
Domača naloga 6: dušeno nihanje

Vaje iz predmeta Numerične metode v fiziki

Igor Grešovnik

Kazalo:

<i>1</i>	<i>Naloga 6a – Nihanje</i>	<i>2</i>
<i>1.1</i>	<i>Enačbe nihanja</i>	<i>2</i>
<i>1.2</i>	<i>Numerično reševanje problema</i>	<i>3</i>
1.2.1	Reševanje sistema diferencialnih enačb 1. reda z začetnimi pogoji z metodo Runge Kutta	4
<i>1.3</i>	<i>Programska rešitev</i>	<i>6</i>
<i>1.4</i>	<i>Rezultati</i>	<i>6</i>
1.4.1	Preverjanje natančnosti in zanesljivosti metode	7
1.4.2	Konvergenca metode	10
<i>2</i>	<i>Dodatna domača naloga – vsiljeno nihanje</i>	<i>13</i>

1 NALOGA 6A – NIHANJE

1.1 Enačbe nihanja



Nedušeno nihanje uteži na vzmeti opišemo z enačbo

$$F = m \frac{d^2 x}{dt^2} = -k x. \quad (1)$$

Tu je m masa uteži, k pa razteznostni koeficient vzmeti. Ko utež pomaknemo iz ravnovesne lege vzmeti, ta deluje nanjo s silo, ki je sorazmerna odmiku iz ravnovesne lege. Ko jo spustimo, se zato prične pospešeno gibati proti ravnovesni legi. Če ni trenja ali upora, utež sinusno niha:

$$x(t) = A \sin(2\pi\nu t + \phi). \quad (2)$$

V zgornji enačbi je A amplituda (odmik od ravnovesne lege, ko utež spustimo), ν frekvenca in ϕ poljubni fazni zamik, ki je odvisen od tega, kdaj začnemo meriti čas. Frekvenca nihala je

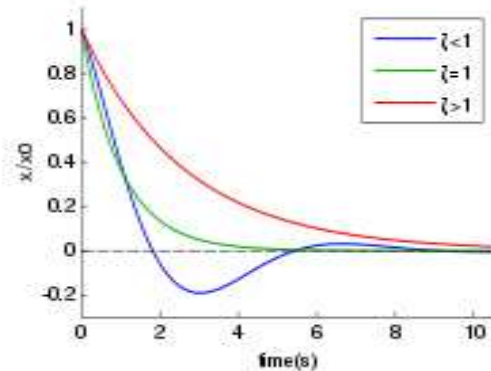
$$\nu = \frac{1}{2\pi} \sqrt{\frac{k}{m}}. \quad (3)$$

Namesto frekvence večkrat navajamo kotno hitrost $\omega = 2\pi\nu$.

V opisanem harmoničnem nihalu se potencialna energija vzmeti brez izgub pretvarja v kinetično energijo uteži in obratno. V neidealnih primerih so prisotne izgube – nihalo oddaja delo okolici zaradi trenja ali upora. Nihanje je dušeno in ga v primeru viskoznega dušenja opišemo z naslednjo enačbo:

$$\frac{d^2 x}{dt^2} + 2\kappa \frac{dx}{dt} + \omega_0^2 x = 0. \quad (4)$$

Intenzivnost takšnega nihala pojema s časom. Hitrost pojemanja je tem večja, čim večji je faktor κ . Vpliv dušenja je odvisen predvsem od razmerja $\zeta = \kappa / \omega_0$. Pri $\zeta < 1$ sistem niha z malo drugačno frekvenco od osnovne in z amplitudo, ki se postopno približuje 0. Nad kritičnim dušenjem ($\zeta > 1$) se sistem eksponentno približuje ravnovesni legi. Pri kritičnem dušenju ($\zeta = 1$) je približevanje ravnovesni legi najhitrejšo izmed vseh primerov, ko sistem ne zaniha (slika).



1.2 Numerično reševanje problema

Osnovna ideja je, da diferencialno enačbo 2. reda prevedemo na sistem dveh enačb 1. reda. Sistem rešimo po metodi Runge Kutta.

Odvod x po t uvedemo kot novo spremenljivko:

$$\frac{dx}{dt} = v(t) \quad (5)$$

Ko to vstavimo v prvotno enačbo (4), dobimo

$$\frac{dv}{dt} = -2\kappa v - \omega_0^2 x \quad (6)$$

Enačbi (5) in (6) predstavljata sistem dveh sklopljenih diferencialnih enačb. 1. reda. Začetne pogoje za prvotno enačbo, recimo

$$\begin{aligned} x(0) &= A_0 \\ \frac{dx}{dt}(0) &= 0 \end{aligned} \quad (7)$$

Lahko neposredno prepišemo v začetne pogoje sistema diferencialnih enačb. V celoti problem prevedemo na sistem dveh diferencialnih enačb 1. Reda z začetnimi pogoji:

$$\begin{aligned}\frac{dx(t)}{dt} &= v(t) \\ \frac{dv}{dt} &= -2\kappa v - \omega_0^2 x, \\ x(0) &= A_0 \\ u(0) &= 0\end{aligned}\tag{8}$$

Uvedba nove spremenljivke v ima v tem primeru zelo jasen fizikalen pomen, ni pa vedno tako.

