

Numerične metode 1

Bor Plestenjak

Jadranska 21, soba 4.04

govorilne ure: petek 11-12 oziroma po dogovoru

bor.plestenjak@fmf.uni-lj.si

asistent: Matej Petković

Spletna učilnica FMF

<http://www-lp.fmf.uni-lj.si/plestenjak/vaje/vaje.htm>

osnovna literatura:

B. Plestenjak, Razširjen uvod v numerične metode, DMFA-založništvo, 2015

Režim

- 2 sklopa domačih nalog : 20% pisne ocene
 - uporaba Matlaba
 - preverjanje poteka v računalniški učilnici v obliki kvizov v spletni učilnici
- 3 pisni izpiti : 80% pisne ocene
 - za pozitivno pisno oceno je potrebno:
 - a) iz domačih nalog in pisnega izpita zbrati skupaj vsaj 50%,
 - b) pisati pozitiven pisni izpit (najmanj 40% od potrebnih 50% mora biti zbranih s pisnim izpitom)
 - pozitivna pisna ocena velja do konca izpitnega obdobja (14.9.2018).
 - pisno oceno lahko enkrat popravljate (velja boljša ocena), naslednjič velja zadnja ocena
- ustni izpit
 - pogoj za pristop je pozitivna pisna ocena

Numerično reševanje

Pri *numeričnem reševanju* dane naloge iščemo rešitev v numerični obliki. To pomeni, da npr. namesto $\sqrt{3}$ iščemo 1.73205....

Numerična metoda je postopek, s katerim iz vhodnih numeričnih podatkov s končnim zaporedjem elementarnih operacij izračunamo numerični približek za rezultat določenega problema.

Elementarne operacije so odvisne od okolja, mi bomo pod to šteli

$$+, -, /, * \text{ in } \sqrt{}.$$

Za zgled bomo uporabljali program [Matlab](#). Doma lahko namesto njega uporabljate prosto dostopen program [Octave](#).

Numerično računanje ni isto kot eksaktno računanje

- Vzamemo kalkulator in izračunamo

$$100 * (100/3 - 33) - 100/3.$$

Rezultat je (ne)pričakovan, npr. v Matlabu dobimo

$$2.3448 \cdot 10^{-13}.$$

- Za množenje naj bi veljala asociativnost, toda, če npr. vzamemo

$$x = 0.1234567890$$

$$y = 0.0987654321$$

$$z = 0.9911991199,$$

v Matlabu dobimo

$$x * y * z - z * y * x = 1.7347 \cdot 10^{-18}.$$

1.1 Uvod

Numerična matematika se ukvarja z razvojem in analizo algoritmov za numerično reševanje matematičnih problemov.

Numerične metode 1:

- *linearni sistemi*: poišči $x \in \mathbb{R}^n$, ki reši $Ax = b$ za $A \in \mathbb{R}^{n \times n}$ in $b \in \mathbb{R}^n$.
- *nelinearni sistemi*: poišči rešitev enačbe $f(x) = y$, kjer je $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$.
- *linearni problem najmanjših kvadratov*: poišči $x \in \mathbb{R}^n$, ki minimizira $\|Ax - b\|_2$ za $A \in \mathbb{R}^{m \times n}$ in $b \in \mathbb{R}^m$, kjer je $m > n$.

Numerične metode 2 (Marjeta Knez):

- *lastne vrednosti*: izračunaj lastne vrednosti in vektorje matrike $A \in \mathbb{R}^{n \times n}$.
- *interpolacija*: poišči polinom, ki gre skozi točke $(x_0, f(x_0)), \dots, (x_n, f(x_n))$.
- *integriranje*: izračunaj $\int_a^b f(t)dt$.
- *diferencialne enačbe*: reši začetni problem $y' = f(x, y)$, $y(x_0) = y_0$.

Nadaljevanje nudijo predmeti na 2. stopnji.

Kdaj uporabljamo numerične metode

Ko drugih možnosti ne poznamo, npr.:

- iskanje ničel polinoma pete stopnje: $x^5 + 3x - 1 = 0$,
- reševanje transcendentne enačbe: $x + \ln x = 0$,
- računanje določenega integrala: $\int_0^1 e^{x^2} dx$,
- večina nelinearnih enačb, diferencialnih enačb,

Kadar so udobnejše oz. manj zahtevne od analitičnih rešitev, npr.:

- računanje inverzne matrike velikosti 100×100 ,
- Cardanove formule za ničle kubičnega polinoma.

