Uvod v numerične metode

Bor Plestenjak

soba 4.04

bor.plestenjak@fmf.uni-lj.si

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

asistent: Gašper Jaklič

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

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, ki iz vhodnih numeričnih podatkov s končnim zaporedjem elementarnih operacij izračuna 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.

Namesto Matlaba lahko doma uporabljate prosto dostopen program Octave.

Numerično računanje se razlikuje od eksaktnega računanja

Računamo s števili, ki so predstavljena v plavajoči vejici.

- 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. Če vzamemo npr.

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

  potem v Matlabu dobimo

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

Numerična matematika

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

Ukvarjali se bomo z naslednjimi problemi:

- *linearni sistem*: poišči $x \in \mathbb{R}^n$, ki reši sistem $Ax = b$ za $A \in \mathbb{R}^{n \times n}$ in $b \in \mathbb{R}^n$.
- *nelinearni sistem*: 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 $m > n$.
- *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 integral $\int_a^b f(t)dt$.
- *diferencialne enačbe*: reši začetni problem $y' = f(x, y)$, $y(x_0) = y_0$.

Povezani predmeti

Prva stopnja:

- Numerična linearna algebra

Druga stopnja:

- Iterativne numerične metode v linearni algebri
- Numerična aproksimacija in interpolacija
- Računalniško podprto (geometrijsko) oblikovanje
- Numerična integracija in navadne diferencialne enačbe
- Numerično reševanje parcialnih diferencialnih enačb
- Numerične metode za linearne sisteme upravljanja

Ozadje

Številne fizikalne, tehnološke, in druge procese lahko simuliramo na računalniku. Postopek je sestavljen iz naslednjih korakov:

1. razvoj matematičnega modela,
2. razvoj numeričnih metod za numerično reševanje modela,
3. implementacija numeričnih metod,
4. simuliranje procesov z računalnikom,
5. predstavitev dobljenih numeričnih rezultatov v pregledni obliki,
6. analiza rezultatov, po potrebi se vrnemo na enega izmed prejšnjih korakov.

Mi se bomo ukvarjali v glavnem s točkama 2. in 3.

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,
- diferencialne enačbe nadomestimo z algebraičnimi enačbami,
- nelinearni problem nadomestimo z linearnim,
- zapletene funkcije nadomestimo z enostavnejšimi, npr. s polinomi,
- splošne matrike nadomestimo z matrikami z enostavnejšo obliko.

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*: na enostavnih problemih vedno deluje pravilno.
- *robustnost*: 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).
- *uporabnost*: lahko jo uporabimo na širokem spektru problemov.
- *prijaznost do uporabnika*: je dobro dokumentirana in ima enostaven uporabniški vmesnik.

Numerične metode se stalno razvijajo. Dejavniki razvoja so:

- novi problemi,
- novi pristopi in novi algoritmi,
- razvoj računalnikov,
- razvoj paralelnih računalnikov oz. večjedrnih procesorjev.

Absolutna in relativna napaka

Pri numeričnem računanju vedno dobimo numerični približek za točno rešitev problema. Razlika med približkom in točno vrednostjo je napaka približka.

Ločimo absolutno in relativno napako.

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

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

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*.

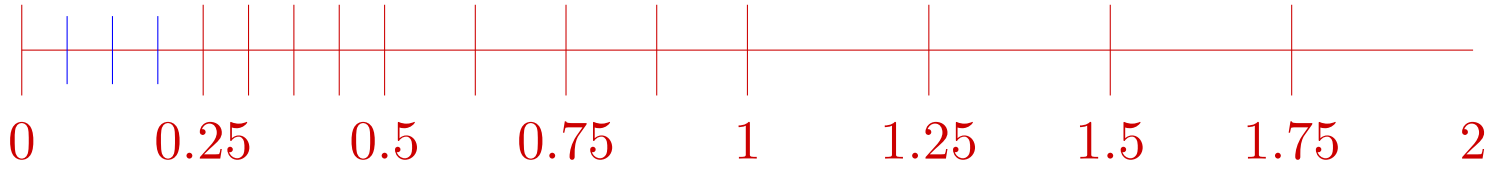
Zapis označimo s $P(b, t, L, U)$.

