

Numerička matematika

1. predavanje

Saša Singer

`singer@math.hr`

`web.math.pmf.unizg.hr/~singer`

PMF – Matematički odsjek, 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**, s najboljim namjerama :-).

Sadržaj predavanja (nastavak)

- Uvodna priča o greškama:
 - Problemi numeričke matematike (zašto ona postoji).
 - Pojam greške, apsolutna i relativna greška.
 - Izvori grešaka — model, ulazni podaci (mjerjenje), 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 — kolokviji

Numerička matematika je u kolokvijskom razredu **A2**.

Službeni termini svih **kolokvija** su:

- **Prvi** kolokvij: **utorak, 2. 4. 2013.**, u **12** sati.
- **Drugi** kolokvij: **ponedjeljak, 27. 5. 2013.**, u **12** sati.
- **Popravni** kolokvij: **ponedjeljak, 10. 6. 2013.**, u **12** sati.

Uputa: “**izbjegnite**” popravni — obavite to **ranije!**

Informacije

Trenutno,

- **nema** posebnih informacija.

Zapravo, ima **jedna** — za **razliku** od prošle **dvije** godine:

- do **1. kolokvija** imamo “**standardnih**” **7** tjedana nastave,

- a **ne** samo **6** ili punih **8**.

Zato predavanja i vježbe idu **normalnim** ritmom.

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**, s najboljim namjerama :-).

Na samom početku

- **Moja malenkost** (u punom “sjaju”):

prof. 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/>**
odn. **<http://web.math.pmf.unizg.hr/~singer/>**

- **Konzultacije** (službeno):

- samo za **NM**: **ponedjeljak u 13 sati** (iza predavanja),

- **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. “egzaktom” 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, matična 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**, ili čak **15**.

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%,
- eventualna završna provjera znanja (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/isto kao na **Prog1** (nužna **prijava** za početak).

Pogledajte — već su “**žive**”, na adresi

<http://web.math.hr/nastava/unm/zadace.php>

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

- uzlaznim redom: **0, 1, 3, 4, 6, 7, 9, 10**.

Rok za **predaju** zadaća je

- dan **drugog** kolokvija, do **početka** kolokvija.

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 (tzv. “završna provjera znanja”):

- 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 završnoj provjeri znanja dobiti i **negativnu** ocjenu — tj. **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 **završnoj** provjeri znanja kao i za redovite 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 prošlih 5 godina, a stizati će i nova (kako nastaju).

Materijale za predavanja doc. **Grubišića** možete naći na

<http://web.math.hr/~luka/Nastava/>

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

Za totalno zbunjivanje, postoji i tzv. “srednja” skripta

http://web.math.hr/~singer/num_anal/num_alg.pdf

Zbunjivanje se riješava usporedbom sadržaja (ima dosta presjeka, ali i značajne simetrične razlike).

Literatura (4)

Dodatna literatura (piše u “službenom” popisu):

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

Pošteno, “nisam sretan” s njom — ima dosta grešaka.

Ako već treba neka preporuka, onda ovo:

- W. Cheney, D. Kincaid,
Numerical Mathematics and Computing (4. edition),
Brooks/Cole Publishing Company, Pacific Groove, 1999.

Ja imam 4. izdanje (možda ima i novije).

Knjiga je matematički vrlo korektna, ima algoritme i hrpu zadataka. Fali mali dio numeričke linearne algebre.

Literatura (5)

Sljedeće preporuke, posebno za **numeričku linearnu algebru**:

- **G. W. Stewart**,

- **Afternotes on Numerical Analysis**,
SIAM, Philadelphia, 1996.

- **Afternotes goes to Graduate School — Lectures on Advanced Numerical Analysis**,
SIAM, Philadelphia, 1996.

Fale zadaci i dio dokaza, ali je **prezentacija** vrlo dobra.

- **I. C. F. Ipsen**,

- **Numerical Matrix Analysis — Linear Systems and Least Squares Problems**,
SIAM, Philadelphia, 2009.

Kratka i **vrlo pregledna** knjiga, ima sve **osnovne dokaze**.

Literatura (6)

Za “čitače” njemačkog, 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!)

Može i hrpa

- malo starijih knjiga iz numeričke analize (matematike).

Samo da naglasak nije na nekom programskom jeziku ili alatu!

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**” trenutno ima samo **jednog demonstratora**:

- **Mario Berljafa**

Kroz neko vrijeme, kad se raspored ustabili,

- za **upute za dogovor i termine**,

- pogledajte **oglase** na oglasnoj ploči ili na **webu kolegija**.

Demos lijepo **moli** da mu se **najavite** mailom koji dan ranije!

Napomena uz kolokvije

Kolokviji iz prošlih godina “više” 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 **riješениh zadataka**.

Hm, ... znam što sad slijedi:

- predavanja su na **webu** — to je **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” tih materijala:

- **prije** predavanja, **pogledajte** i **isprintajte** ih — zgodno je 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 **svoga**:

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

Moderni **software** “**zna**” svašta

- računanje — numeričko i simboličko, vizualizacija, itd.

Međutim, to 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 jednačbi,
- 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 jednačbi ...

Problemi numeričke matematike (nastavak)

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

☛ poput sustava **linearnih** jednažbi (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 perturbacije 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** (mjerjenja 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 $E_{\text{abs}}(x, \hat{x}) = |\Delta x|$.

Katkad se $\Delta x = \hat{x} - x$ zove “prava” greška (predznak bitan).

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 jednačbi se **lineariziraju**, 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 **drugo**j točki.

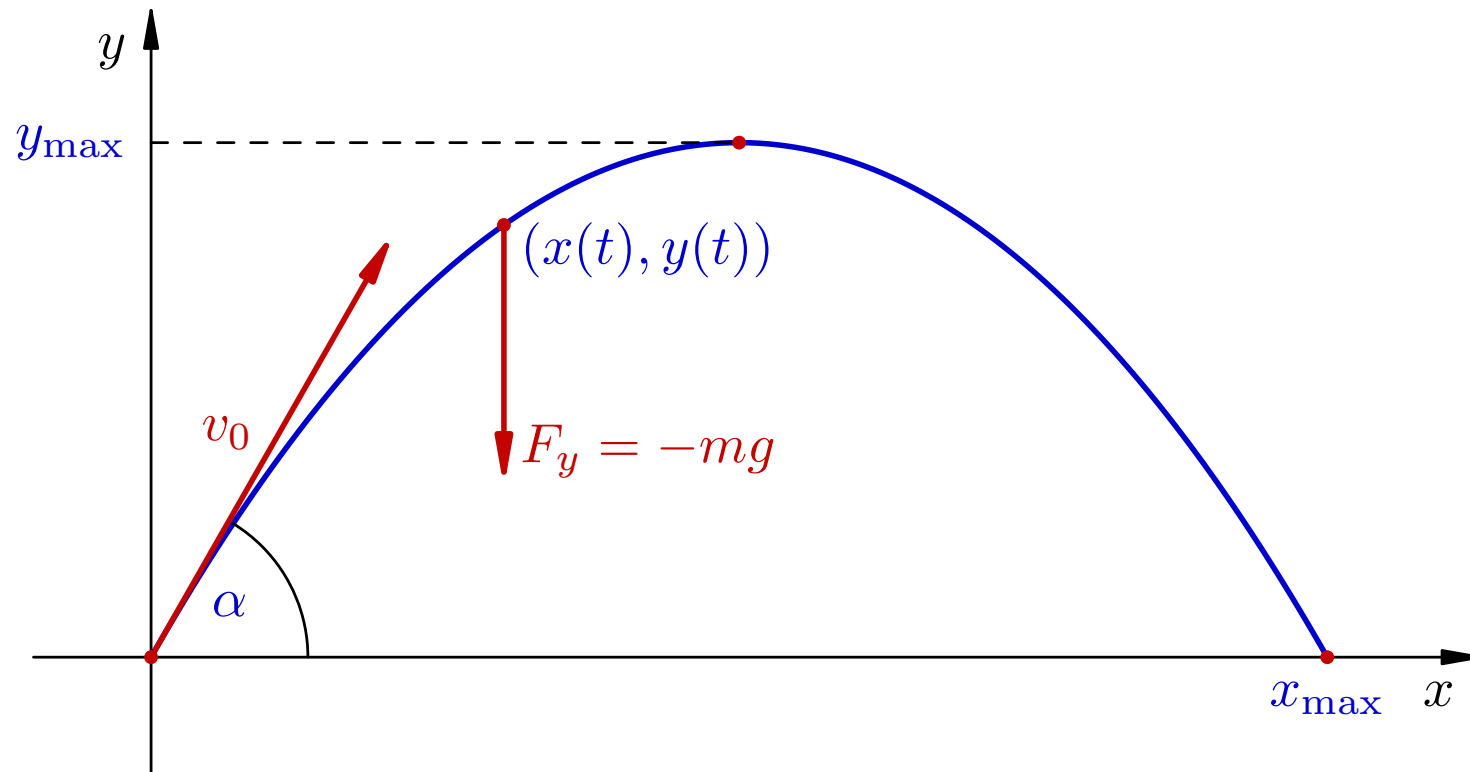
Najjednostavniji model za ovaj problem je poznati **kosi hitac**. Projektil ispaljujemo prema cilju,

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

Slikica (v. sljedeću stranicu)!

