

Poglavje 6

Navadne diferencialne enačbe

6.1 Uvod

Splošna rešitev navadne diferencialne enačbe n -tega reda

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

je n -parametrična družina funkcij. Kadar želimo iz splošne rešitve izločiti eno izmed partikularnih rešitev, potrebujemo dodatno informacijo, ki je navadno podana kot vrednost te izbrane partikularne rešitve in/ali njenih odvodov pri eni ali več vrednostih neodvisne spremenljivke x . Pri enačbi n -tega reda je navadno dovolj n dodatnih pogojev. Če so ti pogoji dani kot vrednosti izbrane rešitve in njenih odvodov v eni sami točki, govorimo o *začetnem problemu*. Če so te vrednosti dane v dveh ali več različnih točkah, imamo *robni problem*.

V tem poglavju se bomo ukvarjali predvsem z reševanjem začetnih problemov pri diferencialnih enačbah prvega reda

$$y' = f(x, y) \quad \text{in} \quad y(x_0) = y_0. \quad (6.1)$$

Metode, ki jih bomo uporabljali za reševanje problema (6.1), bomo kasneje posplošili tudi za reševanje začetnih problemov pri sistemih navadnih diferencialnih enačb in enačb višjega reda, pa tudi za reševanje enostavnih robnih problemov. Ker enačbo (6.1) lahko analitično rešimo le v posebno enostavnih primerih, se moramo navadno zadovoljiti le z numerično rešitvijo. Na intervalu $[x_0, b]$, na katerem želimo izračunati rešitev, si izberemo zaporedje točk

$$x_1 < x_2 < \dots < x_n < \dots,$$

razdalje med njimi pa označimo s $h_i = x_{i+1} - x_i$. Če so vse točke x_i med seboj enako oddaljene, je $h_i = h$ konstanta. Rešitev danega začetnega problema (6.1) dobimo kot zaporedje približkov y_i za vrednosti točne rešitve $y(x_i)$ v izbranih točkah. Za odvod funkcije v točki približne rešitve bomo uporabljali skrajšano oznako $f_i = f(x_i, y_i)$.

Pri numeričnem reševanju začetnih problemov nas predvsem zanima razlika med numerično in točno rešitvijo

$$g_n = y_n - y(x_n),$$

ki ji pravimo *globalna napaka*. Na njeno velikost vplivajo različni faktorji, kot so: izbrana numerična metoda, velikost koraka h_i , obnašanje točne rešitve $y(x)$ začetnega problema, računalnik, s katerim računamo, ... Tu si bomo ogledali le, kako na velikost globalne napake vplivata izbor metode in velikost koraka, ostale vplive bomo zanemarili. Globalna napaka je posledica napake, ki jo naredimo na vsakem posameznem koraku. *Lokalna napaka* metode v točki x_{n+1} je razlika med numeričnim približkom y_{n+1} in tisto rešitvijo diferencialne enačbe, ki poteka skozi prejšnjo točko numerične rešitve (x_n, y_n) , to je rešitvijo začetnega problema

$$z' = f(x, z), \quad z(x_n) = y_n, \quad (6.2)$$

izračunano v točki x_{n+1} :

$$l_{n+1} = y_{n+1} - z(x_{n+1}).$$

Iz slike 6.1 vidimo, da je globalna napaka numerične rešitve v neki točki posledica globalne napake v prejšnji točki in lokalne napake v zadnjem koraku.

Definicija 6.1.1. Red metode za reševanje diferencialnih enačb je celo število p , za katerega je

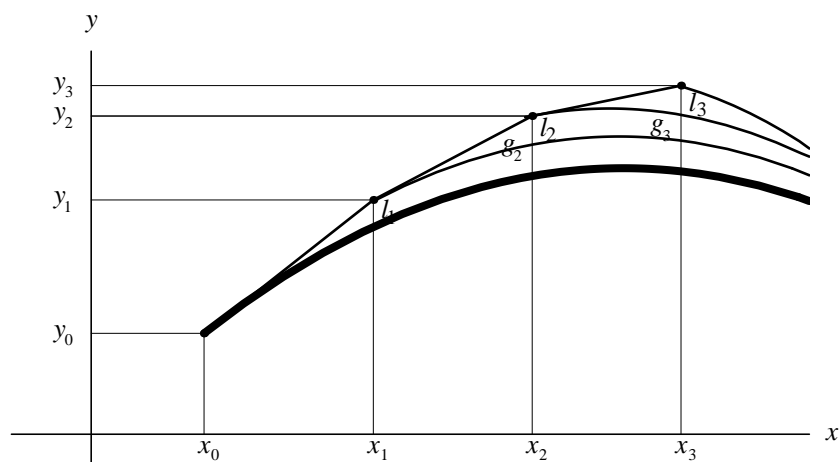
$$l_n = Ch_n^{p+1} + O(h_n^{p+2}).$$

Konstanta C je poznana kot konstanta napake.

6.2 Eulerjeva metoda

Najenostavnejša metoda za reševanje začetnih problemov (6.1) je *Eulerjeva metoda*¹. Izpeljemo jo lahko iz Taylorjeve formule za točno rešitev začetnega

¹Leonhard Euler (1707 Basel – 1783 St Petersburg) švicarski matematik, eden izmed najpomembnejših matematikov. Večino življenja je preživel v Berlinu in St. Petersburgu. Poleg matematike se je ukvarjal tudi z astronomijo, in fiziko.



Slika 6.1: Globalna napaka numerične rešitve je posledica globalne napake v prejšnji točki in lokalne napake v zadnjem koraku

problema

$$y(x_n + h_n) = y(x_n) + h_n y'(x_n) + \frac{h_n^2}{2} y''(\xi_n); \quad \xi \in (x_n, x_n + h_n). \quad (6.3)$$

Če zanemarimo člen s h_n^2 , dobimo pravilo, po katerem iz znane vrednosti y_n izračunamo naslednjo vrednost v zaporedju približkov k rešitvi

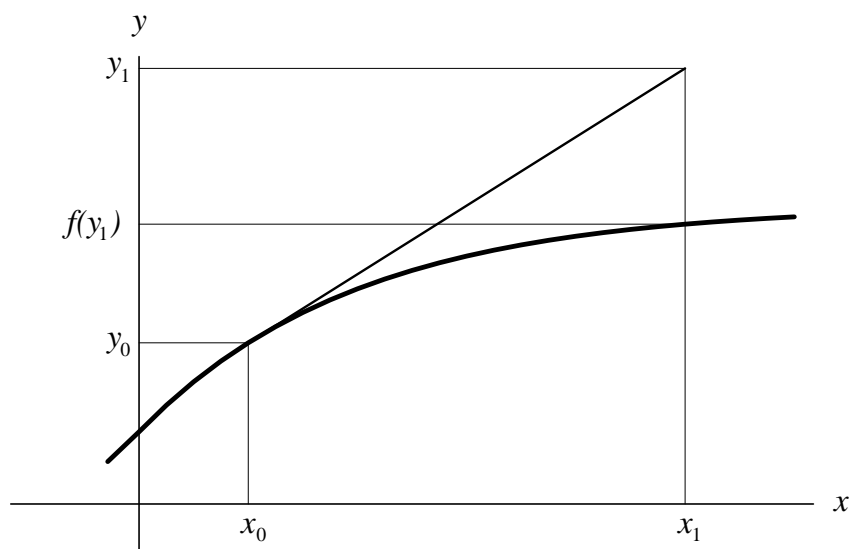
$$y_{n+1} = y_n + h_n f_n. \quad (6.4)$$

Geometrijsko Eulerjeva formula (6.4) pomeni, da rešitev enačbe med točkama x_i in x_{i+1} nadomestimo z odsekom premice iz točke x_i v smeri, ki je določena z odvodom f_i do točke x_{i+1} (slika 6.2).

Iz enačbe (6.3) vidimo, da je lokalna napaka Eulerjeve metode enaka $h^2 y''(\xi)/2$, torej je Eulerjeva metoda prvega reda.

Zapišimo postopek za reševanje začetnega problema z Eulerjevo metodo v obliki algoritma:

Algoritem 6.2.1 (Eulerjeva metoda). Naj bo $y' = f(x, y)$ diferencialna enačba, $y(x_0) = y_0$ začetni pogoj, h dolžina koraka in N naravno število. Naslednji algoritem izračuna zaporedje približkov y_n k vrednostim točne rešitve $y(x_n)$ diferencialne enačbe v izbranih točkah $x_n = x_0 + nh$; $n = 1, \dots, N$ s pomočjo Eulerjeve metode.



Slika 6.2: Rešitev diferencialne enačbe z Eulerjevo metodo

```

y = y_0
x = x_0
for n = 1 : N
    y = y + h * f(x, y)
    x = x + h
end

```

Primer 6.2.1. Poiščimo približno rešitev začetnega problema

$$y' = -y + 1 \quad \text{in} \quad y(0) = 2 \quad (6.5)$$

na intervalu $[0, 1]$ z Eulerjevo metodo pri koraku $h = 0.1$.

Iz rešitve v tabeli 6.1 in na sliki 6.3 vidimo, da numerična rešitev sicer spremlja točno rešitev, vendar je globalna napaka dokaj velika. ■

6.3 Linearne veččlenske metode

Eulerjeva metoda je preprosta, vendar je za resno uporabo premalo natančna. Natančnejše metode lahko dobimo z različnimi posplošitvami Eulerjeve me-

n	x_n	y_n	$y_n - y(x_n)$
0	0.0	2.0000	-0.0000
1	0.1	1.9000	-0.0048
2	0.2	1.8100	-0.0087
3	0.3	1.7290	-0.0118
4	0.4	1.6561	-0.0142
5	0.5	1.5905	-0.0160
6	0.6	1.5314	-0.0174
7	0.7	1.4782	-0.0183
8	0.8	1.4305	-0.0189
9	0.9	1.3874	-0.0191
10	1.0	1.3487	-0.0192

Tabela 6.1: Rešitev začetnega problema z Eulerjevo metodo

tode. V tem razdelku si bomo ogledali metode, pri katerih za izračun vrednosti v naslednji točki (x_{n+1}) poleg vrednosti rešitve y_n in odvoda f_n v točki x_n uporabimo več že prej izračunanih vrednosti in odvodov v točkah x_{n-1}, x_{n-2}, \dots . Take metode imenujemo *linearne veččlenske metode*.