Prevedba na enostavnejši problem

Pri numeričnem reševanju običajno namesto podanega težkega problema rešujemo lažjega, ki ima bodisi enako ali pa bližnjo rešitev:

- sistem linearnih enačb pretvorimo v trikotno obliko,
- neskončne procese nadomestimo s končnimi procesi,
- neskončno razsežne prostore nadomestimo s končno razsežnimi,
- nelinearni problem nadomestimo z linearnim,
- zapletene funkcije nadomestimo z enostavnejšimi, npr. s polinomi.

Dobra numerična metoda

Glavne zahteve za dobro numerično metodo so:

- *zanesljivost in robustnost*: na enostavnih problemih vedno deluje pravilno, običajno deluje na težjih problemih, kadar pa ne deluje, vrne informacijo o tem.
- *natančnost*: izračuna rešitev tako natančno, kot je to možno glede na natančnost podanih začetnih podatkov.
- *ekonomičnost*: časovna (število operacij) in prostorska (poraba spomina).

Numerične metode se stalno razvijajo. Dejavniki razvoja so:

- novi problemi,
- novi pristopi in novi algoritmi,
- razvoj računalnikov,
- razvoj paralelnih računalnikov.

Absolutna in relativna napaka

Pri numeričnem računanju izračunamo numerični približek za točno rešitev.

Razlika med približkom in točno vrednostjo je napaka približka. Ločimo absolutno in relativno napako.

absolutna napaka = približek – točna vrednost,

relativna napaka = $\frac{\text{absolutna napaka}}{\text{točna vrednost}}$.

Naj bo x točna vrednost, $\hat{x}$ pa približek za x .

- Če je $\hat{x} = x + d_a$, potem je $d_a = \hat{x} - x$ *absolutna napaka*.
- Če je $\hat{x} = x(1 + d_r)$ oziroma $d_r = \frac{\hat{x} - x}{x}$, potem je d_r *relativna napaka*.

1.2 Premična pika

V računalniku so števila zapisana v premični piki kot

$$x = \pm m \cdot b^e,$$

kjer je $m = 0.c_1c_2 \dots c_t$ *mantisa* in

- b : *baza* (2, lahko tudi 10 ali 16),
- t : *dolžina mantise*,
- e : *eksponent* v mejah $L \leq e \leq U$,
- c_i : *števke* v mejah od 0 do $b - 1$.

Če je $c_1 \neq 0$, potem je število *normalizirano*, sicer pa *subnormalizirano*.

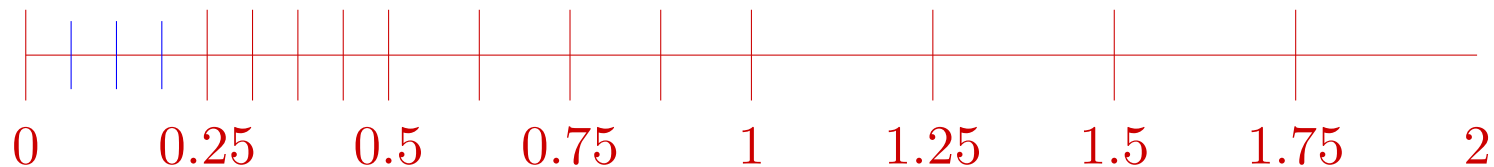
Zapis označimo s $P(b, t, L, U)$.

Npr. $0.1101 \cdot 2^2 = 3.25$.

