Računalo je izračunalo...

Nekoliko fama o računalima:

- računalom se sve može izračunati;
- rješenje se uvijek dobiva u kratkom vremenu;
- računalo uvijek daje točne rezultate.

Izvori grešaka kod rješavanja problema su:

- model,
- metoda za rješavanje modela,
- ulazni podaci (mjerenja),
- aritmetika računala.

Može li greška aritmetike biti dominantna u odnosu na ostale, tako da je rezultat zbog nje besmislen? **MOŽE!**

Griješe ljudi, rijetko računala (u novije vrijeme samo greška dijeljenja u jednoj seriji Pentium procesora 1994. godine).

U računalu postoje dva bitno različita tipa brojeva:

- cijeli brojevi,
- realni brojevi.

Oba skupa su **konačni podskupovi** $\mathbb{Z}$, odnosno $\mathbb{R}$. Za prikaz oba tipa koristi se baza 2.

Cijeli brojevi: duljina n bitova, aritmetika – modularna aritmetika u prstenu ostataka modulo 2^n, samo je sistem ostataka simetričan oko 0, tj.

$$-2^{n-1}, \ldots, -1, 0, 1, \ldots, 2^{n-1} - 1.$$

Realni brojevi: mantisa m i eksponent e:

$$r = \pm m \cdot 2^e,$$

e cijeli broj u određenom rasponu, a m racionalni broj za koji je $1/2 \leq m < 1$ (tj. mantisa započinje s $0.1\ldots$). Za $r = 0$, mantisa je 0.

Eksponent – s-bitni cijeli broj, mantisa – prvih t znamenki iza binarne točke.

mantisa				
$\pm$	m_{-1}	m_{-2}	$\cdots$	m_{-t}

eksponent				
$\pm$	e_{s-2}	e_{s-3}	$\cdots$	e_0

Skup realnih brojeva u računalu parametriziran duljinom mantise i eksponenta, u oznaci $\mathbb{R}(s,t)$.

Ne može se svaki realni broj egzaktno spremiti u računalo. Ako $x \in \mathbb{R}$ (unutar prikazivog raspona) oblika

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

a mantisa ima više od t znamenki, sprema se najbliža aproksimacija $fl(x) \in \mathbb{R}(t,s)$ koja se može prikazati kao

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

Zaokruživanje: greška manja ili jednaka 2^{-t-1+e}. Relativno, greška je manja ili jednaka

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

2^{-t} – jedinična greška zaokruživanja (engl. unit roundoff) – oznaka u.

IEEE standard

	`single`	`double`	`extended`
duljina	32 bita	64 bita	80 bitova
mantisa	$23 + 1$ bit	$52 + 1$ bit	64 bita
eksponent	8 bitova	11 bitova	15 bitova
u	2^{-24}	2^{-53}	2^{-64}
	$5.96 \cdot 10^{-8}$	$1.11 \cdot 10^{-16}$	$5.42 \cdot 10^{-20}$
raspon	$10^{\pm 38}$	$10^{\pm 308}$	$10^{\pm 4932}$

Osnovna pretpostavka: za osnovne aritmetičke operacije ($\circ$ označava $+$, $-$, $*$, $/$) nad $x, y \in \mathbb{R}(s,t)$ vrijedi

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

za sve $x, y \in \mathbb{R}(s,t)$ za koje je $x \circ y$ u dozvoljenom rasponu. Inače postoje rezervirani eksponenti koji označavaju "posebno stanje".

Značenje $fl(\ )$ – rezultat dobiven računalom za operaciju $x \circ y$. Model – izračunata vrijednost $x \circ y$ "jednako dobra" kao zaokružen egzaktni rezultat, u smislu da je u oba slučaja jednaka ocjena relativne greške. Model ne zahtijeva da je za $x \circ y \in \mathbb{R}(s,t)$ greška $\varepsilon = 0$.

Interpretacija desna strane u (1) – egzaktno izvedena operaciju $\circ$ na malo perturbiranim podacima. Koje su operacije opasne ako nam je aritmetika egzaktna, a podaci malo perturbirani, tj. ako je $|\varepsilon_x|$, $|\varepsilon_y| \leq u$?

$\circ$: množenje – benigno

$$x\,(1+\varepsilon_x) * y\,(1+\varepsilon_y) \approx xy\,(1+\varepsilon_x+\varepsilon_y) := xy(1+\varepsilon_*),$$

uz ocjenu $|\varepsilon_*| \leq 2u$.