Adams-Bashforthove metode Vzemimo, da smo pri konstantni dolžini koraka h že izračunali približne rešitve na množici ekvidistantnih točk $x_i = x_0 + ih; i = 0, \dots, n$. Diferencialno enačbo $y' = f(x, y)$ integriramo med x_n in x_{n+1} in dobimo

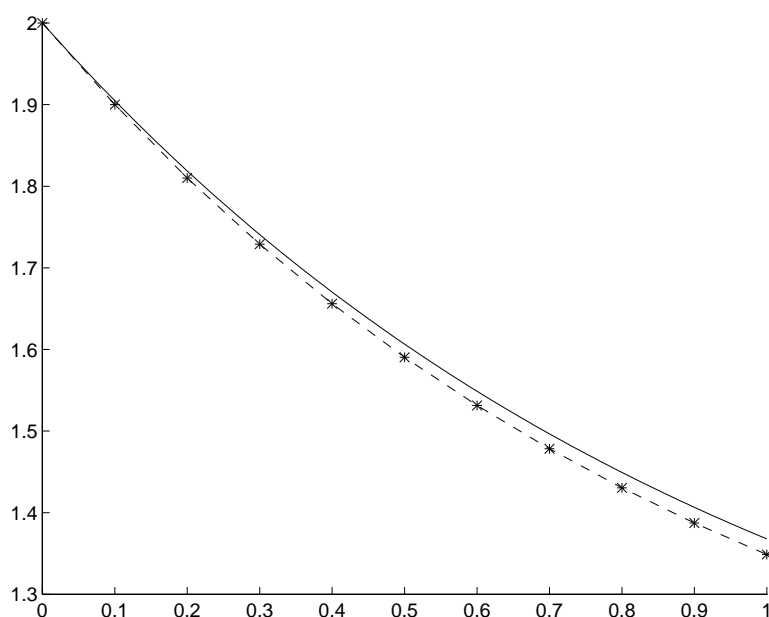
$$\int_{x_n}^{x_{n+1}} y' dx = \int_{x_n}^{x_{n+1}} f(x, y(x)) dx,$$

kar lahko zapišemo kot

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx. \quad (6.6)$$

Če integral na desni izračunamo tako, da $f(x, y(x))$ zamenjamo z interpolacijskim polinomom skozi točke $x_n, x_{n-1}, \dots, x_{n-k+1}$, dobimo k -člensko *Adams²*-

²John Couch Adams (1819 Laneast (Anglija) – 1892 Cambridge), Angleški astronom, velika je njegova vloga pri odkrivanju planeta Neptuna. Metodo za numerično integracijo diferencialnih enačb je objavil skupaj z F. Bashforthom leta 1883.



Slika 6.3: Točna in numerična rešitev začetnega problema (6.5)

*Bashforthovo*³ metodo (AB metodo). (Glej problem 1) Navadno jo zapišemo v obliki

$$y_{n+1} = y_n + h \sum_{i=1}^k \beta_i f_{n-i+1}. \quad (6.7)$$

Napaka k -členske Adams-Bashforthove metode je $C_{k+1}h^{k+1}y^{(k+1)}(\xi_n)$. Konstanti C_{k+1} pravimo *konstanta napake*. Tako ima k -členska Adams-Bashforthova metoda red k . Koeficienti prvih šestih AB metod z ustreznimi konstantami napake so zbrani v tabeli 6.2. Prvo ($k = 1$) izmed njih, $y_{n+1} = y_n + hf_n$, že poznamo, to je Eulerjeva metoda (6.4).

Za reševanje začetnega problema s k -člensko AB metodo ($k \geq 2$) potrebujemo poleg začetne vrednosti y_0 še $k - 1$ vrednosti y_1, \dots, y_{k-1} , ki jih moramo izračunati s kakšno drugo metodo (na primer z metodo Runge-Kutta iz naslednjega razdelka), ki ima isti red natančnosti kot AB metoda.

³Francis Bashforth (1819 Doncaster – 1912 Woodhall Spa (Anglija)), angleški gradbeni inženir in balistik. Na univerzi v Cambridgu je bil sošolec Johna Adamsa. V mladih letih je deloval v gradbeništvu, pozneje je bil profesor uporabne matematike in balistike na vojaški akademiji v Woolwichu.

k	i	1	2	3	4	5	6	C_{k+1}
1	β_i	1						$\frac{1}{2}$
2	$2\beta_i$	3	-1					$\frac{5}{12}$
3	$12\beta_i$	23	-16	5				$\frac{3}{8}$
4	$24\beta_i$	55	-59	37	-9			$\frac{251}{720}$
5	$720\beta_i$	1901	-2774	2616	-1274	251		$\frac{95}{288}$
6	$1440\beta_i$	4277	-7923	9982	-7298	2877	-475	$\frac{19087}{60480}$

Tabela 6.2: Koeficienti Adams–Bashforthovih metod.

Zapišimo algoritem za reševanje začetnega problema z AB metodo:

Algoritem 6.3.1 (AB metoda). Naj bo $y' = f(x, y)$ diferencialna enačba, $y(x_0) = y_0$ začetni pogoj, h korak in N naravno število. Naslednji algoritem izračuna zaporedne približke y_n k vrednostim rešitve $y(x_n)$ diferencialne enačbe v točkah $x_n = x_0 + nh$; $n = 1, \dots, N$ s pomočjo AB metode 4. reda.

```

 $y = y_0$ 
 $x = x_0$ 
 $d(1) = f(x, y)$ 
for  $n = 2 : 4$ 
     $x = x + h$ 
    Izračunaj dodatno vrednost  $y \approx y(x)$ 
     $d(n) = f(x, y)$ 
end
for  $n = 4 : N$ 
     $x = x + h$ 
     $y = y + h * (55 * d(4) - 59 * d(3) + 37 * d(2) - 9 * d(1)) / 24$ 
     $d(5) = f(x, y)$ 
    for  $j = 1 : 4$ 
         $d(j) = d(j + 1)$ 
    end
end

```

Primer 6.3.1. Poiščimo približno rešitev začetnega problema (6.5) na intervalu $[0, 1]$ še z Adams-Bashforthovo metodo 4. reda pri enakem koraku $h = 0.1$ kot v primeru 6.2.1. Manjkajoče začetne vrednosti določimo kar iz točne rešitve $y = e^{-x} + 1$.

n	x_n	y_n	$y_n - y(x_n)$
0	0.0	2.0000	0
1	0.1	1.9049	0
2	0.2	1.8187	0
3	0.3	1.7408	0
4	0.4	1.6703	$2.9 \cdot 10^{-6}$
5	0.5	1.6065	$4.8 \cdot 10^{-6}$
6	0.6	1.5488	$6.8 \cdot 10^{-6}$
7	0.7	1.4966	$8.1 \cdot 10^{-6}$
8	0.8	1.4493	$9.2 \cdot 10^{-6}$
9	0.9	1.4066	$1.0 \cdot 10^{-5}$
10	1.0	1.3679	$1.1 \cdot 10^{-5}$

Tabela 6.3: Rešitev začetnega problema z Adams-Bashforthovo metodo 4. reda

Iz rezultatov v tabeli 6.3 lahko ugotovimo, da je napaka približkov, izračunanih z AB metodo 4. reda, dosti manjša od napake približkov, izračunanih z Eulerjevo metodo. ■

Adams-Moultonove metode Če integral na desni strani enačbe (6.6) izračunamo tako, da interpolacijskim točkam dodamo tudi x_{n+1} , dobimo k -člensko *Adams-Moultonovo*⁴ metodo (AM metodo) (Glej problem 2). Zapišemo jo kot

$$y_{n+1} = y_n + h \sum_{i=0}^k \beta_i^* f_{n-i+1}. \quad (6.8)$$

Napaka k -členske Adams-Moultonove metode je $C_{k+2}^* h^{k+2} y^{(k+1)}(\xi_n)$, kjer konstanti C_{k+2}^* zopet pravimo *konstanta napake*. Tako ima k -členska Adams-Moultonova metoda red $k + 1$. Koeficienti prvih šestih AM metod z ustreznimi konstantami napake so zbrani v tabeli 6.2. Metoda, ki jo dobimo

⁴Forest Ray Moulton (1872 LeRoy, Michigan – 1952 Chicago), ameriški astronom. Adamsovo metodo za reševanje diferencialnih enačb je izpopolnil, da bi natančneje izračunaval balistične trajektorije, objavil jo je leta 1926.

za $k = 0$

$$y_{n+1} = y_n + hf_{n+1} \quad (6.9)$$

je poznana kot *implicitna Eulerjeva metoda*, metoda s $k = 1$

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1})$$

pa je poznana pod imenom *trapezna metoda*.

k	i	0	1	2	3	4	5	C_{k+1}^*
0	β_i^*	1						$-\frac{1}{2}$
1	$2\beta_i^*$	1	1					$-\frac{1}{12}$
2	$12\beta_i^*$	5	8	-1				$-\frac{1}{24}$
3	$24\beta_i^*$	9	19	-5	1			$-\frac{19}{720}$
4	$720\beta_i^*$	251	646	-264	106	-19		$-\frac{3}{160}$
5	$1440\beta_i^*$	475	1427	-798	482	-173	27	$-\frac{863}{60480}$

Tabela 6.4: Koeficienti Adams–Moultonovih metod.