Zgled

Vsa normalizirana pozitivna predstavljljiva števila iz $P(2, 3, -1, 1)$ so:

$0.100_2 \cdot 2^{-1} =$	0.2500	$0.100_2 \cdot 2^0 =$	0.500	$0.100_2 \cdot 2^1 =$	1.00
$0.101_2 \cdot 2^{-1} =$	0.3125	$0.101_2 \cdot 2^0 =$	0.625	$0.101_2 \cdot 2^1 =$	1.25
$0.110_2 \cdot 2^{-1} =$	0.3750	$0.110_2 \cdot 2^0 =$	0.750	$0.110_2 \cdot 2^1 =$	1.50
$0.111_2 \cdot 2^{-1} =$	0.4375	$0.111_2 \cdot 2^0 =$	0.875	$0.111_2 \cdot 2^1 =$	1.75

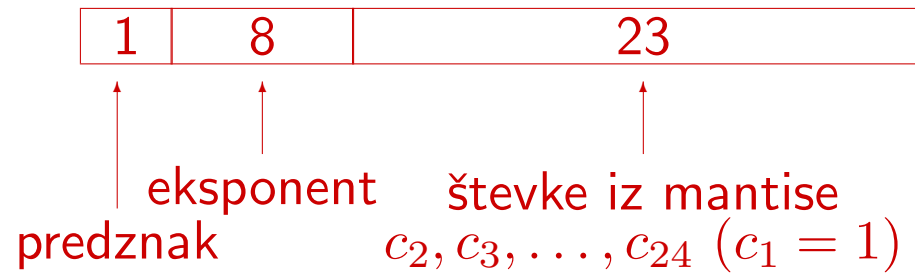


Subnormalizirana števila (možna le pri najmanjšem eksponentu) so:

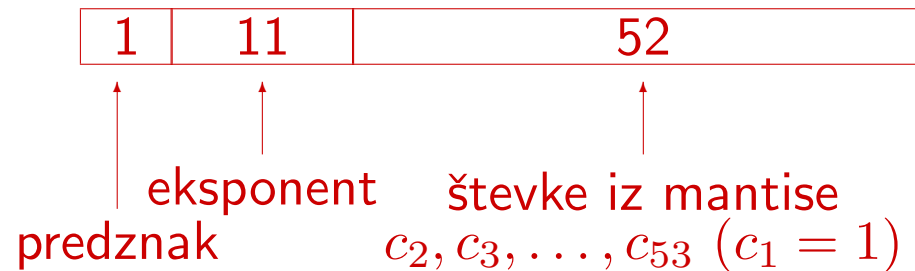
$0.011_2 \cdot 2^{-1} =$	0.01875
$0.010_2 \cdot 2^{-1} =$	0.01250
$0.001_2 \cdot 2^{-1} =$	0.00625

Standard IEEE

- *single*: $P(2, 24, -125, 128)$, število je shranjeno v 32 bitih,



- *double*: $P(2, 53, -1021, 1023)$, število je shranjeno v 64 bitih,



- standard IEEE pozna še števila 0 , ∞ , $-\infty$ in NaN.

Zaokrožanje

Naj bo x pozitivno število z neskončnim zapisom

$$x = 0.d_1d_2 \dots d_t d_{t+1} \dots \cdot b^e,$$

ki leži med najmanjšim in največjim pozitivnim predstavljamim normaliziranim številom. Kandidata za predstavljam približek, ki ga označimo s $\text{fl}(x)$, sta najbližji predstavljam števili z leve in z desne

$$x_- = 0.d_1d_2 \dots d_t \cdot b^e,$$

$$x_+ = (0.d_1d_2 \dots d_t + b^{-t}) \cdot b^e.$$

Pri standardu IEEE uporabimo *zaokrožanje* in za $\text{fl}(x)$ izberemo predstavljam število, ki je najbližje x . V posebnem primeru, ko je x ravno na sredini med x_- in x_+ , med njima izberemo tistega, ki ima sodo zadnjo števk.

desetiško		dvojiško	
x	$\text{fl}(x)$	x	$\text{fl}(x)$
0.16345	0.16	0.10001	0.100
0.16500	0.16	0.10010	0.100
0.16501	0.17	0.10011	0.101
0.17500	0.18	0.10110	0.110

Osnovna zaokrožitvena napaka

Velja

$$\text{fl}(x) = x(1 + \delta) \quad \text{za} \quad |\delta| \leq u,$$

kjer je

$$u = \frac{1}{2} b^{1-t}$$

osnovna zaokrožitvena napaka:

- single: $u = 2^{-24} = 6 \cdot 10^{-8}$,
- double: $u = 2^{-53} = 1 \cdot 10^{-16}$.

