

Matematično-fizikalni Praktikum

Enačbe Hode

Tilen Šket, 28221057

Navodila:

Preizkusi preprosto Eulerjevo metodo ter nato še čim več naprednejših metod (Midpoint, Runge-Kutto 4. reda, Adams-Bashfort-Moultonov prediktor-korektor ...) na primeru z začetnima temperaturama $y(0) = 21$ ali $y(0) = -15$, zunanjo temperaturo $y_{\text{zun}} = -5$ in parametrom $k = 0.1$. Kako velik (ali majhen) korak h je potreben? Izberi metodo (in korak) za izračun družine rešitev pri različnih vrednostih parametra k .

28. november 2024

Kazalo

1	Uvod	3
1.1	Eulerjeva metoda	3
1.2	Midpoint	3
1.3	Runge-Kutta	4
1.4	Adams-Bashfort-Moulton	4
1.5	Opis naloge	5
2	Naloga	5
3	Zaključek	8

1 Uvod

Velikokrat lahko slišimo, da so diferencialne enačbe jezik narave, saj vsak fizikalen opis le te, vsebuje diferencialne enačbe v neki obliki. Lahko pogledamo vsem najbrž znan primer 2. Newton-ov zakon, ki je pogosto zapisan v znani obliki $F = ma$ je primer diferencialne enačbe. Čeprav pri tem zakonu sprva ni videti kje se skrivajo odvodi, se študent fizike hitro poduči, da je pospešil a v resnici drugi odvod naše pozicije, velikokrat označene x , torej $a = \frac{d^2x}{dt^2}$.

Diferencialne enačbe v naravi prihajajo v veliko različnih okusih. Največkrat so odvodi časovni, vendar še zdaleč ne vedno. Obstajajo tudi parcialne diferencialne enačbe (*PDE*), kjer nastopajo odvodi po različnih spremenljivka. Najenostavnejše oblike diferencialnih enačb pa so tako imenovane navadne diferencialne enačbe, znane tudi pod kratico *ODE*, ki izhaja iz angleškega izraza *Ordinary Differential Equation*. Pri tej nalogi, se bomo ukvarjali z enačbami tipa ODE.

Diferencialne enačbe v praksi rešujemo na različne načine. Najbolje je, če je enačbo možno rešiti analitično, saj potem poznamo rešitev do poljubne natančnosti. Vendar je to velikokrat neizvedljivo in moramo namesto tega uporabiti numerične metode.

Problemu kjer poznamo začetne pogoje in ODE sistema, rečemo Eulerjev problem ali v angleščini *Initial Value Problem*, s kratico *IVP*. Slovensko ime je tak tip problema dobil po enem izmed prvih, ki se je ukvarjal z numeričnim reševanjem takšnih problemov, po *Leonhardu Eulerju*. V svojem delu je izdelal tudi metodo za reševanje takšnih problemov, ki se po njemu imenuje *Eulerjeva metoda* in jo pogosto vzamemo za osnovo. Skozi stoletja so se poleg te razvile tudi novejšje, bolj napredne metode, kot so *Heunova*, *Midpoint*, *Runge-Kutta*, *Adams-Bashfort-Moulton* ...

1.1 Eulerjeva metoda

Pri vseh teh metodah naslednjo vrednost funkcije izračunamo iterativno iz prejšnje in njenega odvoda, ki ga dobimo s klicem podanega ODE-ja, ki ga bom od tukaj dalje označeval z f . Izberemo si korak h , ki je razdalja med zaporednima točkama, kjer računamo vrednost iskane funkcije. Predpis izračuna naslednje vrednosti po Eulerjevi metodi je

$$y(x+h) = y(x) + h f(x, y(x)) . \quad (1)$$

1.2 Midpoint

Kot kritika Eulerjevi metodi, opazimo predpostavko, da se vrednost odvoda med posameznima točkama ne spreminja. To vemo, da ni nujno res, zato predpisu dodamo nov člen, kjer odvod med obema točkama povprečimo in tako dobimo nov približek iskani naslednji točki. Omenjen

