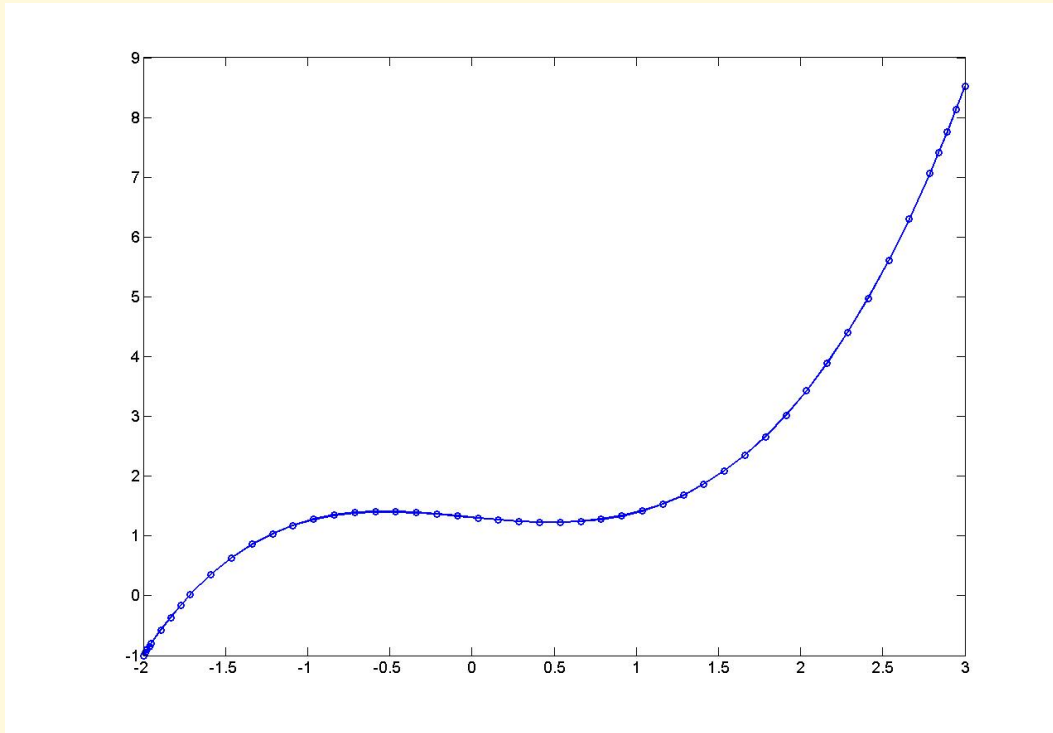
9. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.2 Výsledná aproximace funkce $y(x)$ z příkladu 1.1 určená funkcí ode45



Obsah

10. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Výsledný výraz, který byl funkcí `dsolve` stanoven:

$$y = 5*x^2 - 371/(\exp(2/5)*\exp(x/5)) - 50*x + 250$$

je identický s výrazem přesného řešení (1.6).

Příklad 1.3. Stanovte průběh posouvající síly na konzolovém nosníku, schématicky znázorněném na obr. 1.3, Eulerovou metodou. Konkrétní vstupní údaje jsou uvedeny v tabulce 1.1. Výpočetní krok h zvolte $h = 1$, příp. $h = 0,5$. Výslednou aproximaci porovnejte s přesným řešením.



Obsah

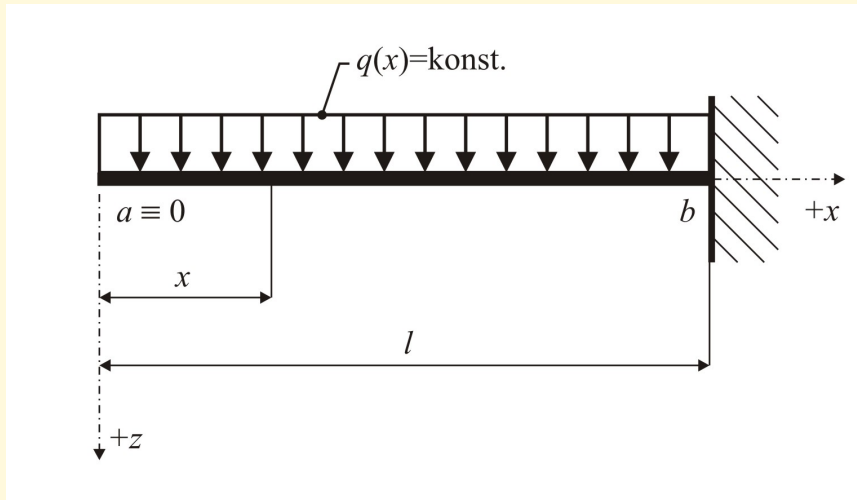
11. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.3 Statické schéma řešeného staticky určitého konzolového nosníku

Spojité silové zatížení q_z :	4 kN/m
Rozpětí konzolového nosníku l :	6 m

Tab. 1.1 Vstupní údaje příkladu 1.3



Obsah

12. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Řešení. Obyčejná diferenciální rovnice prvního řádu vyplývá ze Schwedlerových vztahů:

$$\frac{V_z(x)}{dx} = -q_z(x) = \text{konst} \rightarrow y'(x) = -q_z \cdot x^0. \quad (1.7)$$

Počáteční podmínka vychází ze statické okrajové podmínky, udávající nulovou hodnotu posouvající síly na volném okraji konzolového nosníku, tedy:

$$V_z(x=0) = y(x=0) = 0. \quad (1.8)$$

Výpočet aproximace průběhu posouvající síly Eulerovou metodou je založen na rekurentním vztahu (1.4). Přesné řešení odpovídá analyticky odvozené rovnici pro posouvající sílu $V_z(x)$:

$$V_z(x) = -q_z \cdot x. \quad (1.9)$$

Vzhledem ke skutečnosti, že funkce posouvající síly je lineární, lze v tomto případě získat Eulerovou metodou přesné řešení:



Obsah

13. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

x	y	yp	y-yp
0.0000	0.0000	0.0000	0.0000
0.5000	-2.0000	-2.0000	0.0000
1.0000	-4.0000	-4.0000	0.0000
1.5000	-6.0000	-6.0000	0.0000
2.0000	-8.0000	-8.0000	0.0000
2.5000	-10.0000	-10.0000	0.0000
3.0000	-12.0000	-12.0000	0.0000
3.5000	-14.0000	-14.0000	0.0000
4.0000	-16.0000	-16.0000	0.0000
4.5000	-18.0000	-18.0000	0.0000
5.0000	-20.0000	-20.0000	0.0000
5.5000	-22.0000	-22.0000	0.0000
6.0000	-24.0000	-24.0000	0.0000



Příklad 1.4. Určete na konzolovém nosníku z příkladu 1.3 průběh ohybových momentů Eulerovou metodou. Výpočetní krok h zvolte $h = 1$, příp. $h = 0,5$. Výslednou aproximaci porovnejte s přesným řešením.

