Numerička matematika
1. predavanje

Saša Singer

singer@math.hr
web.math.hr/~singer

PMF – Matematički odjel, Zagreb

Dobar dan, dobro došli

Sadržaj predavanja (početak)

- Uvod u kolegij:
  - Tko sam, što sam i kako do mene.
  - Pravila lijepog ponašanja.
  - Cilj kolegija "Numerička matematika".
  - Pregled sadržaja kolegija.
  - Kolegiji prethodnici — Ponovite!
  - Ostale važne informacije o kolegiju. Posebno:
    - "Pravila igre" ili način polaganja ispita.
    - Literatura.
    - Moja web–stranica.
    - Korisni linkovi — službena web stranica kolegija.
    - Demonstratori.
  - Malo prodike.

Sadržaj predavanja (nastavak)

- Uvodna priča o greškama:
  - Problemi numeričke matematike (zašto ona postoji).
  - Pojam greške, apsolutna i relativna greška.
  - Izvori grešaka — model, ulazni podaci (mjerenje), metoda, zaokruživanje.
    - Ilustracija grešaka na modelnim primjerima.
  - Prikaz brojeva u računalu i greške zaokruživanja (ponavljanje).
  - Greške zaokruživanja osnovnih aritmetičkih operacija.
    - Opasno ili "katastrofalno" kraćenje.
  - "Širenje" grešaka zaokruživanja, stabilni i nestabilni algoritmi.
  - Primjeri iz prakse — posljedice grešaka.

Informacije

Trenutno,

- nema posebnih informacija.

Uvod u kolegij

Sadržaj

- Uvod u kolegij:
  - Tko sam, što sam i kako do mene.
  - Pravila lijepog ponašanja.
  - Cilj kolegija "Numerička matematika".
  - Pregled sadržaja kolegija.
  - Kolegiji prethodnici — Ponovite!
  - Ostale važne informacije o kolegiju. Posebno:
    - "Pravila igre" ili način polaganja ispita.
    - Literatura.
    - Moja web–stranica.
    - Korisni linkovi — službena web stranica kolegija.
    - Demonstratori.
  - Malo prodike.

Na samom početku

- Moja malenkost (u punom "sjaju"):

  doc. dr. sc. Saša Singer

- Službeni osobni podaci:
  - ured (soba, kabinet): 227, drugi kat,
  - e–mail: singer@math.hr
    (Molim plain text poruke.)
  - web stranica: `http://web.math.hr/~singer/`
    (ona "službena": `http://www.math.hr/~singer/`
    je, uglavnom, beskorisna)
- Konzultacije (službeno):
  - petak, 12–14 sati, ili — po dogovoru.

Osnovna pravila "lijepog" ponašanja

Imam nekoliko lijepih zamolbi u rubrici "kultura".

- Prva i osnovna je

  razumna tišina,

  tj. da pričanjem ne ometate izvođenje nastave.
- Zatim, ne kasnite na predavanje.
- Održavajte razuman red u predavaonici.
- Mobilne telefone, molim, utišajte.

Cilj kolegija Numerička matematika

Većina ostalih kolegija na studiju (do sada) bavi se

- tzv. "egzaktnom" ili "pravom" matematikom,

koja izgleda, otprilike, ovako:

- definicija, teorem, dokaz,

uz tek pokoji primjer.

Numerička matematika se ponešto razlikuje od toga:

- orijentirana je prema rješavanju konkretnih praktičnih problema,
- bazirana je na pojmu greške, odnosno, aproksimacije, tj. nije baš "egzaktna".

Cilj kolegija Numerička matematika (nastavak)

Zato kolegij ima nekoliko dosta različitih osnovnih ciljeva:

- spoznavanje neminovnosti pojave grešaka u praktičnom svijetu (izvori i vrste grešaka, važnost ocjene pogreške),
- pregled osnovnih numeričkih metoda za rješavanje nekih "standardnih" problema,
- samostalna primjena tih metoda,
- razvijanje kritičnosti u interpretaciji dobivenih rezultata ("brojevi imaju jedinice").

Ovo zadnje je najvažnije — "da ne bi bilo . . . " (primjeri dolaze na kraju).

Izvedba: više primjera, a manje dokaza!

Pregled sadržaja kolegija

Cijeli kolegij ima 7 "većih" cjelina (poglavlja):

- Uvod u kolegij — greške, uvjetovanost problema, stabilnost algoritama.
- Rješavanje linearnih sustava — tzv. direktne metode (Gaussove eliminacije, LR faktorizacija, faktorizacija Choleskog).
- Aproksimacija i interpolacija — općenito o problemu aproksimacije funkcija, interpolacija polinomom i splineom (splajnom).
- Metoda najmanjih kvadrata — opći diskretni problem, linearizacija, matrična formulacija, QR faktorizacija. Neprekidni problem i ortogonalni polinomi.

Pregled sadržaja kolegija (nastavak)

- Ortogonalni polinomi i generalizirana Hornerova shema.
- Numeričko integriranje — Newton–Cotesove i Gaussove formule.
- Rješavanje nelinearnih jednadžbi — bisekcija, Newton, sekanta, jednostavna iteracija, konstrukcija metoda višeg reda konvergencije.

U nastavnom planu piše još i osma cjelina:

- Uvod u optimizaciju bez ograničenja (1 tjedan).

Međutim, to sigurno nećemo stići — imamo samo 13 tjedana nastave, umjesto ranijih 14.

Kolegiji prethodnici — Ponovite!

Numerička matematika ima prethodnike — to su:

- LA1 = Linearna algebra 1,
- MA2 = Matematička analiza 2.

Stvarno — matematički, trebamo i više od toga:

- LA2 = Linearna algebra 2,
- DRFVV = Diferencijalni račun funkcija više varijabli (parcijalne derivacije, ekstremi).

Dodatno — računarski, trebamo još i Programiranje 1, za:

- prikaz brojeva u računalu, aritmetika računala, greške zaokruživanja,
- pisanje i testiranje osnovnih algoritama.

Pravila polaganja i ocjenjivanja (1)

Elementi ocjenjivanja su:

- domaće zadaće — 10%,
- 1. kolokvij — 40%,
- 2. kolokvij — 50%,
- eventualni završni ispit — 25%.

Zbroj je 125% — nije greška, v. objašnjenje malo niže.

Idemo redom ...

Pravila polaganja i ocjenjivanja (2)

Domaće zadaće iz NM:

- Realizacija ide "automatski" — preko web aplikacije, slično kao na Prog1.

Pogledajte (za jedno 7 dana)

`http://web.math.hr/nastava/unm/zadace.php`

Trenutno ima 7 zadataka iz raznih područja. Bodovi idu prema broju točno riješenih zadataka.

- Zasad: 0, 1, 3, 4, 6, 7, 9, 10.

Rok za predaju zadaća je

- dan drugog kolokvija, do 24 sata (ponoć).

Aplikacija se tada "zatvara za javnost" — bodovi su konačni.

Pravila polaganja i ocjenjivanja (3)

Kolokviji. Tijekom semestra pišu se dva kolokvija.

- 1. kolokvij — ima (najmanje) 40 bodova,
- 2. kolokvij — ima (najmanje) 50 bodova,

tj. oba kolokvija mogu imati "bonus" bodove.

- Na kolokvijima se postavljaju i teorijska pitanja.

Studenti koji ne pristupe nekom od kolokvija tijekom semestra, a svoj nedolazak

- pravovremeno opravdaju na odgovarajući način
  - na pr. medicinskom dokumentacijom,
- kolokvij će polagati u dogovoru s nastavnicima.

Realizacija: Predati molbu s dokumentacijom u referadu.

Pravila polaganja i ocjenjivanja (4)

Za prolaznu ocjenu potrebno je:

- skupiti najmanje 45 bodova (iz kolokvija i zadaća),
- od čega barem 40 bodova mora biti na kolokvijima.

"Prva" ocjena se formira na temelju

- zbroja bodova iz kolokvija i zadaća.

Zato prva 3 elementa ocjenjivanja zbrojeno daju 100%. No,

- možete zaraditi i više od 100 bodova.

Ako ste zadovoljni ocjenom, to je (uglavnom) to!

Pravila polaganja i ocjenjivanja (5)

Završni ispit:

- U načelu — završnog usmenog ispita NEMA.

Mogući izuzeci su:

- po želji — ako niste zadovoljni "prvom" ocjenom,
- po kazni — nastavnik IMA PRAVO pozvati studenta na usmeni ispit (na pr. zbog prepisivanja na kolokviju).

Na završnom ispitu moguće je ostvariti najviše još 25 bodova (v. skalu za ocjene).

Oprez:

- Student može svojim neznanjem na završnom ispitu dobiti i negativnu ocjenu — pasti.

Pravila polaganja i ocjenjivanja (6)

Popravni ispit.

- Studenti koji su tijekom semestra na kolokvijima skupili barem 10 bodova,
- a nisu položili kolegij,

mogu pristupiti popravnom kolokviju.

Popravni kolokvij obuhvaća gradivo cijelog kolegija.

- Na njemu je moguće ostvariti (barem) 100 bodova, tj., opet može biti "bonus" bodova.
- Bodovi iz zadaća se zbrajaju u ocjenu.

Na popravni kolokvij primjenjuje se isto pravilo o završnom ispitu kao i za redovne kolokvije.

Pravila polaganja i ocjenjivanja (7)

Tablica ocjenjivanja:

Bodovi	Ocjena
0 – 44	1
45 – 59	2
60 – 74	3
75 – 89	4
90 i više	5

Onih ≤ 25 bodova na završnom usmenom ispitu znači da

- jako dobrim znanjem možete zaraditi i dvije ocjene više!

Literatura (1)

Osnovna literatura su, naravno,

- predavanja i vježbe,

s popratnim materijalima (predavanja su dostupna na webu).