postopek, je Heunova metoda. Njej ekvivalentna pa je *Midpoint* metoda. Le ta pa vzame predpis iz Eulerjeve metode in naredi le polovico koraka, kjer ponovno izračuna vrednost odvoda in se nato s to vrednostjo odvoda premakne za celoten korak k naslednji točki. Predpis izgleda tako

$$y(x+h) = y(x) + h f\left(x + \frac{h}{2}, y\left(x + \frac{h}{2}\right)\right), \quad (2)$$

z vmesnim računom

$$y\left(x + \frac{h}{2}\right) = y(x) + \frac{h}{2} f(x, y(x)) .$$

1.3 Runge-Kutta

To je v resnici družina numeričnih metod, ki so dandanes ene izmed najbolj uporabljenih, čeprav so stare že več kot 100 let. Za primer opišimo klasično RK-4 metodo. Ponovno uporabljamo podobno idejo, kot pri *Midpoint*, in sicer to, da računamo vrednosti odvoda in same funkcije v nekaterih vmesnih točkah. Vendar tokrat za razliko od prej ne uporabimo le zadnjega približka, vendar upoštevamo vse z nekimi numeričnimi predfaktorji. Predpis za RK-4, vsebuje izračun 4 vmesnih vrednosti, ki jih označimo k_i , kjer je i indeks vrednosti. Predpis tokrat zgleda tako

$$\begin{aligned} k_1 &= f(x, y(x)) , \\ k_2 &= f\left(x + \frac{1}{2}h, y(x) + \frac{h}{2}k_1\right) , \\ k_3 &= f\left(x + \frac{1}{2}h, y(x) + \frac{h}{2}k_2\right) , \\ k_4 &= f(x+h, y(x) + hk_3) , \end{aligned} \quad (3)$$

$$y(x+h) = y(x) + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) . \quad (4)$$

Numerične faktorje pri izračunih poljubne RK metode, lahko zapišemo tudi v tabelo, ki se imenuje *Butcher tableau*. Poleg tega obstaja tudi razširitev metode v *Runge-Kutta-Fehlberg*, ki doda še dinamično spreminjanje koraka h , na podlagi ocene napake, ki smo jo naredili proti napaki, ki jo toleriramo.

1.4 Adams-Bashfort-Moulton

Še zadnja metoda, ki si jo bomo ogledali, pa deluje na nekoliko drugačnem principu, in sicer je prediktor-korektor metoda. Najprej s prediktorjem napovemo, kje je najverjetneje, da se nahaja naslednja točka, nato s korektorjem izbrano pozicijo popravimo, je osnovna ideja te metode. Pri tej metodi moramo najprej z neko drugo izračunati prve štiri točke, s katerimi potem prilagajamo polinom 4. stopnje na te točke in njihove odvode in tako dobimo vrednost v naslednji točki. To se skriva v predpisu Adams-Bashfort-ovega prediktorja

$$y^p(x+h) = y(x) + \frac{1}{24}h (55f(x) - 59f(x-h) + 37f(x-2h) - 9f(x-3h)) .$$

Tukaj je $f(x)$ mišljeno kot $f(x, y(x))$. Nato na podlagi te ocene izračunamo boljšo oceno za naslednjo točko, s pomočjo Adams-Moulton-ovega korektorja

$$y^c(x+h) = y(x) + \frac{1}{24}h(9f^p(x+h) + 19f(x) - 5f(x-h) + f(x-2h)) .$$

Pri tej metodi torej funkcijo f kličemo le dvakrat za vsako novo točko, kar ima v primerih kjer je to zelo ‘drago’ velik pomen.

1.5 Opis naloge

Pri tej nalogi, se bomo ukvarjali z numeričnim reševanjem Eulerjevega problema

$$\frac{dT}{dt} = -k(T - T_{\text{zun}}) , \quad (5)$$

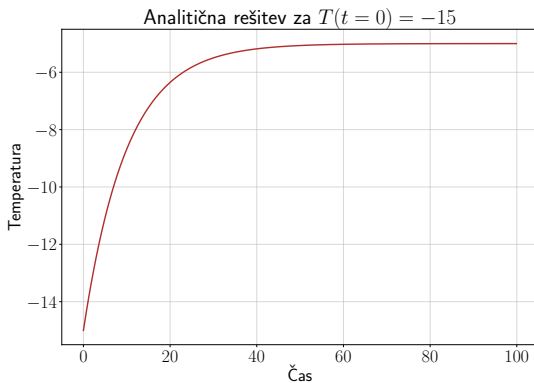
kjer je začetni pogoj $T(t=0) = T_1 = 21$ ali $T(t=0) = T_2 = -15$. Poznamo tudi njeno analitično rešitev