Řešení. Obyčejná diferenciální rovnice prvního řádu vyplývá ze Schwedlerových vztahů:

$$\frac{M_y(x)}{dx} = V_z(x) \rightarrow y'(x) = -q_z \cdot x. \quad (1.10)$$



Obsah

14. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Počáteční podmínka vychází ze statické okrajové podmínky, udávající nulovou hodnotu ohybového momentu na volném okraji konzolového nosníku, tedy:

$$M_y(x = 0) = y(x = 0) = 0. \quad (1.11)$$

Výpočet aproximace průběhu ohybového momentu Eulerovou metodou je založen na rekurentním vztahu (1.4). Přesné řešení odpovídá analyticky odvozené rovnici pro ohybový moment $M_y(x)$:

$$M_y(x) = -\frac{q_z \cdot x^2}{2}. \quad (1.12)$$

Pro výpočetní krok $h = 0,5$ lze Eulerovou metodou získat tyto výsledky:

x	y	yp	y-yp
0.0000	0.0000	0.0000	0.0000
0.5000	0.0000	-0.5000	0.5000
1.0000	-1.0000	-2.0000	1.0000
1.5000	-3.0000	-4.5000	1.5000
2.0000	-6.0000	-8.0000	2.0000
2.5000	-10.0000	-12.5000	2.5000
3.0000	-15.0000	-18.0000	3.0000
3.5000	-21.0000	-24.5000	3.5000
4.0000	-28.0000	-32.0000	4.0000
4.5000	-36.0000	-40.5000	4.5000
5.0000	-45.0000	-50.0000	5.0000
5.5000	-55.0000	-60.5000	5.5000
6.0000	-66.0000	-72.0000	6.0000



Obsah

15. strana ze 46

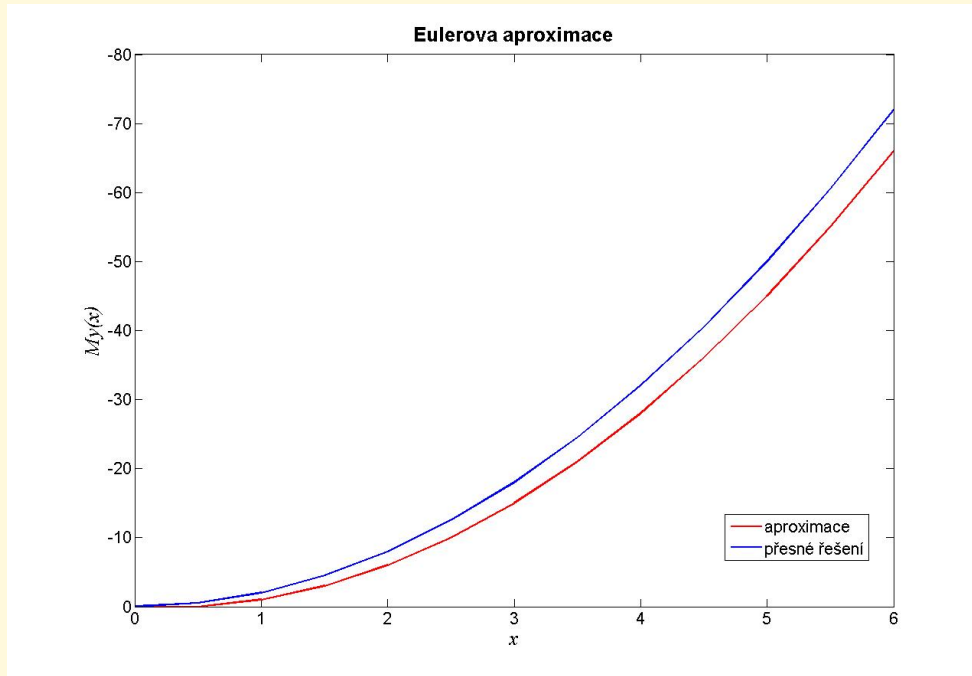


Zavřít dokument

Konec

Celá obrazovka/Okno

Průběh vypočtených ohybových momentů je pak zobrazen na obrázku 1.4.



Obr. 1.4 Výsledná aproximace ohybových momentů na konzolovém nosníku z příkladu 1.3



Obsah

16. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

1.1.2. Metoda Runge-Kutta

Metody založené na výpočetním postupu Runge-Kutta jsou vhodné pro řešení obyčejných diferenciálních rovnic tvaru (1.1). Tyto výpočetní techniky byly vyvinuty na přelomu 19. a 20. století německými matematiky Carlem Rungem a Martinem Wilhelmem Kuttou. Lze je vyjádřit obecným rekurentním vztahem:

$$y_{i+1} = y_i + h \cdot \sum_{i=1}^s (b_i \cdot k_i), \quad (1.13)$$

kde koeficienty k_i jsou dány obecným vztahem:

$$k_i = f\left(x_i + c_i \cdot h, y_i + h \cdot \sum_{j=1}^{i-1} (a_{i,j} \cdot k_j)\right). \quad (1.14)$$

Koeficienty $a_{i,j}$, b_i a c_i pro $i, j = 1, \dots, s$ jsou uvedeny pro Kuttovu metodu třetího řádu ($s = 3$) v tabulce 1.2 a pro klasickou metodu Runge-Kutta čtvrtého řádu ($s = 4$) v tabulce 1.3.

$c_1 = 0$	$a_{1,1} = 0$	$a_{1,2} = 0$	$a_{1,3} = 0$
$c_2 = \frac{1}{2}$	$a_{2,1} = \frac{1}{2}$	$a_{2,2} = 0$	$a_{2,3} = 0$
$c_3 = 1$	$a_{3,1} = -1$	$a_{3,2} = 2$	$a_{3,3} = 0$
	$b_1 = \frac{1}{6}$	$b_2 = \frac{2}{3}$	$b_3 = \frac{1}{6}$

Tab. 1.2 Koeficienty $a_{i,j}$, b_i a c_i Kuttovy metody



Obsah

17. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

$c_1 = 0$	$a_{1,1} = 0$	$a_{1,2} = 0$	$a_{1,3} = 0$	$a_{1,4} = 0$
$c_2 = \frac{1}{2}$	$a_{2,1} = \frac{1}{2}$	$a_{2,2} = 0$	$a_{2,3} = 0$	$a_{2,4} = 0$
$c_3 = \frac{1}{2}$	$a_{3,1} = 0$	$a_{3,2} = \frac{1}{2}$	$a_{3,3} = 0$	$a_{3,4} = 0$
$c_4 = 1$	$a_{4,1} = 0$	$a_{4,2} = 0$	$a_{4,3} = 1$	$a_{4,4} = 0$
	$b_1 = \frac{1}{6}$	$b_2 = \frac{1}{3}$	$b_3 = \frac{1}{3}$	$b_4 = \frac{1}{6}$

Tab. 1.3 Koeficienty $a_{i,j}$, b_i a c_i klasické metody Runge-Kutta

Poznámka 1.5. Pomocí vztahů (1.13) a (1.14) lze vyjádřit i rekurentní výraz (1.4) pro výpočet Eulerovou metodou, která je řádu $s = 1$. Příslušné koeficienty $a_{i,j}$, b_i a c_i jsou obsaženy v tabulce 1.4.

$c_1 = 0$	$a_{1,1} = 0$
	$b_1 = 1$

Tab. 1.4 Koeficienty $a_{i,j}$, b_i a c_i Eulerovy metody

Z principu metody Runge-Kutta vychází řada adaptivních metod. Jedná se např. o metodu Heun-Euler ($s = 2$, tabulka 1.5), metodu Ralstonovu ($s = 3$, tabulka 1.6) nebo metodu Bogacki-Shampine ($s = 4$, tabulka 1.7).