Moja web stranica za Numeričku matematiku je

`http://web.math.hr/~singer/num_mat/`

Tamo su kompletna predavanja od prošle dvije godine, a stizat će i nova (kako nastaju).

Materijale za predavanja doc. Grubišića možete naći na

`http://web.math.hr/~luka/indexh.html`

Napomena: to nije zamjena za "živu" nastavu (v. kasnije)!

Literatura (2)

Postoji i "stvarna" literatura — u "pisanom" obliku:

- tzv. "skripta" iz Numeričke matematike (ili analize).

Skraćena verzija skripte — 1. dio (prvih 7 tjedana):

`http://web.math.hr/~singer/num_mat/num_mat1.pdf`

Skraćena verzija skripte — 2. dio (drugih 6 tjedana):

`http://web.math.hr/~singer/num_mat/num_mat2.pdf`

Da se ne uplašite veličine: i tu ima "previše" materijala.
Jednom će se (možda) dovesti u red.

Literatura (3)

Ako nekog zanima, originalna "velika" skripta je:

- Z. Drmač i ostali,
  Numerička analiza (skripta),
  PMF–MO, 2003.

Izravni "link" na "veliku" skriptu je

`http://web.math.hr/~singer/num_mat/num_anal.pdf`

Literatura (4)

Dodatna literatura:

- K. E. Atkinson,
  An Introduction to Numerical Analysis (second edition),
  John Wiley and Sons, 1989.

i hrpa malo starijih knjiga iz numeričke analize (matematike).

Možete potražiti i knjigu:

- Wolfgang Dahmen i ostali,
  Numerik für Ingenieure und Naturwissenschaftler,

Pripadni nastavni materijali su na stranici

`http://www.igpm.rwth-aachen.de/node/161/`

(koristite Firefox!)

Korisni linkovi

Službena web stranica kolegija je:

http://web.math.hr/nastava/unm/

"User's guide for everything" — Wikipedia:

http://en.wikipedia.org/

Korisni linkovi — nastavak

Na molbu Sanje Singer i Vedrana Novakovića, za goste je otvorena i web stranica kolegija Matematika 3 i 4 na FSB-u.

Tamo možete naći dodatne materijale za neke dijelove NM,

- posebno — vježbe i riješene zadatke.

Predavanja su "malo nježnija" od naših. Početna stranica je

`http://e-ucenje.fsb.hr/`

Zatim potražite "Katedra za matematiku" i onda:

- odete (kliknete) na kolegije Matematika 3 i 4,
- kliknete na gumb "Prijava kao gost",
- na stranici potražite blok 3 "Numerička matematika".

Iskoristite! Naravno, smijete pogledati i ostalo!

Demonstratori

Kolegij "Numerička matematika" ima demonstratore:

- Ervin Duraković i Marin Mišur.

Kroz neko vrijeme, kad se raspored ustabili,

- za upute za dogovor i termine,
- pogledajte oglase na oglasnoj ploči ili na webu kolegija.

Napomena uz kolokvije

Kolokviji iz prošlih godina "vise" na webu kolegija, na adresi

`http://web.math.hr/nastava/unm/kolokviji.php`

Pogledajte ih unaprijed, isplati se!

Napomena: Očekujte da će

- "teorija" nositi još više bodova.

Nekoliko dobrih razloga za to:

- Relativno efikasna zamjena za obavezni "usmeni" ispit (zamislite da ga ima . . .).
- Smanjuje se negativni efekt "glupih" grešaka u računanju (kuckanje po kalkusu),
- a stimulira razumijevanje teorije s predavanja.

Napomena uz vježbe i predavanja

Na kraju, zaboravite na "famu" da se

- NM polaže isključivo vježbama,
- pa na predavanja ne treba ni dolaziti.

Da bi to "prošlo", na kolokvijima morate

- izračunati sve što treba — i to bez grešaka.

Možda je lakše znati ponešto "teorije".

Usput, predavanja sadrže i hrpu riješenih zadataka.
Hm, . . . znam što sad slijedi:

- predavanja su na webu — dodatni razlog da na njih ne treba dolaziti!

Kako hoćete . . . neću popisivati "za bodove"!

Materijali na webu i "živa" nastava

Međutim, najkorisnija stvar na predavanjima je

- ono što onako "usput" ispričam,
- a ne piše na folijama (slajdovima).

Naravno, i to da me se može prekinuti i ponešto pitati!

Materijali na webu imaju sasvim drugu svrhu.

- Ne trebate bjesomučno pisati sve što kažem,
- najveći dio već piše!

Savjet = "uputstvo za uporabu" materijala:

- prije predavanja, pogledajte i isprintajte ih (4 ili 6 slajdova po stranici, kako vam paše),
- a dodatne bilješke pišite na tim papirima.

Programski paketi, biblioteke i sl.

A programska podrška? Ima svega:

- Mathematica, Matlab, BLAS, LAPACK, ...

Moderni software "zna" svašta

- računanje — numeričko i simboličko, vizualizacija, itd.

Međutim, namjerno nisam spominjao! Da se razumijemo,

- dozvoljeno je koristiti, ako znate, ali ...

Numerička matematika nije mjesto za "kurs" iz korištenja raznih programskih paketa, biblioteka i sl.

- Prvo treba naučiti matematiku
- i vidjeti ponešto primjera (nije bitno kako su nastali).

Onda ste "zreli" za dalje.

Programski paketi, biblioteke i sl. — nastavak

Ako vam numerika ikad zatreba u životu,

- na vama će biti odgovornost za primjenu stvari.

Morate prvo znati

- što radite, i što se može dogoditi s rezultatima,

pa tek onda kako to realizirati

- koji paket/biblioteku koristiti, koju metodu/rutinu koristiti (obično ih je nekoliko, za istu ili sličnu stvar), itd.

Čuvajte se "crnih kutija" koje "znaju sve".

- Nekritička primjena bilo čega — i može biti BUUUUM!

Rijetko ćete baš pisati neki numerički kôd. Ali, da znate,

- to je posao za dobro školovane matematičare!

Ima li pitanja?

Slušam . . .

Numerička matematika

Problemi numeričke matematike

U matematici postoji niz problema koje

- ne znamo ili ne možemo egzaktno riješiti,

tj. prisiljeni smo tražiti približno rješenje.

Neki klasični "zadaci" u numeričkom računanju su:

- rješavanje sustava linearnih i nelinearnih jednadžbi,
- računanje integrala,
- računanje aproksimacije neke zadane funkcije (zamjena podataka nekom funkcijom),
- minimizacija (maksimizacija) zadane funkcije, uz eventualna ograničenja (obično, u domeni),
- rješavanje diferencijalnih i integralnih jednadžbi ...

Problemi numeričke matematike (nastavak)

Neke probleme čak znamo egzaktno riješiti (bar u principu),

- poput sustava linearnih jednadžbi (ponoviti LA1),

no to predugo traje, pa koristimo računala.

Međutim, tada imamo dodatni problem, jer

- računala ne računaju egzaktno, već približno!

Oprez, tada ni osnovne aritmetičke operacije nisu egzaktne.

Dakle, ključni pojam u numerici je

- približna vrijednost, odnosno, greška.

Ciljevi numeričke matematike

U skladu s tim, osnovni zadatak numeričke matematike je naći (dati) odgovore na sljedeća pitanja:

- kako riješiti neki problem — metoda,
- koliko je "dobro" izračunato rješenje — točnost, ocjena greške.

Malo preciznije, za svaku od navedenih klasa problema, treba proučiti sljedeće "teme" — potprobleme:

1. Uvjetovanost problema — osjetljivost problema na greške, prvenstveno u početnim podacima (tzv. teorija pertubacije ili smetnje — vezana uz sam problem).
2. Konstrukcija standardnih numeričkih metoda za rješavanje danog problema.

Ciljevi numeričke matematike (nastavak)

Kad jednom "stignemo" do numeričkih metoda, treba još proučiti sljedeće "teme" — potprobleme:

3. Stabilnost numeričkih metoda — njihova osjetljivost na "smetnje" problema.

4. Efikasnost pojedine numeričke metode — orijentirano prema implementaciji na računalu:
   - broj računskih operacija i potreban memorijski prostor za rješavanje problema (= Složenost).

5. Točnost numeričkih metoda, u smislu neke "garancije" točnosti izračunatog rješenja.

Ilustracija ovih "potproblema" na primjerima — malo kasnije.

Greške

Greške

Pri numeričkom rješavanju nekog problema javljaju se različiti tipovi grešaka:

- greške modela — svođenje realnog problema na neki "matematički" problem,
- greške u ulaznim podacima (mjerenja i sl.),
- greške numeričkih metoda za rješavanje "matematičkog" problema,
- greške "približnog" računanja — obično su to
  - greške zaokruživanja u aritmetici računala.

Greške modela su "izvan" dosega numeričke matematike.

- Spadaju u fiziku, kemiju, biologiju, tehniku, ekonomiju, ...

Mjere za grešku

Oznake:

- prava vrijednost — x,
- izračunata ili približna vrijednost — $\hat{x}$.

Standardni naziv: $\hat{x}$ je aproksimacija za x.

Trenutno, nije bitno odakle (iz kojeg skupa) su x i $\hat{x}$.

- Zamislite da su to "obični" realni brojevi — $x, \hat{x} \in \mathbb{R}$.

Mjere za grešku (nastavak)

Apsolutna greška:

- mjeri udaljenost izračunate vrijednosti $\hat{x}$ obzirom na pravu vrijednost x.

Ako imamo vektorski prostor i normu, onda je

- udaljenost = norma razlike.

Dakle, apsolutna greška je definirana ovako:

$$E_{\text{abs}}(x, \hat{x}) := |\hat{x} - x|.$$

Često se koristi i oznaka $\Delta x = \hat{x} - x$ (na pr. u analizi), pa je $E_{\text{abs}}(x, \hat{x}) = |\Delta x|$.