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

Zgled

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

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

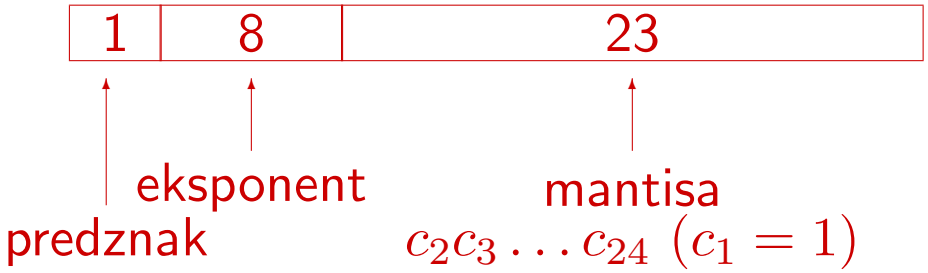


Subnormalizirana števila (možna le pri najmanjšem eksponentu) so:

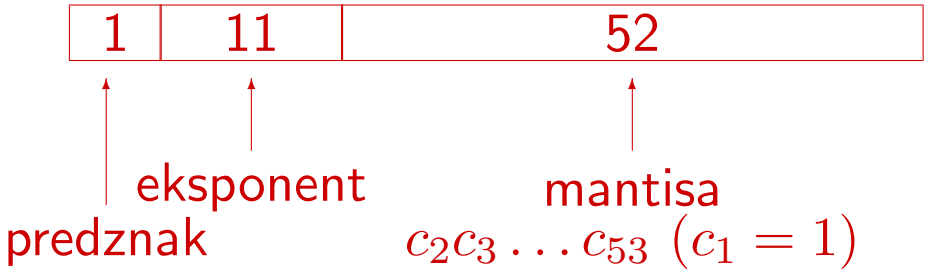
$$
\begin{array}{ll}
0.011_2 \cdot 2^{-1} = & 0.1875 \\
0.010_2 \cdot 2^{-1} = & 0.1250 \\
0.001_2 \cdot 2^{-1} = & 0.0625
\end{array}
$$

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 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 Ariane 5, ki naj bi nadomestila manjšo raketo Ariane 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 Ariane 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

1.3 Vrste napak pri numeričnem računanju

Radi bi izračunali vrednost y funkcije $f : X \to Y$ pri danem x.

Z numerično metodo dobimo približek $\widehat{y}$ za y, razlika

$$D = y - \widehat{y}$$

pa je *celotna napaka* približka.

Izvori napake so:

- nenatančnost začetnih podatkov,
- napaka metode,
- zaokrožitvene napake med računanjem.

Izračun $\sin(\pi/10)$ z osnovnimi operacijami v $P(10, 4, -5, 5)$

točna vrednost je $\sin(\pi/10) = 0.309016994\ldots$

1. Začetni podatek ni predstavljivo število.

   Namesto z $x = \pi/10$ računamo z $\overline{x} = fl(\pi/10) = 0.3142 \cdot 10^0$

2. Kako z osnovnimi operacijami izračunamo vrednost funkcije $\sin$?

   Namesto $\overline{y} = \sin(\overline{x})$ izračunamo $\widetilde{y} = g(\overline{x})$ za $g(x) = x - x^3/6$.

3. Končni rezultat je odvisen od zaporedja operacij za izračun $g(\overline{x})$.

$$\begin{array}{rcccccl} a_1 & = & fl(\overline{x} \cdot \overline{x}) & = & fl(0.09872164) & = & 0.9872 \cdot 10^{-1} \\ a_2 & = & fl(a_1 \cdot \overline{x}) & = & fl(0.03101154) & = & 0.3101 \cdot 10^{-1} \\ a_3 & = & fl(a_2/6) & = & fl(0.0051683\ldots) & = & 0.5168 \cdot 10^{-2} \\ \widehat{y} & = & fl(\overline{x} - a_3) & = & fl(0.309032) & = & 0.3090 \cdot 10^0 \end{array}$$

Absolutna napaka končnega približka je $D = y - \widehat{y} = 1.7 \cdot 10^{-5}$.