Obsah

18. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

$$\begin{array}{c|cc} c_1 = 0 & a_{1,1} = 0 & a_{1,2} = 0 \\ c_2 = 1 & a_{2,1} = 1 & a_{2,2} = 0 \\ \hline & b_1 = 1 & b_2 = 0 \end{array}$$

Tab. 1.5 Koeficienty $a_{i,j}$, b_i a c_i metody Heun-Euler

$$\begin{array}{c|ccc} c_1 = 0 & a_{1,1} = 0 & a_{1,2} = 0 & a_{1,3} = 0 \\ c_2 = \frac{1}{2} & a_{2,1} = \frac{1}{2} & a_{2,2} = 0 & a_{2,3} = 0 \\ c_3 = \frac{3}{4} & a_{3,1} = 0 & a_{3,2} = -\frac{3}{4} & a_{3,3} = 0 \\ \hline & b_1 = \frac{2}{9} & b_2 = \frac{3}{9} & b_3 = \frac{4}{9} \end{array}$$

Tab. 1.6 Koeficienty $a_{i,j}$, b_i a c_i Ralstonovy metody

$$\begin{array}{c|ccccc} c_1 = 0 & a_{1,1} = 0 & a_{1,2} = 0 & a_{1,3} = 0 & a_{1,4} = 0 \\ c_2 = \frac{1}{2} & a_{2,1} = \frac{1}{2} & a_{2,2} = 0 & a_{2,3} = 0 & a_{2,4} = 0 \\ c_3 = \frac{3}{4} & a_{3,1} = 0 & a_{3,2} = \frac{3}{4} & a_{3,3} = 0 & a_{3,4} = 0 \\ c_4 = 1 & a_{4,1} = \frac{2}{9} & a_{4,2} = \frac{1}{3} & a_{4,3} = \frac{4}{9} & a_{4,4} = 0 \\ \hline & b_1 = \frac{7}{24} & b_2 = \frac{1}{4} & b_3 = \frac{1}{3} & b_4 = \frac{1}{8} \end{array}$$

Tab. 1.7 Koeficienty $a_{i,j}$, b_i a c_i metody Bogacki-Shampine

Obsah

19. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.6. Metodou Runge-Kutta stanovte přibližné řešení obyčejné diferenciální rovnice z příkladu 1.1:

$$y'(x) = x^2 - 0,2 \cdot y(x), \quad (1.15)$$

v intervalu $\langle -2; 3 \rangle$ s počáteční podmínkou $y(-2) = -1$. Výpočetní krok h postupně volte $h = 1$, $h = 0,5$, $h = 0,1$ příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením.

Řešení. Vztahy (1.13) a (1.14), popisující podstatu metody Runge-Kutta, mohou být s využitím hodnot koeficientů $a_{i,j}$, b_i a c_i z tabulky 1.3 aplikovány pro velikost výpočetního kroku $h = 1$ např. následujícím způsobem:

```
f=inline('x^2-0.2*y');
a=-2; b=3;
c=-1;
h=1;
n=(b-a)/h;
x(1)=a; y(1)=c;
for i=1:n
    x(i+1)=x(i)+h;
    K1=h*f(x(i),y(i));
    K2=h*f(x(i)+h/2,y(i)+K1/2);
    K3=h*f(x(i)+h/2,y(i)+K2/2);
    K4=h*f(x(i)+h,y(i)+K3);
    y(i+1)=y(i)+(K1+2*K2+2*K3+K4)/6;
end
[x' y'], plot(x,y,'r');
```



Obsah

20. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Porovnájí-li se dosažené výsledky s přesným řešením (viz obrázek 1.5):

x	y	yp	y-yp
-2.0000	-1.0000	-1.0000	0.0000
-1.0000	1.2508	1.2509	-0.0001
0.0000	1.3112	1.3113	-0.0001
1.0000	1.3910	1.3909	0.0001
2.0000	3.2994	3.2990	0.0004
3.0000	8.5175	8.5167	0.0008

ukáže se podstatně větší přesnost vypočtených aproximací, nežli tomu bylo v případě Eulerovy metody.



Příklad 1.7. Stanovte přibližné řešení obyčejné diferenciální rovnice z příkladu 1.1:

$$y'(x) = x^2 - 0,2 \cdot y(x) , \quad (1.16)$$

v intervalu $\langle -2; 3 \rangle$ s počáteční podmínkou $y(-2) = -1$ s využitím ostatních metod, vycházejících z klasické metody Runge-Kutta, tedy Kuttovy metody třetího řádu, metody Heun-Euler, Ralstonovy metody a metody Bogacki-Shampine. Výpočetní krok h postupně volte $h = 1$, $h = 0,5$, $h = 0,1$ příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením.



Obsah

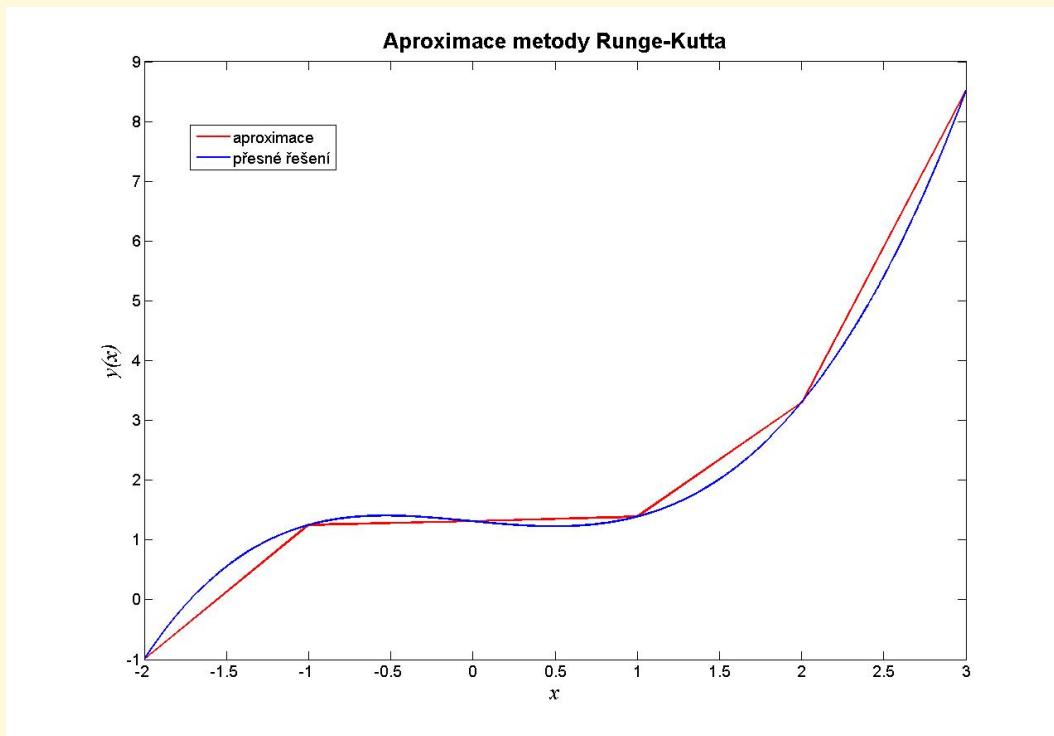
21. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.5 Výsledná aproximace metodou Runge-Kutta



Obsah

22. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.8. Určete na konzolovém nosníku z příkladu 1.3 průběh ohybových momentů metodou Runge-Kutta. Výpočetní krok h zvolte $h = 1$, příp. $h = 0,5$. Výslednou aproximaci porovnejte s přesným řešením.

Řešení. Výpočet aproximace průběhu ohybových momentů metodou Runge-Kutta vychází z rekurentních vztahů (1.13) a (1.14) a příslušných hodnot koeficientů $a_{i,j}$, b_i a c_i z tabulky 1.3.