Mjere za grešku (nastavak)

Primjer. Dojam o "veličini" greške:

- ako smo umjesto 1 izračunali 2, to nam se čini lošije nego
- ako smo umjesto 100 izračunali 101.

Relativna greška:

- mjeri relativnu točnost aproksimacije $\hat{x}$ obzirom na veličinu broja x,
- na pr. koliko se vodećih znamenki brojeva x i $\hat{x}$ podudara.

Relativna greška definirana je za $x \neq 0$,

$$E_{\text{rel}}(x, \hat{x}) := \frac{|\hat{x} - x|}{|x|}.$$

Često se koristi i oznaka δ_x. Katkad se u nazivniku javlja $|\hat{x}|$.

Mjere za grešku (nastavak)

Ideja relativne greške: ako $\hat{x}$ napišemo kao $\hat{x} = x(1+\rho)$, onda je njegova relativna greška

$$E_{\text{rel}}(x, \hat{x}) := |\rho|.$$

Dakle, relativna greška mjeri

- koliko se faktor $(1+\rho)$ apsolutno razlikuje od 1.

Sad možemo detaljnije opisati one četiri vrste grešaka:

- greške modela,
- greške u ulaznim podacima (mjerenjima),
- greške metoda za rješavanje modela,
- greške aritmetike računala.

Greške modela

Greške modela mogu nastati:

- zbog zanemarivanja utjecaja nekih sila,
  - na primjer, zanemarivanje utjecaja otpora zraka ili trenja (v. primjer),
- zbog zamjene kompliciranog modela jednostavnijim,
  - na primjer, sustavi nelinearnih običnih ili parcijalnih diferencijalnih jednadžbi se linearizirаju, da bi se dobilo barem približno rješenje,
- zbog upotrebe modela u graničnim slučajevima,
  - na primjer, kod matematičkog njihala se $\sin x$ aproksimira s x, što vrijedi samo za male kutove.

Modelni primjer — Problem gađanja

Primjer. Imamo top (ili haubicu) u nekoj točki — recimo, ishodištu.

- Treba pogoditi cilj koji se nalazi u nekoj drugoj točki.

Najjednostavniji model za ovaj problem je poznati kosi hitac. Projektil ispaljujemo prema cilju,

- nekom početnom brzinom v_0 (vektor),
- pod nekim kutem α, obzirom na horizontalnu ravninu.

Slikica!

Cijela stvar se odvija pod utjecajem gravitacije (prema dolje). Ako zanemarimo otpor zraka, dobijemo "obični" kosi hitac.

Modelni primjer — Jednadžba

Osnovna jednadžba je

$$F = ma,$$

gdje je m masa projektila (neće nam trebati na početku), a

- a je akceleracija — vektor u okomitoj (x, y)-ravnini,
- F je sila gravitacije, prema dolje, tj. $F_x = 0$ i $F_y = -mg$.

Gornja jednadžba je diferencijalna jednadžba drugog reda u vremenu. Ako je $(x(t), y(t))$ položaj projektila u danom trenutku, jednadžba ima oblik po komponentama:

$$m\,\frac{d^2x}{dt^2} = F_x, \quad m\,\frac{d^2y}{dt^2} = F_y.$$

Akceleracija je druga derivacija položaja.

Modelni primjer — Rješenje jednadžbe

Neka je projektil ispaljen u trenutku $t_0 = 0$.

Nakon integracije, za brzinu v = prva derivacija položaja, imamo jednadžbu

$$mv = F \cdot t + mv_0,$$

ili, po komponentama (masa se skrati)

$$v_x = \frac{dx}{dt} = v_0 \cos\alpha, \quad v_y = \frac{dy}{dt} = v_0 \sin\alpha - gt.$$

Još jednom integriramo (početni položaj je $x_0 = 0$, $y_0 = 0$). Za položaj projektila u trenutku t dobivamo:

$$x(t) = v_0 t \cos\alpha, \quad y(t) = v_0 t \sin\alpha - \frac{1}{2} g t^2.$$

Reklo bi se — znamo sve!

Modelni primjer — Još neke relacije

Jednadžba "putanje" projektila u (x, y)-ravnini je

$$y = x \operatorname{tg} \alpha - \frac{g}{2v_0^2 \cos^2 \alpha} x^2.$$

To je parabola, s otvorom nadolje, koja prolazi kroz ishodište.

Najveća visina projektila je

$$y_{\max} = \frac{(v_0 \sin \alpha)^2}{2g},$$

a maksimalni domet na horizontalnoj x-osi je

$$x_{\max} = \frac{v_0^2 \sin 2\alpha}{g}.$$

Modelni primjer — Stvarnost

Nažalost, s ovim modelom nećemo ništa pogoditi.

- Fali otpor zraka, tlak pada s visinom, vjetrovi i sl.

Praksa:

- Koeficijent za otpor ovisi o obliku projektilu — mjeri se.
- Izračunate tablice se eksperimentalno "upucavaju" i korigiraju.
- Primjena u praksi ide obratno — znam daljinu, tražim kut.

Greške modela (nastavak)

Primjer. Među prvim primjenama jednog od prvih brzih paralelnih računala na svijetu (ASCI Blue Pacific) bilo je

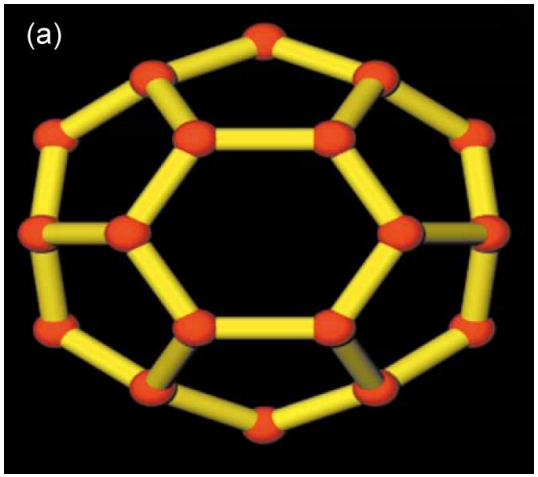
- određivanje trodimenzionalne strukture i elektronskog stanja ugljik-36 fulerena.

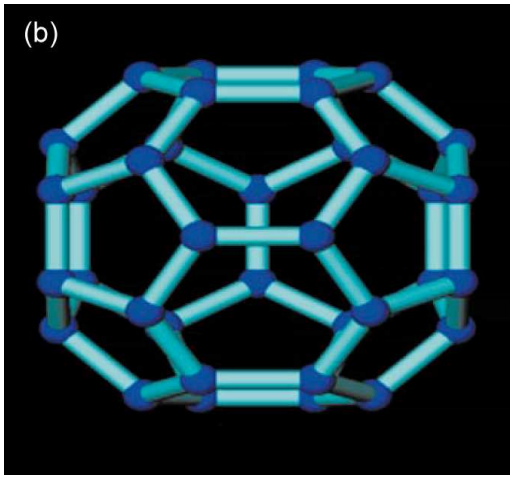
Primjena spoja je višestruka:

- supravodljivost na visokim temperaturama,
- precizno doziranje lijekova u stanice raka.

Greške modela (nastavak)

Prijašnja istraživanja kvantnih kemičara dala su dvije moguće strukture tog spoja.





Te dvije strukture imaju različita kemijska svojstva.

Greške modela (nastavak)

Stanje stvari:

- eksperimentalna mjerenja pokazivala su da je struktura (a) stabilnija,
- teoretičari su tvrdili da je stabilnija struktura (b).

Prijašnja računanja,

- zbog pojednostavljivanja i interpolacije,

kao odgovor davala su prednost "teoretskoj" strukturi.

Definitivan odgovor,

- proveden računanjem bez pojednostavljivanja,

pokazao je da je struktura (a) stabilnija.

Greške u ulaznim podacima

Greške u ulaznim podacima javljaju se zbog

- nemogućnosti ili besmislenosti točnog mjerenja (Heisenbergove relacije neodređenosti).
- Primjer, tjelesna temperatura se obično mjeri na desetinku stupnja Celziusa točno. Pacijent je podjednako loše ako ima tjelesnu temperaturu $39.5°$ ili $39.513462°$.

Bitno praktično pitanje:

- Mogu li male greške u ulaznim podacima bitno povećati grešku rezultata?

Nažalost MOGU!

- Takvi problemi zovu se loše uvjetovani problemi.

Greške u ulaznim podacima (nastavak)

Primjer.
Zadana su dva sustava linearnih jednadžbi — recimo, umjesto ispravnih (prvih) koeficijanata, izmjerili smo druge:

$$\begin{aligned} 2x &+ 6y &&= 8 \\ 2x &+ 6.0001y &&= 8.0001, \end{aligned}$$

i

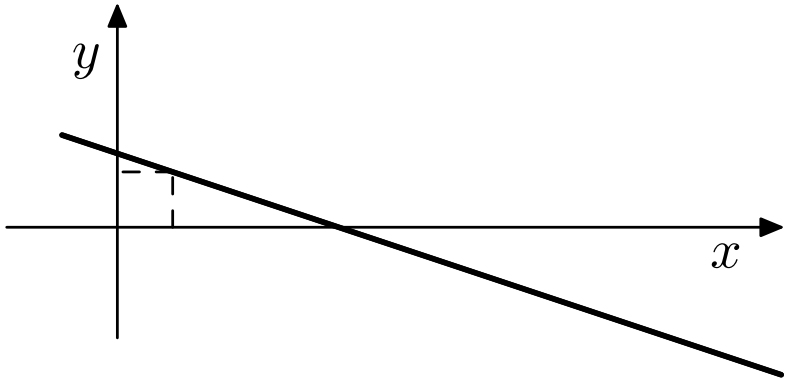
$$\begin{aligned} 2x &+ 6y &&= 8 \\ 2x &+ 5.99999y &&= 8.00002. \end{aligned}$$

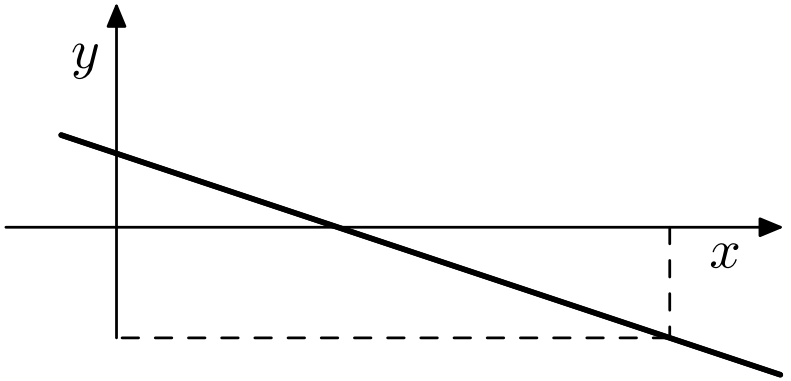
Perturbacije koeficijenata: reda veličine 10^{-4}. Je li se rezultat također promijenio za red veličine 10^{-4}?

