Numerická matematika

Zadání 25.

Řešení diferenciální rovnice

Rungovou – Kuttovou metodou

Václav Bubník, xbubni01, sk. 60 FIT VUT v Brně, 2004

Obsah

1. Teorie

1.1 Diferenciální rovnice

Diferenciální rovnice se využívají k popisu matematických modelů systémů, jejichž stavové proměnné se podle jistého zákona mění v závislosti na okamžitém stavu systému. Řešení, pokud existuje, není jediné. Abychom zaručili jednoznačnost řešení, požaduje ještě splnění tzv. počáteční podmínky.

Obyčejná diferenciální rovnici zapisujeme ve tvaru :

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

Hodnota proměnné x udává zpravidla buďto časový okamžik nebo prostorovou souřadnici. Diferenciální rovnicí je dáno směrové pole. V každém bodě roviny kterým prochází některé řešení této rovnice, je hodnota f (x, y) rovna směrnici tečny ke grafu tohoto řešení.

Počáteční podmínku zadáváme (pro levý kraj intervalu) ve tvaru :

$$y(x_0) = y_0$$

Některé diferenciální rovnice jsou analyticky řešitelné jen obtížně nebo je analyticky nelze řešit vůbec. Takové rovnice řešíme numericky kde se spokojíme s přibližným řešením. Řešení diferenciálních rovnic analyticky nám také umožňuje určit globální a lokální chyby použité numerické metody.

1.2 Diskretizace

Základem Rungovy – Kuttovy metody je diskretizace proměnných. Přibližné řešení se nekonstruuje jako spojitá funkce, ale hodnoty počítáme v konečném počtu bodů z intervalu <a,b>. Bodům říkáme uzlové body nebo uzly sítě. Množině všech bodů $\{x_0, x_1, \ldots , x_n\}$ říkáme síť. Rozdílu dvou sousedních bodů $h = x_{i+1} - x_i$ říkáme krok. Body sítě jsou obvykle od sebe stejně vzdálené tj. ekvidistantní a mluvíme o metodě s konstantním krokem. V případě, že jsou mezi body vzdálenosti různé a krok tak není stejný pro celý interval <a, b>, jedná se o metodu s proměnným krokem.

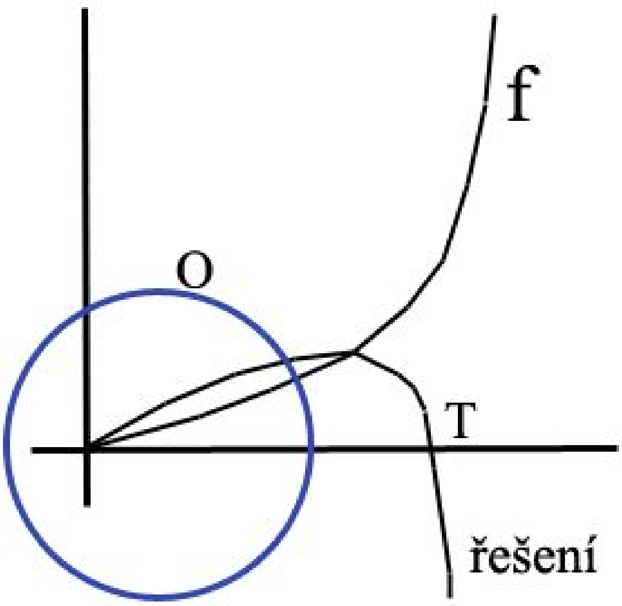
1.3 Jednokrokové numerické metody

Metoda se označuje jako jednokroková, počítáme – li přibližnou hodnotu řešení v daném bodě pouze pomocí hodnoty z předchozího kroku jak je tomu i u Rungovy – Kuttovy metody.

1.4 Rungova – Kuttova metoda 4. řádu

Rungova – Kuttova metoda čtvrtého řádu je modifikovanou Eulerovou metodou a spojuje vyvážení přesnosti a numerickou náročnost. Řešíme ji na konečném intervalu <a, b> s jeho počátečním zvoleným dělením krokem **h**. Hledaná aproximace je kombinací několika hodnot funkce f vypočítaných ve čtyřech strategicky zvolených bodech (x, y) na intervalu <x_n, x_{n+1}>. Získáme tak přesnější numerické řešení než u ostatních jednokrokových metod.

Přesnost dále podstatně ovlivňuje zvolený krok h, který souvisí s rychlostí nárůstu globální chyby. Zatímco v okolí počátku **O** je metoda relativně přesná, existuje bod **T** (závislý na zvoleném kroku) kde po nakupení chyb přechází řešení do záporné oblasti.



Obrázek 1: nestabilní rovnice (f = 0.2 x^2)

Rekurzivní zápis metody :

Hodnota závisí na hodnotě předchozího kroku plus přírůstek. Přírůstek počítáme ve tvaru odvození přes Taylorovu řadu.