Relativna napaka končnega približka je $D/y = 5.5 \cdot 10^{-5}$.

Osnovna zaokrožitvena napaka je $5 \cdot 10^{-4}$.

Celotno napako lahko razdelimo na tri dele

Neodstranljiva napaka: Namesto z x računamo s približkom $\overline{x}$ in namesto $y = f(x)$ izračunamo $\overline{y} = f(\overline{x})$. Neodstranljiva napaka je $D_n = y - \overline{y}$.

D_n je posledica napak začetnih podatkov.

Zgled: *Računanje* $\sin(\pi/10)$ *z osnovnimi operacijami v* $P(10, 4, -5, 5)$

Namesto z $x = \pi/10$ *računamo z* $\overline{x} = 0.3142 \cdot 10^0$

$$D_n = y - \overline{y} = \sin(\pi/10) - \sin(0.3142) = -3.9 \cdot 10^{-5}$$

Napaka metode: Namesto f računamo vrednost funkcije g, ki jo lahko izračunamo s končnim številom operacij. Namesto $\overline{y} = f(\overline{x})$ tako izračunamo $\widetilde{y} = g(\overline{x})$. Napaka metode je $D_m = \overline{y} - \widetilde{y}$.

Pri sami numerični metodi pogosto neskončen proces nadomestimo s končnim (seštejemo le končno členov neskončne vrste, po končnem številu korakov prekinemo iterativno metodo).

Zgled: *Namesto* $\sin(\overline{x})$ *izračunamo* $g(\overline{x})$ *za* $g(x) = x - x^3/6$.

$$D_m = \overline{y} - \widetilde{y} = 2.5 \cdot 10^{-5}$$

Celotna napaka

Zaokrožitvena napaka: Pri računanju $\widetilde{y} = g(\overline{x})$ se pri vsaki računski operaciji pojavi zaokrožitvena napaka, tako da namesto $\widetilde{y}$ izračunamo $\widehat{y}$. Sama vrednost $\widehat{y}$ je odvisna od vrstnega reda operacij in načina izračuna $g(\overline{x})$. Zaokrožitvena napaka je $D_z = \widetilde{y} - \widehat{y}$.

Zgled: *D_z je odvisna je od vrstnega reda in načina računanja $g(\overline{x})$. Primer:*

$$\begin{aligned} a_1 &= fl(\overline{x} \cdot \overline{x}) = fl(0.09872164) = 0.9872 \cdot 10^{-1} \\ a_2 &= fl(a_1 \cdot \overline{x}) = fl(0.03101154) = 0.3101 \cdot 10^{-1} \\ a_3 &= fl(a_2/6) = fl(0.0051683\ldots) = 0.5168 \cdot 10^{-2} \\ \widehat{y} &= fl(\overline{x} - a_3) = fl(0.309032) = 0.3090 \cdot 10^{0} \end{aligned}$$

$$D_z = \widetilde{y} - g(\overline{x}) = 3.0 \cdot 10^{-5}$$

Celotna napaka: Končna napaka je $D = D_n + D_m + D_z$. Velja

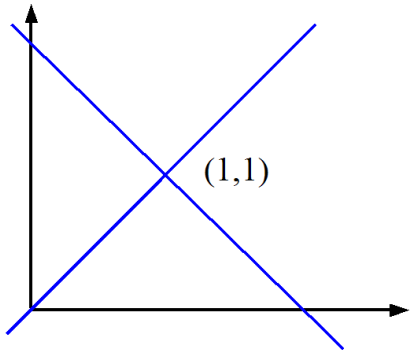
$$|D| \leq |D_n| + |D_m| + |D_z|.$$

Zgled: *Celotna napaka je* $D = D_n + D_m + D_z = 1.6 \cdot 10^{-5}$

1.4 Občutljivost problema

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

a) $$\begin{aligned} x + y &= 2 \\ x - y &= 0 \end{aligned} \implies x = y = 1.$$