$$T(t) = T_{\text{zun}} + e^{-kt}(T(0) - T_{\text{zun}}) ,$$

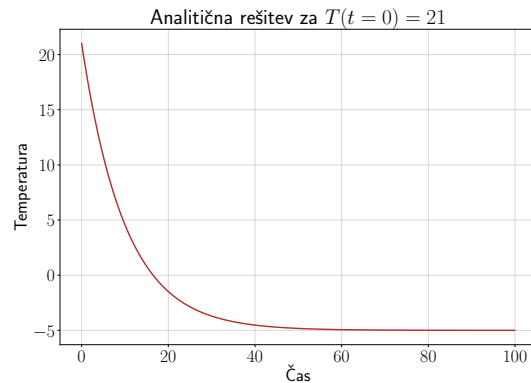
Parametri v enačbi znašajo $T_{\text{zun}} = -5$, $k = 0,1$.

2 Naloga

Pri primerjavi različnih metod reševanja, bom analitično rešitev vzel kot pravilno in na podlagi le te določil napako ostalih metod. Zato si najprej oglejmo, kakšna je analitična rešitev. Za oba začetna pogoja sem rešitvi izrisal na sliki 1.



(a) Analitična rešitev za $T_0 = -15$

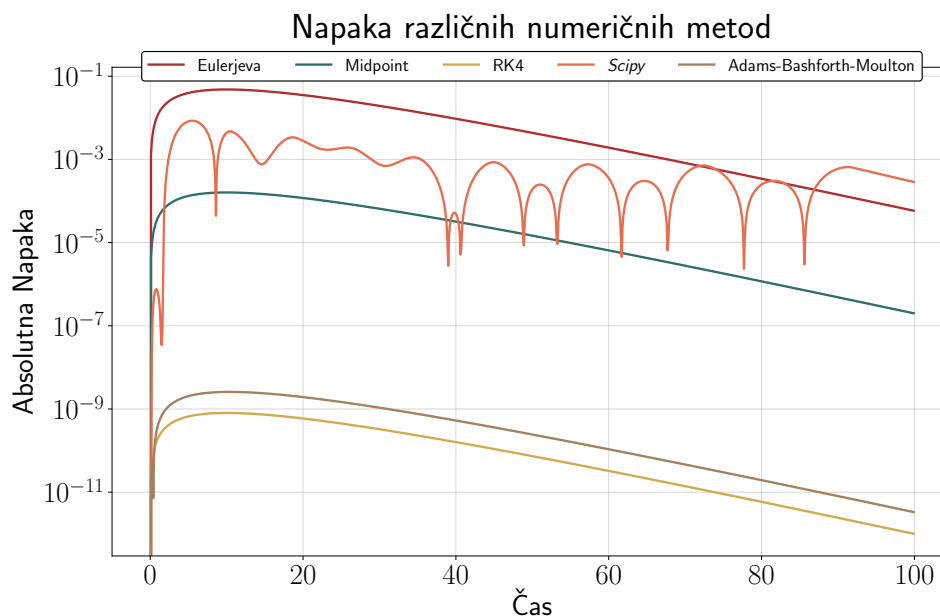


(b) Analitična rešitev za $T_0 = 21$

Slika 1: Analitični rešitvi enačbe pri dveh različnih začetnih pogojih.

Enačbo sem nato rešil z skupaj 5 različnimi numeričnimi metodami, in sicer z Eulerjevo, *Midpoint*, RK-4, Adams-Bashfort-Moulton in metodo, ki je vgrajena v *Scipy* knjižnico v

Pythonu. Primerjal sem napake, ki se naberejo skozi čas pri teh metodah, pri koraku $h = 0,1$. To je prikazano na sliki 2.