1.2.1 Reševanje sistema diferencialnih enačb 1. reda z začetnimi pogoji z metodo Runge Kutta

Podobno kot pri reševanju ene enačbe imamo pri sistemu enačb odvode podane kot funkcijo odvisnih in neodvisnih spremenljivk, ki nastopajo v problemu. Zato lahko enačbe zelo neposredno izpeljemo po analogiji s primerom, ko imamo samo eno enačbo.

Najbolj enostavna metoda za reševanja takšnega sistema je Eulerjeva, vendar želimo uporabiti kakšno bolj natančno metodo. Pri metodi Runge-Kutta računamo približek za rešitev enačbe

$$y' = f(x, y)\tag{9}$$

po naslednji formuli:

$$\begin{aligned}k_1 &= h f(x, y) \\ k_2 &= h f\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right) \\ k_3 &= h f\left(x + \frac{h}{2}, y + \frac{k_2}{2}\right) \\ k_4 &= h f(x + h, y + k_3) \\ y(x+h) &= y(x) + \frac{(k_1 + 2k_2 + 2k_3 + k_4)}{6} + O(h^5)\end{aligned}\tag{10}$$

Pri razširitvi na sistem diferencialnih enačb obravnavamo f kot vektorsko funkcijo. Sistem diferencialnih enačb je lahko podan kot

$$\begin{aligned}
y_1' &= f_1(x, y_1, y_2, \dots, y_n) \\
y_2' &= f_2(x, y_1, y_2, \dots, y_n) \\
&\dots \\
y_n' &= f_n(x, y_1, y_2, \dots, y_n)
\end{aligned}
\tag{11}$$

z začetnimi pogoji

$$y_i(x_0) = y_{i0} \tag{12}$$

Sistem lahko zapišemo v vektorski obliki:

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(x_0) = \mathbf{y}_0 \tag{13}$$

S posplošitvijo metode Runge-Kutta za eno diferencialno enačbo dobimo naslednji postopek reševanja sistema diferencialnih enačb prvega reda:

$$\begin{aligned}
&y_i(x_0) = y_{i0}, \quad i = 1, \dots, n \text{ - začetni pogoj} \\
&j = 0, 2, \dots, m-1: \\
&k_{1,i} = h f_i(x, y_1, y_2, \dots, y_n), \quad i = 1, \dots, n \\
&k_{2,i} = h f_i\left(x + \frac{h}{2}, y_1 + \frac{k_{1,1}}{2}, y_2 + \frac{k_{1,2}}{2}, \dots, y_n + \frac{k_{1,n}}{2}\right), \quad i = 1, \dots, n \\
&k_{3,i} = h f_i\left(x + \frac{h}{2}, y_1 + \frac{k_{2,1}}{2}, y_2 + \frac{k_{2,2}}{2}, \dots, y_n + \frac{k_{2,n}}{2}\right), \quad i = 1, \dots, n \\
&k_{4,i} = h f_i(x + h, y_1 + k_{3,1}, y_2 + k_{3,2}, \dots, y_n + k_{3,n}), \quad i = 1, \dots, n \\
&y_i(x+h) = y_i(x) + \frac{(k_{1,i} + 2k_{2,i} + 2k_{3,i} + k_{4,i})}{6} + O(h^5), \quad i = 1, \dots, n
\end{aligned}
\tag{14}$$

Po zgornjih formulah računamo vrednosti neznanih funkcij določenih s sistemom diferencialnih enačb postopno v točkah x_1, x_2, \dots, x_m . Indekse pri točkah x smo izpustili zaradi preglednosti, namesto tega pišemo $x_j = x, x_{j+1} = x+h, y_i(x_j) = y_i = y_i(x), y_i(x_{j+1}) = y_i(x+h)$.

Zapišimo metodo Runge-Kutta še za primer z dvema enačbama:

$$\begin{aligned}
y' &= f(x, y, z) & y(x_0) &= y_0 \\
z' &= g(x, y, z) & z(x_0) &= z_0
\end{aligned}
\tag{15}$$

Iz (14) dobimo za primer $n=2$ naslednje enačbe (namesto vektorskih koeficientov $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4$ uvedemo pare koeficientov $k_1, l_1, k_2, l_2, k_3, l_3, k_4, l_4$):

$$\begin{aligned}
y(x_0) &= y_0, & y(x_0) &= z_0 \\
k_1 &= h f(x, y, z) & l_1 &= h g(x, y, z) \\
k_2 &= h f\left(x+h/2, y+k_1/2, z+l_1/2\right) & l_2 &= h g\left(x+h/2, y+k_1/2, z+l_1/2\right) \\
k_3 &= h f\left(x+h/2, y+k_2/2, z+l_2/2\right) & l_3 &= h g\left(x+h/2, y+k_2/2, z+l_2/2\right) \\
k_4 &= h f(x+h, y+k_3, z+l_3) & l_4 &= h g(x+h, y+k_3, z+l_3) \\
y(x+h) &= y(x) + \frac{(k_1+2k_2+2k_3+k_4)}{6} + O(h^5) \\
z(x+h) &= z(x) + \frac{(l_1+2l_2+2l_3+l_4)}{6} + O(h^5)
\end{aligned} \tag{16}$$