Greške u ulaznim podacima (nastavak)

- Rješenje prvog problema: $x = 1$, $y = 1$.
- Rješenje drugog problema: $x = 10$, $y = -2$.

Grafovi presjecišta dva pravca za prvi i drugi sustav:





Greške metoda za rješavanje problema

Najčešće nastaju kad se nešto beskonačno zamjenjuje nečim konačnim. Razlikujemo dvije kategorije:

- greške diskretizacije koje nastaju zamjenom kontinuuma konačnim diskretnim skupom točaka, ili "beskonačno" malu veličinu h ili $\varepsilon \to 0$ zamijenjujemo nekim "konačno" malim brojem;
- greške odbacivanja koje nastaju "rezanjem" beskonačnog niza ili reda na konačni niz ili sumu, tj. odbacujemo ostatak niza ili reda.

Greške metoda za rješavanje problema (nast.)

Tipični primjeri greške diskretizacije:

- aproksimacija funkcije f na $[a, b]$, vrijednostima te funkcije na konačnom skupu točaka (tzv. mreži) $\{x_1, \ldots, x_n\} \subset [a, b]$,
- aproksimacija derivacije funkcije f u nekoj točki x. Po definiciji je

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

a za približnu vrijednost uzmemo dovoljno mali $h \neq 0$ i

$$f'(x) \approx \frac{\Delta f}{\Delta x} = \frac{f(x+h) - f(x)}{h}.$$

Greške metoda za rješavanje problema (nast.)

Tipični primjeri greške odbacivanja:

- zaustavljanje iterativnih procesa nakon dovoljno velikog broja n iteracija (recimo kod računanja nultočaka funkcije);
- zamjena beskonačne sume konačnom kad je greška dovoljno mala (recimo kod sumiranja Taylorovih redova).

Taylorov red, Taylorov polinom, ...

Za dovoljno glatku funkciju f, Taylorov red oko točke x_0

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

možemo aproksimirati Taylorovim polinomom p

$$f(x) = p(x) + R_{n+1}(x), \qquad p(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k,$$

pri čemu je $R_{n+1}(x) = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$ greška odbacivanja, a ξ neki broj između x_0 i x. $R_{n+1}(x)$ obično ocjenjujemo po apsolutnoj vrijednosti.

Taylorov red, Taylorov polinom, ... (nastavak)

Primjer.

- Funkcije e^x i $\sin x$ imaju Taylorove redove oko točke 0 koji konvergiraju za proizvoljan $x \in \mathbb{R}$.
- Zbrajanjem dovoljno mnogo članova tih redova, možemo, barem u principu, dobro aproksimirati vrijednosti funkcija e^x i $\sin x$.
- Traženi Taylorovi polinomi s istim brojem članova (ali ne istog stupnja) su

$$e^x \approx \sum_{k=0}^{n} \frac{x^k}{k!}, \qquad \sin x \approx \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!}.$$

Taylorov red, Taylorov polinom, ... (nastavak)

Za grešku odbacivanja trebaju nam derivacije:

$$(e^x)^{(n)} = e^x, \qquad (\sin x)^{(n)} = \sin\left(x + \frac{n\pi}{2}\right),$$

pa su pripadne greške odbacivanja

$$R_{n+1}(x) = \frac{e^\xi x^{n+1}}{(n+1)!}, \qquad R_{2n+3}(x) = \frac{\sin(\xi + \frac{2n+3}{2}\pi)x^{2n+3}}{(2n+3)!},$$

Pretpostavimo sada da je $x > 0$. Iz $\xi \leq x$ dobivamo

$$|R_{n+1}(x)| \leq \frac{e^x x^{n+1}}{(n+1)!}, \qquad |R_{2n+3}(x)| \leq \frac{x^{2n+3}}{(2n+3)!}.$$

Taylorov red, Taylorov polinom, ... (nastavak)

Zbrojimo li članove reda sve dok apsolutna vrijednost prvog odbačenog člana ne padne ispod zadane točnosti $\varepsilon > 0$, napravili smo grešku odbacivanja manju ili jednaku

$$\begin{cases} e^x \varepsilon, & \text{za } e^x, \\ \varepsilon, & \text{za } \sin x. \end{cases}$$

U prvom slučaju očekujemo

- malu relativnu grešku,

a u drugom slučaju očekujemo

- malu apsolutnu grešku.

Provjerimo to eksperimentalno — u aritmetici računala!

Prikaz brojeva u računalu i greške zaokruživanja

Tipovi brojeva u računalu

U računalu postoje dva bitno različita tipa brojeva:

- cijeli brojevi
- realni brojevi.

Oba skupa su konačni podskupovi odgovarajućih skupova $\mathbb{Z}$ i $\mathbb{R}$ u matematici.

Kao baza za prikaz oba tipa koristi se baza 2.

Cijeli brojevi u računalu

Cijeli brojevi bez predznaka — sažetak

Ako imamo n bitova za prikaz brojeva, onda je skup svih prikazivih cijelih brojeva bez predznaka jednak

$$\mathbb{Z}_{2^n} = \{\, 0,\ 1,\ 2,\ \ldots\ ,\ 2^n - 2,\ 2^n - 1 \,\}.$$

Prikaz broja $B \in \mathbb{Z}_{2^n}$ dobiva se iz "proširenog" zapisa tog broja u bazi 2, s točno n binarnih znamenki.

Aritmetika cijelih brojeva bez predznaka je modularna aritmetika u prstenu $(\mathbb{Z}_{2^n}, \oplus_{2^n}, \odot_{2^n})$:

- operacije $+$, $-$ i $\cdot$ daju cjelobrojni rezultat modulo 2^n,
- operacije cjelobrojnog dijeljenja s ostatkom div i mod daju iste rezultate kao da dijelimo u $\mathbb{Z}$ (ili $\mathbb{N}_0$).

Cijeli brojevi s predznakom — sažetak

Ako imamo n bitova za prikaz brojeva, onda je skup svih prikazivih cijelih brojeva bez predznaka jednak

$$\mathbb{Z}_{2^n}^{-} = \{ -2^{n-1},\ -2^{n-1}+1,\ \ldots\ ,\ -2,\ -1,\ 0,\ 1,\ \ldots\ ,\ 2^{n-1}-2,\ 2^{n-1}-1 \}.$$

Za prikaz broja $B \in \mathbb{Z}_{2^n}^{-}$ vrijedi:

- nenegativni brojevi $B = 0, \ldots, 2^{n-1}-1$ imaju isti prikaz kao i bez predznaka,
- negativni brojevi $B = -1, \ldots, -2^{n-1}$ imaju isti prikaz kao i brojevi $2^n + B$ bez predznaka.

Cijeli brojevi s predznakom — sažetak

Osim toga, prikaz suprotnog broja $-B$ dobivamo tako da

- komplementiramo prikaz samog broja i dodamo 1 modulo 2^n.

Aritmetika cijelih brojeva s predznakom je modularna aritmetika modulo 2^n na sustavu ostataka $\mathbb{Z}_{2^n}^-$.

- To vrijedi za operacije $+$, $-$ i $\cdot$.

Operacije cjelobrojnog dijeljenja s ostatkom div i mod daju iste rezultate kao da dijelimo u $\mathbb{Z}$,

- ali treba provjeriti kako se dobiva proširenje ovih operacija s $\mathbb{N}_0 \times \mathbb{N}$ na $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$.

Dijeljenje cijelih brojeva s predznakom

Eksperiment:

- test–program `divmod.c` (pokaži!),
- Intelov C++ compiler (verzija 9.1.032), na IA–32.

Rezultati $q = a \operatorname{div} b$ i $r = a \bmod b$ za $a = \pm 5$, $b = \pm 3$:

a	b	q	r
5	3	1	2
-5	3	-1	-2
5	-3	-1	2
-5	-3	1	-2

Operacije div i mod interpretiramo na $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$.

Veza cjelobrojnog i običnog dijeljenja

Ključ za interpretaciju:

- kvocijent se uvijek "zaokružuje" prema nuli,

$$q = \operatorname{sign}\left(\frac{a}{b}\right) \cdot \left\lfloor \left| \frac{a}{b} \right| \right\rfloor,$$

- ostatak ima isti predznak kao i a.

$$r = \operatorname{sign}(a) \cdot (|a| \bmod |b|).$$

Za ostatak r ovdje vrijedi:

- $0 \leq r < |b|$, za $a \geq 0$,
- $-|b| < r \leq 0$, za $a < 0$.

Veza cjelobrojnog i običnog dijeljenja

Razlog: standardno ograničenje na ostatak $0 \leq r < b$, tj. $r \in \mathbb{Z}_b$, odgovara cijelim brojevima bez predznaka.