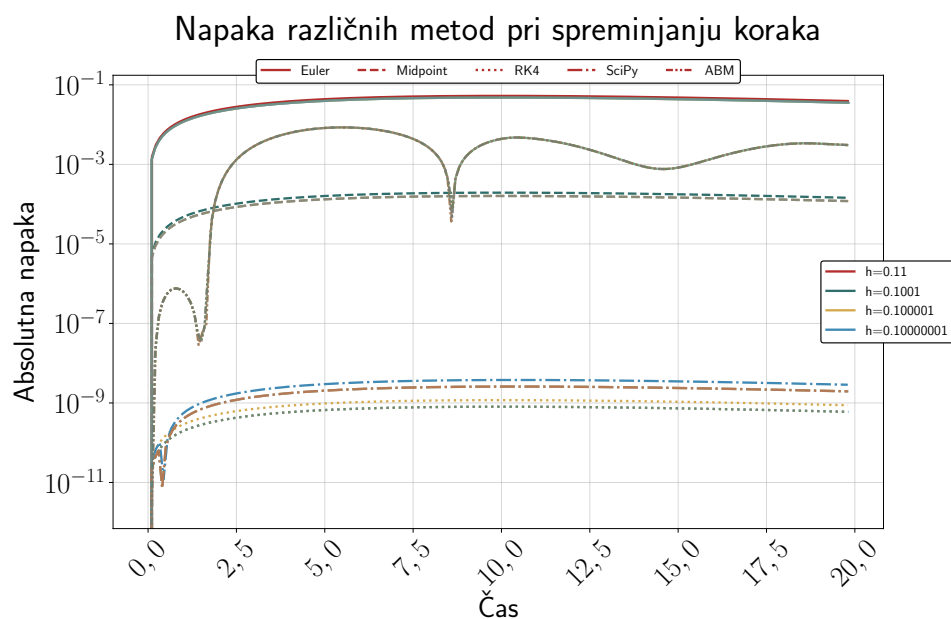


Slika 2: Absolutne napake različnih numeričnih metod. Presenetljiva opazka iz tega grafa je, da je *Scipy* metoda toliko slabša od nekaterih drugih, ki sem jih zakodiral sam. Napake so bile izračunane pri začetnem pogoju T_1 , vendar pri drugem ni očitnih razlik.

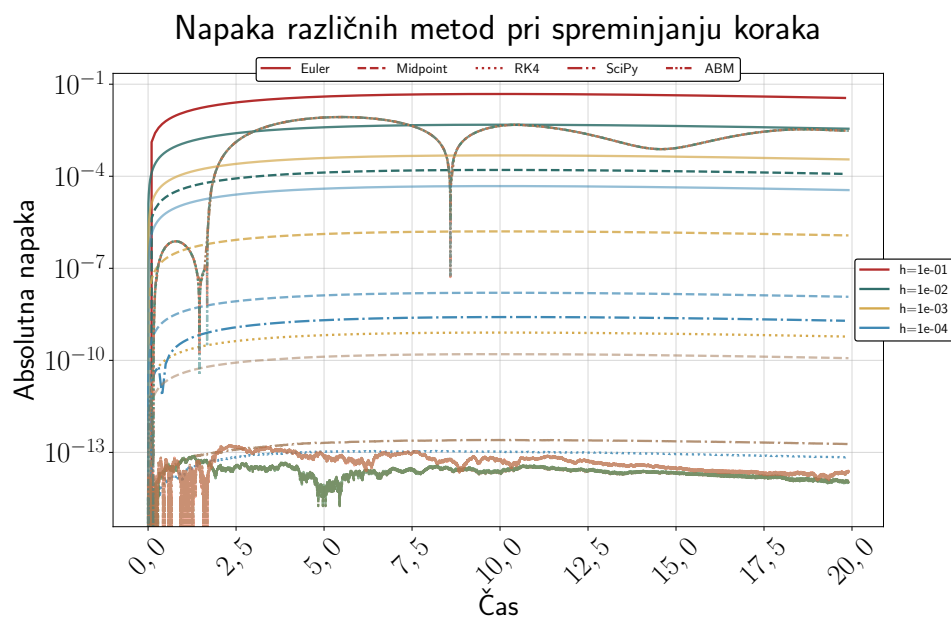
Nato sem pogledal, kako se napako posameznih numeričnih metod spreminjajo z velikostjo koraka. In sicer, sem našel, da je korak okoli $h = 0,1$ nekakšna mejna vrednost, zato sem pogledal, kako se spreminjajo napake, še korak za le malo povečam od te vrednosti. Rezultati so izrisani na sliki 3.