Zgornje enačbe lahko neposredno uporabimo pri reševanju sistema dveh diferencialnih enačb. Druga možnost je, da sprogramiramo bolj splošno metodo za reševanje poljubnega števila diferencialnih enačb 1. reda po formulah (16) in to metodo uporabimo za primer, ko sta enačbi samo dve. Prvi način je nekoliko lažji za implementacijo, zato pa z drugim načinom dobimo bolj splošno metodo, ki jo lahko uporabimo za večji nabor sorodnih problemov. Pri reševanju naloge bomo ubrali drugo pot.

1.3 Programska rešitev

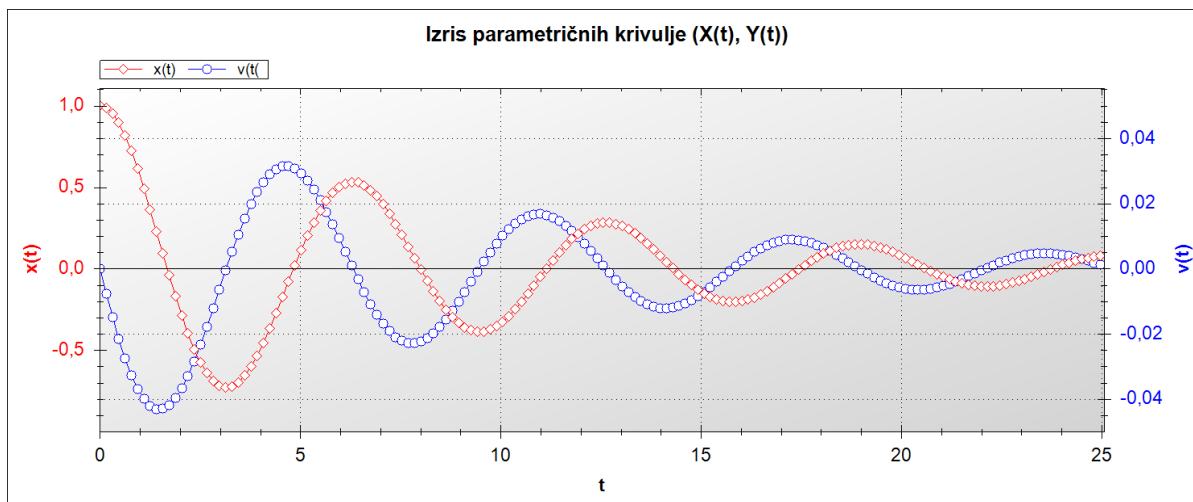
Rešitev naloge je v projektu *csharp/gresovnik/03_DifferentialEquations*.

Glavni program je v datoteki *ProgramDifferential.cs*, metoda Runge-Kutta za sisteme diferencialnih enačb 1. reda z začeni pogoji pa je v *RungeKutta.cs*. Iz datoteke *00library/0definitions.cs* se uporablja definicija delegata *ScalarFunction*.

V razredu *ProgramDifferential* (datoteka *ProgramDifferential.cs*) projekta *DifferentialEquations* se nastavijo vsi parametri problema, v metodi *Main* se poženejo izračuni in na koncu izrišejo rezultati (ti se izpišejo tudi na konzolo). Za riasnje grafov je uporabljena zunanja knjižnica *ZedGraph* iz *csharp/external*.

1.4 Rezultati

Na sliki spodaj so rezultati za štiri nihaje s koeficientom dušenja $\kappa = 0,1$. Število podintervalov na vsak nihaj je 40.