Međutim, kod brojeva s predznakom imamo i negativne brojeve, pa (možda) ima smisla dozvoliti da i ostaci budu negativni (u nekim slučajevima).

Prednosti ovakve "definicije" operacija div i mod na skupu $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$:

- bez obzira na predznake od a i b, dobivamo
- iste apsolutne vrijednosti kvocijenta q i ostatka r,

tj. samo predznaci od q i r ovise o predznacima od a i b.

Ovo je i najčešća realizacija cjelobrojnog dijeljenja u praksi. Noviji standard za C (tzv. C99) propisuje ovakvo ponašanje.

Aritmetika cijelih brojeva: klasične greške

Cijeli brojevi — klasične greške

Primjer. Računanje $n!$ u cjelobrojnoj aritmetici.

Za prirodni broj $n \in \mathbb{N}$, funkciju faktorijela definiramo na sljedeći način:

$$1! = 1,$$
$$n! = n \cdot (n-1)!, \quad n \geq 2.$$

Napišimo program koji računa broj $50!$ u cjelobrojnoj aritmetici (tip `int`).

Cijeli brojevi — klasične greške (nastavak)

```
#include <stdio.h>

int main(void) {
    int i, f50 = 1;  /* n = 32 za int */

    for (i = 2; i <= 50; ++i)
        f50 *= i;

    printf(" f50 = %d\n", f50);  /* f50 = 0 */

    return 0;
}
```

Izlaz programa je: `f50 = 0`. Zašto?

Cijeli brojevi — klasične greške (nastavak)

Za početak, $50!$ je ogroman broj. Točna vrijednost je

$$50! = 30414\ 09320\ 17133\ 78043\ 61260\ 81660$$
$$64768\ 84437\ 76415\ 68960\ 51200\ 00000\ 00000,$$

i ima 65 dekadskih znamenki. Dakle, sigurno nije prikaziv u cjelobrojnoj aritmetici.

Granice za tip `int` ($n = 32$ bita) iz zaglavlja `limits.h` su

$$\texttt{INT\_MAX} = 2147483647, \quad \texttt{INT\_MIN} = (-\texttt{INT\_MAX} - 1).$$

Objasnimo još zašto je `f50 = 0` u našem programu. Cjelobrojna aritmetika u kojoj računamo je modularna aritmetika — modulo 2^{32}.

Cijeli brojevi — klasične greške (nastavak)

To znači da naš program kaže da je

$$50! = 0 \bmod 2^{32},$$

ili da 2^{32} dijeli $50!$.

Zadatak. Nađite najveću potenciju broja 2 koja dijeli $50!$, ili, općenito $n!$. Rješenje:

$$m = \left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{2^2} \right\rfloor + \left\lfloor \frac{n}{2^3} \right\rfloor + \left\lfloor \frac{n}{2^4} \right\rfloor + \left\lfloor \frac{n}{2^5} \right\rfloor + \cdots .$$

Za $n = 50$ imamo $m = 25 + 12 + 6 + 3 + 1 = 47$.

Zadatak. Nađite najmanji broj n za koji je $n! = 0$ u cjelobrojnoj aritmetici s ℓ bitova za prikaz brojeva.

Tablica $n!$ za $\ell = 16$ i $\ell = 32$ bita

Usporedimo rezultate za $n!$ dobivene u cjelobrojnoj aritmetici s $\ell = 16$ i $\ell = 32$ bita s (ispravnim) rezultatom dobivenim u realnoj aritmetici.

$n!$	$\ell = 16$	$\ell = 32$	realna aritmetika
$7!$	5040	5040	5040
$8!$	-25216	40320	40320
$9!$	-30336	362880	362880
$10!$	24320	3628800	3628800
$11!$	5376	39916800	39916800
$12!$	-1024	479001600	479001600

Tablica $n!$ za $\ell = 16$ i $\ell = 32$ bita (nastavak)

$n!$	$\ell = 16$	$\ell = 32$	realna aritmetika
$13!$	-13312	1932053504	6227020800
$14!$	10240	1278945280	87178291200
$15!$	22528	2004310016	1307674368000
$16!$	-32768	2004189184	20922789888000
$17!$	-32768	-288522240	355687428096000
$18!$	0	-898433024	6402373705728000
$19!$	0	109641728	121645100408832000
$20!$	0	-2102132736	2432902008176640000

Prikaz "realnih" brojeva u računalu — IEEE standard

Prikaz realnih brojeva

U računalu se binarni zapis realnog broja pohranjuje u znanstvenom formatu:

$$\text{broj} = \text{predznak} \cdot \text{mantisa} \cdot 2^{\text{eksponent}}.$$

Mantisa se uobičajeno (postoje iznimke!) pohranjuje u tzv. normaliziranom obliku, tj.

$$1 \leq \text{mantisa} < (10)_2.$$

I za pohranu mantise i za pohranu eksponenta rezervirano je konačno mnogo binarnih znamenki. Posljedice:

- prikaziv je samo neki raspon realnih brojeva,
- niti svi brojevi unutar prikazivog raspona nisu prikazivi (mantisa predugačka) $\Longrightarrow$ zaokruživanje.

Prikaz realnih brojeva (nastavak)

Primjer: Znanstveni prikaz binarnih brojeva:

$$1010.11 = 1.01011 \cdot 2^3$$

$$0.0001011011 = 1.01011 \cdot 2^{-4}$$

Primijetite da se vodeća jedinica u normaliziranom obliku ne mora pamtiti (ako je broj $\neq 0$).

- Taj bit se može upotrijebiti za pamćenje dodatne znamenke mantise.

Tada se vodeća jedinica zove skriveni bit (engl. hidden bit) — jer se ne pamti.

Ipak ovo je samo pojednostavljeni prikaz realnih brojeva.

Stvarni prikaz realnih brojeva

Najznačajnija promjena obzirom na pojednostavljeni prikaz:

- eksponent se prikazuje u "zamaskiranoj" ili "pomaknutoj" formi (engl. "biased form").

To znači da se stvarnom eksponentu

- dodaje konstanta — takva da je "pomaknuti" eksponent uvijek pozitivan.

Ta konstanta ovisi o broju bitova za eksponent i bira se tako da je prikaziva

- recipročna vrijednost najmanjeg pozitivnog normaliziranog broja.

Takav "pomaknuti" eksponent naziva se karakteristika, a normaliziranu mantisu neki zovu i signifikand.

Oznake

Oznake:

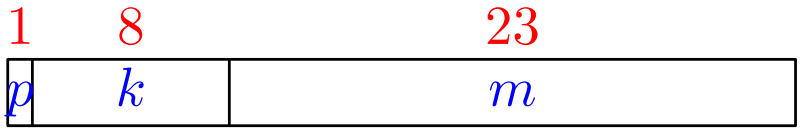
- Crveno — duljina odgovarajućeg polja (u bitovima), bitove brojimo od 0 zdesna nalijevo (kao i obično),
- p — predznak: 0 za pozitivan broj, 1 za negativan broj,
- k — karakteristika,
- m — mantisa (signifikand).
- Najznačajniji bit u odgovarajućem polju je najljeviji.
- Najmanje značajan bit u odgovarajućem polju je najdesniji.

Stvarni prikaz tipa single

"Najkraći" realni tip je tzv. realni broj jednostruke točnosti — u C-u poznat kao `float`.

On ima sljedeća svojstva:

- duljina: 4 byte-a (32 bita), podijeljen u tri polja.



- u mantisi se ne pamti vodeća jedinica ako je broj normaliziran,
- stvarni eksponent broja e, $e \in \{-126, \ldots, 127\}$,
- karakteristika $k = e + 127$, tako da je $k \in \{1, \ldots, 254\}$,
- karakteristike $k = 0$ i $k = 255$ koriste se za "posebna stanja".

Stvarni prikaz tipa single (nastavak)

Primjer: Broj $(10.25)_{10}$ prikažite kao broj u jednostrukoj točnosti.

$$(10.25)_{10} = \left(10 + \frac{1}{4}\right)_{10} = (10 + 2^{-2})_{10}$$
$$= (1010.01)_2 = 1.01001 \cdot 2^3.$$

Prema tome je:

$$p = 0$$
$$k = e + 127 = (130)_{10} = (2^7 + 2^1)_{10} = 1000\ 0010$$
$$m = 0100\ 1000\ 0000\ 0000\ 0000\ 000$$

Prikazi nule: $k = 0$, $m = 0$

Realni broj nula ima dva prikaza:

- mantisa i karakteristika su joj nula,

a predznak može biti

- 0 — "pozitivna nula", ili
- 1 — "negativna nula".

Ta dva prikaza nule su:

$$+0 = 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$$

$$-0 = 1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000$$

Smatra se da su vrijednosti ta dva broja jednake (kad se uspoređuju).

Denormalizirani brojevi: $k = 0$, $m \neq 0$

Ako je $k = 0$, a postoji barem jedan znak mantise koji nije nula, onda se kao eksponent uzima -126. Mantisa takvog broja nije normalizirana i počinje s $0.m$.

Takvi brojevi zovu se denormalizirani brojevi.

Primjer: Kako izgleda prikaz realnog broja

$$0.000\ 0000\ 0000\ 0000\ 0000\ 1011 \cdot 2^{-126}?$$

Rješenje:

$$p = 0$$

$$k = 0000\ 0000$$

$$m = 000\ 0000\ 0000\ 0000\ 0000\ 1011$$

Plus i minus beskonačno: $k = 255$, $m = 0$

Ako je $k = 255$, a mantisa jednaka 0, onda

- $p = 0$ — prikaz $+\infty$, skraćena oznaka `+Inf`,
- $p = 1$ — prikaz $-\infty$, skraćena oznaka `-Inf`.

