Numerične metode 1

Bor Plestenjak

soba 4.04

bor.plestenjak@fmf.uni-lj.si

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

asistent: Andrej Muhič

Režim

- 2 sklopa domačih nalog - 20% pisne ocene
  - potrebno rešiti in oddati v predvidenem roku
- 3 pisni izpiti - 80% pisne ocene
- za pozitivno pisno oceno je potrebno skupaj iz domačih nalog in pisnega izpita zbrati vsaj 50%
- ustni izpit
  - pogoj za pristop je pozitivna pisna ocena

2. semester bom na študijskem dopustu - ustni izpiti v poletnem izpitnem roku odpadejo!

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 $f : \mathbb{R}^n \to \mathbb{R}^n$.
- *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 Krajnc):

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

Numerično reševanje

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

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 zglede 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

$$\begin{aligned} x &= 0.1234567890 \\ y &= 0.0987654321 \\ z &= 0.9911991199, \end{aligned}$$

v Matlabu dobimo

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

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.

Problem prevedemo na lažji problem

Glavni princip pri numeričnem reševanju je, da namesto podanega težkega problema rešujemo lažjega, ki ima ali enako ali pa zelo bližnjo rešitev. Npr.:

- 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,

Zanimalo nas bo torej:

- kakšna je tipska oblika problema in kako jo učinkovito in stabilno rešimo,
- kako splošni problem učinkovito in stabilno prevedemo na tipsko obliko.

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 vedno izračunamo numerični približek za točno rešitev problema.

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

$$\text{absolutna napaka} \quad = \quad \text{približek} - \text{točna vrednost},$$

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

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

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

1.2 Plavajoča vejica

V računalniku so števila zapisana v plavajoči vejici kot

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

kjer je $m = 0.c_1c_2 \ldots c_t$ *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*.

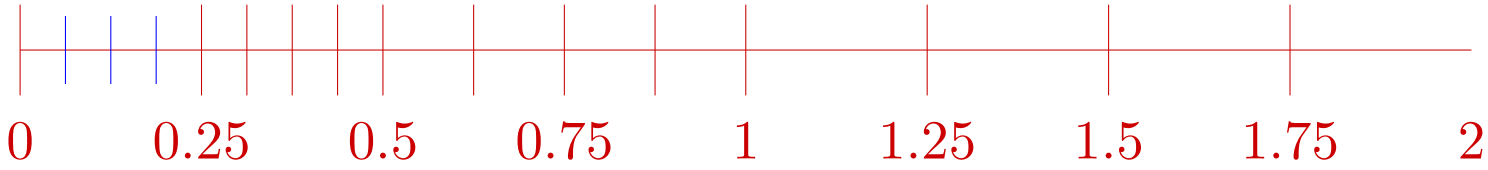
Zapis označimo s $P(b, t, L, U)$.

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

Zgled

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

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

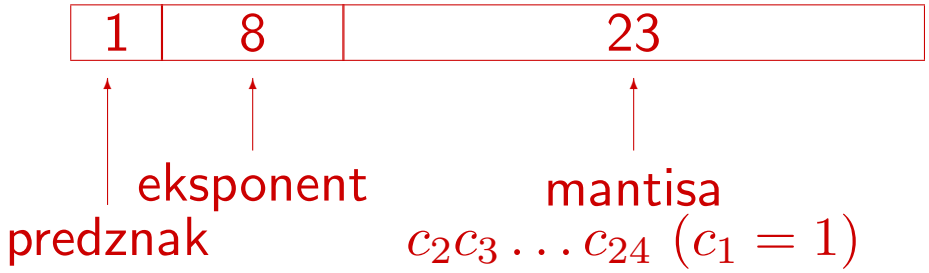


Subnormalizirana števila (možna le pri najmanjšem eksponentu) so:

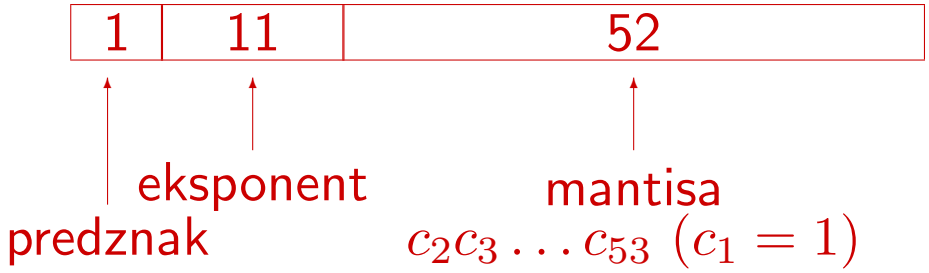
$$
\begin{array}{ll}
0.011_2 \cdot 2^{-1} = & 0.01875 \\
0.010_2 \cdot 2^{-1} = & 0.01250 \\
0.001_2 \cdot 2^{-1} = & 0.00625
\end{array}
$$

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.

Osnovna zaokrožitvena napaka

Števila, ki niso predstavljiva, predstavimo s približki, ki jih dobimo z zaokrožanjem. Naj bo x število in $fl(x)$ najbližje predstavljivo število. Velja

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

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 o osnovni zaokrožitveni napaki

Izrek 1. *Če število x leži znotraj intervala predstavljivih števil, potem velja*

$$\frac{|fl(x) - x|}{|x|} \leq \frac{u}{1+u}.$$

Računanje po standardu IEEE

Standard IEEE zagotavlja, da velja:

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

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

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

Nesreča rakete Arianne

4. junija 1996 je pri prvem poletu rakete Arianne 5, ki naj bi nadomestila manjšo raketo Arianne 4, prišlo do nesreče. Raketa je po 40 sekundah zavila s prave poti in eksplodirala. Izkazalo se je, da je do nesreče prislo zaradi prekoračitve obsega. Ker program ni imel testiranja prekoračitve, se je sesul, s tem pa tudi celoten polet.
Podrobna analiza je pokazala, da je del programa, ki je povzročil napako, prišel iz programa za Arianne 4, kjer je vedno deloval brez napak. Tokrat pa je močnejša raketa povzročila, do so bile izmerjene količine prevelike in prišlo je do prekoračitve obsega.

Več lahko najdete na:
http://www.fmf.uni-lj.si/~jaklicg/arianne.htm