Za razliko od AB metod, nastopa pri AM metodah neznana vrednost y_{n+1} na obeh straneh enačbe (6.8), tako da moramo na vsakem koraku rešiti nelinearno enačbo

$$y_{n+1} = y_n + h\beta_0^* f(x_{n+1}, y_{n+1}) + h \sum_{i=1}^k \beta_i^* f_{n-i+1}. \quad (6.10)$$

Metodam, pri katerih moramo vrednost y_{n+1} določiti kot rešitev enačbe, pravimo *implicitne* metode, za razliko od *eksplicitnih* metod (kot so AB metode), pri katerih izračunamo y_{n+1} direktno.

V praksi rešujemo enačbo (6.10) običajno z navadno iteracijo. Če imamo dober začetni približek, ki ga navadno dobimo z ustrezno eksplicitno AB metodo istega reda, ki ji v tem primeru pravimo *prediktor*, in dovolj majhen korak h , je večinoma dovolj že ena iteracija (implicitni AM formuli v tem primeru pravimo *korektor*). Taka kombinacija AB in AM metode je znana pod imenom *metoda prediktor-korektor*. Zapišimo še ustrezen algoritem:

Algoritem 6.3.2 (ABM prediktor-korektor). Naj bo $y' = f(x, y)$ diferencialna enačba, $y(x_0) = y_0$ začetni pogoj, h korak in N naravno število. Naslednji algoritem izračuna zaporedje približkov y_n k vrednostim točne rešitve

$y(x_n)$ diferencialne enačbe v izbranih točkah $x_n = x_0 + nh$; $n = 1, \dots, N$ s pomočjo ABM metode prediktor-korektor 4. reda.

```

 $x = x_0, y = y_0$ 
 $d(1) = f(x, y)$ 
for  $n = 2 : 4$ 
     $x = x + h$ 
    Izračunaj dodatno vrednosti  $y \approx y(x)$ 
     $d(n) = f(x, y)$ 
end
for  $n = 4 : N$ 
     $x = x + h$ 
     $yp = y + h * (55 * d(4) - 59 * d(3) + 37 * d(2) - 9 * d(1))/24$ 
     $dp = f(x, yp)$ 
     $y = y + h * (9 * dp + 19 * d(4) - 5 * d(3) + d(2))/24$ 
     $d(5) = f(x, y)$ 
    for  $j = 1 : 4$ 
         $d(j) = d(j + 1)$ 
    end
end

```

Primer 6.3.2. Poiščimo približno rešitev začetnega problema (6.5) na intervalu $[0, 1]$ še z ABM metodo prediktor-korektor 4. reda pri enakem koraku $h = 0.1$ kot v obeh prejšnjih primerih 6.2.1 in 6.3.1. Manjkajoče začetne vrednosti določimo kar iz točne rešitve $y = e^{-x} + 1$.

Iz rezultatov v tabeli 6.5 vidimo, da je napaka približkov, izračunanih z metodo prediktor-korektor za velikostni razred manjša od napak približkov, ki smo jih izračunali samo s prediktorjem (primer 6.3.1). ■

6.4 Metode Runge-Kutta

Pri Eulerjevi metodi (6.4) vzamemo za vrednost odvoda rešitve na intervalu $[x_n, x_{n+1}]$ kar vrednost odvoda rešitve v začetni točki intervala $f'(x_n)$. Pri

n	x_n	y_n	$y_n - y(x_n)$
0	0.0	2.0000	0
1	0.1	1.9048	0
2	0.2	1.8187	0
3	0.3	1.7408	0
4	0.4	1.6703	$-3.1 \cdot 10^{-7}$
5	0.5	1.6065	$-5.6 \cdot 10^{-7}$
6	0.6	1.5488	$-7.5 \cdot 10^{-6}$
7	0.7	1.4966	$-9.1 \cdot 10^{-6}$
8	0.8	1.4493	$-1.0 \cdot 10^{-6}$
9	0.9	1.4066	$-1.1 \cdot 10^{-6}$
10	1.0	1.3679	$-1.2 \cdot 10^{-6}$

Tabela 6.5: Rešitev začetnega problema z ABM metodo prediktor-korektor 4. reda

metodah Runge⁵-Kutta⁶ dosežemo večjo natančnost tako, da vrednosti odvoda izračunamo v več točkah na intervalu x_n, x_{n+1} in pri računanju vrednosti rešitve v točki x_{n+1} upoštevamo njihovo primerno obteženo povprečje. Kot primer izračunajmo koeficiente a, b_1, b_2 in c preproste dvostopenjske metode Runge-Kutta

$$y_{n+1} = y_n + b_1 k_1 + b_2 k_2, \quad (6.11)$$

kjer sta hk_1 in hk_2 približka za vrednost odvoda v točkah x_n in $x_n + ch$. Izračunamo ju iz

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + ch, y_n + ak_1). \end{aligned}$$

Točno rešitev $y(x_n + h)$ razvijemo v Taylorjevo vrsto po potencah h in dobimo

⁵Carle David Tolmé Runge (1856 Bremen – 1927 Göttingen), nemški matematik in fizik, profesor na univerzah v Hannoveru in Göttingenu. Svoj prvi članek o numeričnem reševanju diferencialnih enačb je objavil leta 1895.

⁶Martin Wilhelm Kutta (1867 Pitschen (sedaj Byczyna, Poljska) – 1944 Fürstfeldbruck), nemški matematik, profesor na univerzi v Stuttgartu. Na osnovi Rungejeve ideje je objavil izboljšano verzijo metode Runge-Kutta leta 1901.

$$\begin{aligned}
y(x_n + h) &= y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \dots \\
&= y_n + hf_n + \frac{h^2}{2}(f_x + ff_y)_n \\
&\quad + \frac{h^3}{6}(f_{xx} + 2ff_{xy} + f_{yy}f^2 + f_xf_y + f_y^2f)_n + \dots,
\end{aligned} \tag{6.12}$$

kjer smo z indeksom n označili, da je potrebno vse funkcije izračunati v točki (x_n, y_n) .

Da bi dobili Taylorjevo vrsto za numerično rešitev (6.11), razvijemo najprej k_2 :

$$k_2 = hf_n + h^2(cf_x + aff_y)_n + h^3 \left(\frac{c^2}{2}f_{xx} + acff_{xy} + \frac{a^2}{2}f_{yy} \right)_n + \dots,$$

kar vstavimo v (6.11), da dobimo

$$\begin{aligned}
y_{n+1} &= y_n + (b_1 + b_2)hf_n + b_2h^2(cf_x + aff_y)_n + \\
&\quad b_2h^3 \left(\frac{c^2}{2}f_{xx} + acff_{xy} + \frac{a^2}{2}f^2f_{yy} \right)_n + \dots
\end{aligned} \tag{6.13}$$

S primerjavo vrst (6.12) in (6.13) ugotovimo, da se vrsti ujemata v členih s h in h^2 , če so izpolnjene enačbe

$$\begin{aligned}
b_1 + b_2 &= 1 \\
ab_2 = cb_2 &= \frac{1}{2}.
\end{aligned}$$

Ta sistem ima enoparametrično družino rešitev. Najpreprostejši rešitvi dobimo, če si izberemo $b_2 = 1/2$ ali $b_2 = 1$. V prvem primeru imamo metodo

$$\begin{aligned}
k_1 &= hf(x_n, y_n) \\
k_2 &= hf(x_n + h, y_n + k_1) \\
y_{n+1} &= y_n + (k_1 + k_2)/2,
\end{aligned} \tag{6.14}$$

v drugem pa

$$\begin{aligned}
k_1 &= hf(x_n, y_n) \\
k_2 &= hf(x_n + h/2, y_n + k_1/2) \\
y_{n+1} &= y_n + k_2,
\end{aligned} \tag{6.15}$$

kar sta obe metodi Runge-Kutta drugege reda.

Splošno s -stopenjsko eksplisitno metodo Runge-Kutta lahko zapišemo kot

$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

kjer števila k_1, \dots, k_s izračunamo zaporedoma kot

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + c_2 h, y_n + a_{2,1} k_1) \\ &\vdots \\ k_s &= hf(x_n + c_s h, y_n + \sum_{j=1}^{s-1} a_{s,j} k_j). \end{aligned}$$

Koeficiente $a_{i,j}$, b_i in c_i lahko pregledno zapišemo v obliki tabele (*Butcherjeva tabela*⁷)

0	0				
c_2	$a_{2,1}$	0			
c_3	$a_{3,1}$	$a_{3,2}$	0		
\vdots	\vdots	\ddots			
c_s	$a_{s,1}$	$a_{s,2}$	\cdots	$a_{s,s-1}$	0
	b_1	b_2	\cdots	b_{s-1}	b_s

kjer so koeficienti c_i določeni s koeficienti a_{ij} kot

$$c_i = \sum_{j=1}^{i-1} a_{ij}.$$

Pri večjem številu stopenj s postanejo enačbe, ki določajo koeficiente zelo zapletene. Najpopularnejša in najpogosteje uporabljana je metoda Runge-Kutta četrtega reda

⁷John Butcher (1933 Auckland (Nova Zelandija)), novozelandski matematik, profesor računalništva in matematike na univerzi v Aucklandu. Najzaslužnejši za razvoj teorije in prakse metod Runge Kutta. Leta 1963 je objavil članek, ki je omogočil teoretično analizo in nadaljni razvoj teh metod.

$$\begin{aligned}
k_1 &= hf_n \\
k_2 &= hf(x_n + h/2, y_n + k_1/2) \\
k_3 &= hf(x_n + h/2, y_n + k_2/2) \\
k_4 &= hf(x_n + h, y_n + k_3) \\
y_{n+1} &= y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6.
\end{aligned}$$

Zapišimo jo še kot algoritem

Algoritem 6.4.1 (Runge-Kutta 4. reda). Naj bo $y' = f(x, y)$ diferencialna enačba, $y(x_0) = y_0$ začetni pogoj in N naravno število. Naslednji algoritem izračuna vektor približkov y_n k vrednostim rešitve $y(x_n)$ diferencialne enačbe v točkah $x_n = x_0 + nh$; $n = 1, \dots, N$ s pomočjo metode Runge-Kutta 4. reda.

```

x = x0
y = y0
for n = 1 : N
    k1 = h * f(x, y)
    k2 = h * f(x + h/2, y + k1/2)
    k3 = h * f(x + h/2, y + k2/2)
    k4 = h * f(x + h, y + k3)
    y = y + (k1 + 2 * k2 + 2 * k3 + k4)/6
    x = x + h
end

```

Primer 6.4.1. Poiščimo približno rešitev začetnega problema (6.5) na intervalu $[0, 1]$ tudi z metodo Runge-Kutta 4. reda z enako dolžino koraka $h = 0.1$ kot v prejšnjih primerih.