1.4.1 Preverjanje natančnosti in zanesljivosti metode

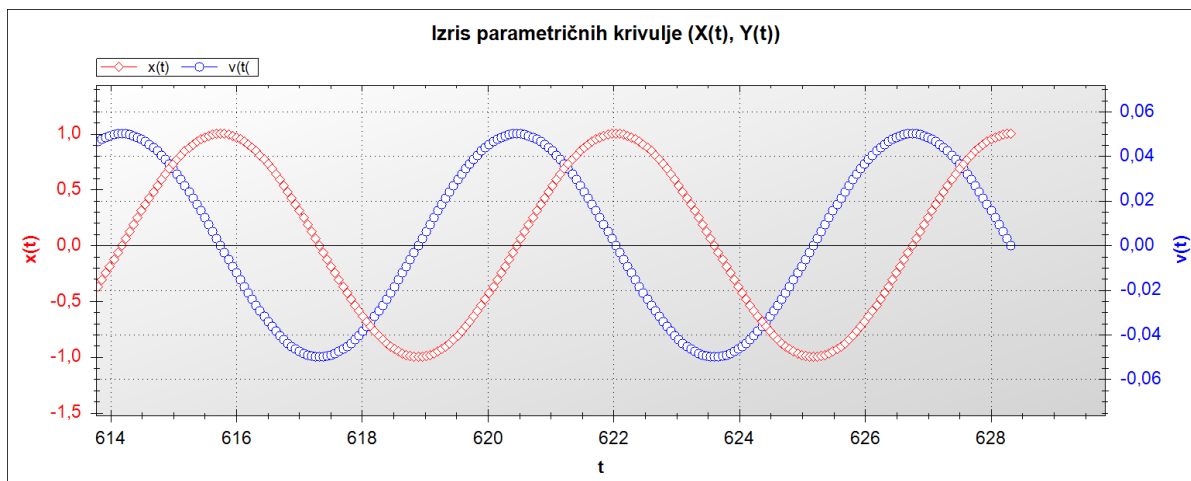
Pri številnih numeričnih postopkih obstajajo koristne ocene za velikost napak. Te lahko uporabimo za to, da zagotovimo natančnost željenih rezultatov. Pri bolj kompleksnih postopkih je poleg teoretičnih ocen napak priporočljivo uporabiti še dodatne preizkuse uporabljenih metod. S tem preverimo tudi pravilnost implementacije postopkov in velikokrat odkrijemo odstopanja, ki bi jih težko teoretično napovedali.

Natančnost metode je vedno dobro preveriti s problemom, kjer je rešitev zelo natančno znanana. Če ni dušenja ($\alpha = 0$), je rešitev periodično harmonično nihanje, vendar ima potem problem v marsičem drugačne lastnosti.

V tem primeru ($\alpha = 0$) lahko na primer preverimo, če amplituda izračunanega nihanja po veliko nihajih znatno odstopa od začetne in ali se morda ustvarja fazni zamik. Pri diferencialnih enačbah se napake metode akumulirajo, podobno kot pri integraciji, zato je preverjanje natančnosti po veliko nihajih lahko dober preizkus metode. Vendar pa vsebuje problem nedušenega nihanja precej simetrije¹, kar bi lahko bilo razlog za to, da se napake med sabo skozi nihaje tudi uničujejo. To je povezano z izjavo iz prejšnjega odstavka, da se narava problema spremeni, če odvezamo dušenje. Vsekakor se moramo zavedati vsaj tega, da majhne napake pri nedušenem nihanju ne pomenijo nujno, da bi dobili enako majhne napake pri prvotnem problemu.

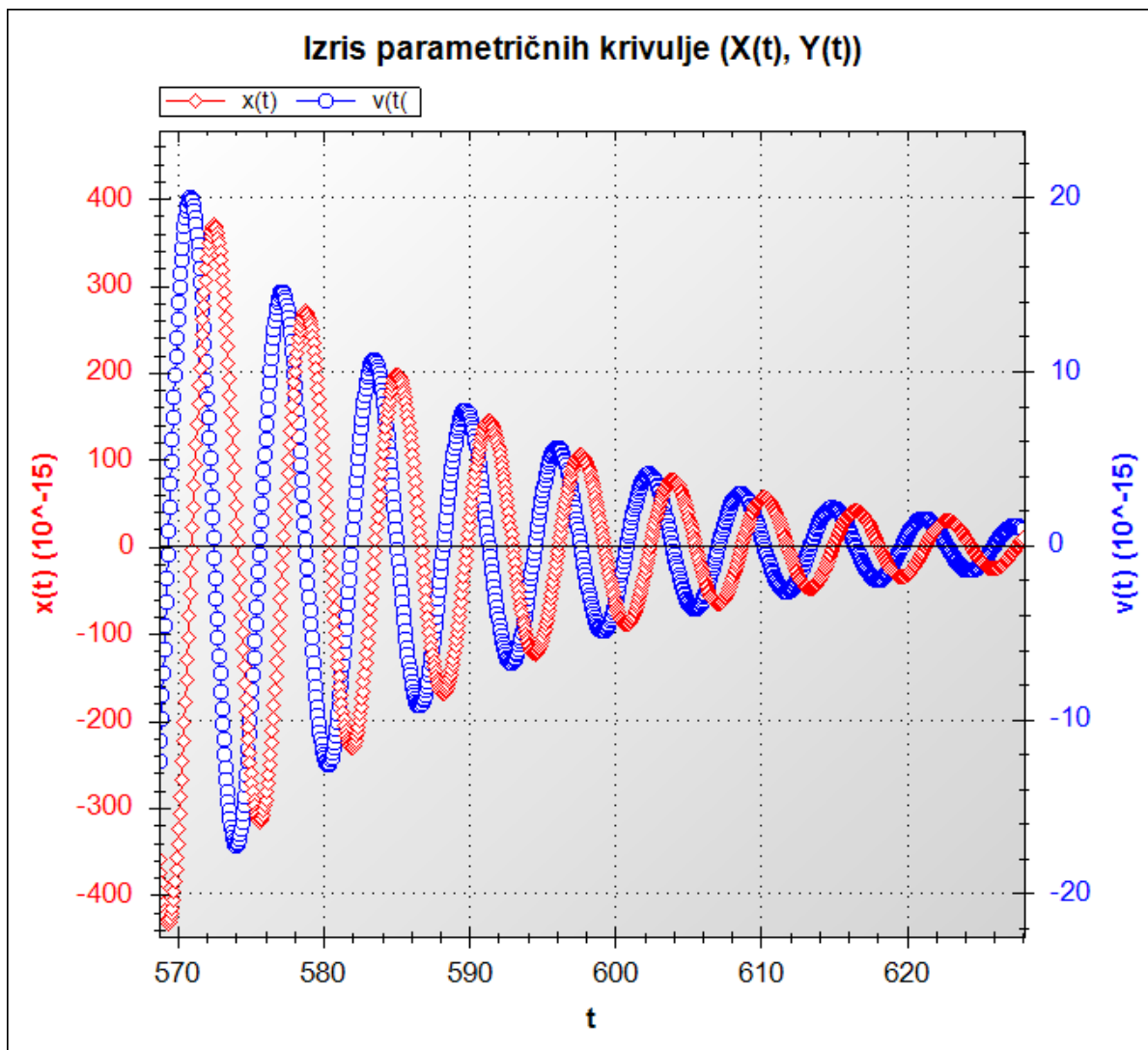
Na spodnji sliki je prikazano nedušeno nihanje do 100 nihajev ($\omega_0 t = 100 \cdot 2\pi$). Relativna napaka pomika pos totem nihaju je reda velikosti 10^{-5} , pri računanju smo uporabili 100 vozlišč na nihaj.