$\circ$: dijeljenje – benigno

$$\frac{x\,(1+\varepsilon_x)}{y\,(1+\varepsilon_y)} \approx \frac{x}{y}\,(1+\varepsilon_x)\,(1-\varepsilon_y) := \frac{x}{y}\,(1+\varepsilon_/),$$

uz ocjenu $|\varepsilon_/| \leq 2u$.

x i y proizvoljnog predznaka. Za zbrajanje (oduzimanje) vrijedi:

$$x\,(1+\varepsilon_x) + y\,(1+\varepsilon_y) = (x+y)\Big(1 + \frac{x\varepsilon_x + y\varepsilon_y}{x+y}\Big),$$

ako je $x + y \neq 0$, pa definiramo

$$\varepsilon_\pm := \frac{x\varepsilon_x + y\varepsilon_y}{x+y} = \frac{x}{x+y}\,\varepsilon_x + \frac{y}{x+y}\,\varepsilon_y.$$

x i y brojevi istog predznaka – benigno

$$\left|\frac{x}{x+y}\right|,\ \left|\frac{y}{x+y}\right| \leq 1, \tag{2}$$

pa je $|\varepsilon_\pm| \leq 2u$.

x i y imaju različite predznake – opasno, jer kvocijenti u (2) mogu biti proizvoljno veliki.

OPASNOST – rezultat zbrajanja brojeva suprotnog predznaka = broj koji je po apsolutnoj vrijednosti mnogo manji od polaznih podataka.

Primjer na računalu u bazi 10. Mantisa: 4 dekadske znamenke, eksponent 2 i

$$x = 0.88866 = 0.88866 \cdot 10^0, \quad y = 0.88844 = 0.88844 \cdot 10^0.$$

Umjesto brojeva x i y, spremili smo

$$fl(x) = 0.8887 \cdot 10^0, \quad fl(y) = 0.8884 \cdot 10^0$$

i napravili malu relativnu grešku. Oduzimamo znamenku po znamenku mantise, pa normaliziramo

$$0.8887 \cdot 10^0 - 0.8884 \cdot 10^0 = 0.0003 \cdot 10^0 = 0.3??? \cdot 10^{-3},$$

? – znamenke koje više ne možemo restaurirati, pa računalo na ta mjesta upisuje 0. Pravi rezultat $0.22 \cdot 10^{-3}$ – prva značajna znamenka pogrešna!

Oduzimanje bilo egzaktno za $fl(x)$ i $fl(y)$, ali rezultat je pogrešan. Katastrofa — ako $0.3??? \cdot 10^{-3}$ uđe u naredna zbrajanja i oduzimanja i ako se pritom "skrati" i ta trojka.

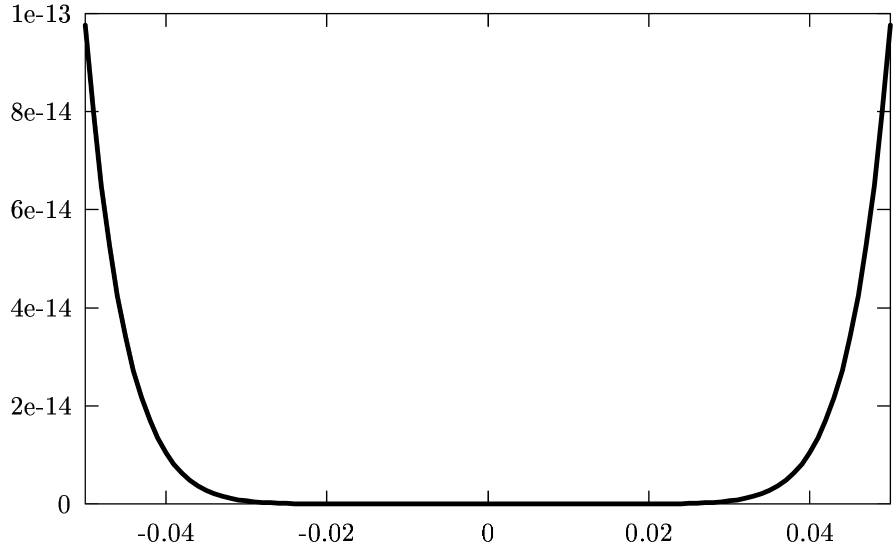
Primjer. Vrijednost $f(x) = (x - n)^{10}$, gdje je $n \in \mathbb{N}_0$ računamo računalom u okolini točke n, prvo direktno po formuli kako je napisano, a zatim korištenjem razvoja u Taylorov red oko točke 0. Što je točnije?