Iz rezultatov v tabeli 6.6 vidimo, da je napaka pri računanju z metodo Runge-Kutta nekoliko manjša od napake, ki smo jo dobili pri metodi prediktor-korektor (tabela 6.5). ■

Rezultati, dobljeni z metodo Runge-Kutta, so praviloma nekoliko boljši, kot tisti, ki jih dobimo z ABM metodo prediktor-korektor. Cena, ki jo moramo za to plačati, pa so štiri računanja odvodov v primerjavi s samo dvema

n	x_n	y_n	$y_n - y(x_n)$
0	0.0	1.0000	0
1	0.1	1.9048	$8.2 \cdot 10^{-8}$
2	0.2	1.8187	$1.5 \cdot 10^{-7}$
3	0.3	1.7408	$2.0 \cdot 10^{-7}$
4	0.4	1.6703	$2.4 \cdot 10^{-7}$
5	0.5	1.6065	$2.7 \cdot 10^{-7}$
6	0.6	1.5488	$3.0 \cdot 10^{-7}$
7	0.7	1.4966	$3.1 \cdot 10^{-7}$
8	0.8	1.4493	$3.3 \cdot 10^{-7}$
9	0.9	1.4066	$3.3 \cdot 10^{-7}$
10	1.0	1.3679	$3.3 \cdot 10^{-7}$

Tabela 6.6: Rešitev začetnega problema z metodo Runge-Kutta 4. reda

pri metodi prediktor-korektor. Po drugi strani pa je metoda Runge-Kutta enostavnejša za uporabo, saj pri njej ne potrebujemo dodatnih vrednosti odvodov v prejšnjih točkah, enostavno pa je tudi spreminjanje dolžine koraka.

6.5 Kontrola koraka

Pri računanju približka k točni rešitvi začetnega problema nas predvsem zanima velikost globalne napake. Njene velikosti med računanjem ne moremo direktno ocenjevati, lahko pa ocenjujemo velikost lokalnih napak, od katerih je odvisna tudi globalna napaka. Če bomo z izbiro dolžine koraka primerno omejili lokalne napake, bo navadno tudi globalna napaka ostala majhna.

Skoraj pri vseh metodah za reševanje začetnih problemov lahko uporabimo podoben način, ki smo ga spoznali že pri računanju integralov. Vzemimo, da smo z metodo p -tega reda izračunali vrednost $y_{n+1,h}$ z dolžino koraka h . Lokalna napaka (6.2) je torej

$$l_{n+1} = y_{n+1,h} - z(x_{n+1}) \approx Ch^{p+1},$$

kjer je $z(x_{n+1})$ rešitev začetnega problema (6.2). Izračunajmo še vrednost $y_{n+1,h/2}$ tako, da iz točke x_n naredimo dva koraka dolžine $h/2$, torej

$$y_{n+1,h/2} - z(x_{n+1}) \approx 2Ch^{p+1}2^{-p-1}.$$

Zadnji dve enačbi odštejemo:

$$y_{n+1,h} - y_{n+1,h/2} \approx Ch^{p+1}(1 - 2^{-p}),$$

od koder dobimo približek za lokalno napako vrednosti $y_{n+1,h}$

$$l_{n+1} \approx \frac{y_{n+1,h} - y_{n+1,h/2}}{1 - 2^{-p}},$$

lokalna napaka točnejše vrednosti $y_{n+1,h/2}$ pa je še za faktor 2^p manjša.

Na podlagi vnaprej predpisane zgornje meje ε za lokalno napako, lahko sedaj sodimo o primernosti dolžine koraka h . Če je ugotovljena lokalna napaka manjša od produkta εh , potem vrednost $y_{n+1,h}$ (še bolje pa $y_{n+1,h/2}$) sprejmemo, sicer začnemo računati ponovno iz prejšnje vrednosti y_n , vendar z manjšo dolžino koraka. Če pa je ugotovljena lokalna napaka dosti manjša od εh , lahko v naslednjem koraku h povečamo.

Opisani način kontrole lokalne napake in dolžine koraka je uporaben pri vseh metodah (linearnih veččlenskih in Runge-Kutta), zahteva pa nekaj dodatnega računanja, saj moramo vrednost rešitve v vsaki točki računati dvakrat. Oba tipa metod pa omogočata tudi učinkovitejše računanje približne vrednosti lokalne napake, kar si bomo sedaj ogledali.

Metode prediktor-korektor Pri metodah prediktor-korektor lahko izračunamo približno vrednost napake iz razlike med vrednostjo rešitve, izračunano z eksplisitno metodo (prediktorjem) in z implicitno metodo (korektorjem) popravljeno vrednostjo. Vzemimo, da sta obe formuli, prediktor in korektor, istega reda p . Približni vrednosti za lokalni napaki prediktorja in korektorja

$$\begin{aligned} y_{n+1}^p - z(x_{n+1}) &\approx C_p h^{p+1} \\ l_{n+1} = y_{n+1} - z(x_{n+1}) &\approx C_k h^{p+1}, \end{aligned}$$

odštejemo in dobimo

$$y_{n+1} - y_{n+1}^p \approx (C_k - C_p) h^{p+1},$$

od koder izračunamo približno vrednost lokalne napake

$$l_{n+1} \approx \frac{C_k}{C_k - C_p} (y_{n+1} - y_{n+1}^p).$$

Kadar je tako izračunana lokalna napaka manjša od εh , sprejmemo vrednost y_{n+1} kot približek za rešitev v točki x_{n+1} . Kadar pa je ugotovljena napaka večja, moramo zmanjšati dolžino koraka h in začeti znova od vrednosti y_n , kar pa je pri veččlenskih metodah povezano z dodatnimi težavami, saj moramo posebej izračunati vrednosti odvodov rešitve v zaporednih točkah, ki so med seboj oddaljene za h . Če korak razpolovimo, lahko nekatere izmed znanih vrednosti uporabimo tudi pri novi dolžini koraka, ostale pa lahko izračunamo s pomočjo interpolacije. Kadar je ugotovljena napaka znatno manjša (za več kot faktor 2^p) od dovoljene, pa lahko dolžino koraka povečamo, vendar se tudi v tem primeru znajdemo pred podobnimi težavami kot pri zmanjšanju dolžine koraka. V tem primeru je najbolje dolžino koraka podvojiti, saj tako lahko uporabimo vrednosti odvodov, ki smo jih že izračunali, seveda pa si moramo v tem primeru zapomniti več kot samo zadnjih k vrednosti odvoda.

Metode Runge-Kutta Pri metodah Runge-Kutta se je v zadnjih letih za sprotno ocenjevanje lokalne napake uveljavila tehnika *vgnezdenih parov metod*. To sta dve metodi Runge-Kutta, ki imata isto matriko koeficientov a_{ij} , $i = 1, \dots, j-1$; $j = 1, \dots, s$ in različna vektorja uteži b_i in b_i^* ; $i = 1, \dots, s$. Koeficienti in uteži morajo biti izbrani tako, da sta metodi različnih redov, naprimer metoda z utežmi b_i je reda p , metoda z utežmi b_i^* pa reda $p+1$.

Najpreprostejši primer takega vgnezdenega para sta metodi prvega in drugega reda, ki ju lahko zapišemo kot Butcherjevo tabelo

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

kjer smo v oglatem oklepaju zapisali uteži metode višjega reda. To pomeni, da je metoda nižjega reda (v tem primeru prvega reda) Eulerjeva metoda, metoda višjega reda (v tem primeru drugega) pa metoda (6.14). Ker je metoda višjega reda točnejša od metode nižjega reda, izračunamo približno vrednost lokalne napake kot razliko dveh približkov

$$\begin{aligned} y_{n+1} &= y_n + k_1 \\ y_{n+1}^* &= y_n + (k_1 + k_2)/2, \end{aligned}$$

torej

$$l_{n+1} \approx y_{n+1}^* - y_{n+1} = (-k_1 + k_2)/2,$$

Na ta način smo sicer izračunali približno vrednost napake metode nižjega reda, vendar običajno za rešitev y_{n+1} vzamemo točnejši približek y_{n+1}^* .

Pri praktičnem računanju se zelo obnese metoda, poznana pod imenom DOPRI5, ki sta jo konstruirala Dormand in Prince (1980). Njen red je 5(4), njeni koeficienti pa so v tabeli 6.7.