¹ V praksi velikokrat izkoriščajo povezavo med fizikalnimi zakoni in metodo reševanja problemov. Pri simulaciji mehanskih sistemov na primer obstajajo metode, ki natančno ohranjajo energijo, gibalno ali vrtilno količino.



9989: $t = 627,564548481097$, $x(t) = 0,728909798803644$
 9990: $t = 627,627380334169$, $x(t) = 0,770458066732506$
 9991: $t = 627,690212187241$, $x(t) = 0,808965689415052$
 9992: $t = 627,753044040312$, $x(t) = 0,844280694899082$
 9993: $t = 627,815875893384$, $x(t) = 0,87626371103077$
 9994: $t = 627,878707746456$, $x(t) = 0,904788515492723$
 9995: $t = 627,941539599528$, $x(t) = 0,929742533945667$
 9996: $t = 628,0043714526$, $x(t) = 0,951027284307804$
 9997: $t = 628,067203305671$, $x(t) = 0,968558765418484$
 9998: $t = 628,130035158743$, $x(t) = 0,982267788552312$
 9999: $t = 628,192867011815$, $x(t) = 0,992100250475352$
 10000: $t = 628,255698864887$, $x(t) = 0,998017346965808$
 10001: $t = 628,318530717959$, $x(t) = 0,999995725956508$

Drug zanimiv preizkus metode je lahko reševanje enačbe dušenega nihanja po veliko nihajih. Amplituda nihanja pri dušenju eksponentno pojemata, zato se intenzivnost nihanja po velikem številu nihajev drastično zmanjša. Če se izračuni dobro ujemajo z analitično rešitvijo po veliko nihajih, je to dobra referenca za zanesljivost metode. Spodnja slika prikazuje izračunano dušeno nihanje po 100 nihajih (pod sliko je še tabela zadnjih izračunanih pomikov). Amplituda v tem času pade na približno 10^{-14} začetne. Pri računanju je bilo spet uporabljenih 100 vozlišč na vsak nihaj.



9981: $t = 627,061893656523$, $x(t) = -1,20371295494624E-14$
 9982: $t = 627,124725509595$, $x(t) = -1,06622666138548E-14$
 9983: $t = 627,187557362666$, $x(t) = -9,25406794496155E-15$
 9984: $t = 627,250389215738$, $x(t) = -7,81828243306116E-15$
 9985: $t = 627,31322106881$, $x(t) = -6,36073149166894E-15$
 9986: $t = 627,376052921882$, $x(t) = -4,88728570086048E-15$
 9987: $t = 627,438884774954$, $x(t) = -3,40384140345983E-15$
 9988: $t = 627,501716628025$, $x(t) = -1,91629734646206E-15$
 9989: $t = 627,564548481097$, $x(t) = -4,30531459303483E-16$
 9990: $t = 627,627380334169$, $x(t) = 1,04762214051943E-15$
 9991: $t = 627,690212187241$, $x(t) = 2,51239582544252E-15$
 9992: $t = 627,753044040312$, $x(t) = 3,95811073220051E-15$
 9993: $t = 627,815875893384$, $x(t) = 5,37919854746465E-15$
 9994: $t = 627,878707746456$, $x(t) = 6,77022272210414E-15$
 9995: $t = 627,941539599528$, $x(t) = 8,12589903418966E-15$
 9996: $t = 628,0043714526$, $x(t) = 9,4411154239198E-15$
 9997: $t = 628,067203305671$, $x(t) = 1,07109510269967E-14$
 9998: $t = 628,130035158743$, $x(t) = 1,19306943365911E-14$
 9999: $t = 628,192867011815$, $x(t) = 1,30958604279018E-14$