k1 = $f(x_n, y_n)$

k2 = $f(x_n + 1/2\ h, y_n + 1/2\ h*k1)$

k3 = $f(x_n + 1/2\ h, y_n + 1/2\ h*k2)$

k4 = $f(x_n + h, y_n + h*k3)$

$y_{n+1} = y_n + 1/6\ h\ (k1 + 2*k2 + 2*k3 + k4)$

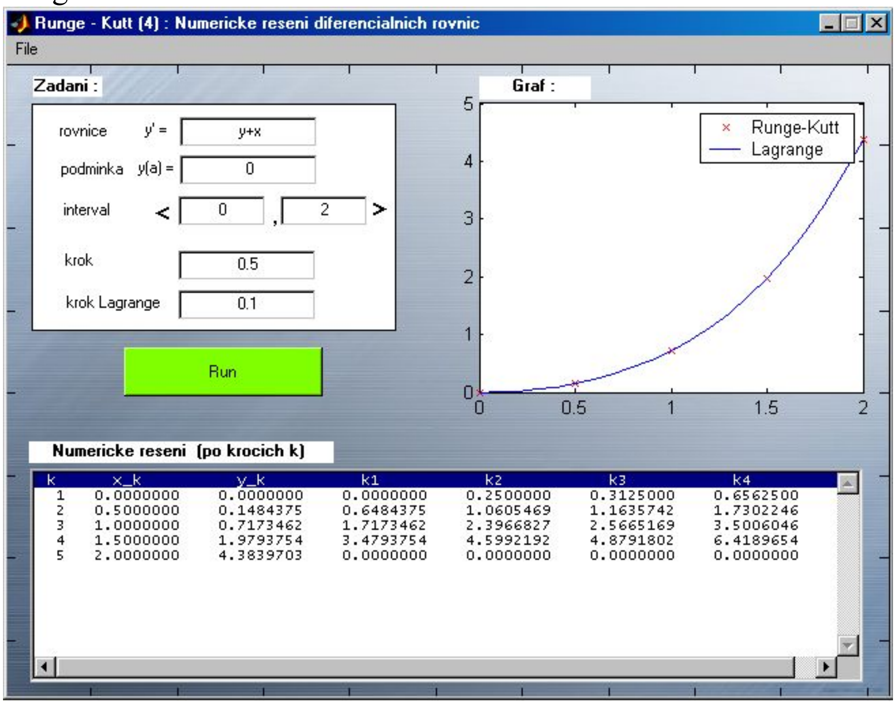
k1, k2, k3, k4 … výpočet čtyř pomocných výsledků

2. Program

2.1 Části programu

Program obsahuje dvě verze :

1. Dávkový *.m soubor **program.m** pro textový výstup a přímé zadávání parametrů přiřazováním do proměnných.
2. Aplikaci Matlabu (*.fig), spustitelnou přes **gui_rk4.m,** s grafickým uživatelským rozhraním. Po zadání úlohy v části „Zadání“ se výpočet spustí tlačítkem „Run“. Uživatelské vstupy zadávané klávesnicí do editačních okének jsou při spuštění zkontrolovány. Aplikace tak případně uživateli oznámí, které pole zapomněl vyplnit nebo jej vyplnil nepovolenou hodnotou tj. kde program očekává číselný vstup nebylo číslo a podobně. Při neuvedení hodnoty kroku Lagrangeova interpolačního polynomu nebude program polynom počítat a provede pouze Rungovu – Kuttovu metodu.



Obě verze jsou funkčně ekvivalentní. Používají společné funkce; pro výpočet numerickou metodou v souboru **runge_kutt4.m** a pro proložení vypočítaných hodnot Lagrangeovým polynomem funkci v souboru **my_lagrange.m**.

2.2 Spuštění

V Matlabu spustit dávku **gui_rk4.m**.

3. Řešený příklad

Řešte počáteční diferenciální rovnici **$y'=x / y$** na intervalu **<1, 2.5>** s počáteční podmínkou **$y(1) = 3$** a krokem **$h = 0,5$**.

Přesné řešení : $y^2 = x^2 + c$ $y(1) = 3$ $3^2 = 1^2+c$ $9 = 1+c$ $c=8$

$y = (x^2 + 8)^{0,5}$ je použito k určení lokální chyby e_i

Numerické řešení :

i	x_i	y_i	***k***	e_i
1	1	**3**	$k1 = x_i/y_i = 1 / 3 =$ **0,3333333**	0
	1,25		$k2 = (x_i+h/2) / (y_i+h/2*k1) = 1,25 / (3 + 0,25 * 0,3333333) =$ **0,4054054**	
	1,25		$k3 = (x_i+h/2) / (y_i+h/2*k2) = 1,25 / (3 + 0,25 * 0,4054054) =$ **0,4030501**	
	1,5		$k4 = (x_i+h) / (y_i+h*k3) = 1,5 / (3 + 0,5 * 0,4030501) =$ **0,4685267**	

$y_{i+1} = y_i + h/6$ **(k1 + 2***k2 + 2*k3 + k4)

$y_2 = 3 + 0,5 / 6 *(0,3333333 + 2*0,4054054 + 2*0,4030501+0,4685267) =$ **3,20156425**

i	x_i	y_i	***k***	e_i
2	1,5	**3,2015643**	k1 = 1,5 / 3,2015643 = **0,4685210**	0,0000022
	1,75		k2 = 1,75 / (3,2015643 + 0,25 * 0,4685210) = **0,5273158**	
	1,75		k3 = 1,75 / (3,2015643 + 0,25 * 0,5273158) = **0,5249906**	
	2,0		k4 = 2,0 / (3,2015643 + 0,5 * 0,5249906) = **0,5773573**	

$y_3 = 3,2015643 + 0,5 / 6 *(0,4685210 + 2*0,5273158 + 2*0,5249906+0,5773573) =$ **3,4641052**

i	x_i	y_i	***k***	e_i
3	2,0	**3,4641052**	k1 = 2,0 / 3,4641052 = **0,5773497**	0,0000036
	2,25		k2 = 2,25 / (3,4641052 + 0,25 * 0,5773497) = **0,6235377**	
	2,25		k3 = 2,25 / (3,4641052 + 0,25 *0,6235377) = **0,6215487**	
	2,5		k4 = 2,5 / (3,4641052 + 0,5 * 0,6215487) = **0,6622728**	

$y_4 = 3,4641052 + 0,5 / 6 *(0,5773497 + 2*0,6235377 + 2*0,6215487+0,6622728) =$ **3,7749215**

i	x_i	y_i	***k***	e_i
4	2,5	**3,7749215**		0,0000043