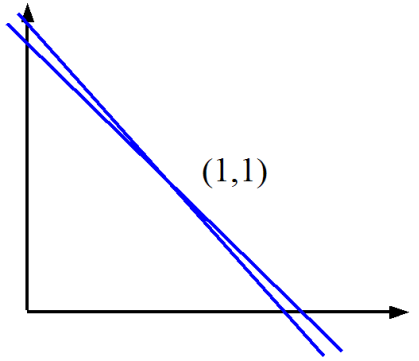


Zmotimo desno stran:

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

Ta sistem je neobčutljiv.

b) $$\begin{aligned} x + 0.99y &= 1.99 \\ 0.99x + 0.98y &= 1.97 \end{aligned} \implies x = y = 1.$$



Zmotimo desno stran:

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

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 $1, 2, \ldots, 20$, 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 vse ničle enostavne in lepo separirane, majhna motnja povzroči velike spremembe.

Wilkinson je s tem primerom pokazal, da računanje lastnih vrednosti preko karakterističnega polinoma ni stabilno.

Stopnja občutljivosti

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

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

Velja

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

torej je $|f'(x)|$ *absolutna občutljivost* f v točki x.

Za oceno relativne napake dobimo

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

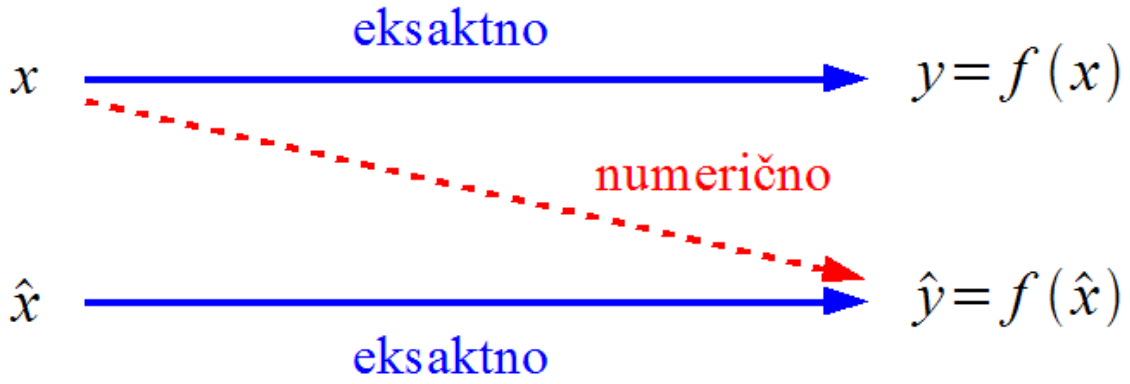
torej je $\dfrac{|f'(x)| \cdot |x|}{|f(x)|}$ *relativna občutljivost* f v točki x.

1.5 Stabilnost metode

Pri računskem procesu pravimo, da je *stabilen* oz. *nestabilen*, ločimo pa *direktno* in *obratno stabilnost*. S tem se ukvarja *analiza zaokrožitvenih napak*.

- direktna stabilnost: Iz x namesto $y = f(x)$ izračunamo $\widehat{y}$. Če je za vsak x razlika med y in $\widehat{y}$ majhna (absolutno oz. relativno), je proces direktno stabilen (absolutno oz. relativno), sicer pa nestabilen.

- obratna stabilnost: Iz x namesto $y = f(x)$ izračunamo $\widehat{y}$. Sedaj se vprašamo, za koliko moramo spremeniti argument x v $\widehat{x}$, da bo $f(\widehat{x}) = \widehat{y}$. Če je za vsak x razlika med x in $\widehat{x}$ 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.

Če rešimo neobčutljiv problem z obratno stabilno metodo, potem se izračunani rezultat ne razlikuje dosti od točnega.

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.