Pro výpočetní krok $h = 0,5$ lze metodou Runge-Kutta získat oproti Eulerovy metody podstatně přesnější výsledky:

x	y	yp	y-yp
0.0000	0.0000	0.0000	0.0000
0.5000	-0.5000	-0.5000	0.0000
1.0000	-2.0000	-2.0000	0.0000
1.5000	-4.5000	-4.5000	0.0000
2.0000	-8.0000	-8.0000	0.0000
2.5000	-12.5000	-12.5000	0.0000
3.0000	-18.0000	-18.0000	0.0000
3.5000	-24.5000	-24.5000	0.0000
4.0000	-32.0000	-32.0000	0.0000
4.5000	-40.5000	-40.5000	0.0000
5.0000	-50.0000	-50.0000	0.0000
5.5000	-60.5000	-60.5000	0.0000
6.0000	-72.0000	-72.0000	0.0000

Průběh vypočtených ohybových momentů je pak zobrazen na obrázku 1.6.



Obsah

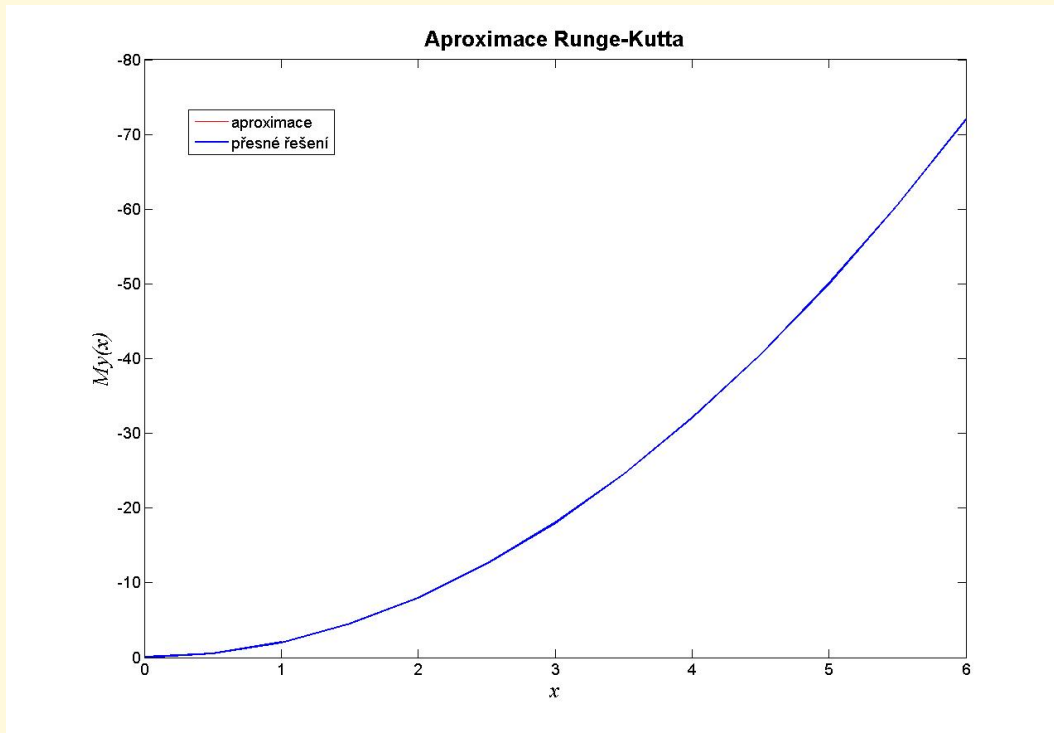
23. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.6 Výsledná aproximace ohybových momentů na konzolovém nosníku z příkladu 1.8



Obsah

24. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.9. Určete na konzolovém nosníku z příkladu 1.3 průběh ohybových momentů s využitím ostatních metod, vycházejících z klasické metody Runge-Kutta, tedy Kuttovy metody třetího řádu, metody Heun-Euler, Ralstonovy metody a metody Bogacki-Shampine. Výpočetní krok h postupně volte $h = 1$, $h = 0,5$, $h = 0,1$ příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením.

1.1.3. Metoda skákající žáby

Metoda skákající žáby je příkladem dvoukrokové metody, která vychází z rekurentního vzorce:

$$y_{i+1} = y_{i-1} + 2 \cdot h \cdot f(x_i, y_i) . \quad (1.17)$$

K výpočtu hodnoty y_{i+1} je tedy nutno znát hodnoty funkce y_i a y_{i-1} ve dvou předchozích bodech. Při zahájení výpočtu se obě hodnoty v počátečním úseku y_0 a y_1 stanoví z počáteční podmínky a s využitím některé z jednokrokových metod.

Název metody skákající žáby vystihuje skutečnost, že hodnota vypočtené aproximace osciluje okolo řešení přesného.

Příklad 1.10. Metodou skákající žáby stanovte přibližné řešení obyčejné diferenciální rovnice z příkladu 1.1:

$$y'(x) = x^2 - 0,2 \cdot y(x) , \quad (1.18)$$

v intervalu $\langle -2; 3 \rangle$ s počáteční podmínkou $y(-2) = -1$. Pro určení hodnoty funkce ve druhém bodu výpočtu $y(-2 + h)$ použijte Eulerovy metody. Výpočetní krok h postupně volte $h = 1$, $h = 0,5$, $h = 0,1$ příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením.