10000: $t = 628,255698864887$, $x(t) = 1,42022071834183E-14$
10001: $t = 628,318530717959$, $x(t) = 1,52457504613105E-14$

1.4.2 Konvergenca metode

Napako metode lahko velikokrat dobro ocenimo tako, da pogledamo, kako se razlike med zaporednimi rezultati manjšajo, ko manjšamo korak računanja oziroma večamo število iteracij pri iterativnih metodah. Pristop temelji na tem, da ima metoda določen red konvergence, ki se v neki okolici rešitve jasno izrazi in se zato v bližini rešitve napaka hitro manjša.

Pri metodi Runge Kutta 4. reda je napaka v posameznem koraku reda h^5 , celotna akumulirana napaka pa je reda h^4 . Recimo, da pri tej metodi manjšamo dolžino koraka h v geometrijskem zaporedju (npr. vsakič rapolovimo korak) in primerjamo izračunane vrednosti iskane funkcije pri določenem času. V limiti zaradi 4. reda konvergence velja naslednja enačba za napako:

$$E(h) = R(h) - R^* \approx K h^4, \quad (17)$$

kjer je K neznana konstanta. V zgornji enačbi je h korak uporabljen pri numeričnem postopku, R je numerična rešitev problema pri določenem koraku, R^* je natančna rešitev problema (ki je velikokrat ne poznamo, ker problem ni analitično rešljiv), $E(h)$ pa je napaka numerične rešitve pri danem koraku. Iz tega dobimo:

$$E\left(\frac{h}{2}\right) \approx K \left(\frac{h}{2}\right)^4 = \frac{1}{16} K h^4. \quad (18)$$

Za razliko napak pri izbranem in polovičnem koraku potem velja

$$E(h) - E\left(\frac{h}{2}\right) = R(h) - R\left(\frac{h}{2}\right) \approx K h^4 \left(1 - \frac{1}{16}\right) = \frac{15}{16} K h^4 \approx \frac{15}{16} E(h). \quad (19)$$

Iz tega lahko dobimo oceno napake pri koraku h , ko imamo približka izračunana po numeričnem postopku enkrat s korakom h in enkrat s korakom $h/2$:

$$E(h) \approx \frac{16}{15} \left(R(h) - R\left(\frac{h}{2}\right) \right) \quad (20)$$

Vidimo, da gre napaka v danem koraku proti $16/15$ razlike med rezultatom pri danem koraku in rezultatom pri polovičnem koraku. Lokalno torej napaka pri določeni dolžini koraka ni veliko večja kot razlika med izračunano vrednostjo pri danem koraku in vrednostjo pri polovičnem koraku. Zgornjega rezultata pa ne moremo kar na slepo uporabiti, ampak se moramo prepričati, da je izračunan rezultat pri izhodiščnem koraku h že dovolj blizu rešitve, da metoda hitro konvergira pri zmanjševanju h in lahko vzamemo, da enačba (17) dovolj dobro velja s konstantnim koeficientom K (to je bila predpostavka pri izpeljavi, velja pa v limiti). To vidimo tako, da večkrat zapored dvakrat zmanjšamo korak metode in če je očitno, da rezultati konvergirajo v skladu z redom konvergence

(torej, da gre razlika izračunanih vrednosti dovolj hitro proti 0), lahko formulo uporabimo za oceno napake. Upoštevati je treba tudi, da koraka ne moremo poljubno zmanjševati, ker pri nekem koraku začnejo zaokrožitvene napake prevladovati nad napako metode (ocena velja za slednjo) in postane ocena neuporabna, poleg tega se tam celotna napaka začne povečevati z nadaljnjim zmanjševanjem koraka, tako da to nima več smisla¹.

Če večkrat zaporedoma zmanjšamo dolžino koraka za polovico, se blizu rešitve tudi razlike med zaporednimi izračuni manjšajo podobno kot napaka. Iz enačbe (19) izluščimo

$$R(h) - R\left(\frac{h}{2}\right) \approx \frac{15}{16} K h^4. \quad (21)$$

Če isto enačbo zapišemo za desetkrat manjši korak (torej za $h/2$ namesto za h), dobimo

$$R\left(\frac{h}{2}\right) - R\left(\frac{h}{2^2}\right) \approx \frac{15}{16} K \left(\frac{h}{2}\right)^4 = \frac{1}{16} \cdot \frac{15}{16} K h^4. \quad (22)$$