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

Modelni primjer — Slika za kosi hitac



Uzmimo da se cilj nalazi na “istoj visini” — u točki $(x_{\max}, 0)$.

🔴 Udaljenost x_{\max} znamo, a traži se početni kut α .

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 jednačbe

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

Nakon integracije, za **brzinu** $v =$ **prva** derivacija položaja, imamo jednačbu

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

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:

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

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 projektila — mjeri se.
- Izračunate tablice se eksperimentalno “upucavaju” i korigiraju.
- Primjena u praksi ide obratno — znam daljinu, tražim kut.

Na primjer, za obični **kosi hitac**, traženi **kut** je

$$\alpha = \frac{1}{2} \arcsin \frac{x_{\max} g}{v_0^2}.$$

Modelni primjer — Stvarni podaci

Primjer. Za ilustraciju, uzmimo podatke za pravu haubicu kalibra 155 mm, model H155 M65, uz najjače 7. punjenje (maksimalna količina baruta u čahuri).

Početna brzina ispaljene granate je $v_0 = 564 \text{ m/s}$.

Bez otpora zraka, maksimalni domet se postiže za kut $\alpha = 45^\circ = \pi/4$ i iznosi

$$d_{\max} = \frac{564^2 \sin(\pi/2)}{9.81} \approx 32\,425.69 \text{ m.}$$

Stvarni maksimalni domet postiže se za kut $\alpha = 45^\circ 10'$ i iznosi “samo”

$$d_{\max} = 14\,854 \text{ m.}$$

Greške modela (nastavak)

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

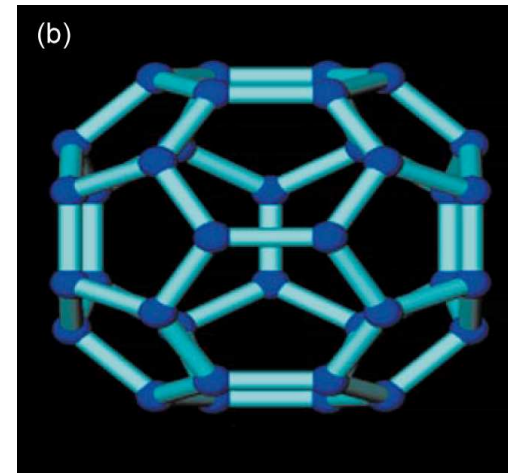
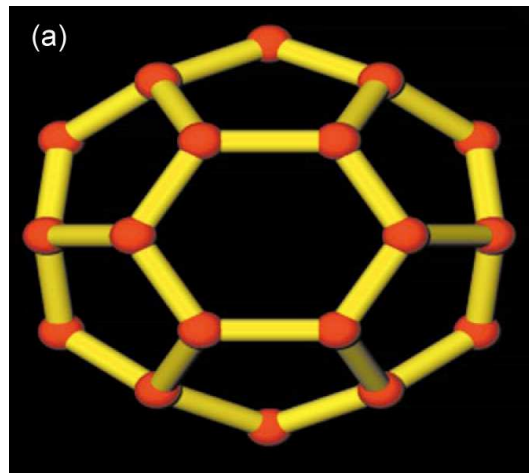
- određivanje trodimenzionalne strukture i elektronskog stanja **ugljik-36 fulerena**.

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).
- Na primjer, tjelesna temperatura se obično mjeri na **desetinku** stupnja Celzusa točno. Pacijent je podjednako loše ako ima temperaturu **39.5° C** ili **39.513462° C**.

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**:

$$2x + 6y = 8$$

$$2x + 6.0001y = 8.0001,$$

i

$$2x + 6y = 8$$

$$2x + 5.99999y = 8.00002.$$

Samo **druga** jednadžba se “malo” **promijenila**.

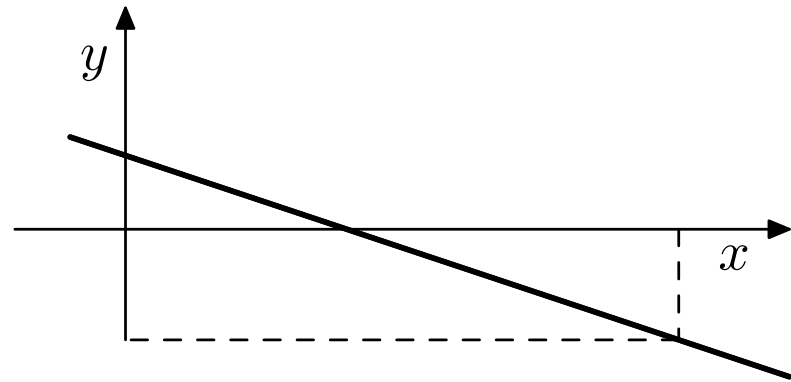
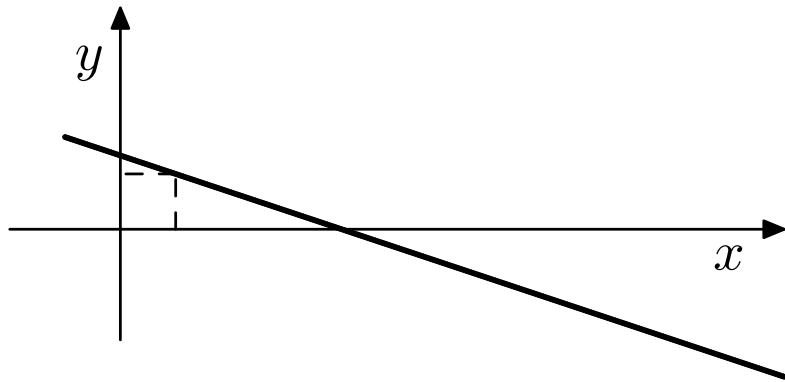
Perturbacije **koeficijenata** su reda veličine 10^{-4} .

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:



Koja dva pravca???

- U oba sustava, pravci su “skoro” paralelni, pa se njihovi grafovi ne razlikuju na slikama — i baš tu je problem!

Greške metoda za rješavanje problema

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

- **greške diskretizacije** koje nastaju
 - zamjenom **kontinuum**a (neprebrojiv skup) **konačnim diskretnim** skupom točaka,
 - ili “**beskonačno**” malu veličinu h ili $\varepsilon \rightarrow 0$ 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) $\{x_1, \dots, x_n\} \subset [a, b]$,
- aproksimacija derivacije funkcije f u nekoj točki x . Po definiciji je

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

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 greška postane dovoljno mala (recimo, kod sumiranja Taylorovih redova — v. sljedeći primjer).