Primjer: Prikaz broja $+\infty$ $(-\infty)$ je

$$p = 0 \qquad (p = 1)$$
$$k = 1111\ 1111$$
$$m = 000\ 0000\ 0000\ 0000\ 0000\ 0000$$

Nije broj: $k = 255$, $m \neq 0$

Ako je $k = 255$ i postoji bar jedan bit mantise različit od nule, onda je to signal da se radi o pogrešci (recimo — dijeljenje s nulom, vađenje drugog korijena iz negativnog broja i sl.)

Tada se takva pogreška kodira znakom za Not a Number ili, skraćeno, s `NaN`.

Primjer.

$$p = 0$$

$$k = 1111\ 1111$$

$$m = 000\ 0000\ 0000\ 0101\ 0000\ 0000$$

Greške zaokruživanja

Postoje realni brojevi koje ne možemo egzaktno spremiti u računalo, čak i kad su unutar prikazivog raspona brojeva. Takvi brojevi imaju predugačku mantisu.

Primjer: Realni broj (u binarnom zapisu)

$$a = 1.0001\ 0000\ 1000\ 0011\ 1001\ 0111$$

ima 25 znamenki mantise i ne može se egzaktno spremiti u realni broj jednostruke preciznosti `float` u C-u, koji ima $23 + 1$ znamenki za mantisu.

Procesor tada pronalazi dva najbliža prikaziva susjeda a_-, a_+, broju a, takva da vrijedi

$$a_- < a < a_+.$$

Greške zaokruživanja (nastavak)

U našem primjeru je:

$$a = 1.0001\ 0000\ 1000\ 0011\ 1001\ 0111$$

$$a_- = 1.0001\ 0000\ 1000\ 0011\ 1001\ 011$$

$$a_+ = 1.0001\ 0000\ 1000\ 0011\ 1001\ 100$$

Nakon toga, zaokružuje se rezultat. Zaokruživanje može biti:

- prema najbližem broju (standardno, engl. default, za IA-32 procesore) — ako su dva susjeda jednako udaljena od a, izabire parni od ta dva broja (zadnji bit je 0),
- prema dolje, tj. prema $-\infty$,
- prema gore, tj. prema ∞,
- prema nuli, tj. odbacivanjem "viška" znamenki.

Greške zaokruživanja (nastavak)

Standardno zaokruživanje u našem primjeru:

$$a = 1.0001\ 0000\ 1000\ 0011\ 1001\ 0111$$

$$a_- = 1.0001\ 0000\ 1000\ 0011\ 1001\ 011$$

$$a_+ = 1.0001\ 0000\ 1000\ 0011\ 1001\ 100$$

Ovdje su a_- i a_+ jednako udaljeni od a, pa je zaokruženi a jednak a_+, jer a_+ ima parni zadnji bit (jednak je 0).

Jedinična greška zaokruživanja

Ako je $x \in \mathbb{R}$ unutar raspona brojeva prikazivih u računalu, onda se, umjesto x, sprema zaokruženi prikazivi broj $fl(x)$.

Time smo napravili grešku zaokruživanja $\leq \frac{1}{2}$ "zadnjeg bita" mantise, i taj broj se zove

- jedinična greška zaokruživanja (engl. unit roundoff).

Standardna oznaka je u. Za `float` je

$$u = 2^{-24} \approx 5.96 \cdot 10^{-8}.$$

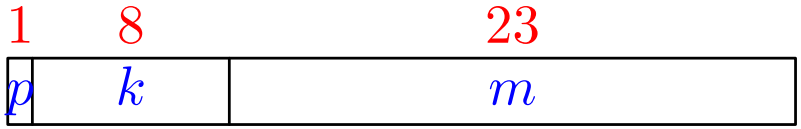
Vrijedi

$$fl(x) = (1 + \varepsilon)x, \quad |\varepsilon| \leq u,$$

gdje je ε relativna greška napravljena tim zaokruživanjem. Dakle, imamo vrlo malu relativnu grešku.

Prikaz brojeva jednostruke točnosti — sažetak

IEEE tip single = `float` u C-u:



Vrijednost broja je

$$v = \begin{cases} (-1)^p * 2^{(k-127)} * (1.m) & \text{ako je } 0 < k < 255, \\ (-1)^p * 2^{(-126)} * (0.m) & \text{ako je } k = 0 \text{ i } m \neq 0, \\ (-1)^p * 0 & \text{ako je } k = 0 \text{ i } m = 0, \\ (-1)^p * \texttt{Inf} & \text{ako je } k = 255 \text{ i } m = 0, \\ \texttt{NaN} & \text{ako je } k = 255 \text{ i } m \neq 0. \end{cases}$$

Raspon tipa float

Najveći prikazivi pozitivni broj je
`FLT_MAX` $\approx 3.402823466 \cdot 10^{38}$, s prikazom

$$p = 0$$

$$k = 1111\ 1110$$

$$m = 111\ 1111\ 1111\ 1111\ 1111\ 1111$$

Najmanji prikazivi normalizirani pozitivni broj je
`FLT_MIN` $\approx 1.175494351 \cdot 10^{-38}$, s prikazom

$$p = 0$$

$$k = 0000\ 0001$$

$$m = 000\ 0000\ 0000\ 0000\ 0000\ 0000$$

Raspon tipa float

Simboličke konstante `FLT_MAX`, `FLT_MIN` i još poneke vezane uz tip `float`, definirane su u datoteci zaglavlja `float.h` i mogu se koristiti u C programima.

Uočite:

- `1/FLT_MIN` je egzaktno prikaziv (nađite prikaz),
- `1/FLT_MAX` nije egzaktno prikaziv i zalazi u denormalizirane brojeve (tzv. "gradual underflow").

Najmanji prikazivi denormalizirani pozitivni broj je $2^{-126} \cdot 2^{-23} = 2^{-149} \approx 1.4013 \cdot 10^{-45}$, s prikazom

$$p = 0$$

$$k = 0000\ 0000$$

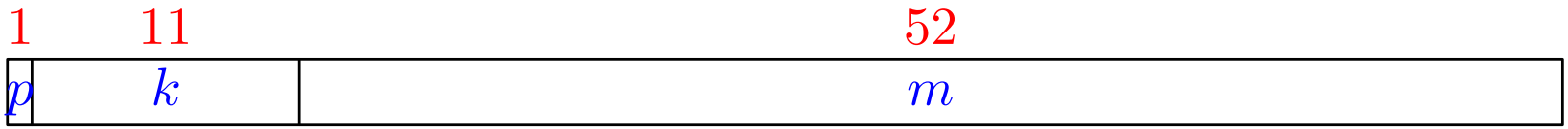
$$m = 000\ 0000\ 0000\ 0000\ 0000\ 0001$$

Stvarni prikaz tipa double

"Srednji" realni tip je tzv. realni broj dvostruke točnosti — u C-u poznat kao `double`.

On ima sljedeća svojstva:

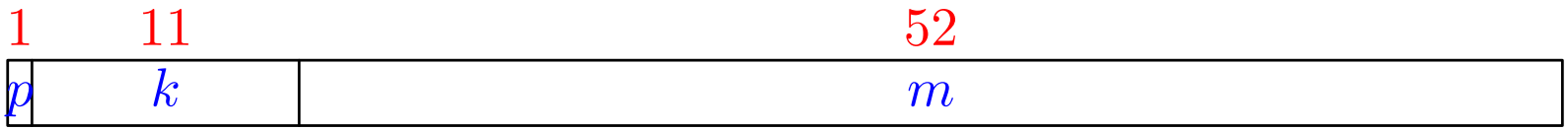
- Duljina: 8 byte-a (64 bita), podijeljen u tri polja.



- u mantisi se ne pamti vodeća jedinica ako je broj normaliziran,
- stvarni eksponent broja e, $e \in \{-1022, \ldots, 1023\}$,
- karakteristika $k = e + 1023$, tako da je $k \in \{1, \ldots, 2046\}$,
- karakteristike $k = 0$ i $k = 2047$ — "posebna stanja".

Prikaz brojeva dvostruke točnosti — sažetak

IEEE tip double = `double` u C-u:



Vrijednost broja je

$$v = \begin{cases} (-1)^p * 2^{(k-1023)} * (1.m) & \text{ako je } 0 < k < 2047, \\ (-1)^p * 2^{(-1022)} * (0.m) & \text{ako je } k = 0 \text{ i } m \neq 0, \\ (-1)^p * 0 & \text{ako je } k = 0 \text{ i } m = 0, \\ (-1)^p * \texttt{Inf} & \text{ako je } k = 2047 \text{ i } m = 0, \\ \texttt{NaN} & \text{ako je } k = 2047 \text{ i } m \neq 0. \end{cases}$$

Jedinična greška i raspon tipa double

Jedinična greška zaokruživanja za `double` je

$$u = 2^{-53} \approx 1.11 \cdot 10^{-16}.$$

Broj $1 + 2u$ je najmanji prikazivi broj strogo veći od 1. Postoji

$$\texttt{DBL\_EPSILON} = 2u \approx 2.2204460492503131 \cdot 10^{-16}.$$

Najveći prikazivi pozitivni broj je

$$\texttt{DBL\_MAX} \approx 1.7976931348623158 \cdot 10^{308}.$$

Najmanji prikazivi normalizirani pozitivni broj je

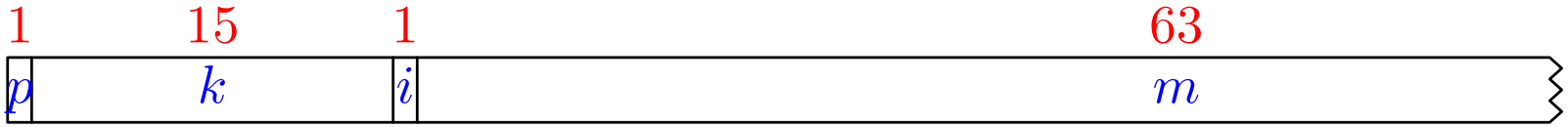
$$\texttt{DBL\_MIN} \approx 2.2250738585072014 \cdot 10^{-308}.$$

Tip extended

Stvarno računanje (na IA–32) se obično radi u "proširenoj" točnosti — u C-u možda dohvatljiv kao `long double`.