Z deljenjem obeh enačb dobimo

$$R\left(\frac{h}{2}\right) - R\left(\frac{h}{2^2}\right) \approx \frac{1}{16} \left(R(h) - R\left(\frac{h}{2}\right) \right). \quad (23)$$

Iz povedanega lahko izluščimo strategijo za oceno napake metode, ki je ključnega pomena za praktično uporabnost numeričnega postopka. Najprej rešimo problem z zmerno dolžino koraka, za katero po občutku pričakujemo, da bi natančnost lahko bila blizu zahtevane. Nič hudega ni, če se pri tem precej uštejemo, ker bomo to ugotovili v nadaljnjem postopku. Nato problem še večkrat zaporedoma rešimo z večjo natančnostjo, pri čemer vsakič zmanjšamo korak za enak faktor. Uporabimo lahko kar faktor 2, v tem primeru veljata oceni (20) in (23), vendar lahko podobne ocene izpeljemo tudi za večje faktorje. Na splošno velja, da se pri metodah višjega reda splača uporabiti manjši faktor (npr. 2 ali 4 pri metodah 4. reda), pri metodah nižjega reda pa večji (npr. 10 pri metodah 1. reda). Ko manjšamo korak, primerjamo razlike med zaporednimi rezultati. Ko je vidno, da se razlike manjšajo v skladu z enačbo (23) (oz. ustrezno spremenjeno enačbo za metode drugih redov ali za drugačen faktor zmanjševanja koraka), lahko uporabimo enačbo (20) za oceno napake. Ko je ocenjena napaka pod zahtevano, lahko postopek zmanjševanja koraka ustavimo in uporabimo dobljeno rešitev skupaj z oceno napake.

V praksi koraka ne moremo poljubno zmanjševati koraka zaradi zaokrožitvenih napak, ki se v našem primeru povečujejo z izboljševanjem natančnosti metode (torej z zmanjševanjem dolžine koraka), ker se s tem povečuje število osnovnih aritmetičnih operacij, ki prispevajo k zaokrožitvenim napakam. Ko velikostni red zaokrožitvenih napak postane primerljiv z velikostnim redom napake metode, oceni (20) in (23) ne držita več in tudi nima smisla nadalje zmanjševati koraka, ker s tem ne pridobimo pri natančnosti. Ko dosežemo mejni velikostni red koraka, pri

¹ V tem dokumentu je ocena za napako samo izpeljana in ni dejansko uporabljena. V poročilu za končno nalogo bi na tem mestu sledila tabela izračunanih vrednostih pri različnih korakih h za izbrano število časov in bi bila podana ocena napake na podlagi tabele.

omenjenem postopku, navadno jasno vidimo po tem, da se razlike med rezultati ne obnašajo več približno v skladu z enačbo (23). Problem lahko nastane, če že začnemo s premajhnim korakom. Zaradi tega se splača v okviru preverjanja metode vedno začeti z razmeroma velikim korakom in korak zmanjševati vse do meje, ko zaokrožitvene napake prevladajo nad napako metode, če to dopušča hitrost računalnika.

1.4.2.1 Opomba: konvergenca pri metodah nižjega reda

Podobno izpeljavo lahko naredimo tudi za metode nižjih redov. Za Eulerjevo metodo, ki je 1. reda, imamo

$$E(h) = R(h) - R^* \approx Ch, \quad (24)$$

iz česar lahko izpeljemo oceno za napako

$$E(h) \approx 2 \left(R(h) - R\left(\frac{h}{2}\right) \right) \quad (25)$$

Ocena je podobna kot za metodo 4. reda, le z manjšim faktorjem. Problem pri metodah nižjega reda pa je v tem, da se napaka ne zmanjšuje dovolj hitro, ko manjšamo dolžino koraka – pri razpolovljenem koraku se zmanjša le za polovico, medtem ko se pri metodi 4. reda zmanjša za četrtno. Zato je tu težje preveriti, ali je trenutni korak že v območju dovolj blizu rešitve, ko lahko neznani koeficient K vzamemo za konstanten in tako dovolj dobro veljaja predpostavka uporabljena pri izpeljavi enačbe (25).

Pristop, ki ga lahko uberemo pri metodah nižjega reda, je, da rešitev izračunano pri danem h primerjamo z rešitvijo pri večkrat razpolovljenem koraku, oziroma (kar je ekvivalentno) z rešitvijo pri koraku, ki je zmanjšan za večji faktor (to lahko seveda naredimo tudi pri metodah višjega reda, vendar velikokrat ni potrebe za to in lahko prihranimo računski čas, če uporabljamo manjši faktor).

Če npr. trikrat zmanjšamo korak za faktor 2, je to ekvivalentno enkratnemu zmanjšanju koraka za faktor $2^3=8$. Približno za enak faktor se pri metodi 1. reda zmanjša napaka, če smo dovolj blizu rešitve, da približno velja enačba (25) s konstantnim koeficientom K . Če zmanjšamo korak še dvakrat (torej za faktor 16), dobimo z metodo 1. reda podobno zmanjšanje napake – za faktor 16 – kot pri metodi 4. reda ob razpolovitvi dolžine koraka.