Izrek 1. Če za število x velja, da $|x|$ leži na intervalu med najmanjšim in največjim pozitivnim predstavljamim normaliziranim številom, potem velja

$$\frac{|\text{fl}(x) - x|}{|x|} \leq u.$$

Dokaz. Predpostavimo, da je x pozitiven in enak $x = 0.d_1 \dots d_t d_{t+1} d_{t+2} \dots \cdot b^e$

To razdelimo na $x = (y+z)b^e$, kjer je $y = 0.d_1 \dots d_t$ in $z = 0.0 \dots 0d_{t+1}d_{t+2} \dots$

Naj bo $\text{fl}(x) = mb^e$, kjer je $m = 0.c_1 \dots c_t$.

a) Če je prišlo do zaokrožanja navzdol, je $z \leq \frac{1}{2}b^{-t}$ in $m = y$. Sledi

$$\frac{|\text{fl}(x) - x|}{|x|} = \frac{z}{y+z} \leq \frac{z}{y} \leq \frac{\frac{1}{2}b^{-t}}{b^{-1}} = u.$$

Pri oceni smo izbrali največji z in najmanjši y .

b) Pri zaokrožanju navzgor je $\frac{1}{2}b^{-t} \leq z < b^{-t}$ in $m = y + b^{-t}$. Sledi

$$\frac{|\text{fl}(x) - x|}{|x|} = \frac{b^{-t} - z}{y+z} \leq \frac{b^{-t} - z}{y} \leq \frac{b^{-t} - \frac{1}{2}b^{-t}}{b^{-1}} = u.$$

Računanje po standardu IEEE

Standard IEEE zagotavlja, da velja:

- $\text{fl}(x \oplus y) = (x \oplus y)(1 + \delta)$, $|\delta| \leq u$ za $\oplus = +, -, /, *$,
- $\text{fl}(\sqrt{x}) = \sqrt{x}(1 + \delta)$, $|\delta| \leq u$.

Izjema je, če pride do *prekoračitve* (overflow) ali *podkoračitve* (underflow) obsega predstavljenih števil. V tem primeru dobimo po IEEE:

- prekoračitev: $\pm\infty$,
- podkoračitev: 0.

1.3 Občutljivost problema

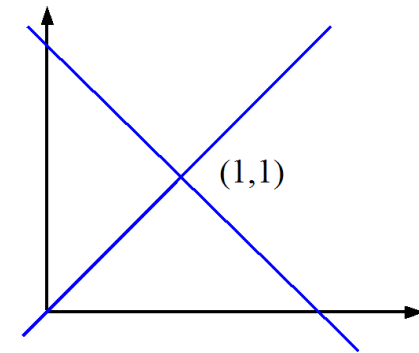
Če se rezultat pri majhni spremembi argumentov (*motnji* oz. *perturbaciji*) ne spremeni veliko, je problem *neobčutljiv*, sicer pa je *občutljiv*.

$$\text{a) } \begin{array}{l} x + y = 2 \\ x - y = 0 \end{array} \implies x = y = 1.$$

Zmotimo desno stran:

$$\begin{array}{l} x + y = 1.9999 \\ x - y = 0.0002 \end{array} \implies x = 1.00005, y = 0.99985.$$

Ta sistem je neobčutljiv.

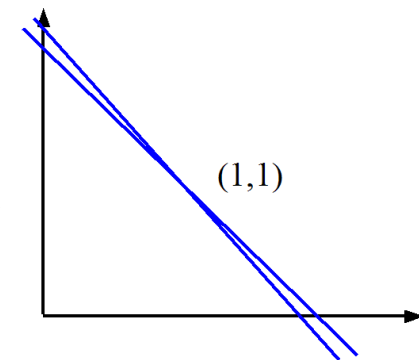


$$\text{b) } \begin{array}{l} x + 0.99y = 1.99 \\ 0.99x + 0.98y = 1.97 \end{array} \implies x = y = 1.$$

Zmotimo desno stran:

$$\begin{array}{l} x + 0.99y = 1.9899 \\ 0.99x + 0.98y = 1.9701 \end{array} \implies x = 2.97, y = -0.99.$$

Ta sistem je zelo občutljiv.