Obsah

25. strana ze 46



Zavřít dokument

Konec

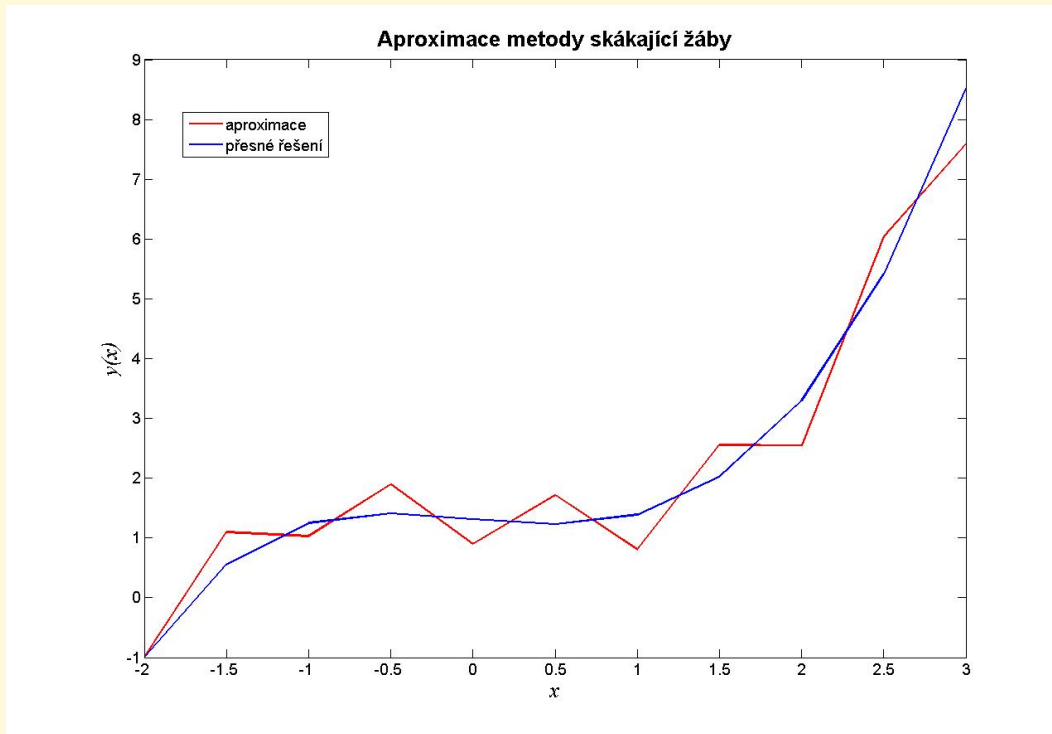
Celá obrazovka/Okno

Řešení. Výpočet metodou skákající žáby je založen na rekurentním vztahu (1.17). Výsledný zdrojový text programu pro velikost výpočetního kroku $h = 0,5$ může vypadat následujícím způsobem:

```
f=inline('x^2-0.2*y');  
a=-2; b=3;  
c=-1;  
h=0.5;  
n=(b-a)/h;  
x(1)=a; y(1)=c;  
x(2)=x(1)+h; y(2)=y(1)+h*f(x(1),y(1));  
for i=2:n  
    x(i+1)=x(i)+h;  
    y(i+1)=y(i-1)+2*h*f(x(i),y(i));  
end  
[x' y'], plot(x,y,'r');
```

Na obrázku 1.7 je zobrazena vypočtená aproximace metodou skákající žáby pro výpočetní krok $h = 0,5$. Na obrázku je patrná charakteristická vlastnost výpočetního postupu metody skákající žáby, a to oscilace okolo přesného řešení.

[Obsah](#)[26. strana ze 46](#)[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)



Obr. 1.7 Výsledná aproximace metody skákající žáby s výpočetním krokem $h = 0,5$



Obsah

27. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Dosažené numerické výsledky i odchylky od přesného řešení jsou tyto:

x	y	yp	y-yp
-2.0000	-1.0000	-1.0000	0.0000
-1.5000	1.1000	0.5553	0.5447
-1.0000	1.0300	1.2509	-0.2209
-0.5000	1.8940	1.4064	0.4876
0.0000	0.9012	1.3113	-0.4101
0.5000	1.7138	1.2271	0.4866
1.0000	0.8084	1.3909	-0.5824
1.5000	2.5521	2.0169	0.5352
2.0000	2.5480	3.2990	-0.7509
2.5000	6.0425	5.4127	0.6298
3.0000	7.5895	8.5167	-0.9272



Obsah

28. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

1.2. Obyčejné diferenciální rovnice druhého řádu

Podobně jako v případě obyčejné diferenciální rovnice prvního řádu (1.1) lze řešit i obyčejné diferenciální rovnice druhého řádu:

$$y''(x) = f(x, y(x), y'(x)). \quad (1.19)$$

Typů diferenciálních rovnic 2. řádu je mnoho typů. Např. lze řešit obyčejné diferenciální rovnice s konstantními koeficienty a, b a c , které lze vyjádřit ve tvaru:

$$a \cdot y''(x) + b \cdot y'(x) + c \cdot y(x) = f(x). \quad (1.20)$$

Numerické řešení rovnic typu (1.20) spočívá v jejich převedení na soustavu dvou diferenciálních rovnic:

$$z(x) = y'(x) \quad (1.21)$$

a

$$z'(x) = \frac{f(x) - b \cdot z(x) - c \cdot y(x)}{a}. \quad (1.22)$$

Nutností řešení jsou v daném případě dvě počáteční podmínky, např.:

$$y(a) = c \quad (1.23)$$

a

$$z(b) = d. \quad (1.24)$$



Obsah

29. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.11. Určete na konzolovém nosníku z příkladu 1.3 průběh ohybových momentů řešením obyčejné diferenciální rovnice 2.řádu, vycházející ze Schwedlerových vztahů:

$$\frac{M_y(x)}{dx^2} = -q_z(x) = \text{konst} \rightarrow y''(x) = -q_z \cdot x^0. \quad (1.25)$$

Počáteční podmínky vychází ze statických okrajových podmínek, které udávají nulovou hodnotu posouvající síly i ohybového momentu na volném okraji konzolového nosníku, tedy:

$$V_z(x = 0) = y'(x = 0) = 0 \quad (1.26)$$

a

$$M_y(x = 0) = y(x = 0) = 0. \quad (1.27)$$

Pro numerické řešení použijte Eulerovu metodu. Výpočetní krok h zvolte $h = 1$, příp. $h = 0,5$. Výslednou aproximaci porovnejte s přesným řešením.



Obsah

30. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Řešení. Celý výpočetní postup je zřejmý ze zápisu programu s výpočetním krokem $h = 0,5$ do m-souboru programu MATLAB:

```
f=inline('-4*x^0');  
q=4; a=0; b=6;  
c=0; d=0; h=0.5;  
n=(b-a)/h;  
x(1)=a; y(1)=c;  
y2(1)=d;  
for i=1:n  
    x(i+1)=x(i)+h;  
    y(i+1)=y(i)+h*f(x(i));  
    y2(i+1)=y2(i)+h*(y(i));  
end  
[x' y2'], plot(x,y2,'r');
```

Zápis zdrojového textu programu lze vylepšit následujícím způsobem (řešená funkce $y(x)$ i její derivace $y'(x)$ je uložena v jedné proměnné):

```
f=inline('-4*x^0');  
q=4; a=0; b=6;  
c=0; d=0;  
h=0.5; x=a:h:b;  
f=inline('-4*x^0');  
y(1,:)= [d c];
```

[Obsah](#)[31. strana ze 46](#)[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

```

for i=2:length(x)
    k1=[y(i-1,2) -q];
    y(i,:)=y(i-1,:)+k1*h;
end
[x' y(:,1)], plot(x,y(:,1),'r');

```

Pro výpočetní krok $h = 0,5$ lze Eulerovou metodou získat tyto výsledky:

x	y	YP	y-YP
0.0000	0.0000	0.0000	0.0000
0.5000	0.0000	-0.5000	0.5000
1.0000	-1.0000	-2.0000	1.0000
1.5000	-3.0000	-4.5000	1.5000
2.0000	-6.0000	-8.0000	2.0000
2.5000	-10.0000	-12.5000	2.5000
3.0000	-15.0000	-18.0000	3.0000
3.5000	-21.0000	-24.5000	3.5000
4.0000	-28.0000	-32.0000	4.0000
4.5000	-36.0000	-40.5000	4.5000
5.0000	-45.0000	-50.0000	5.0000
5.5000	-55.0000	-60.5000	5.5000
6.0000	-66.0000	-72.0000	6.0000



Obsah

32. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.12. Diferenciální rovnici (1.25) z příkladu 1.11 pro určení průběhu ohybových momentů vyřešte s využitím klasické metody Runge-Kutta.