0							
$\frac{1}{5}$	$\frac{1}{5}$						
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	
	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0
	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$

Tabela 6.7: Koeficienti metode DOPRI5 (Dormand in Prince (1980))

Pri metodah Runge-Kutta ni težav pri spreminjanju dolžine koraka, zato jo lahko sprti spreminjamo glede na to, kako se obnaša lokalna napaka. Vzemimo, da računamo z metodo reda p , torej je lokalna napaka $l_n \approx Ch_n^{p+1}$. Dolžino naslednjega koraka želimo izbrati tako, da bo lokalna napaka $Ch_{n+1}^{p+1} \approx \varepsilon h_{n+1}$. Od tod lahko eliminiramo konstanto napake C in dobimo

$$\frac{h_{n+1}^{p+1}}{h_n^{p+1}} \approx \frac{\varepsilon h_{n+1}}{|l_n|},$$

zato moramo dolžino naslednjega koraka izbrati kot

$$h_{n+1} = h_n \sqrt[p]{\frac{\varepsilon h_n}{|l_n|}},$$

raje pa, da bo napaka zanesljivo manjša od maksimalno dovoljene, nekoliko manj. Zapišimo celoten algoritem.

Algoritem 6.5.1 (DOPRI5). Naj bo $y' = f(x, y)$ diferencialna enačba, $y(x_0) = y_0$ začetni pogoj, b končna točka intervala, na katerem nas zanima

ε	$\max y_n - y(x_n) $	število korakov
10^{-0}	$2.8 \cdot 10^{-0}$	4
10^{-1}	$7.7 \cdot 10^{-2}$	5
10^{-2}	$1.9 \cdot 10^{-3}$	6
10^{-3}	$3.1 \cdot 10^{-4}$	8
10^{-4}	$4.5 \cdot 10^{-5}$	11
10^{-5}	$5.9 \cdot 10^{-6}$	16
10^{-6}	$7.0 \cdot 10^{-7}$	25
10^{-7}	$8.0 \cdot 10^{-8}$	40
10^{-8}	$8.6 \cdot 10^{-9}$	68
10^{-9}	$9.1 \cdot 10^{-10}$	118
10^{-10}	$9.4 \cdot 10^{-11}$	205
10^{-11}	$9.6 \cdot 10^{-12}$	358
10^{-12}	$9.8 \cdot 10^{-13}$	631

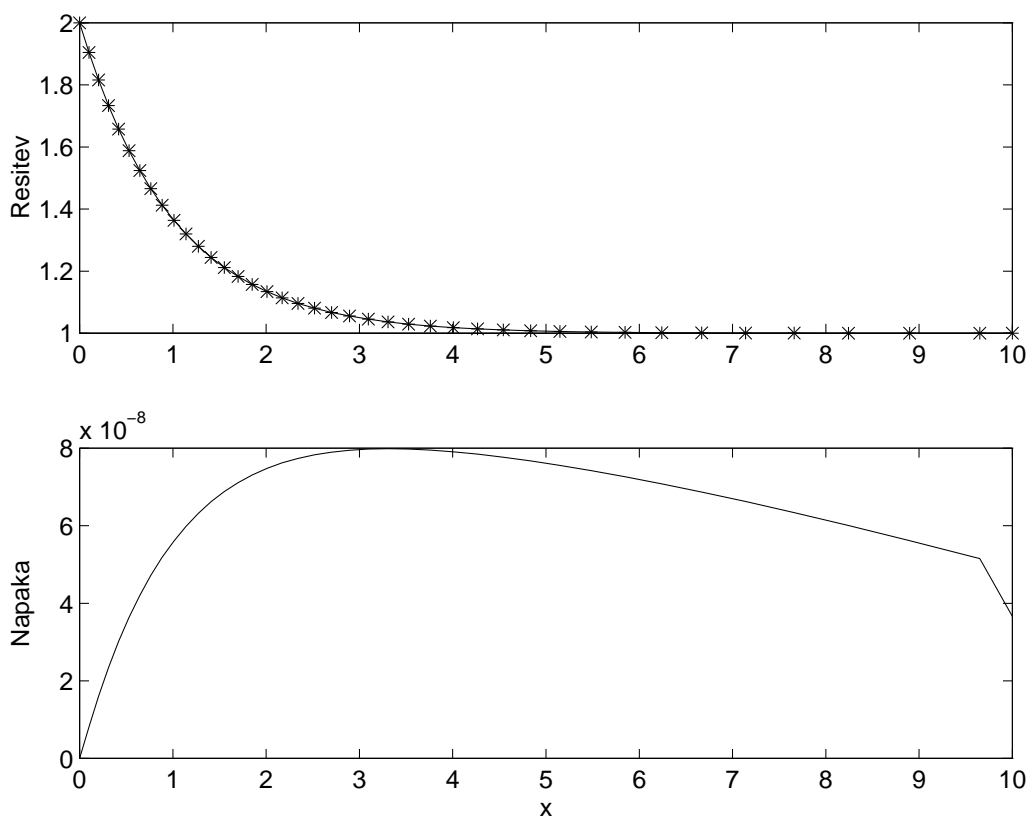
Tabela 6.8: Rešitev začetnega problema z metodo DOPRI5

rešitev in ε zgornja meja za lokalno napako na enotskem intervalu (intervalu dolžine 1). Naslednji algoritem izračuna vektor približkov y_n k vrednostim rešitve $y(x_n)$ diferencialne enačbe v točkah $x_n = x_{n-1} + h_n$; $n = 1, \dots, N$, $x_N = b$ s pomočjo metode DOPRI5.

```

x = x0
y = y0
n = 1
σ = 0.9
h = (b - x)
while x < b
    k1 = h * f(x, y)
    k2 = h * f(x + h/5, y + k1/5)
    k3 = h * f(x + h * 0.3, y + k1 * 0.075 + k2 * 0.225)
    k4 = h * f(x + h * 0.8, y + 44 * k1/45 - 56 * k2/15 + 32 * k3/9)
    k5 = h * f(x + 8 * h/9, y + 19372 * k1/6561 - 25360 * k2/2187 +
        64448 * k3/6561 - 212 * k4/729)
    k6 = h * f(x + h, y + 9017 * k1/3168 - 355 * k2/33 +
        46732 * k3/5247 + 49 * k4/176 - 5103 * k5/18656)
    k7 = h * f(x + h, y + 35 * k1/384 + 500 * k3/1113 +
        125 * k4/192 - 2187 * k5/6784 + 11 * k6/84)

```



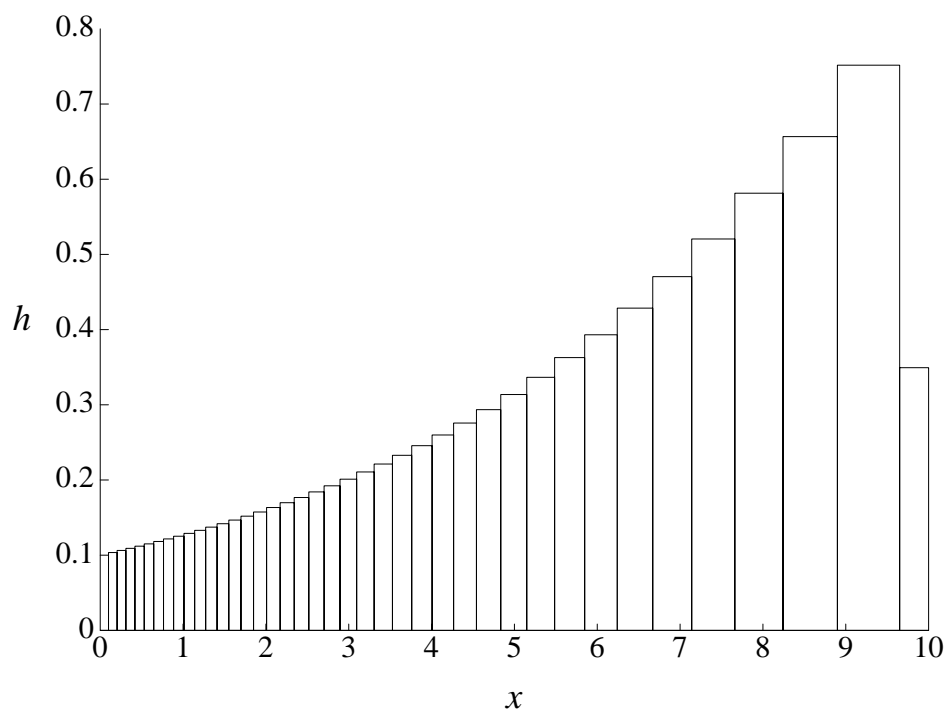
Slika 6.4: Rešitev začetnega problema (6.5) z metodo DOPRI5 (zgoraj) in globalna napaka (spodaj) pri $\varepsilon = 10^{-7}$

```

l = 71 * k1/57600 - 71 * k3/16695 + 71 * k4/1920
  -17253 * k5/339200 + 22 * k6/525 - k7/40
if |l| < ε * h
  y = y + (5179 * k1/57600 + 7571 * k3/16695 +
    393 * k4/640 - 92097 * k5/339200 + 187 * k6/2100 + k7/40)
  x = x + h
  h = σ * h *  $\sqrt[5]{\varepsilon h / |l|}$ 
  if x + h > b
    h = b - x
  end
else
  h = h/2
end

```

end



Slika 6.5: Zaporedje dolžin korakov pri reševanju začetnega problema (6.5) z metodo DOPRI5 pri $\varepsilon = 10^{-7}$

Primer 6.5.1. Poiščimo približno rešitev začetnega problema (6.5) na intervalu $[0, 10]$ tudi z metodo DOPRI5 in pri različnih zgornjih mejah za lokalno napako.