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

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

pri čemu je $R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}$ **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

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

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

Za grešku odbacivanja trebaju nam derivacije:

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

pa su pripadne greške odbacivanja

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

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

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

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!**

Red za eksponencijalnu funkciju, $x = 12\pi$

Za $x = 12\pi$ i $\varepsilon = 5 \cdot 10^{-16}$ trebamo 132 člana reda ($n = 131$)

$$|\text{greška odbacivanja}| \leq 2.5101 \cdot 10^0$$

$$|\text{maksimalni član}| = 1.5329 \cdot 10^{15} \quad (n = 37).$$

Dobivamo:

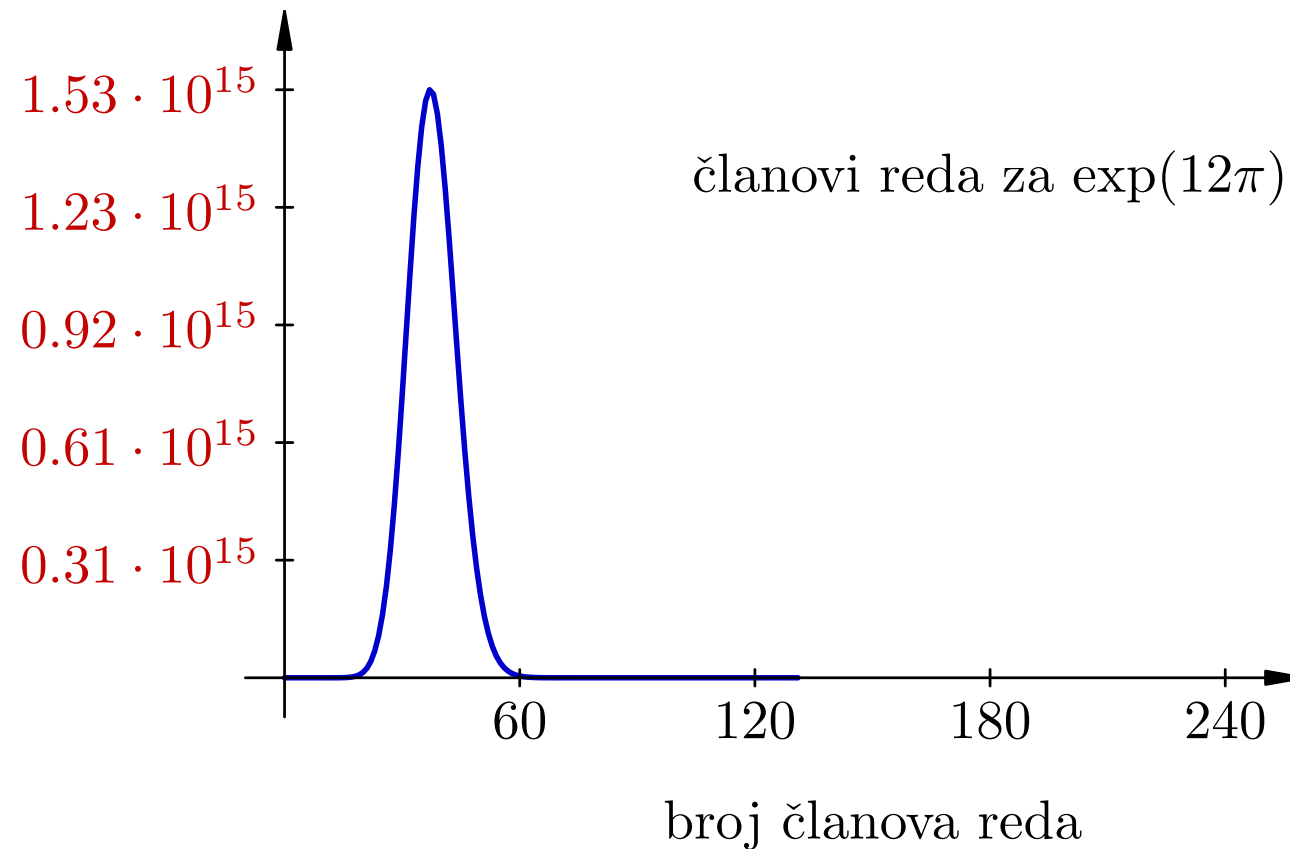
$$\exp(12\pi)_{\text{funkcija}} = 2.3578503968558192 \cdot 10^{16}$$

$$\exp(12\pi)_{\text{Taylor}} = 2.3578503968558196 \cdot 10^{16}$$

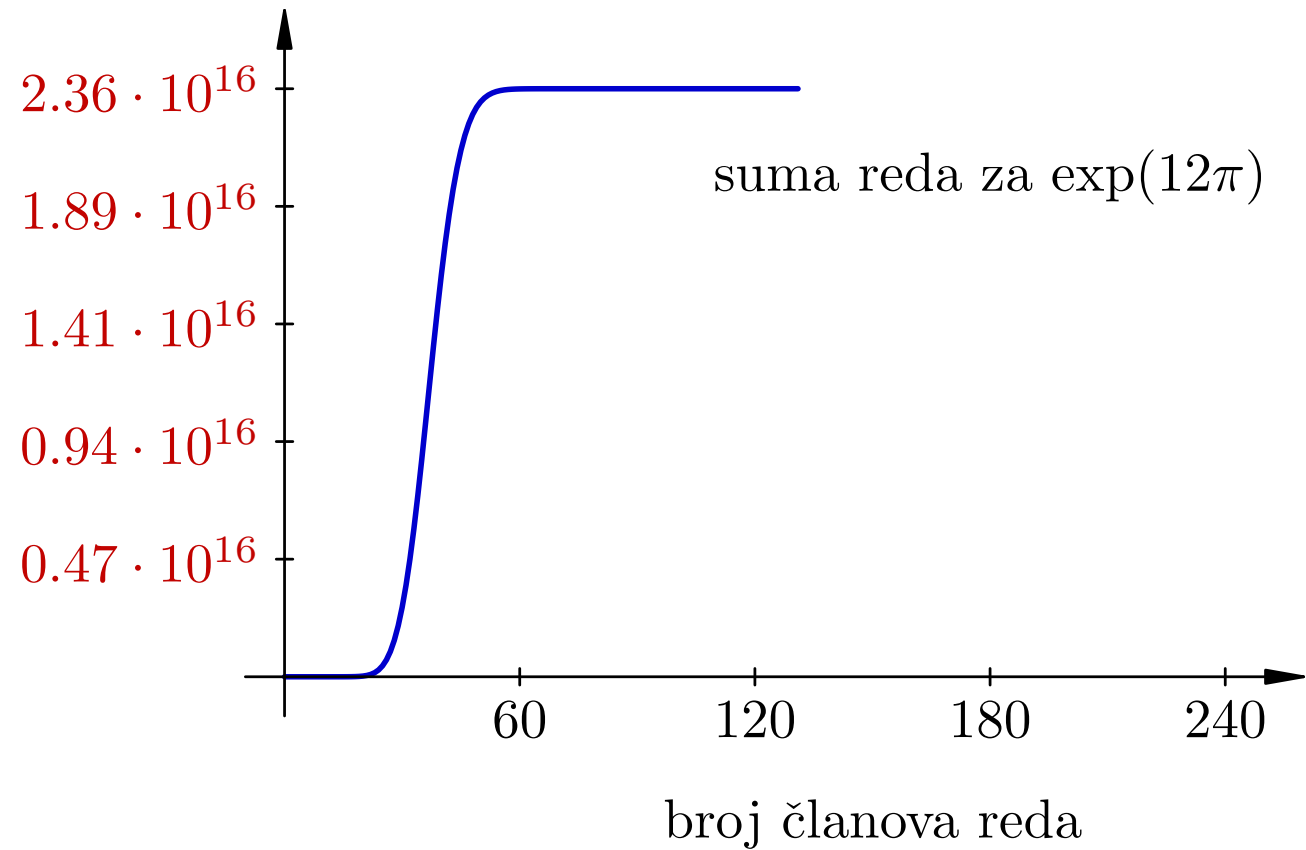
$$\text{prava greška} = -4.0000000000000000 \cdot 10^0$$

$$\text{relativna greška} = 1.6964604732064335 \cdot 10^{-16}.$$

Članovi reda za $\exp(12\pi)$



Suma reda za $\exp(12\pi)$



Red za eksponencijalnu funkciju, $x = 24\pi$

Za $x = 24\pi$ i $\varepsilon = 5 \cdot 10^{-16}$ trebamo 236 članova reda ($n = 235$)

$$|\text{greška odbacivanja}| \leq 5.0445 \cdot 10^{16}$$

$$|\text{maksimalni član}| = 2.5555 \cdot 10^{31} \quad (n = 75).$$

Dobivamo:

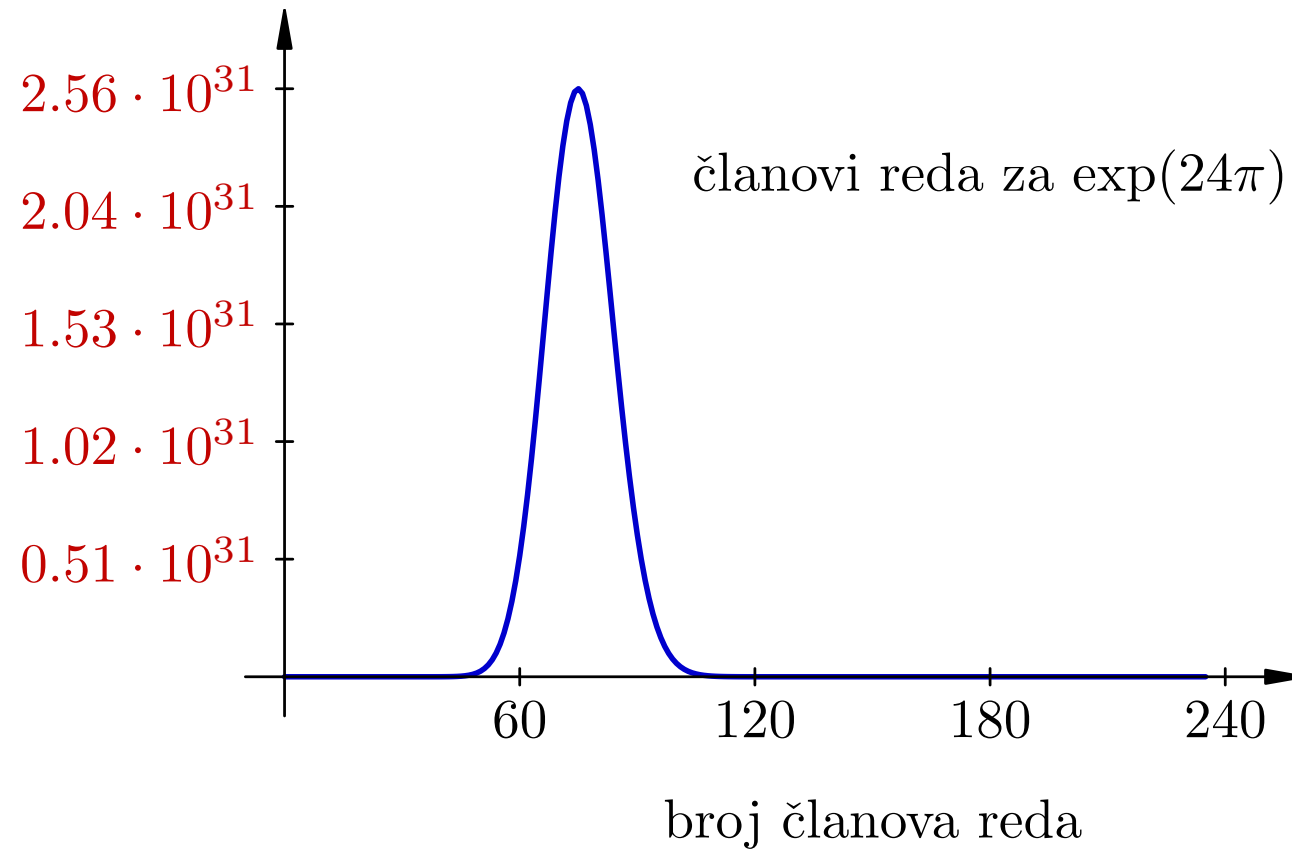
$$\exp(24\pi)_{\text{funkcija}} = 5.5594584939531437 \cdot 10^{32}$$

$$\exp(24\pi)_{\text{Taylor}} = 5.5594584939531445 \cdot 10^{32}$$

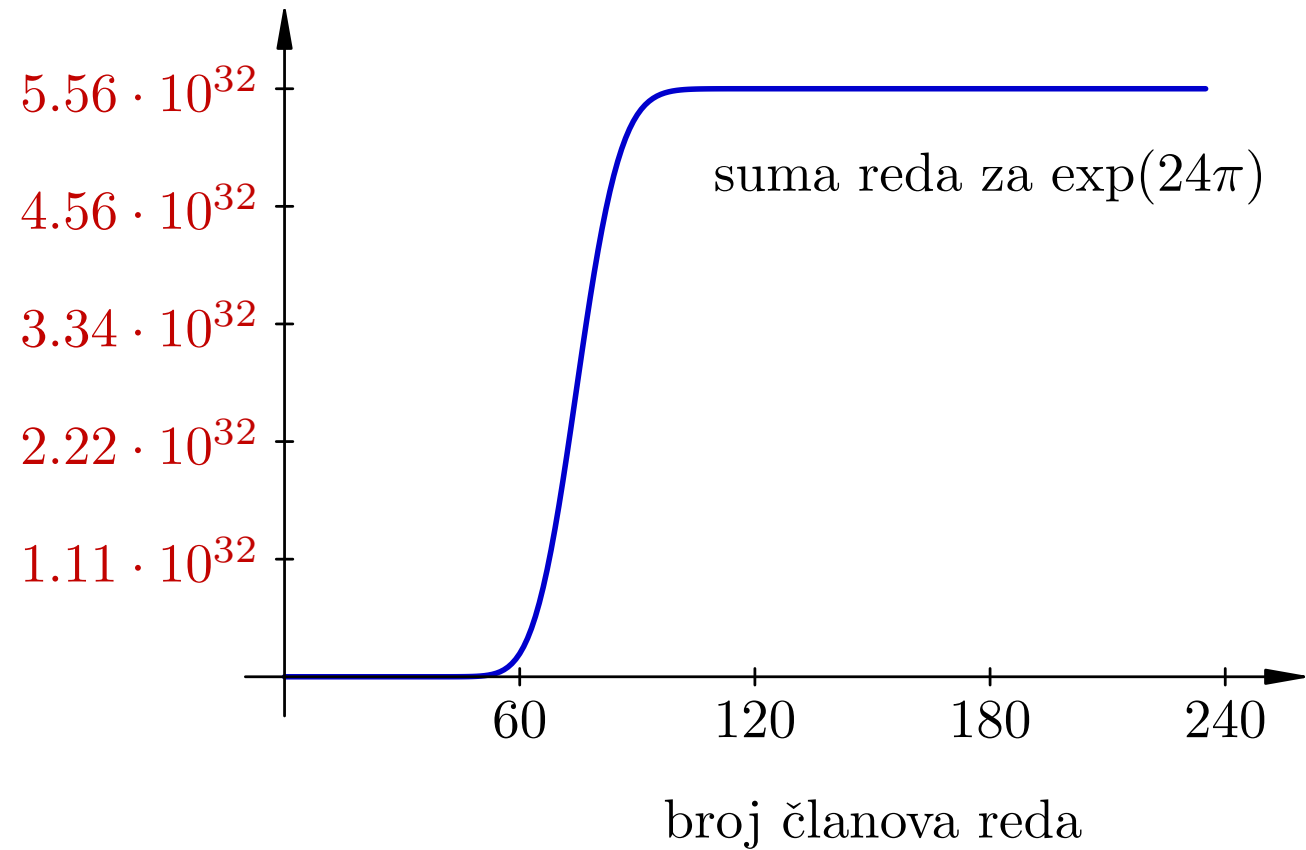
$$\text{prava greška} = -7.2057594037927936 \cdot 10^{16}$$

$$\text{relativna greška} = 1.2961261266057264 \cdot 10^{-16}.$$

Članovi reda za $\exp(24\pi)$



Suma reda za $\exp(24\pi)$



Red za funkciju sinus, $x = 12\pi$

Za $x = 12\pi$ i $\varepsilon = 5 \cdot 10^{-16}$ trebamo 66 članova reda ($n = 131$)

$$|\text{greška odbacivanja}| \leq 3.0175 \cdot 10^{-17}$$

$$|\text{maksimalni član}| = 1.5329 \cdot 10^{15} \quad (n = 37).$$

Dobivamo:

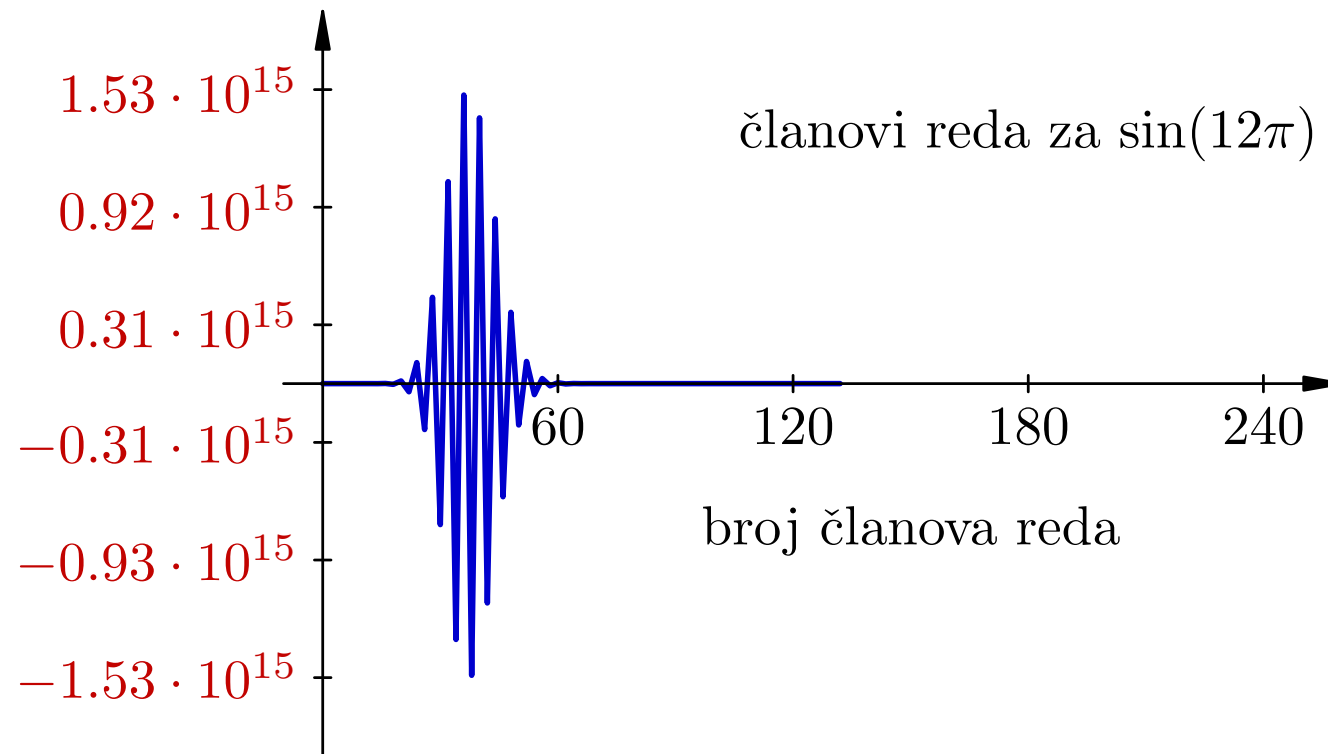
$$\sin(12\pi)_{\text{funkcija}} = -1.4695276245868527 \cdot 10^{-15}$$

$$\sin(12\pi)_{\text{Taylor}} = -4.1381632107344454 \cdot 10^{-2}$$

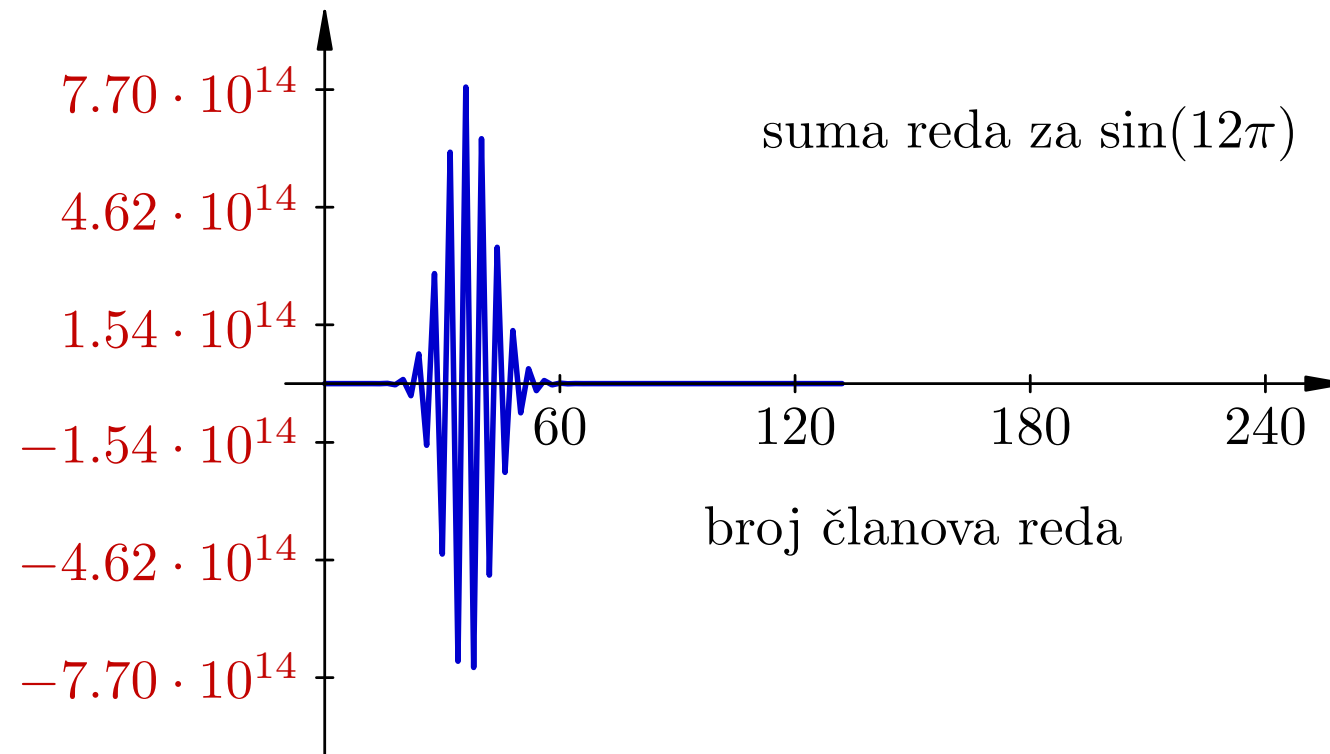
$$\text{prava greška} = 4.1381632107342983 \cdot 10^{-2}$$

$$\text{relativna greška} = 2.8159819124854586 \cdot 10^{13}.$$

Članovi reda za $\sin(12\pi)$



Suma reda za $\sin(12\pi)$



Red za funkciju sinus, $x = 24\pi$

Za $x = 24\pi$ i $\varepsilon = 5 \cdot 10^{-16}$ trebamo 118 članova reda ($n = 235$)

$$|\text{greška odbacivanja}| \leq 2.8867 \cdot 10^{-17}$$

$$|\text{maksimalni član}| = 2.5555 \cdot 10^{31} \quad (n = 75).$$

Dobivamo:

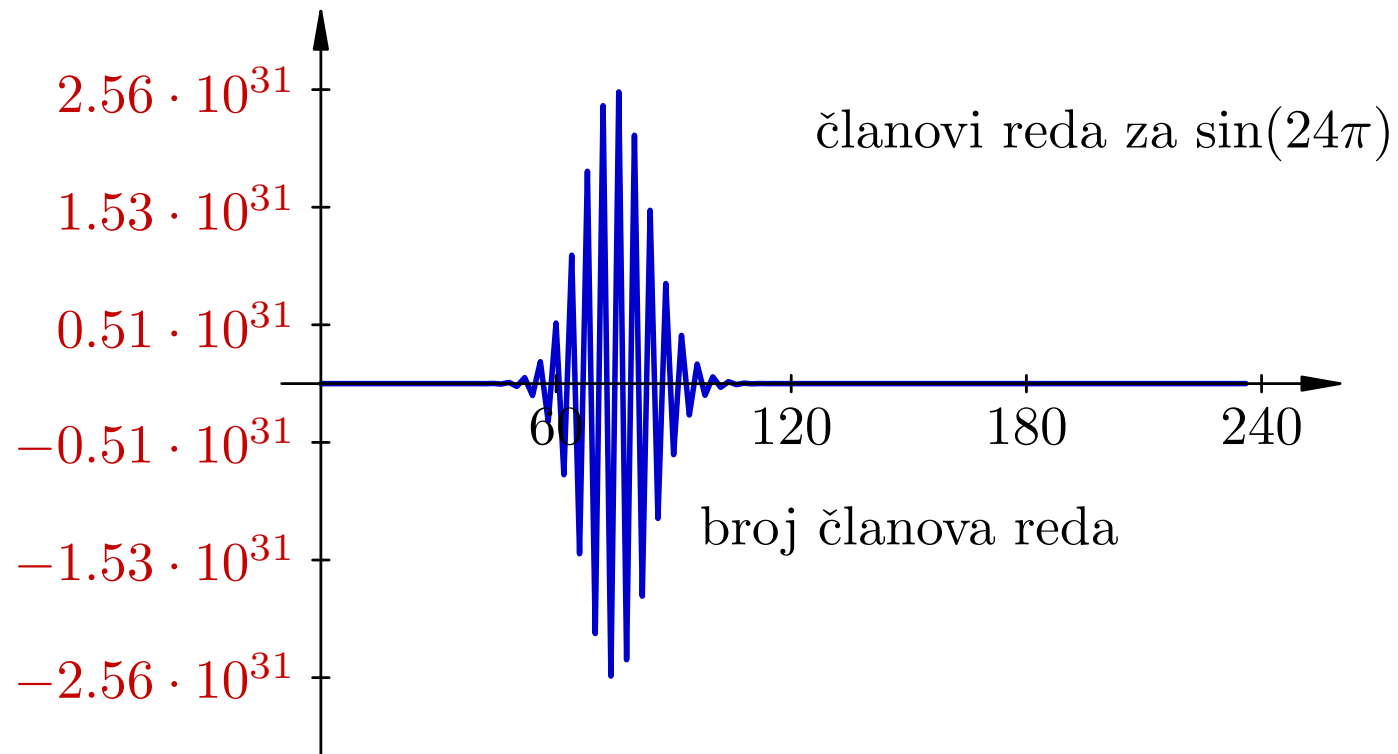
$$\sin(24\pi)_{\text{funkcija}} = -2.9390552491737054 \cdot 10^{-15}$$