Taylorov red oko 0 je

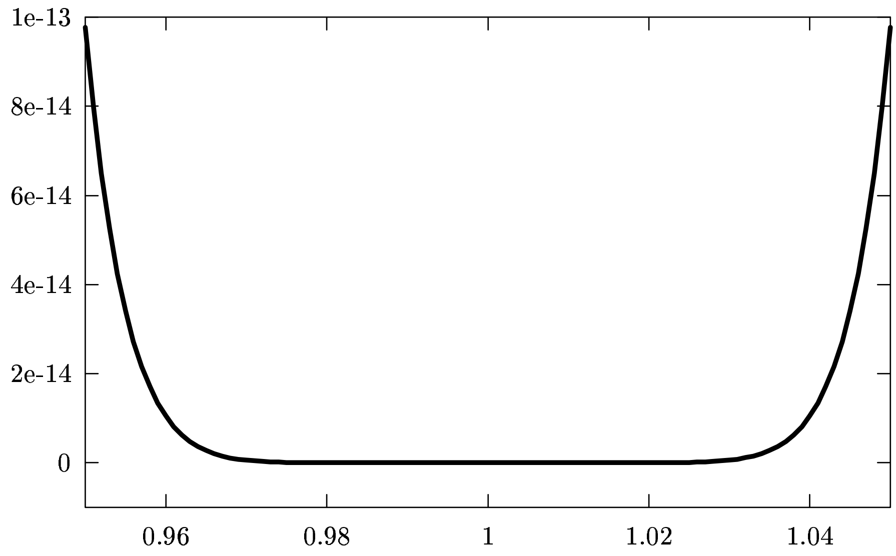
$$(x - n)^{10} = \sum_{k=0}^{10} \binom{10}{k} x^k (-n)^{10-k}. \qquad (3)$$

U okolini točke n lijeva strana (3) je mali broj. Koeficijenti desne strane (3) su alternirajući i rastu s porastom n. U sumi mora doći do kraćenja, pa je rezultat bolje izračunati direktno. Grafovi desnih strana u (3) za $n = 0, 1, \ldots, 5$.