Vprašal sem se tudi, kako vpliva red velikosti koraka na napako posameznih metodo. Graf teh odvisnosti sem izrisal na sliki 4. Iz te slike pa vidimo tudi, da metoda *Scipy* ni odvisna od velikosti koraka, iz česar lahko sklepamo, da uporablja dinamično velikost koraka. Pri dovolj majhnem koraku, pa napaka pride tudi na velikostni red napake zaokroževanja pri uporabi float, pri določenih metodah.

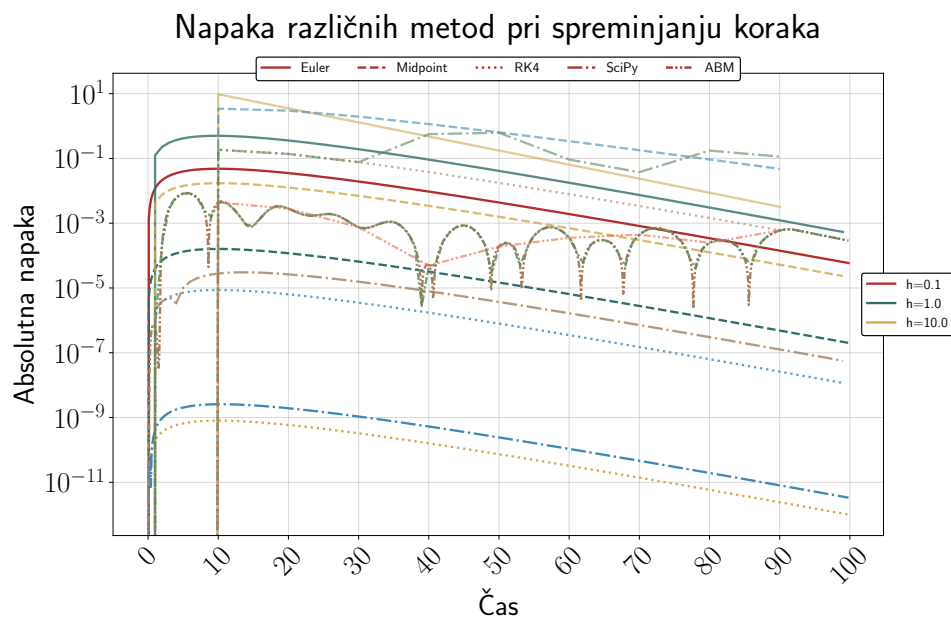
Poglejmo si še, kaj pa se z metodami dogaja, če korak še močno povečamo. Ta primer je izrisan na grafu 5. Opazimo, da Eulerjeva metoda na neki točki prehití nekaj ostalih metod. Medtem ko sama RK4 metoda, še na tej skali deluje zelo dobro.



Slika 3: Graf napak, pri različnih vrednostih koraka v okolici $h = 0,1$.



Slika 4: Graf napak, pri različnih redih velikosti koraka. Opazimo, da je *SciPy* metoda neodvisna od koraka, torej najbrž uporablja dinamično velikost koraka. Opazimo tudi, da pri premajhnih velikostih koraka, napaka pri določenih metodah pride na red velikosti napake zaokroževanja.



Slika 5: Graf napak različnih metod, pri zelo velikih korakih. Zanimiva opazka je, da na neki točki Eulerjeva metoda niti ni več najslabša.

3 Zaključek

Tale zaključek ponovno pišem v noči iz srede na četrtek. Skupen čas, ki sem ga porabil za to nalogo je nekje 6 ur, od česar sta 2 iz predavanj prejšnji četrtek. Ta zaključek bo kratek, saj imam jutri kolokvij in potrebujem spanec, lahko noč!