Sedaj pogledajmo, kaj se zgodi pri metodi 1. reda, ko smo dovolj blizu rešitve in zmanjšujemo korak za nekoliko večji faktor, npr. vsakič za desetkrat. Iz enačbe (25) vidimo, da se pri vsakem zmanjšanju koraka za faktor 10 tudi napaka zmanjša za faktor 10. Ustrezno se zmanjšujejo tudi razlike rezultatov med zaporednimi zmanjševanji koraka, zato je bolje vidno, kdaj dosežemo območje, ko lahko uporabimo enačbo (25) s konstantnim koeficientom K . Na že znani način lahko izpeljemo tudi enačbo za oceno napake v tem območju:

$$E(h) = R(h) - R^* \approx Ch, \quad (26)$$

$$E\left(\frac{h}{10}\right) \approx C\left(\frac{h}{10}\right) \approx \frac{1}{10} E(h). \quad (27)$$

$$E(h) - E\left(\frac{h}{10}\right) = R(h) - R\left(\frac{h}{10}\right) \approx Ch\left(1 - \frac{1}{10}\right) = \frac{9}{10}Ch \approx \frac{9}{10}E(h). \quad (28)$$

$$E(h) \approx \frac{10}{9}\left(R(h) - R\left(\frac{h}{10}\right)\right) \quad (29)$$

Ocena je podobna kot v enačbah (20) in (25), le z drugimi faktorji.

Če večkrat zaporedoma zmanjšamo dolžino koraka za desetino, se blizu rešitve tudi razlike med zaporednimi izračuni manjšajo podobno kot napaka (izpeljava je spet podobna kot pri metodah višjega reda):

$$R\left(\frac{h}{10}\right) - R\left(\frac{h}{10^2}\right) \approx \frac{1}{10}\left(R(h) - R\left(\frac{h}{10}\right)\right). \quad (30)$$

Strategija za oceno napake metode je podobna kot pri metodah višjega reda: začnemo pri nekem zmernem koraku, zmanjšujemo dolžino koraka za izbrani faktor in primerjamo razlike med zaporednimi izračuni pri različnih dolžinah koraka. Razlika je le v tem, da je pri metodah nižjega reda navadno bolj smiselno zmanjševati korak za večji faktor kot pri metodah višjega reda. Pri metodi nižjega reda moramo namreč korak zmanjšati za večji faktor, da dosežemo podobno zmanjšanje napake metode oz. razlike med zaporednimi rezultati kot pri metodi višjega reda.

2 DODATNA DOMAČA NALOGA – VSILJENO NIHANJE

Pri vsiljenem nihanju delujemo na nihalo s časovno odvisno zunanjo silo $F_f(t)$. Vzemimo, da se sila periodično spreminja na naslednji način:

$$F_f(t) = F_0 \sin(\omega_f t). \quad (31)$$

V tem primeru dobimo naslednjo enačbo za gibanje vzmetnega nihala:

$$m \frac{d^2 x}{dt^2} = -kx - \kappa \frac{dx}{dt} + F_0 \sin(\omega_f t). \quad (32)$$

Izračunaj amplitudo takšnega vsiljenega nihanja vzmetnega nihala po 100 periodah zunanje sile pri kotnih frekvencah vsiljenega nihanja

$$\omega_f = \omega_0 k_\omega^n. \quad (33)$$

za $k_\omega = 1, 2$ in $n = -5, -4, -3, \dots, 3, 4, 5$. Pri tem je ω_0 lastna kotna frekvenca vzmetnega nihala,

$$\omega_0 = \sqrt{\frac{k}{m}}. \quad (34)$$

Podatki so naslednji:

$$m = 1 \text{ kg}$$

$$k = 100 \text{ N/m}$$

$$\kappa = 0,1 \text{ kg m/s}$$

$$F_0 = 10 \text{ N}$$

Zahtevane amplitude oceni tako, da najprej izračunaš pomik uteži nihala $x(t)$ v času od 0 do $200\pi/\omega_f$ (100 period zunanje sile), nato pa poiščeš prvi lokalni ekstrem od zadnjega izračunanega pomika proti času 0. Absolutno vrednost tega lokalnega ekstrema vzameš za zahtevano amplitudo.

Pri reševanju diferencialnih enačb vsako periodo nihanja zunanje sile razdeli vsaj na 60 podintervalov. Fina delitev je potrebna tudi za dobro oceno ekstrema, saj pomke izračunamo le v vozliščih. Druga možnost bi bila uporaba interpolacije pri ocenjevanju ekstremov, to pa je malo bolj komplicirano.

Koristne povezave:

<http://en.wikipedia.org/wiki/Oscillation>

http://en.wikipedia.org/wiki/Harmonic_oscillator

<http://www.cmt.phys.kyushu-u.ac.jp/virtuallab/phys/physmath/decrease-e.html>