$$\sin(24\pi)_{\text{Taylor}} = 3.6199983145905898 \cdot 10^{13}$$

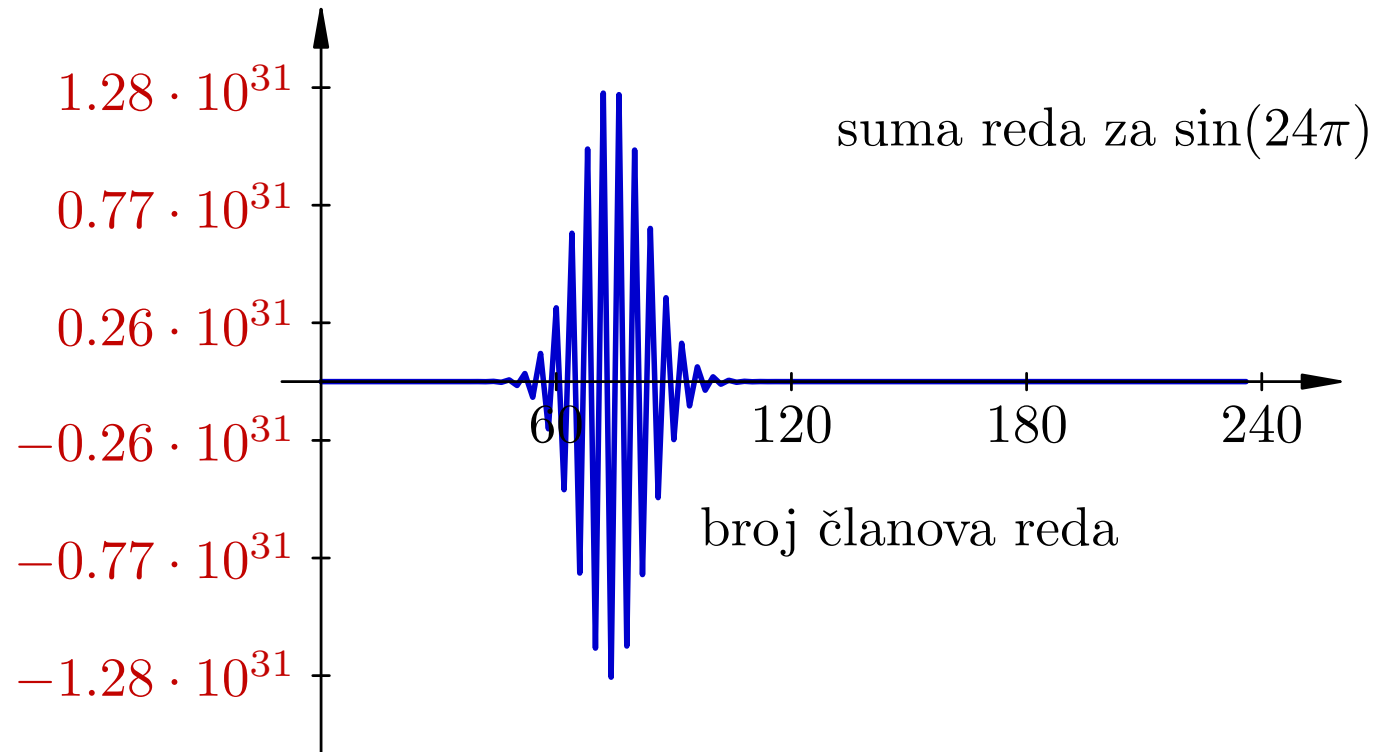
$$\text{prava greška} = -3.6199983145905898 \cdot 10^{13}$$

$$\text{relativna greška} = 1.2316877389794990 \cdot 10^{28}.$$

Članovi reda za $\sin(24\pi)$



Suma reda za $\sin(24\pi)$



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

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

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 s predznakom jednak

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

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

- nenegativni brojevi $B = 0, \dots, 2^{n-1} - 1$ imaju isti prikaz kao i bez predznaka,
- negativni brojevi $B = -1, \dots, -2^{n-1}$ 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 B 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, za svaki slučaj, treba provjeriti kako se dobiva proširenje ovih operacija s $\mathbb{N}_0 \times \mathbb{N}$ na $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$,
- tj. radi li compiler prema C99 standardu.

Dijeljenje cijelih brojeva s predznakom

Eksperiment:

- test-program `divmod.c` (pokaži!),
- Intel C++ compiler, gcc compiler (Code::Blocks).

Rezultati $q = a \operatorname{div} b$ i $r = a \operatorname{mod} b$ 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 (*standard*)

Ključ za interpretaciju:

- **kvocijent** se uvijek “zaokružuje” prema nuli,

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

- **ostatak** ima isti predznak kao i a .

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

Za ostatak r ovdje vrijedi:

- ako je $a \geq 0$, onda je $r \in \mathbb{Z}_b$, tj. $0 \leq r < |b|$,
- ako je $a < 0$, onda je $r \in -\mathbb{Z}_b$, tj. $-|b| < r \leq 0$.

Ovo je “**Euklidov** teorem” za cijele brojeve u \mathbb{C} -u!

Dijeljenje cijelih brojeva (*standard*)

Novi C99 standard propisuje ovakav izbor ostataka, tj.

- ovakvo ponašanje cjelobrojnog dijeljenja za cijele brojeve s predznakom.

Zato zaboravite raniju definiciju, iako se “novi” mod ponaša drugačije nego u matematici.

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 (misli se i na ostale programske jezike).

Cijeli brojevi u C-u — sažetak

U programskom jeziku C:

- cijelim brojevima **bez predznaka** odgovara tip koji se zove **unsigned int**, ili, skraćeno, **unsigned**,
- cijelim brojevima **s predznakom** odgovara tip koji se zove **int**.

Ovi tipovi postoje u nekoliko raznih **veličina**:

- standardna, **short**, **long**, a katkad i druge (**long long**).

Razlike su u **broju** bitova n predviđenih za prikaz.

- Zapis konstanti (vrijednost, navođenje tipa).
- Zapis operacija $+$, $-$ i \cdot znakovima $+$, $-$ i $*$.
- Zapis operacija **div** i **mod** znakovima $/$ i $\%$.

Cijeli brojevi u C-u — sažetak

U programskom jeziku C:

- cijelim brojevima s predznakom odgovara tip koji se zove `int`.

Ovaj tip postoji u nekoliko raznih **veličina**, a razlike su u **broju** bitova n predviđenih za prikaz.

Na 32-bitnim arhitekturama računala imamo sljedeće tipove:

- standardni `int` ($n = 32$),
- `short` ($n = 16$),
- `long` ($n = 32$), tj. **isto** kao standardni `int`,
- a katkad i druge, poput `long long` ($n = 64$).

Zapis konstanti (vrijednost, navođenje tipa) — sljedeći put.

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:

$$1! = 1,$$

$$n! = n \cdot (n - 1)!, \quad \text{za } n \geq 2.$$

Napišimo **C** 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?

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 $\mathbb{R}(t, w)$.

mantisa (signifikand)

1.	b_{-1}	b_{-2}	\dots	b_{-t}
----	----------	----------	---------	----------

eksponent (karakteristika)

e_{w-1}	e_{w-2}	\dots	e_1	e_0
-----------	-----------	---------	-------	-------

Ne može se svaki realni broj egzaktano spremiti u računalo.

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

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

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

Realni brojevi

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

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

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

$$\left| \frac{x - fl(x)}{x} \right| \leq \frac{2^{-t-1+e}}{2^0 \cdot 2^e} = 2^{-t-1},$$

tj. imamo vrlo **malu** relativnu grešku.

Veličinu 2^{-t-1} 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 $fl(x) \in \mathbb{R}(t, w)$ i vrijedi

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

gdje je ε **relativna** greška napravljena tim zaokruživanjem.