Řešení. Celý výpočetní postup je opět zřejmý ze zápisu programu s výpočetním krokem $h = 0,5$ do m-souboru programu MATLAB:

```
f=inline('-4*x^0');  
q=4; a=0; b=6; c=0; d=0; h=0.5;  
n=(b-a)/h;  
x(1)=a; y(1)=c; y2(1)=d;  
for i=1:n  
K1=f(x(i));  
K2=f(x(i)+h/2);  
K3=f(x(i)+h/2);  
K4=f(x(i)+h);  
y(i+1)=y(i)+h*((K1+K4)/6+(K2+K3)/3);  
K1=y(i);  
K2=(y(i)+y(i+1))/2;  
K3=(y(i)+y(i+1))/2;  
K4=y(i+1);  
y2(i+1)=y2(i)+h*((K1+K4)/6+(K2+K3)/3);  
x(i+1)=x(i)+h;  
end  
[x' y2'], plot(x,y2,'r');
```

I v tomto případě je možno zápis zdrojového textu programu vylepšit uložením řešení

[Obsah](#)

33. strana ze 46

[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

funkce $y(x)$ i její derivace $y'(x)$ do jediné proměnné:

```
f=inline('-4*x^0');  
q=4; a=0; b=6;  
c=0; d=0;  
h=0.5; x=a:h:b;  
f=inline('-4*x^0');  
y(1,:)= [d c];  
for i=2:length(x)  
    K1=f(x(i-1));  
    K2=f(x(i-1)+h/2);  
    K3=f(x(i-1)+h/2);  
    K4=f(x(i-1)+h);  
    y(i,2)=y(i-1,2)+h*((K1+K4)/6+(K2+K3)/3);  
    K1=y(i-1,2);  
    K2=(y(i-1,2)+y(i,2))/2;  
    K3=(y(i-1,2)+y(i,2))/2;  
    K4=y(i,2);  
    y(i,1)=y(i-1,1)+h*((K1+K4)/6+(K2+K3)/3);  
end  
[x' y(:,1)], plot(x,y(:,1),'r');
```

Pro výpočetní krok $h = 0,5$ lze metodou Runge-Kutta získat oproti Eulerově metodě podstatně přesnější výsledky:

[Obsah](#)[34. strana ze 46](#)[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

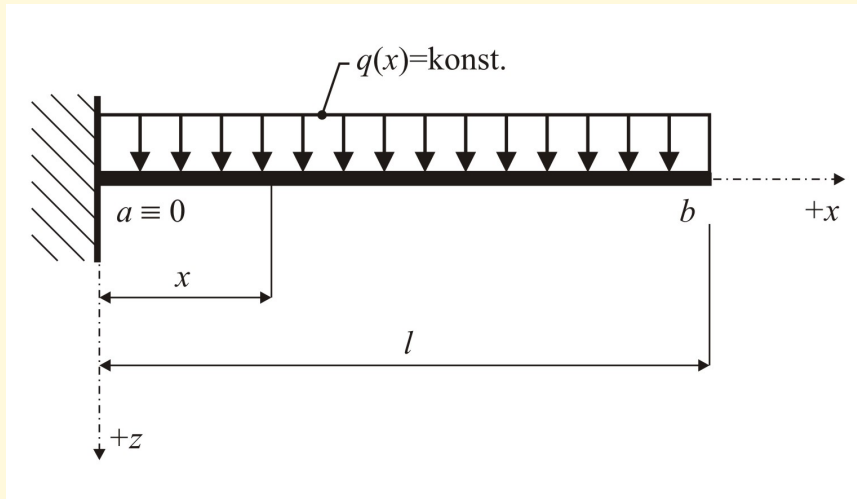
x	y	yp	y-yp
0.0000	0.0000	0.0000	0.0000
0.5000	-0.5000	-0.5000	0.0000
1.0000	-2.0000	-2.0000	0.0000
1.5000	-4.5000	-4.5000	0.0000
2.0000	-8.0000	-8.0000	0.0000
2.5000	-12.5000	-12.5000	0.0000
3.0000	-18.0000	-18.0000	0.0000
3.5000	-24.5000	-24.5000	0.0000
4.0000	-32.0000	-32.0000	0.0000
4.5000	-40.5000	-40.5000	0.0000
5.0000	-50.0000	-50.0000	0.0000
5.5000	-60.5000	-60.5000	0.0000
6.0000	-72.0000	-72.0000	0.0000



Příklad 1.13. Diferenciální rovnici (1.25) z příkladu 1.11 pro určení průběhu ohybových momentů vyřešte s využitím ostatních metod, které vycházejí z klasické metody Runge-Kutta, tedy Kuttovy metody třetího řádu, metody Heun-Euler, Ralstonovy metody a metody Bogacki-Shampine.

[Obsah](#)[35. strana ze 46](#)[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

Příklad 1.14. Stanovte tvar ohybové čáry konzolového nosníku, schématicky znázorněném na obr. 1.8. Konkrétní vstupní údaje jsou uvedeny v tabulce 1.8. K numerickému řešení použijte Eulerovu metodu. Výpočetní krok h zvolte $h = 0,5$, $h = 0,25$, $h = 0,1$, příp. $h = 0,01$. Výslednou aproximaci porovnejte s přesným řešením.



Obr. 1.8 Statické schéma řešeného staticky určitého konzolového nosníku



Obsah

36. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Spojité silové zatížení q_z :	4 kN/m
Rozpětí konzolového nosníku l :	3 m
Šířka obdélníkového průřezu b :	0,02 m
Výška obdélníkového průřezu h :	0,15 m
Moment setrvačnosti I_y :	$\frac{1}{12} \cdot 0,02 \cdot 0,15^3 = 5,625 \cdot 10^{-6} \text{ m}^4$
Modul pružnosti v tahu a tlaku E :	$2,1 \cdot 10^{11} \text{ Pa}$

Tab. 1.8 Vstupní údaje příkladu 1.14

Řešení. Obyčejná diferenciální rovnice druhého řádu nabývá tvaru:

$$EI_y w_z(x)'' = -M_y(x), \quad (1.28)$$

kde EI_y je ohybová tuhost nosníku (konstantní a nenulová).

Počáteční podmínky vychází z deformačních okrajových podmínek, které udávají nulovou hodnotu průhybu i pootočení ve vetknutí konzolového nosníku, tedy:

$$\varphi_y(x=0) = w_z'(x=0) = 0 \quad (1.29)$$

a

$$w_z(x=0) = y(x=0) = 0. \quad (1.30)$$

Z podmínek rovnováhy lze určit nejprve velikost silové reakce ve vetknutí konzoly:

$$R_{a,z} = q_z \cdot l (\uparrow), \quad (1.31)$$



Obsah

37. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

momentovou reakci ve vetknutí konzolového nosníku:

$$M_{a,y} = \frac{q_z \cdot l^2}{2} (\odot), \quad (1.32)$$

a nakonec i samotnou rovnici ohybového momentu:

$$M_y(x) = -\frac{q_z \cdot l^2}{2} + q_z \cdot l \cdot x - \frac{q_z \cdot x^2}{2} = q_z \cdot \left(-\frac{l^2}{2} + l \cdot x - \frac{x^2}{2} \right), \quad (1.33)$$

která figuruje v řešení diferenciální rovnici (1.28).