Iz rezultatov v tabeli 6.8 vidimo, da zgornja meja za lokalno napako dobro omejuje globalno napako. V zadnji koloni je navedeno število korakov (zavrženih in sprejetih), ki so bili potrebni za izračun rezultata na intervalu $[0, 10]$. Na sliki 6.4 je rešitev problema (6.5) z algoritmom 6.5.1 in njena globalna napaka, na sliki 6.5 pa so dolžine posameznih korakov, vse pri $\varepsilon = 10^{-7}$. ■

6.6 Sistemi diferencialnih enačb

Sistem m diferencialnih enačb

$$\begin{aligned} y_1' &= f_1(x, y_1, y_2, \dots, y_m) \\ y_2' &= f_2(x, y_1, y_2, \dots, y_m) \\ &\dots \quad \dots \\ y_m' &= f_m(x, y_1, y_2, \dots, y_m) \end{aligned} \quad (6.16)$$

za m neznanih funkcij y_1, \dots, y_m ima v splošnem m -parametrično družino rešitev, zato potrebujemo še m dodatnih pogojev, ki naj bodo dani kot začetni pogoji

$$y_i(x_0) = y_{i,0} \quad i = 1, \dots, m. \quad (6.17)$$

Začetni problem (6.16, 6.17) lahko krajše zapišemo v vektorski obliki

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(x_0) = \mathbf{y}_0, \quad (6.18)$$

kjer je $\mathbf{y} = [y_1, \dots, y_m]^T$, $\mathbf{f} = [f_1, \dots, f_m]^T$ in $\mathbf{y}_0 = [y_{1,0}, \dots, y_{m,0}]^T$.

Za reševanje sistema enačb, zapisanega v vektorski obliki (6.18), lahko uporabljamo iste metode kot za eno samo enačbo, le pri zapisu algoritma moramo paziti na to, da imamo opravka z vektorji in ne skalarji. Zapišimo kot zgled varianto algoritma 6.3.2, prirejeno za reševanje sistema diferencialnih enačb.

Algoritem 6.6.1 (ABM prediktor-korektor za sistem). Naj bo

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

sistem m diferencialnih enačb, $\mathbf{y}(x_0) = \mathbf{y}_0$ začetni pogoj, h korak in N naravno število. Naslednji algoritem izračuna zaporedje vektorjev približkov \mathbf{y}_n k vrednostim rešitve $\mathbf{y}(x_n)$ diferencialne enačbe v točkah $x_n = x_0 + nh$; $n = 1, \dots, N$ s pomočjo ABM metode prediktor-korektor 4. reda.

```

x = x0
y = y0
d = zeros(m, 5)
d(:, 1) = f(x, y)
for n = 2 : 4
    x = x + h

```

```

Izračunaj dodatne vrednosti  $y \approx y(x)$ 
 $d(:, n) = f(x, y)$ 
end
for  $n = 4 : N$ 
     $x = x + h$ 
     $yp = y + h * (55 * d(:, 4) - 59 * d(:, 3) + 37 * d(:, 2) - 9 * d(:, 1))/24$ 
     $dp = f(x, yp)$ 
     $y = y + h * (9 * dp + 19 * d(:, 4) - 5 * d(:, 3) + d(:, 2))/24$ 
     $d(:, 5) = f(x, y)$ 
    for  $j = 1 : 4$ 
         $d(:, j) = d(:, j + 1)$ 
    end
end
end

```

		RK4		ABM4	
n	x_n	$g_n(y)$	$g_n(z)$	$g_n(y)$	$g_n(z)$
0	0.0				
1	0.1	$3.9 \cdot 10^{-7}$	$5.8 \cdot 10^{-7}$	$-1.3 \cdot 10^{-7}$	$2.9 \cdot 10^{-7}$
2	0.2	$6.6 \cdot 10^{-7}$	$1.1 \cdot 10^{-6}$	$-3.2 \cdot 10^{-7}$	$4.8 \cdot 10^{-7}$
3	0.3	$8.1 \cdot 10^{-7}$	$1.6 \cdot 10^{-6}$	$-5.5 \cdot 10^{-7}$	$6.0 \cdot 10^{-7}$
4	0.4	$8.5 \cdot 10^{-7}$	$2.0 \cdot 10^{-6}$	$-8.2 \cdot 10^{-7}$	$6.3 \cdot 10^{-7}$
5	0.5	$7.8 \cdot 10^{-7}$	$2.3 \cdot 10^{-6}$	$-1.1 \cdot 10^{-6}$	$5.8 \cdot 10^{-7}$
6	0.6	$6.2 \cdot 10^{-7}$	$2.6 \cdot 10^{-6}$	$-1.4 \cdot 10^{-6}$	$4.7 \cdot 10^{-7}$
7	0.7	$3.7 \cdot 10^{-7}$	$2.7 \cdot 10^{-6}$	$-1.7 \cdot 10^{-6}$	$2.9 \cdot 10^{-7}$
8	0.8	$5.5 \cdot 10^{-8}$	$2.8 \cdot 10^{-6}$	$-1.9 \cdot 10^{-6}$	$6.3 \cdot 10^{-8}$
9	0.9	$-3.2 \cdot 10^{-7}$	$2.7 \cdot 10^{-6}$	$-2.2 \cdot 10^{-6}$	$-2.1 \cdot 10^{-7}$
10	1.0	$-7.3 \cdot 10^{-7}$	$2.6 \cdot 10^{-6}$	$-2.5 \cdot 10^{-6}$	$-8.2 \cdot 10^{-7}$

Tabela 6.9: Globalne napake komponent začetnega problema (6.19) z metodo Runge-Kutta 4. reda in z Adams-Bashforth-Moultonovo metodo prediktor-korektor 4. reda ($h = 0.1$)

Primer 6.6.1. Poiščimo približno rešitev začetnega problema

$$\begin{aligned}y' &= y - 2z - 2e^{-x} + 2 \\z' &= 2y - z - 2e^{-x} + 1; \\y(0) &= 1 \\z(0) &= 1\end{aligned}\tag{6.19}$$

na intervalu $[0, 1]$ z metodo Runge-Kutta in z Adams-Bashforth-Moultonovo metodo prediktor-korektor, obe z isto dolžino koraka $h = 0.1$. Začetne vrednosti za Adams-Bashforth-Moultonovo metodo so dobljene iz točne rešitve $y = e^{-x}$, $z = 1$ na intervalu $[-0.3, 0]$. Zapišimo še algoritem za izračun vrednosti funkcije f v tem primeru:

```
function  $f = f(x, y)$ 
     $f = [y(1) - 2 * y(2) - 2 * \mathbf{exp}(-x) + 2;$ 
         $2 * y(1) - y(2) - 2 * \mathbf{exp}(-x) + 1];$  ■
```

6.7 Enačbe višjih redov

Diferencialne enačbe višjih redov navadno rešujemo tako, da jih zapišemo kot sistem enačb prvega reda. To najlažje dosežemo tako, da za nove spremenljivke izberemo odvode neznane funkcije. Vzemimo diferencialno enačbo m -tega reda

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

Z uvedbo novih spremenljivk

$$\begin{aligned}y_1 &= y \\y_2 &= y' \\y_3 &= y'' \\&\vdots \\y_m &= y^{(m-1)}\end{aligned}$$

dobimo sistem enačb prvega reda

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

ki ga rešujemo s pomočjo že znanih metod.

6.8 Stabilnost

Začnimo z enostavnim primerom:

Primer 6.8.1. Rešimo začetni problem

$$y' = -8y; \quad y(0) = 1 \tag{6.20}$$

na intervalu $x \in (0, 1)$ za različne dolžine koraka h .

Iz rezultatov na sliki 6.6 vidimo, da se numerične rešitve obnašajo zelo različno. Pri dolžinah koraka $h < 1/8$ numerične rešitve monotonno padajo proti vrednosti 0, prav tako kakor točna rešitev $y = e^{-8x}$. Pri dolžinah koraka $1/8 < h < 1/4$ rešitve oscilirajo okoli točne rešitve in se ji približujejo, ko $x \rightarrow \infty$. Za dolžine korakov $h > 1/4$ pa rešitve oscilirajo okoli točne rešitve, njihove absolutne vrednosti pa neomejeno naraščajo (pravimo, da so nestabilne). ■

Poizkusimo pojasniti tako obnašanje rešitev. Ko rešujemo *testno diferencialno enačbo*

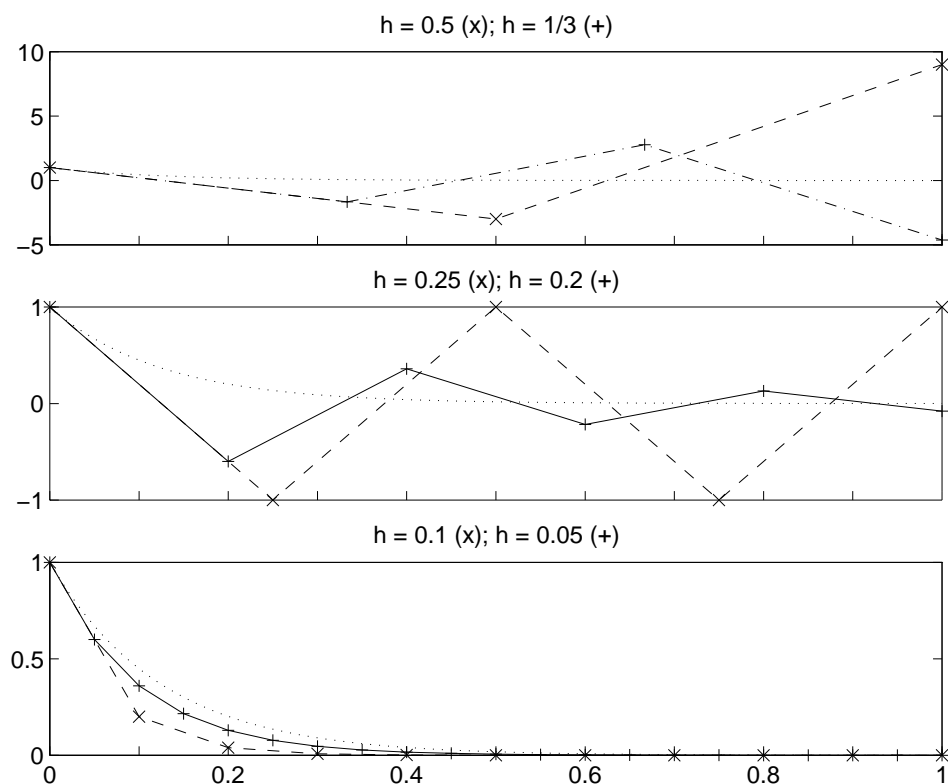
$$y' = \lambda y; \quad \lambda < 0 \tag{6.21}$$

z Eulerjevo metodo, dobimo med dvema zaporednima približkoma relacijo

$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)y_n, \tag{6.22}$$

medtem ko točna rešitev $y = e^{\lambda x}$ zadošča podobni relaciji

$$y(x_{n+1}) = e^{h\lambda}y(x_n).$$



Slika 6.6: Rešitve testne enačbe $y' = -8y$ z Eulerjevo metodo pri različnih dolžinah koraka