Prikaz “realnih” brojeva u računalu — IEEE standard

Stvarni prikaz realnih brojeva — IEEE-754

Stvarni prikaz realnih brojeva ima **tri dijela** i svaki od njih ima svoju **duljinu** — broj bitova predviđenih za prikaz tog dijela.

- **predznak s** — uvijek zauzima **jedan** bit, i to **najviši**;
- **karakteristika k** — zauzima sljedećih w bitova (w = engl. “width”, širina pomaknutog eksponenta);
- **signifikand m** — zauzima sljedećih t bitova (t = engl. “trailing”, završni ili razlomljeni dio od m).

Po starom standardu — ako se **pamti** vodeći (cjelobrojni) bit mantise, on je **prvi** (vodeći) u m , a duljina je $t + 1$.

Još se koristi i standardna oznaka

- **preciznost $p := t + 1$** — to je **ukupni broj vodećih značajnih** bitova cijele mantise.

Stvarni prikaz realnih brojeva — IEEE–754

Karakteristika k se interpretira kao cijeli broj bez predznaka, tako da je $k \in \{0, \dots, 2^w - 1\}$. “Rubne” vrijednosti za k označavaju tzv. posebna stanja:

- $k = 0$ — nula i denormalizirani brojevi,
- $k = 2^w - 1$ — beskonačno i nije broj.

Sve ostale vrijednosti $k \in \{1, \dots, 2^w - 2\}$ koriste se za prikaz normaliziranih brojeva različitih od nule.

Veza između karakteristike k i stvarnog eksponenta e je:

$$k = e + bias, \quad bias = 2^{w-1} - 1.$$

Dakle, dozvoljeni eksponenti e moraju biti između

$$e_{\min} = -(2^{w-1} - 2) \quad \text{i} \quad e_{\max} = 2^{w-1} - 1.$$

Standardni tipovi realnih brojeva — IEEE-754

Novi standard IEEE-754 standard ima sljedeće tipove za prikaz realnih brojeva:

ime tipa	binary32	binary64	binary128
duljina u bitovima	32	64	128
$t =$	23	52	112
$w =$	8	11	15
$u = 2^{-p}$	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	$10^{\pm 38}$	$10^{\pm 308}$	$10^{\pm 4932}$

Broj u je tzv. jedinična greška zaokruživanja (v. malo kasnije).

Najveći tip binary128 još uvijek ne postoji u većini procesora.

Standardni tipovi realnih brojeva — extended

Većina **PC** procesora još uvijek ima posebni dio — tzv. **FPU** (engl. Floating-Point Unit). On **stvarno** koristi

- tip **extended** iz **starog** IEEE standarda, koji odgovara tipu **extended binary64** u **novom** **IEEE-754** standardu.

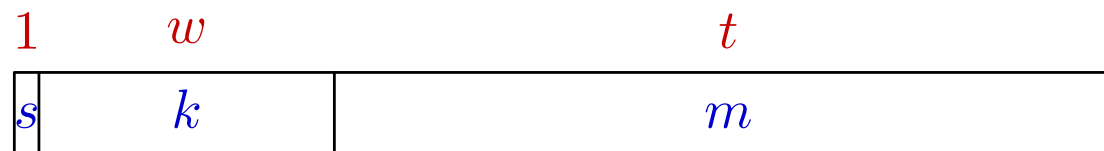
Dio primjera koje ćete vidjeti napravljen je baš u **tom tipu!**

ime tipa	extended
duljina u bitovima	80
$t + 1 =$	$63 + 1$
$w =$	15
$u = 2^{-p}$	2^{-64}
$u \approx$	$5.42 \cdot 10^{-20}$
raspon brojeva \approx	$10^{\pm 4932}$

Oznake

Oznake:

- **Crveno** — duljina odgovarajućeg polja u **bitovima**, bitove brojimo od **0**, zdesna nalijevo (kao i obično),
- **s** — predznak: **0** za pozitivan broj, **1** za negativan broj,
- **k** — karakteristika,
- **m** — mantisa (signifikand).



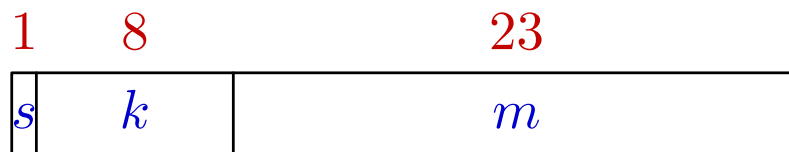
- Najznačajniji bit u odgovarajućem polju je **najljeviji**, a najmanje značajan bit je **najdesniji**.

Stvarni prikaz tipa single (binary32)

“Najkraći” realni tip je tzv. realni broj **jednostruke** točnosti. U C-u se taj tip zove **float**. Savjet: **ne koristiti** u praksi!

On ima sljedeća svojstva:

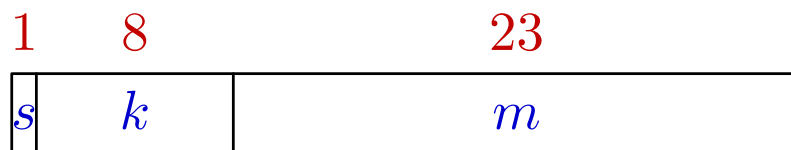
- duljina: 4 byte-a (32 bita), podijeljen u tri polja.



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

Prikaz brojeva jednostruke točnosti — sažetak

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



Vrijednost broja je

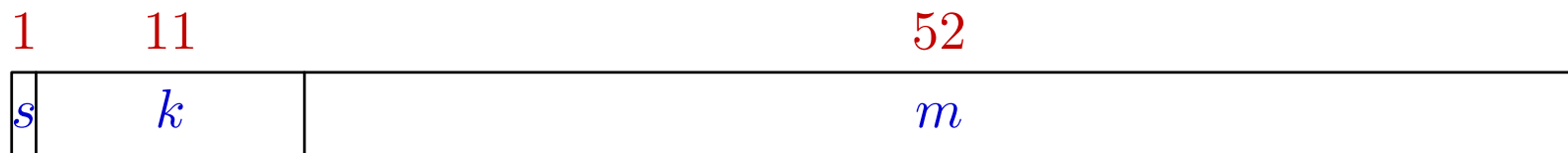
$$v = \begin{cases} (-1)^s * 2^{(k-127)} * (1.m) & \text{ako je } 0 < k < 255, \\ (-1)^s * 2^{(-126)} * (0.m) & \text{ako je } k = 0 \text{ i } m \neq 0, \\ (-1)^s * 0 & \text{ako je } k = 0 \text{ i } m = 0, \\ (-1)^s * \text{Inf} & \text{ako je } k = 255 \text{ i } m = 0, \\ \text{NaN} & \text{ako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

Stvarni prikaz tipa double (binary64)

“Srednji” realni tip je tzv. realni broj **dvostruke** točnosti. U C-u se taj tip zove **double**. Savjet: njega **treba koristiti!**

On ima sljedeća svojstva:

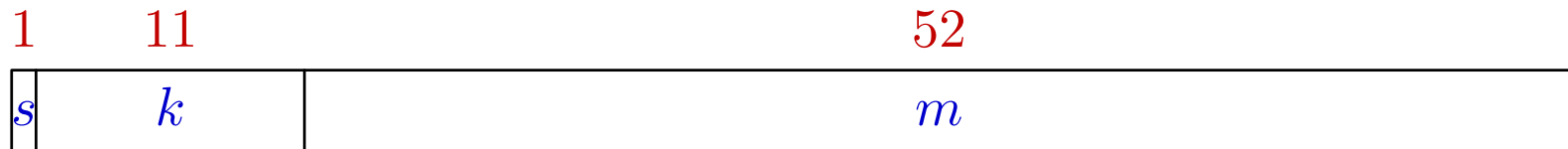
- Duljina: 8 byte-a (64 bita), podijeljen u **tri** polja.