Výpočet Eulerovou metodou pak lze provést pro výpočetní krok $h = 0,25$ s využitím funkce `horner` a následujícího sledu příkazů:

```
qz=4000; l=3; E=2.1*10^11;
sirka=0.02; vyska=0.15; Iy=1/12*sirka*vyska^3;
M=[-qz/2*l^2 qz*l -qz/2];
a=0; b=1;
c=0; d=0;
h=0.25;
n=(b-a)/h;
x(1)=a; y(1)=c; y2(1)=d;
for i=1:n
    x(i+1)=x(i)+h;
    y(i+1)=y(i)-h*horner(2,M,x(i+1))/(E*Iy);
    y2(i+1)=y2(i)+h*y(i)*1000;
end
[x' y2'], plot(x,y2,'r');
```



Obsah

38. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Dosažené výsledky lze porovnat s přesnou hodnotou, vycházející z analyticky určené rovnice ohybové čáry:

$$w_z(x) = \frac{q_z}{EI_y} \cdot \left(\frac{l^2 \cdot x^2}{4} - \frac{l \cdot x^3}{6} + \frac{x^4}{24} \right). \quad (1.34)$$

Pro výpočetní krok $h = 0,25$ pak vycházejí následující numerické hodnoty výsledného řešení:

x	y1	y1p	y [mm]	yp [mm]	y-yp
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.2500	0.0032	0.0035	0.0000	0.4503	-0.4503
0.5000	0.0058	0.0064	0.8003	1.7019	-0.9017
0.7500	0.0080	0.0088	2.2619	3.6161	-1.3542
1.0000	0.0097	0.0107	4.2593	6.0670	-1.8078
1.2500	0.0110	0.0122	6.6799	8.9424	-2.2625
1.5000	0.0119	0.0133	9.4246	12.1429	-2.7183
1.7500	0.0126	0.0141	12.4074	15.5826	-3.1752
2.0000	0.0130	0.0147	15.5556	19.1887	-3.6332
2.2500	0.0133	0.0150	18.8095	22.9018	-4.0923
2.5000	0.0134	0.0152	22.1230	26.6755	-4.5525
2.7500	0.0134	0.0152	25.4630	30.4767	-5.0138
3.0000	0.0134	0.0152	28.8095	34.2857	-5.4762

Na obrázku 1.9 je zobrazena vypočtená aproximace ohybové čáry konzolového nosníku určená Eulerovou metodou pro výpočetní krok $h = 0,25$.



Obsah

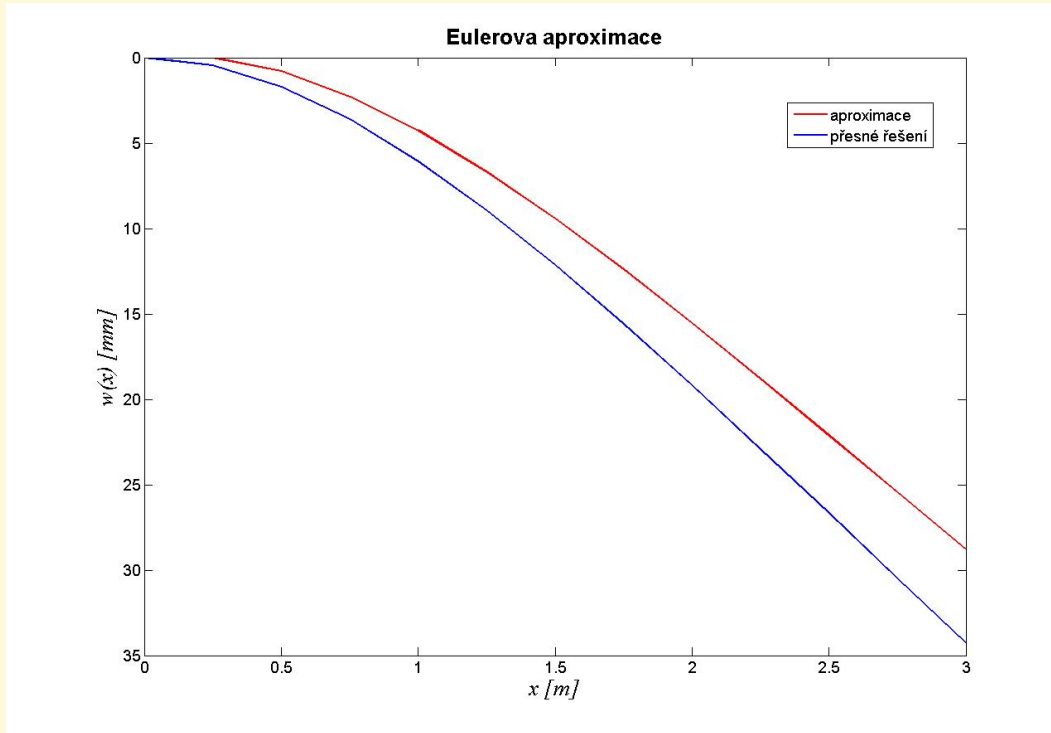
39. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.9 Výsledná aproximace ohybové čáry konzolového nosníku s výpočetním krokem $h = 0,25$



Obsah

40. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Příklad 1.15. Určete aproximaci ohybové čáry konzolového nosníku z příkladu 1.14 metodou Runge-Kutta.

Řešení. Výpočet aproximace ohybové čáry může být proveden m-skriptem:

```
qz=4000; l=3; E=2.1*10^11; sirka=0.02; vyska=0.15;
Iy=1/12*sirka*vyska^3; M=[-qz/2*l^2 qz*l -qz/2];
a=0; b=1; h=0.25; n=(b-a)/h;
c=0; d=0; x(1)=a; y(1)=c; y2(1)=d;
for i=1:n
    x(i+1)=x(i)+h;
    K1=horner(2,M,x(i))/(E*Iy);
    K2=horner(2,M,x(i)+h/2)/(E*Iy);
    K3=horner(2,M,x(i)+h/2)/(E*Iy);
    K4=horner(2,M,x(i)+h)/(E*Iy);
    y(i+1)=y(i)-h*((K1+K4)/6+(K2+K3)/3);
    K1=y(i)*1000;
    K2=(y(i)+y(i+1))*1000/2;
    K3=(y(i)+y(i+1))*1000/2;
    K4=y(i+1)*1000;
    y2(i+1)=y2(i)+h*((K1+K4)/6+(K2+K3)/3);
end
[x' y2'], plot(x,y2,'r');
```

Obsah

41. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Pro výpočetní krok $h = 0,5$ lze metodou Runge-Kutta získat oproti Eulerově metodě podstatně přesnější výsledky:

x	y1	y1p	y [mm]	yp [mm]	y-yp
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.2500	0.0035	0.0035	0.4376	0.4503	-0.0127
0.5000	0.0064	0.0064	1.6777	1.7019	-0.0243
0.7500	0.0088	0.0088	3.5813	3.6161	-0.0347
1.0000	0.0107	0.0107	6.0229	6.0670	-0.0441
1.2500	0.0122	0.0122	8.8900	8.9424	-0.0524
1.5000	0.0133	0.0133	12.0833	12.1429	-0.0595
1.7500	0.0141	0.0141	15.5170	15.5826	-0.0656
2.0000	0.0147	0.0147	19.1182	19.1887	-0.0705
2.2500	0.0150	0.0150	22.8274	22.9018	-0.0744
2.5000	0.0152	0.0152	26.5983	26.6755	-0.0772
2.7500	0.0152	0.0152	30.3979	30.4767	-0.0788
3.0000	0.0152	0.0152	34.2063	34.2857	-0.0794



Příklad 1.16. Určete aproximaci ohybové čáry konzolového nosníku z příkladu 1.14 s využitím ostatních metod, vycházejících z klasické metody Runge-Kutta, tedy Kuttovy metody třetího řádu, metody Heun-Euler, Ralstonovy metody a metody Bogacki-Shampine.



Obsah

42. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Příklad 1.17. Stanovte přibližné řešení obyčejné diferenciální rovnice druhého řádu:

$$y''(t) + 100 \cdot y'(t) + 10000 \cdot y(t) = 10000 \cdot |\sin(377 \cdot t)| \quad (1.35)$$

v intervalu $\langle 0; 0,08 \rangle$ s počátečními podmínkami $y(0) = 0$ a $y'(0) = 0$. Pro numerické řešení použijte Eulerovu metodu a klasickou metodu Runge-Kutta 4. řádu. Výpočetní krok h postupně volte $h = 0,01$, $h = 0,0025$ a $h = 0,0001$. Výslednou aproximaci porovnejte s řešením pomocí funkce `ode45` programu MATLAB .