On ima sljedeća svojstva:

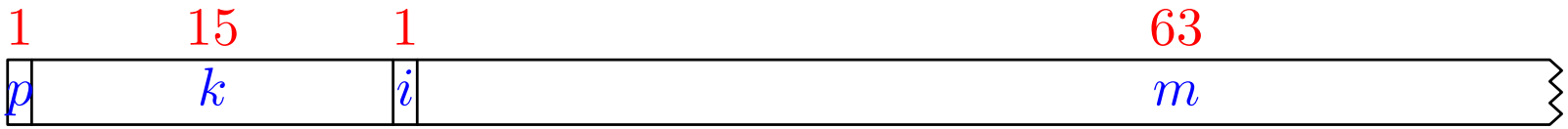
- Duljina: 10 byte-a (80 bita), podijeljen u četiri polja.



- u mantisi se pamti vodeći bit i mantise,
- stvarni eksponent broja e, $e \in \{-16382, \ldots, 16383\}$,
- karakteristika $k = e + 16383$, tako da je $k \in \{1, \ldots, 32766\}$,
- karakteristike $k = 0$ i $k = 32767$ — "posebna stanja".

Prikaz brojeva proširene točnosti — sažetak

IEEE tip extended:



Vrijednost broja je

$$v = \begin{cases} (-1)^p * 2^{(k-16383)} * (i.m) & \text{ako je } 0 \leq k < 32767, \\ (-1)^p * \texttt{Inf} & \text{ako je } k = 32767 \text{ i } m = 0, \\ \texttt{NaN} & \text{ako je } k = 32767 \text{ i } m \neq 0. \end{cases}$$

Uočite da prva mogućnost uključuje:

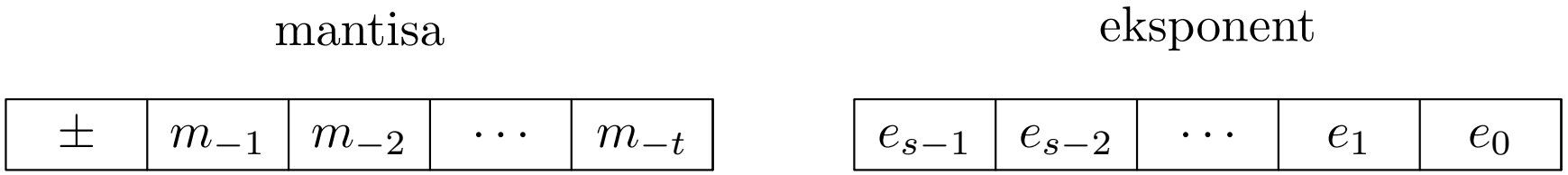
- $+0$, -0 i denormalizirane brojeve (za $k = 0$),

jer se pamti vodeći "cjelobrojni" bit i mantise.

Prikaz realnih brojeva
sažetak

Realni brojevi

Skup svih realnih brojeva prikazivih u računalu je omeđen, a parametriziramo ga duljinom mantise i eksponenta i označavamo s $\mathbb{R}(t, s)$.



Ne može se svaki realni broj egzaktno spremiti u računalo.

Ako je broj $x \in \mathbb{R}$ unutar prikazivog raspona i

$$x = \pm \left(\sum_{k=1}^{\infty} b_{-k} 2^{-k} \right) 2^e$$

i mantisa broja ima više od t znamenki, ...

Realni brojevi

. . . bit će spremljena aproksimacija tog broja $fl(x) \in \mathbb{R}(t, s)$ koja se može prikazati kao

$$fl(x) = \pm \left(\sum_{k=1}^{t} b_{-k}^{*} 2^{-k} \right) 2^{e^*}.$$

Slično kao kod decimalne aritmetike

- ako je prva odbačena znamenka 1, broj zaokružujemo nagore,
- a ako je 0, nadolje.

Time smo napravili apsolutnu grešku manju ili jednaku od "pola zadnjeg prikazivog bita", tj. 2^{-t-1+e}.

Relativna greška zaokruživanja

Gledajući relativno, greška je manja ili jednaka

$$\left|\frac{x - f\ell(x)}{x}\right| \leq \frac{2^{-t-1+e}}{2^{-1} \cdot 2^{e}} = 2^{-t},$$

tj. imamo vrlo malu relativnu grešku.

Veličinu 2^{-t} zovemo jedinična greška zaokruživanja (engl. unit roundoff) i uobičajeno označavamo s u.

Za $x \in \mathbb{R}$ unutar prikazivog raspona, umjesto x sprema se zaokruženi broj $f\ell(x) \in \mathbb{R}(t, s)$ i vrijedi

$$f\ell(x) = (1 + \varepsilon)x, \quad |\varepsilon| \leq u,$$

gdje je ε relativna greška napravljena tim zaokruživanjem.

IEEE standard za prikaz brojeva

Prikaz realnih brojeva u računalu zove se prikaz s pomičnim zarezom/točkom (engl. floating point representation), a aritmetika je aritmetika pomičnog zareza/točke (engl. floating point arithmetic).

Veličine s i t prema novom IEEE standardu:

format	32-bitni	64-bitni	128-bitni
duljina mantise	23 bita	52 bita	112 bita
duljina eksponenta	8 bitova	11 bitova	15 bitova
jedinična gr. zaokr.	2^{-24}	2^{-53}	2^{-113}
$u \approx$	$5.96 \cdot 10^{-8}$	$1.11 \cdot 10^{-16}$	$9.63 \cdot 10^{-35}$
raspon brojeva $\approx$	$10^{\pm 38}$	$10^{\pm 308}$	$10^{\pm 4932}$

IEEE standard za prikaz brojeva (nastavak)

Većina PC računala (procesora) još ne podržava 128-bitni prikaz i aritmetiku.

Umjesto toga, FPU (Floating–point unit) stvarno koristi

- tzv. tip `extended` iz starog IEEE standarda.

Dio primjera koje ćete vidjeti napravljen je baš u tom tipu!

`format`	80-`bitni`
duljina mantise	64 bita
duljina eksponenta	15 bitova
jedinična gr. zaokr.	2^{-64}
$u \approx$	$5.42 \cdot 10^{-20}$
raspon brojeva $\approx$	$10^{\pm 4932}$

Realna aritmetika računala (IEEE standard)

IEEE standard za aritmetiku računala

Aritmetika računala nije egzaktna, jer rezultat operacije mora biti prikaziv.

IEEE standard propisuje i svojstva aritmetike — dozvoljene greške.
Za osnovne aritmetičke operacije ($\circ$ označava $+$, $-$, $*$, $/$) nad $x, y \in \mathbb{R}(t, s)$ vrijedi

$$fl(x \circ y) = (1 + \varepsilon)\,(x \circ y), \quad |\varepsilon| \leq u,$$

za sve $x, y \in \mathbb{R}(t, s)$ za koje je $x \circ y$ u dozvoljenom rasponu.

Dobiveni rezultat je tada prikaziv, tj. vrijedi $fl(x \circ y) \in \mathbb{R}(t, s)$ i ima malu relativnu grešku.

Svojstva aritmetike računala

Za aritmetiku računala ne vrijedi:

- asocijativnost zbrajanja i množenja,
- distributivnost množenja prema zbrajanju.

Jedino vrijedi:

- komutativnost za zbrajanje i množenje.

Primjer neasocijativnosti zbrajanja

Primjer. Asocijativnost zbrajanja u računalu ne vrijedi.

Znamo (odn. uskoro ćete znati) da je tzv. harmonijski red

$$1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{i} + \cdots$$

divergentan, tj. suma mu je "beskonačna".

No, nitko nas ne spriječava da računamo konačne početne komade ovog reda, tj. njegove parcijalne sume

$$S_n := 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-1} + \frac{1}{n}.$$

A kojim redom zbrajamo? (Zbrajanje je binarna operacija!)

Primjer neasocijativnosti zbrajanja (nastavak)

U realnim brojevima je potpuno svejedno kojim poretkom zbrajanja računamo ovu sumu, jer vrijedi asocijativnost.

$$a + (b + c) = (a + b) + c = a + b + c.$$

Uostalom, sam zapis izraza bez zagrada

$$S_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-1} + \frac{1}{n}$$

već "podrazumijeva" asocijativnost. U suprotnom, morali bismo zagradama naglasiti poredak operacija.

Ovdje imamo točno $n - 1$ binarnih operacija zbrajanja, i možemo ih napraviti kojim redom hoćemo.

Primjer neasocijativnosti zbrajanja (nastavak)

Drugim riječima, u prethodni izraz za S_n

- možemo rasporediti zagrade na bilo koji način, samo da svi plusevi budu "binarni", tj. zbrajaju dva objekta, a objekt je broj ili (podizraz u zagradama).

Na pr., zbrajanju "unaprijed" odgovara raspored zagrada

$$S_{n,1} := \left(\cdots \left(\left(1 + \frac{1}{2} \right) + \frac{1}{3} \right) + \cdots + \frac{1}{n-1} \right) + \frac{1}{n},$$

a zbrajanju "unatrag" odgovara raspored zagrada

$$S_{n,2} := 1 + \left(\frac{1}{2} + \left(\frac{1}{3} + \cdots + \left(\frac{1}{n-1} + \frac{1}{n} \right) \cdots \right) \right).$$

Primjer neasocijativnosti zbrajanja (nastavak)

Koliko takvih rasporeda zagrada ima — bit će napravljeno u Diskretnoj matematici. Bitno je da svi daju isti rezultat.

Komutativnost nam uopće ne treba. Ako i nju iskoristimo, dobivamo još puno više načina za računanje ove sume, i svi, naravno, opet daju isti rezultat.