Wilkinsonov zgled

Polinom

$$p(x) = (x - 1)(x - 2) \cdots (x - 20) = x^{20} - 210x^{19} + \cdots + 20!$$

ima ničle $1, 2, \dots, 20$, polinom

$$g(x) = p(x) - 2^{-23}x^{19}$$

pa ima ničle

$$\begin{aligned} x_9 &= 8.91752 \\ x_{10,11} &= 10.0953 \pm 0.64310i \\ &\vdots \\ x_{16,17} &= 16.7307 \pm 2.81263i \\ x_{18,19} &= 19.5024 \pm 1.94033i \\ x_{20} &= 20.8469 \end{aligned}$$

Čeprav so ničle enostavne in lepo separirane, majhna motnja povzroči velike spremembe.

Stopnja občutljivosti

Stopnjo občutljivosti merimo z razmerjem med velikostjo spremembe rezultata in velikostjo spremembe podatkov.

Zgled: Naj bo $f : \mathbb{R} \rightarrow \mathbb{R}$ zvezna in odvedljiva funkcija. Zanima nas razlika med $f(x)$ in $f(x + \delta x)$, kjer je δx majhna motnja.

Absolutna občutljivost: Iz ocene

$$|f(x + \delta x) - f(x)| \approx |f'(x)| \cdot |\delta x|,$$

sledi, da je $|f'(x)|$ *absolutna občutljivost* f v točki x .

Relativna občutljivost: Iz ocene

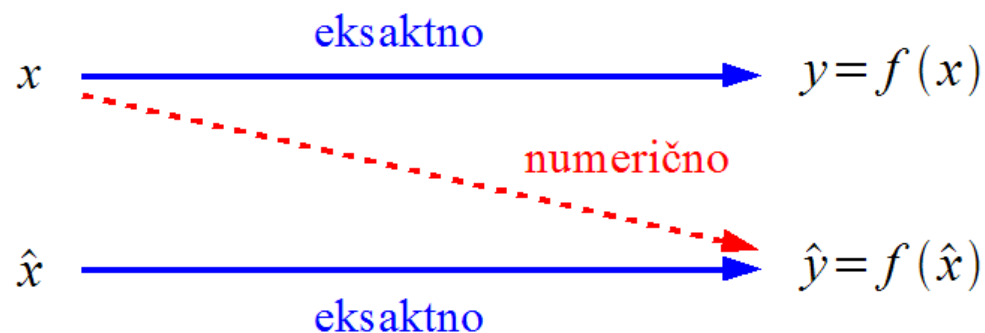
$$\frac{|f(x + \delta x) - f(x)|}{|f(x)|} \approx \frac{|f'(x)| \cdot |x|}{|f(x)|} \cdot \frac{|\delta x|}{|x|}$$

sledi, da je $\frac{|f'(x)| \cdot |x|}{|f(x)|}$ *relativna občutljivost* f v točki x .

1.4 Stabilnost metode

Pri računskem procesu pravimo, da je *stabilen* oz. *nestabilen*, ločimo pa *direktno* in *obratno stabilnost*.

- **direktna analiza:** Iz x namesto $y = f(x)$ izračunamo $\hat{y}$. Če je za vsak x razlika med y in $\hat{y}$ majhna (absolutno oz. relativno), je proces direktno stabilen (absolutno oz. relativno), sicer pa nestabilen.
- **obratna analiza:** Iz x namesto $y = f(x)$ izračunamo $\hat{y}$. Sedaj se vprašamo, za koliko moramo spremeniti argument x v $\hat{x}$, da bo $f(\hat{x}) = \hat{y}$. Če je za vsak x razlika med x in $\hat{x}$ majhna (absolutno oz. relativno), je proces obratno stabilen (absolutno oz. relativno), sicer pa nestabilen.



Občutljivost, stabilnost in natančnost

Algoritem je stabilen, če so rezultati, ki jih vrne, relativno neobčutljivi na motnje, ki se pojavijo zaradi zaokrožitvenih napak med samim računanjem.

Obratno stabilen algoritem tako vrne točno rešitev bližnjega problema.

Če je problem občutljiv, se točna rešitev bližnjega problema lahko zelo razlikuje od točne rešitve začetnega problema in izračunani rezultat je nenatančen.

Nenatančnost je tako lahko posledica:

- uporabe stabilnega algoritma na občutljivem problemu,
- uporabe nestabilnega algoritma na neobčutljivem problemu.

Natančnost je zagotovljena, kadar neobčutljiv problem rešimo s stabilno numerično metodo.