Ker je za $h < 0$ faktor $|e^{h\lambda}| < 1$, je točna rešitev padajoča funkcija. Seveda želimo, da se podobno obnaša tudi numerična rešitev, zato mora tudi faktor $1 + h\lambda$ v enačbi (6.22) zadoščati podobni neenačbi kot točna rešitev, torej $|1 + h\lambda| < 1$. Od tod dobimo da mora biti $-2 < h\lambda < 0$. Interval $[-2, 0]$ zato imenujemo *interval absolutne stabilnosti* za Eulerjevo metodo.

Na podoben način je definiran interval absolutne stabilnosti tudi za druge metode: interval absolutne stabilnosti je največji interval $[-a, 0]$, da je numerična rešitev testne diferencialne enačbe (6.21) omejena za vsak $-a \leq h\lambda \leq 0$. Za metode, ki smo jih srečali v tem poglavju, so intervali absolutne stabilnosti zbrani v tabeli 6.10.

Izkaže se, da je obnašanje metode pri reševanju testne enačbe (6.21) značilno za to, kako globalna napaka iz prejšnjega koraka vpliva na globalno napako v naslednjem koraku. Kadar je $h\lambda$ zunaj intervala absolutne sta-

Metoda	interval abs. stabilnosti
AB1 (Eulerjeva)	$[-2, 0]$
AB2	$[-1.1, 0]$
AB3	$[-0.65, 0]$
AB4	$[-0.4, 0]$
AM1 (Implicitna Eulerjeva)	$[-\infty, 0]$
AM2 (trapezna)	$[-\infty, 0]$
AM3	$[-6, 0]$
AM4	$[-3, 0]$
ABM4	$[-1.4, 0]$
RK2 (obe metodi)	$[-2, 0]$
RK4 (klasična Runge-Kutta)	$[-2.8, 0]$
DOPRI5	$[-3.3, 0]$

Tabela 6.10: Intervali absolutne stabilnosti

bilnosti, se globalna napaka iz koraka v korak povečuje. Zato moramo pri reševanju diferencialnih enačb paziti, da je korak metode vedno v mejah, ki jih določa interval absolutne stabilnosti, sicer lahko napaka v rešitvi nekontrolirano narašča.

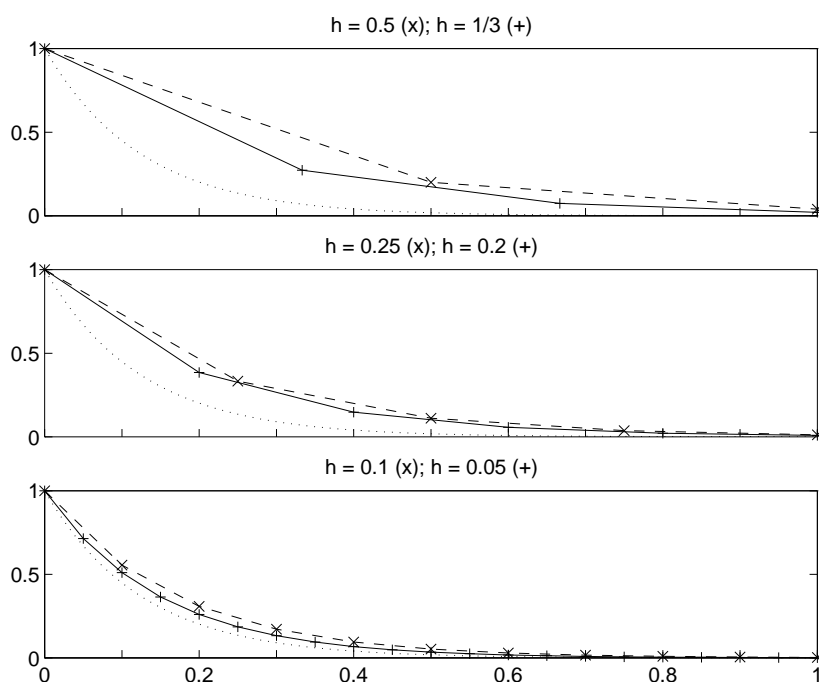
Primer 6.8.2. Rešimo začetni problem (6.20) še z implicitno Eulerjevo metodo (6.9).

Iz rezultatov na sliki 6.7 vidimo, da pri reševanju z implicitno Eulerjevo metodo zaradi njene stabilnosti (interval absolutne stabilnosti implicitne Eulerjeve metode je $(-\infty, 0]$, glej tabelo 6.10) izračunana rešitev dobro sledi pravi rešitvi tudi pri večjih dolžinah koraka. ■

6.9 Robni problemi

Opisali bomo *strelsko metodo*, ki je dovolj splošna, da z njo lahko rešujemo tako linearne kot tudi nelinearne robne probleme. Za primer vzemimo robni problem

$$\begin{aligned} y' &= f(x, y, z) \\ z' &= g(x, y, z); \end{aligned} \tag{6.23}$$



Slika 6.7: Rešitve testne enačbe $y' = -8y$ z implicitno Eulerjevo metodo pri različnih dolžinah koraka

$$y(a) = A, \quad y(b) = B.$$

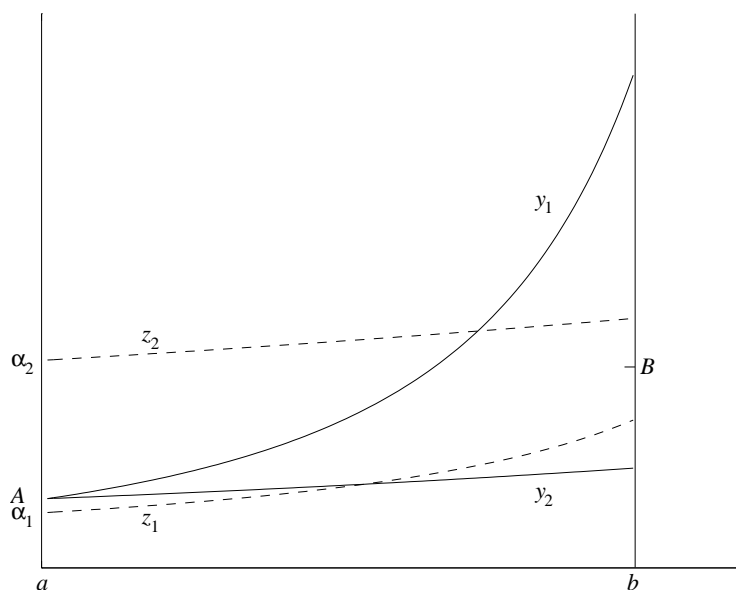
Ker želimo za reševanje problema (6.23) uporabiti kakšno izmed obravnavanih metod za reševanje začetnih problemov, moramo manjkajočo vrednost $z(a)$ nadomestiti z neznanim parametrom, npr. α in njegovo vrednost določiti tako, da bo funkcija y pri $x = b$ zavzela predpisano vrednost B .

Izberimo za α poljubno vrednost in rešimo začetni problem

$$\begin{aligned} y' &= f(x, y, z) \\ z' &= g(x, y, z); \\ y(a) &= A, \quad z(a) = \alpha \end{aligned} \tag{6.24}$$

na intervalu $[a, b]$. Vrednost spremenljivke y pri $x = b$ je tako odvisna od izbire parametra α , torej $y(b) = F(\alpha)$. Vrednost parametra α , za katerega je $y(b) = B$ je torej rešitev nelinearne enačbe

$$F(\alpha) - B = 0. \tag{6.25}$$



Slika 6.8: Dve rešitvi začetnega problema (6.24)

Izberimo za začetek dve različni vrednosti α_0 in α_1 in rešimo začetni problem (6.24) za obe vrednosti (slika 6.8). Rešitev nelinearne enačbe (6.25) sedaj lahko poiščemo s sekantno metodo. Zapišimo algoritem:

Algoritem 6.9.1 (Streška metoda). Naj bo (6.23) dan robni problem, ε največje dopustno odstopanje izračunane rešitve od predpisane vrednosti B , α_0 in α_1 dva začetna približka za $z(a)$ in m naravno število. Naslednji algoritem izračuna pravilno začetno vrednost $z(a) = \alpha$ in rešitev robnega problema, ali pa se po m iteracijah konča brez rezultata ($\alpha = NaN$).