Řešení. Programový systém MATLAB nedisponuje funkcemi pro řešení obyčejných diferenciálních rovnic druhého řádu. Řešení je založeno na převedení úlohy na soustavu dvou diferenciálních rovnic popsaných vztahy (1.21) a (1.22). V případě řešené úlohy se daný rozklad provede s využitím samostatné `m-funkce`, např.:

```
function y=fce(t,y);  
y=[y(2);-100*y(2)-10000*y(1)+10000*abs(sin(377*t))];
```

na kterou se v `m-skriptu` lze odkázat například takto:

```
t0=[0 0]; interval=[0,0.08]; options = odeset('AbsTol',1e-9);  
[t,y]=ode45(@fce,interval,t0,options); plot(t,y(:,1));
```



Obsah

43. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

Program, který provede výpočet řešení diferenciální rovnice 2. řádu Eulerovou metodou, klasickou metodou Runge-Kutta čtvrtého řádu a funkcí ode45 programu MATLAB pak pro výpočetní krok $h = 0,0025$ vypadá následovně:

```
a=0; b=0.08; h=0.0025; n=(b-a)/h;
c=0; d=0; x(1)=a; ye(1)=c; yrk(1)=c; y2e(1)=d; y2rk(1)=d;
for i=1:n
    x(i+1)=x(i)+h;
    ye(i+1)=ye(i)+h*(1E4*abs(sin(377*x(i)))-100*ye(i)-1E4*y2e(i));
    y2e(i+1)=y2e(i)+h*(ye(i));
    K1=h*(1E4*abs(sin(377*x(i)))-100*yrk(i)-1E4*y2rk(i));
    K2=h*(1E4*abs(sin(377*(x(i)+h/2)))-100*(yrk(i)+K1/2)-1E4*y2rk(i));
    K3=h*(1E4*abs(sin(377*(x(i)+h/2)))-100*(yrk(i)+K2/2)-1E4*y2rk(i));
    K4=h*(1E4*abs(sin(377*(x(i)+h)))-100*(yrk(i)+K3)-1E4*y2rk(i));
    yrk(i+1)=yrk(i)+(K1+2*K2+2*K3+K4)/6;
    K1=yrk(i); K2=(yrk(i)+yrk(i+1))/2;
    K3=(yrk(i)+yrk(i+1))/2; K4=yrk(i+1);
    y2rk(i+1)=y2rk(i)+h*((K1+K4)/6+(K2+K3)/3);
end
t0=[0 0]; interval=[0,0.08]; options=odeset('AbsTol',1e-9);
[xm,y2m]=ode45(@fce,interval,t0,options);
[x' y2e' y2rk'], plot(x,y2e,'r',x,y2rk,'b',xm,y2m(:,1),'k');
```

[Obsah](#)

44. strana ze 46

[Zavřít dokument](#)[Konec](#)[Celá obrazovka/Okno](#)

Numerické výsledky, získané Eulerovou metodou a metodou Runge-Kutta pro výpočetní krok $h = 0,0025$, jsou následující:

x	y_eul	y_rk
0.0000	0.0000	0.0000
0.0025	0.0000	0.0126
0.0050	0.0000	0.0610
0.0075	0.0506	0.1415
0.0100	0.1479	0.2242
0.0125	0.2371	0.3098
0.0150	0.3315	0.4091
0.0175	0.4499	0.4969
0.0200	0.5548	0.5663
0.0225	0.6246	0.6364
0.0250	0.7018	0.6956
0.0275	0.7712	0.7291
0.0300	0.7794	0.7547
0.0325	0.7879	0.7785
0.0350	0.8050	0.7803
0.0375	0.7879	0.7698
0.0400	0.7615	0.7662
0.0425	0.7549	0.7513
0.0450	0.7392	0.7238
0.0475	0.6995	0.7071



Obsah

45. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno

0.0500	0.6830	0.6923
0.0525	0.6774	0.6663
0.0550	0.6306	0.6475
0.0575	0.6037	0.6413
0.0600	0.6036	0.6266
0.0625	0.5850	0.6112
0.0650	0.5702	0.6125
0.0675	0.5849	0.6100
0.0700	0.5971	0.6003
0.0725	0.5890	0.6048
0.0750	0.6051	0.6128
0.0775	0.6308	0.6095
0.0800	0.6124	0.6122

Na obrázku 1.10 jsou zobrazeny vypočtené aproximace Eulerovou metodou, metodou Runge-Kutta i funkcí `ode45` programového systému MATLAB pro výpočetní krok $h = 0,0025$.



Obsah

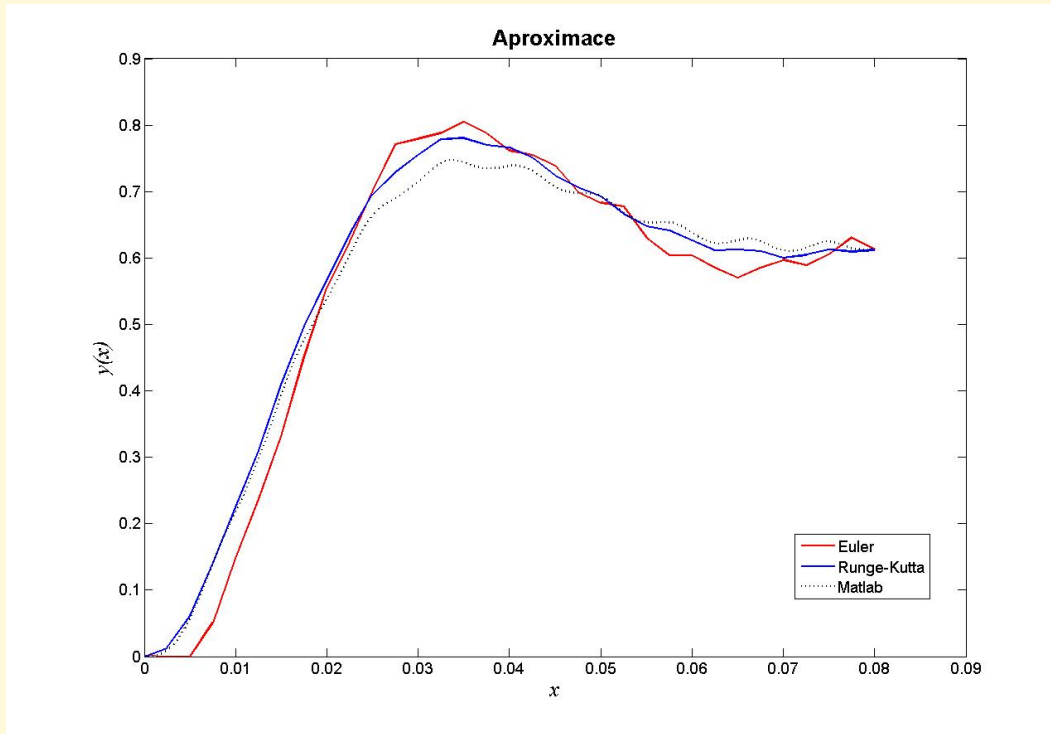
46. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno



Obr. 1.10 Srovnání dosažených aproximací Eulerovou metodou, metodou Runge-Kutta a funkcí ode45 pro výpočetní krok $h = 0,0025$



Obsah

47. strana ze 46



Zavřít dokument

Konec

Celá obrazovka/Okno