Izračunajmo aritmetikom računala navedene dvije sume

- $S_{n,1}$ — unaprijed, i
- $S_{n,2}$ — unatrag,

za $n = 1\,000\,000$, u tri standardne IEEE točnosti single, double i extended. Preciznije, koristimo ova tri tipa za prikaz brojeva, uz pripadne artimetike za računanje.

Primjer neasocijativnosti zbrajanja (nastavak)

Uz skraćene oznake S_1 i S_2 za varijable u kojima zbrajamo pripadne sume, odgovarajući algoritmi za zbrajanje su:

- unaprijed

$$S_1 := 1,$$

$$S_1 := S_1 + \frac{1}{i}, \quad i = 2, \ldots, n,$$

- unatrag

$$S_2 := \frac{1}{n},$$

$$S_2 := \frac{1}{i} + S_2, \quad i = n-1, \ldots, 1.$$

Dakle, zaista ne koristimo komutativnost zbrajanja.

Primjer neasocijativnosti zbrajanja (nastavak)

Dobiveni rezultati za sume S_1, S_2 i pripadne relativne greške su:

tip i suma	vrijednost	rel. greška
single S_1	14.3573579788208008	$2.45740E{-}0003$
single S_2	14.3926515579223633	$5.22243E{-}0006$
double S_1	14.3927267228647810	$6.54899E{-}0014$
double S_2	14.3927267228657545	$-2.14449E{-}0015$
extended S_1	14.3927267228657234	$1.91639E{-}0017$
extended S_2	14.3927267228657236	$-1.08475E{-}0018$

Slovo E u brojevima zadnjeg stupca znači "puta 10 na", pa je, na pr., $-1.08475E{-}0018 = -1.08475 \times 10^{-18}$.

Primjer neasocijativnosti zbrajanja (nastavak)

Izračunate vrijednosti S_1 i S_2 su različite (u sve tri točnosti). Dakle, zbrajanje brojeva u aritmetici računala očito nije asocijativno.

Primijetite da, u sve tri točnosti, zbrajanje unatrag S_2 daje nešto točniji rezultat. To nije slučajno.
Svi brojevi koje zbrajamo su istog predznaka pa zbroj stalno raste, bez obzira na poredak zbrajanja.

- Kad zbrajamo unatrag — od manjih brojeva prema većim, zbroj se pomalo "nakuplja".
- Obratno, kad zbrajamo unaprijed — od velikih brojeva prema manjim, zbroj puno brže naraste. Onda mali dodani član jedva utječe na rezultat (tj. dobar dio znamenki pribrojnika nema utjecaj na sumu).

Širenje grešaka zaokruživanja

Širenje grešaka zaokruživanja

Vidimo da gotovo svaki izračunati rezultat ima neku grešku. Osim toga,

- zaokruživanje se vrši nakon svake pojedine operacije.

(Najlakše je stvar zamišljati kao da zaokruživanje ide "na kraju" operacije, iako je ono "dio operacije".)

Kad imamo puno aritmetičkih operacija (inače nam računalo ne treba), dolazi do tzv.

- akumulacije grešaka zaokruživanja.

Malo pogrešni rezultati (možda već od čitanja), ulaze u operacije, koje opet malo griješe, i tako redom ...

- greške se "šire" kroz sve što računamo!

Primjer katastrofalnog kraćenja

Zakruživanjem ulaznih podataka dolazi do male relativne greške. Kako ona može utjecati na konačan rezultat?

Primjer. Uzmimo realnu aritmetiku "računala" u bazi 10. Za mantisu (značajni dio) imamo $t = 4$ dekadske znamenke, a za eksponent $s = 2$ znamenke (što nije bitno). Neka je

$$x = 8.8866 = 8.8866 \times 10^0,$$
$$y = 8.8844 = 8.8844 \times 10^0.$$

Umjesto brojeva x i y (koji nisu prikazivi), u "memoriju" spremamo brojeve $fl(x)$ i $fl(y)$, pravilno zaokružene na $t = 4$ znamenke

$$fl(x) = 8.887 \times 10^0,$$
$$fl(y) = 8.884 \times 10^0.$$

Primjer katastrofalnog kraćenja (nastavak)

Ovim zaokruživanjem smo napravili malu relativnu grešku (ovdje je $u = 5 \times 10^{-5}$).

Razliku $f\ell(x) - f\ell(y)$ računamo tako da izjednačimo eksponente (što već jesu), oduzmemo značajne dijelove (mantise), pa normaliziramo

$$\begin{aligned} f\ell(x) - f\ell(y) &= 8.887 \times 10^0 - 8.884 \times 10^0 \\ &= 0.003 \times 10^0 = 3.??? \times 10^{-3}. \end{aligned}$$

Kod normalizacije, zbog pomaka "ulijevo", pojavljuju se

- ? = znamenke koje više ne možemo restaurirati (ta informacija se izgubila).

Što sad?

Primjer katastrofalnog kraćenja (nastavak)

Računalo radi isto što bismo i mi napravili:

- na ta mjesta ? upisuje 0.

Razlog: da rezultat bude točan, ako su ulazni brojevi točni. Dakle, ovo oduzimanje je egzaktno i u aritmetici računala.

Konačni rezultat je $fl(x) - fl(y) = 3.000 \times 10^{-3}$.

Pravi rezultat je

$$\begin{aligned} x - y &= 8.8866 \times 10^0 - 8.8844 \times 10^0 \\ &= 0.0022 \times 10^0 = 2.2 \times 10^{-3}. \end{aligned}$$

Već prva značajna znamenka u $fl(x) - fl(y)$ je pogrešna, a relativna greška je ogromna! Uočite da je ta znamenka (3), ujedno, i jedina koja nam je ostala — sve ostalo se skratilo!

Primjer katastrofalnog kraćenja (nastavak)

Prava katastrofa se događa ako $3.??? \times 10^{-3}$ uđe u naredna zbrajanja (oduzimanja), a onda se skrati i ta trojka!

Uočite da je oduzimanje $fl(x) - fl(y)$ bilo egzaktno (a egzaktno je i u aritmetici računala), ali rezultat je pogrešan.

Krivac, očito, nije oduzimanje (kad je egzaktno).

- Uzrok su polazne greške u operandima.

Ako njih nema, tj. ako su operandi egzaktni,

- i dalje (naravno) dolazi do kraćenja,
- ali je rezultat (uglavnom, a po IEEE standardu sigurno) egzaktan,

pa se ovo kraćenje zove benigno kraćenje.

Primjeri "grešaka" iz prakse

Promašaj raketa Patriot

- U Zaljevskom ratu, 25. veljače 1991. godine, Patriot rakete iznad Dhahrana u Saudijskoj Arabiji nisu uspjele oboriti iračku Scud raketu.
- Raketa je pukim slučajem pala na američku vojnu bazu usmrtivši 28 i ranivši stotinjak ljudi.

Promašaj raketa Patriot (nastavak)

Istraga otkriva sljedeće:

- Računalo koje je upravljalo Patriot raketama, vrijeme je brojilo u desetinkama sekunde proteklim od trenutka paljenja.
- Desetinka sekunde binarno

$$0.1_{10} = (0.0\dot{0}01\dot{1})_2.$$

- To računalo prikazivalo je realne brojeve korištenjem nenormalizirane mantise duljine 23 bita.
- Spremanjem 0.1 u registar takvog računala radi se (apsolutna) greška $\approx 9.5 \cdot 10^{-8}$.

Promašaj raketa Patriot (nastavak)

Detalji:

- Računalo je bilo u pogonu 100 sati, pa je ukupna greška zaokruživanja bila

$$100 \cdot 60 \cdot 60 \cdot 10 \cdot 9.5 \cdot 10^{-8} = 0.34\,\mathrm{s}.$$

- Scud raketa putuje brzinom $\approx 1.6\,\mathrm{km/s}$, pa je tražena više od pola kilometra daleko od stvarnog položaja.
- Greška uočena dva tjedna ranije nakon 8 sati rada sustava. Modifikacija programa stigla dan nakon nesreće.
- Posade sustava mogle su (ali nisu) dobiti uputu o "isključivanju i uključivanju računala" svakih nekoliko sati.

Samouništenje Ariane 5

- Raketa Ariane 5 lansirana 4. lipnja 1995. godine iz Kouroua (Francuska Gvajana) nosila je u putanju oko Zemlje komunikacijske satelite vrijednost 500 milijuna USD.
- 37 sekundi nakon lansiranja izvršila je samouništenje.

Samouništenje Ariane 5 (nastavak)

Objašnjenje:

- U programu za vođenje rakete postojala je varijabla koja je horizontalnu brzinu rakete (nije koristila ničemu).
- Greška je nastupila kad je program pokušao pretvoriti preveliki 64-bitni realni broj u 16-bitni cijeli broj.
- Računalo je javilo grešku, što je izazvalo samouništenje.
- Isti program bio korišten u prijašnjoj sporijoj verziji Ariane 4, pa do katastrofe nije došlo.

Potonuće naftne platforme

- Naftna platforma Sleipner A potonula je prilikom sidrenja, 23. kolovoza 1991. u blizini Stavangera.
- Baza platforme su 24 betonske ćelije, od kojih su 4 produljene u šuplje stupove na kojima leži paluba.

Potonuće naftne platforme (nastavak)

- Prilikom uronjavanja baze došlo je do pucanja veza među ćelijama (v. sliku).
- Rušenje je izazvalo potres jačine 3.0 stupnja po Richterovoj ljestvici i štetu od 700 milijuna USD.
- Greška je nastala u projektiranju, primjenom standardnog paketa programa, kad je upotrijebljena metoda konačnih elemenata s nedovoljnom točnošću.
- Proračun je dao naprezanja 47% manja od stvarnih.
- Točnijim proračunom utvrđeno je da su ćelije morale popustiti na dubini od 62 metra, a popustile su na 65 metara!