Reši začetni problem (6.24) za $\alpha = \alpha_0$ z

eno izmed metod za reševanje začetnih problemov

$ys = y$

Reši začetni problem (6.24) za $\alpha = \alpha_1$

$yn = y$

$j = 1$

while ($\text{abs}(yn - B) > \varepsilon$) * ($j < m$)

$j = j + 1$

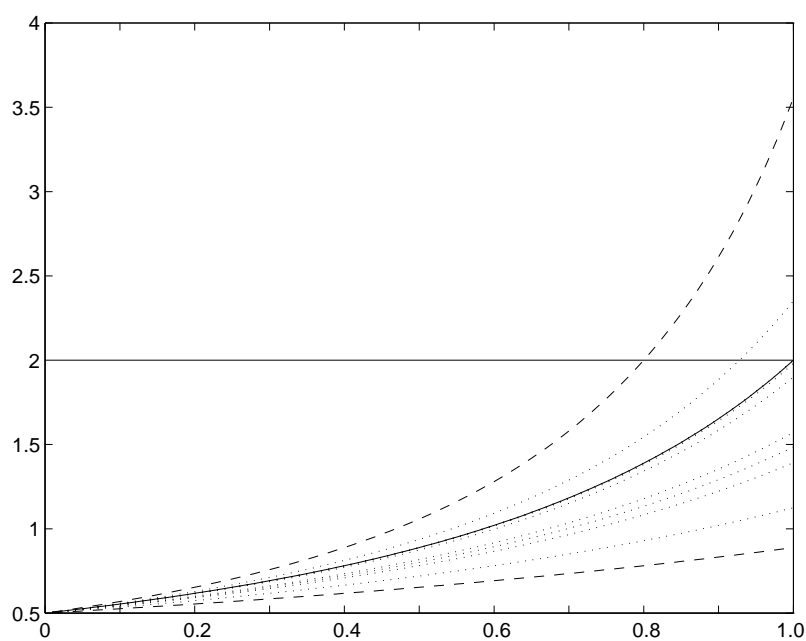
$\alpha = \alpha_0 + (\alpha_1 - \alpha_0) * (B - ys) / (yn - ys)$

Reši začetni problem (6.24) za α

```

ys = yn
yn = y
α0 = α1
α1 = α
end
if abs(yn - B) > ε
    α = NaN
end
end

```



Slika 6.9: Reševanje robnega problema (6.26) s strelsko metodo

Primer 6.9.1. Rešimo robni robni problem

$$\begin{aligned}
 y' &= \frac{y^2}{z}; & y(0) &= \frac{1}{2} \\
 z' &= \frac{y}{2}; & y(1) &= 2
 \end{aligned}
 \tag{6.26}$$

s strelsko metodo.

n	α	$y(1)$
0	1.000	.8889
1	.4000	3.556
2	.7500	1.125
3	.6240	1.392
4	.3369	7.520
5	.5955	1.485
6	.5734	1.572
7	.4639	2.352
8	.5133	1.900
9	.5024	1.981
10	.4998	2.001
11	.5000	2.000

Tabela 6.11: Rezultati reševanja robnega problema (6.26) s strelsko metodo

Računamo z algoritmom 6.9.1, začetne probleme računamo z metodo Runge-Kutta, za manjkajoči začetni vrednosti odvoda izberemo $\alpha_0 = 1$ in $\alpha_1 = 0.4$. Rezultati računanja so povzeti v tabeli 6.11. Končna vrednost $y(1) = 2$ je bila pri natančnosti 10^{-12} dosežena po 12 korakih iteracije. ■

6.10 Povzetek

Izmed problemov, povezanih z reševanjem sistemov navadnih diferencialnih enačb, smo se ukvarjali predvsem z začetnimi problemi za diferencialno enačbo prvega reda. Najenostavnejša metoda za reševanje teh problemov je Eulerjeva metoda, ki pa žal za resno uporabo ni dovolj natančna.

Izmed linearnih veččlenskih metod smo spoznali dve družini Adamsovih metod: eksplicitne Adams-Bashforthove in implicitne Adams-Moultonove metode. Pri eksplicitnih formulah je na vsakem koraku potrebno le enkrat računati vrednost desne strani diferencialne enačbe in njihova uporaba je razmeroma enostavna. Implicitne formule so praviloma natančnejše, vendar je njihova uporaba zapletenejša, saj moramo na vsakem koraku rešiti nelinearno enačbo, zato jih uporabljamo navadno skupaj z ustrezno eksplicitno metodo kot metodo prediktor-korektor. V tem primeru moramo na vsakem koraku

dvakrat računati vrednost desne strani diferencialne enačbe. Glavni pomanjkljivosti teh metod sta: zapleteno spreminjanje dolžine koraka in potreba po posebni proceduri, s katero na začetku izračunamo dodatne vrednosti (za kar navadno uporabimo ustrezno metodo Runge-Kutta).

Metode tipa Runge-Kutta so enočlenske in zato enostavnejše za uporabo. Pri njih lahko enostavno menjamo dolžino koraka. So tudi dovolj natančne, njihova pomanjkljivost pa je, da moramo na vsakem koraku večkrat računati vrednost desne strani diferencialne enačbe, zato je reševanje počasnejše.

Reševanje začetnih problemov za sisteme diferencialnih enačb poteka podobno, le formule, ki smo jih spoznali za reševanje začetnih problemov z eno samo diferencialno enačbo, moramo uporabiti za vsako komponento posebej.

Diferencialne enačbe višjega reda navadno rešujemo tako, da jih zapišemo kot sistem diferencialnih enačb prvega reda.

Reševanje robnih problemov prevedemo na reševanje zaporedja začetnih problemov, kjer dodatne začetne vrednosti izbiramo tako, da zadovoljimo preostale robne pogoje.

Spoznali smo tudi, kako lahko sproti, med samim reševanjem ocenjujemo lokalno napako metode in obenem prilagajamo dolžino koraka metode tako, da so lokalne napake manjše od dovoljenih.

6.11 Problemi

1. V identiteti (6.6) nadomesti funkcijo $f(x, y(x))$ z interpolacijskim polinomom skozi točki x_n in x_{n-1} in integriraj. Tako dobiš dvostopenjsko Adams-Bashforthovo metodo

$$y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1}),$$

katere lokalna napaka je enaka $\frac{5}{12}h^3y'''(\xi_n)$.

2. V identiteti (6.6) nadomesti funkcijo $f(x, y(x))$ z interpolacijskim polinomom skozi točki x_{n+1} in x_n ter integriraj. Tako dobiš dvostopenjsko Adams-Moultonovo (trapezno) metodo

$$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n)$$

z lokalno napako $\frac{-1}{12}h^3y'''(\xi_n)$.

3. Rešujemo začetni problem

$$y' = 2\frac{y}{x} + \frac{3}{x^2}; \quad y(1) = 0.$$

- (a) Poišči točno rešitev začetnega problema.
 - (b) Z Eulerjevo metodo izračunaj približno rešitev na intervalu $[1, 2]$ z dolžinami koraka $h = 0.5$, $h = 0.1$ in $h = 0.01$. Kolikšni sta lokalna in globalna napaka pri $x = 2$?
 - (c) Izračunaj približno rešitev z metodo Runge-Kutta. Kolikšni sta lokalna in globalna napaka pri $x = 2$ v tem primeru?
 - (d) Izračunaj približno rešitev še z metodo prediktor-korektor ABM4 (dodatne začetne vrednosti naj bodo izračunane z metodo Runge-Kutta). Kolikšni sta lokalna in globalna napaka pri $x = 2$ v tem primeru?
4. Rešujemo začetni problem $y' = y + e^x$; $y(0) = 0$.
- (a) Poišči točno rešitev problema.
 - (b) Izračunaj približke za $y(x)$ v točkah $x = 0.1$ in $x = 0.2$ z metodo Runge Kutta drugega reda ($h = 0.1$).
 - (c) Izračunaj približno vrednost rešitve tega začetnega problema na intervalu $[0, 1]$ z Adams-Bashforth-Moultonovo metodo prediktor-korektor 3. reda. Kolikšni sta globalna in lokalna napaka metode pri $x = 0.7$?
5. Algoritem 6.4.1 priredi za reševanje sistema diferencialnih enačb in z njim reši začetni problem (6.19).
6. Dan je začetni problem

$$y'' + 4y' + 13y = 40 \cos x; \quad y(0) = 3; \quad y'(0) = 4$$

- (a) Diferencialno enačbo drugega reda zapiši kot sistem dveh diferencialnih enačb prvega reda.
- (b) Izračunaj točno rešitev.
- (c) Izračunaj rešitev z Eulerjevo metodo na intervalu $[0, 1]$ z dolžinami koraka $h = 0.5$, 0.05 in 0.005 . Kolikšna je globalna napaka v končni točki?

- (d) Ponovi račun z metodo Runge-Kutta. Kolikšna je sedaj napaka?
- (e) Ponovi račun še z metodo prediktor-korektor ABM4 (dodatne začetne vrednosti so izračunane z metodo Runge-Kutta). Kolikšna je globalna napaka v tem primeru?

7. Začetni problem

$$y' + \frac{y}{x} = -x; \quad y(1) = 1$$

na intervalu $(1, 2)$ rešujemo z Milne-Simpsonovo metodo prediktor-korektor

$$y_{n+1}^p = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2})$$

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1}^p + 4f_n + f_{n-1}).$$

- (a) Izračunaj red metode (Navodilo: razvoj po Taylorjevi formuli).
- (b) Zapiši algoritem.
- (c) Izračunaj točno rešitev začetnega problema.
- (d) Izračunaj numerično rešitev s koraki $h = 0.1$, $h = 0.01$ in 0.001 . Za dodatne začetne vrednosti vzemi kar vrednosti prave rešitve. Kako se obnaša globalna napaka?