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

Prikaz brojeva dvostruke točnosti — sažetak

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



Vrijednost broja je

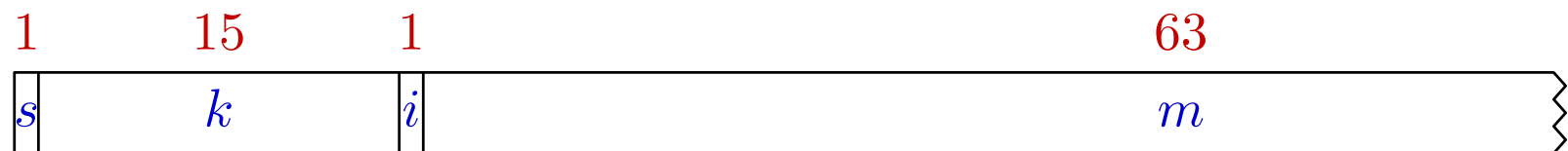
$$v = \begin{cases} (-1)^s * 2^{(k-1023)} * (1.m) & \text{ako je } 0 < k < 2047, \\ (-1)^s * 2^{(-1022)} * (0.m) & \text{ako je } k = 0 \text{ i } m \neq 0, \\ (-1)^s * 0 & \text{ako je } k = 0 \text{ i } m = 0, \\ (-1)^s * \text{Inf} & \text{ako je } k = 2047 \text{ i } m = 0, \\ \text{NaN} & \text{ako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

Tip extended

Stvarno računanje (na IA-32) se obično radi u “proširenoj” točnosti. U C-u je taj tip možda dohvatljiv kao `long double`.

On ima sljedeća svojstva:

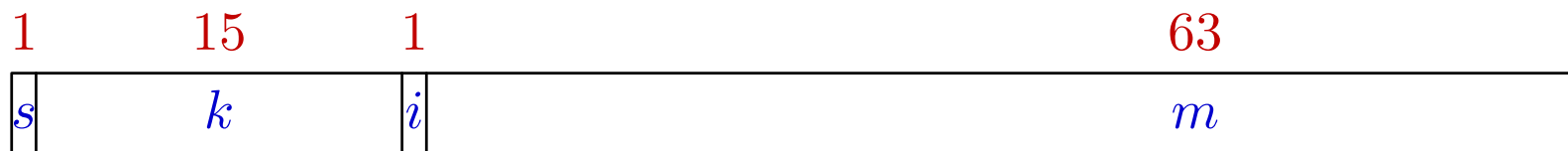
- Duljina: 10 byte-a (80 bita), podijeljen u četiri polja.



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

Prikaz brojeva proširene točnosti — sažetak

IEEE tip *extended*:



Vrijednost broja je

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

Uočite da **prva** mogućnost uključuje:

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

jer se **pamti** vodeći “cjelobrojni” bit i mantise.

Realna aritmetika računala (IEEE standard)

Realna aritmetika računala — standard

Realna aritmetika računala nije egzaktna!

Razlog:

- Rezultat svake operacije mora biti prikaziv,
- pa dolazi do zaokruživanja.

Standard IEEE-754 za realnu aritmetiku računala propisuje da za sve četiri osnovne aritmetičke operacije vrijedi

- ista ocjena greške zaokruživanja kao i za prikaz brojeva,
- tj. da izračunati rezultat ima malu relativnu grešku.

Isto vrijedi i za neke matematičke funkcije, poput $\sqrt{\quad}$, ali ne vrijedi za sve funkcije (na pr. za \cos i \sin u okolini nule).

Realna aritmetika računala — zaokruživanje

Neka je \circ bilo koja od aritmetičkih operacija $+$, $-$, $*$, $/$, i neka su x i y prikazivi operandi (drugih, ionako, nema u računalu).

- Ako su x i y u dozvoljenom, tj. normaliziranom rasponu,
- i ako se egzakti rezultat $x \circ y$, također, nalazi u normaliziranom rasponu (ne mora biti prikaziv),

za računalom izračunati (prikazivi) rezultat $fl(x \circ y)$ onda vrijedi

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

gdje je u jedinična greška zaokruživanja za dani tip brojeva. Ova ocjena odgovara zaokruživanju egzaktnog rezultata!

Prava relativna greška ε ovisi o x , y , operaciji \circ , i stvarnoj realizaciji aritmetike računala.

Posljedice zaokruživanja u realnoj aritmetici

Napomena. Bez pretpostavki o **normaliziranom** rasponu, prethodni rezultat **ne vrijedi** — greška može biti **puno veća!**

Zbog **zaokruživanja**, u realnoj aritmetici računala, nažalost,

- **ne vrijede** uobičajeni **zakoni** za aritmetičke operacije na skupu \mathbb{R} .

Na primjer, za aritmetičke operacije u **računalu**

- **nema asocijativnosti** zbrajanja i množenja,
- **nema distributivnosti** množenja prema zbrajanju.

Dakle, **poredak izvršavanja operacija** je **bitan!**

Zapravo, **jedino** standardno pravilo koje **vrijedi** je

- **komutativnost** za zbrajanje i za množenje.

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** ili **širenja** 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!**

Opasne i bezopasne operacije — sažetak

Jedina opasna operacija — kad rezultat može imati veliku relativnu grešku, je

- oduzimanje bliskih brojeva,
- i to samo kad polazni operandi već imaju neku grešku (samo oduzimanje je tada, najčešće, egzaktno).

Ovaj fenomen zove se opasno ili katastrofalno kraćenje.

Sve ostale operacije su bezopasne — relativna greška rezultata ne raste pretjerano. Posebno,

- dijeljenje malim brojem nije opasno,
- osim kad je mali broj nastao (ranijim) kraćenjem.

Nažalost, u nekim knjigama piše suprotno — i pogrešno.

Ponavljjanje i dodatak

Pogledajte **dodatak** ovom predavanju. Sadrži

- **ponavljanje** gradiva iz **Prog1** o prikazu brojeva u računalu i greškama zaokruživanja,
- još poneke stvari o **širenju** grešaka prilikom aritmetičkih operacija.

Primjeri “grešaka” iz prakse

Promašaj raketa Patriot

U prvom Zaljevskom ratu, 25. veljače 1991. godine, američke rakete Patriot nisu uspjele oboriti iračku Scud raketu iznad Dhahrana u Saudijskoj Arabiji.

- Scud raketa je pukim slučajem pala na američku vojnu bazu — usmrativši 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 (uključivanja) sustava.
- Desetinka sekunde binarno

$$0.1_{10} = (0.00011)_2.$$

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

Ne izgleda puno ... , a kamo li opasno.

Promašaj raketa Patriot (nastavak)

Detalji:

- Računalo je bilo u pogonu 100 sati, pa je ukupna greška zaokruživanja bila (stalno se zbraja, svakih 0.1 sekundi)

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

- Scud raketa putuje brzinom $\approx 1.6 \text{ km/s}$, pa je “tražena” više od pola kilometra daleko od stvarnog položaja.
- Greška je uočena dva tjedna ranije, nakon 8 sati rada jednog drugog sustava. Modifikacija programa stigla je dan nakon nesreće.
- Posade sustava mogle su i dva tjedna ranije dobiti uputu “isključi/uključi računalo” svakih nekoliko sati — ali je nisu dobile.

Samouništenje Ariane 5

Raketa **Ariane 5** lansirana 4. lipnja 1995. godine iz Kouroua (Francuska Gvajana).

- Nosila je u putanju oko Zemlje komunikacijske satelite vrijedne **500** milijuna USD.
- **37** sekundi nakon lansiranja izvršila je **samouništenje**.



Samouništenje Ariane 5 (nastavak)

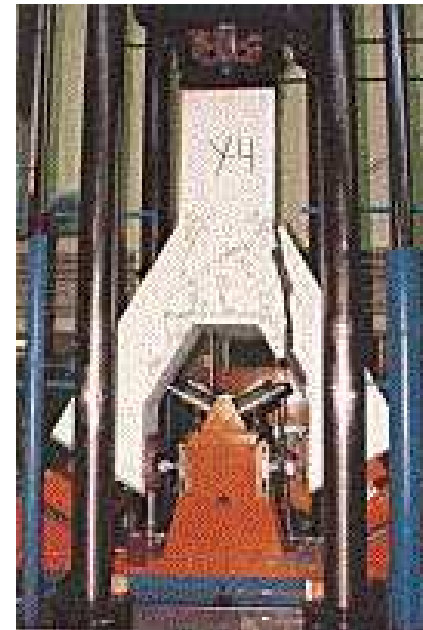
Objašnjenje:

- U programu za vođenje rakete postojala je varijabla koja je registrirala (pamtila) horizontalnu brzinu rakete (stvarno, 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 je korišten u prijašnjoj sporijoj verziji Ariane 4, pa do katastrofe nije došlo.

Potonuće naftne platforme

Naftna platforma **Sleipner A** **potonula** je prilikom prvog sidrenja, 23. kolovoza 1991. godine 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)

Razlozi nesreće:

- Prilikom uronjavanja baze došlo je do **pucanja veza** među ćelijama (v. desnu sliku).
- Rušenje na dno mora 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!