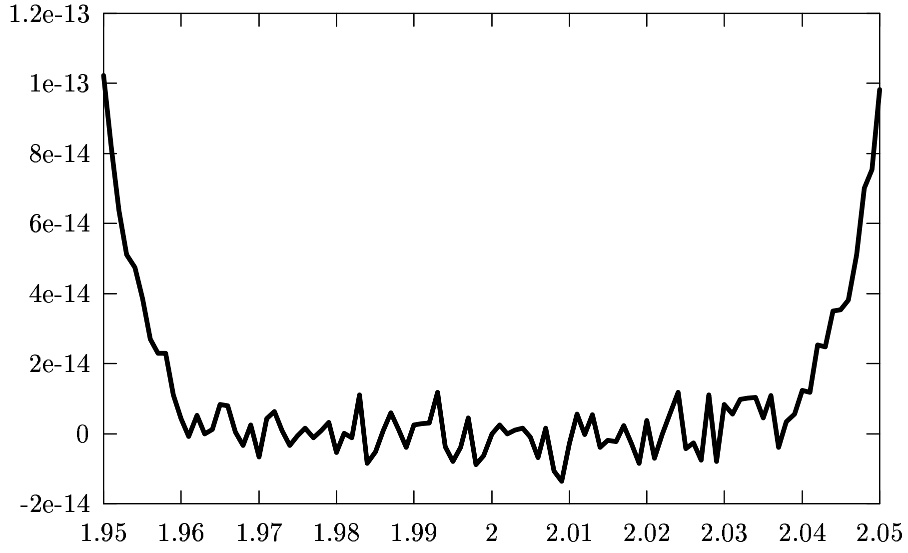
x^{10}



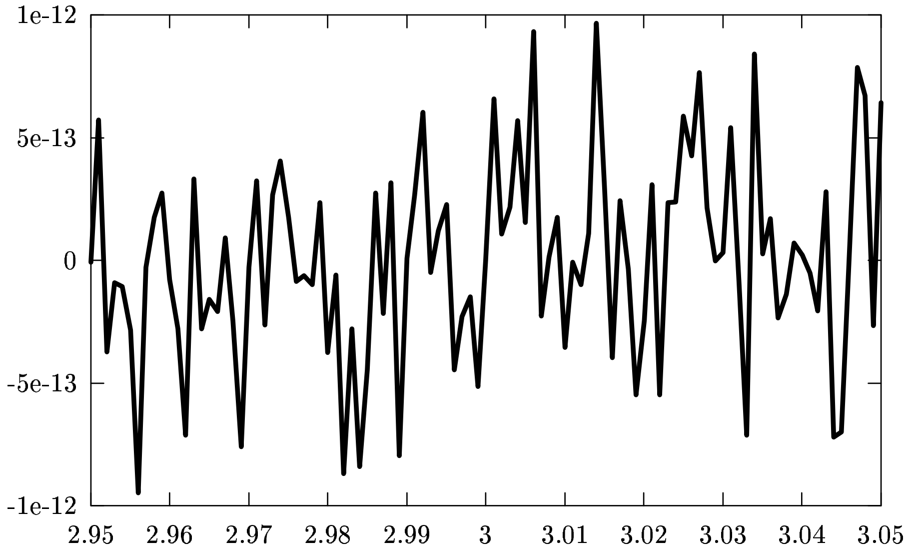
$(x-1)^{10}$



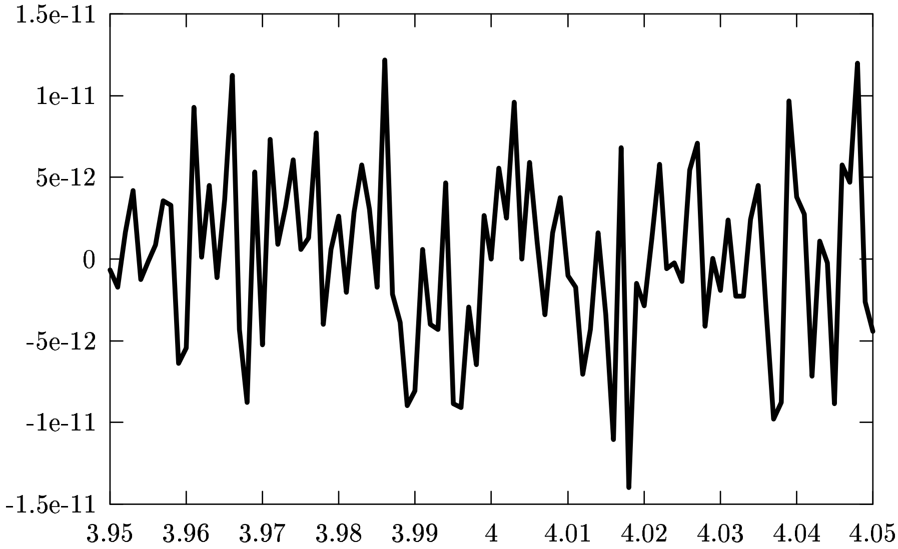
$(x-2)^{10}$


1.2e-13
1e-13
8e-14
6e-14
4e-14
2e-14
0
-2e-14
1.95
1.96
1.97
1.98
1.99
2
2.01
2.02
2.03
2.04
2.05

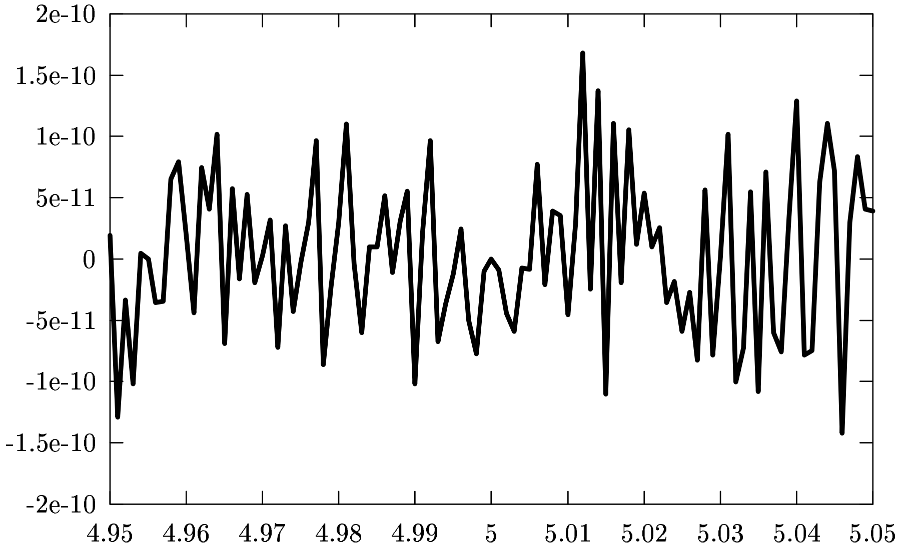
$(x-3)^{10}$


1e-12
5e-13
0
-5e-13
-1e-12
2.95
2.96
2.97
2.98
2.99
3
3.01
3.02
3.03
3.04
3.05

$(x-4)^{10}$



$(x-5)^{10}$



$\square$

Primjer. Taylorovi redovi za e^x i $\sin x$ oko 0 konvergiraju za proizvoljan $x \in \mathbb{R}$. Zbrajanjem dovoljno mnogo članova – u principu možemo dobro aproksimirati vrijednosti funkcija.

Ako to napravimo računalom, rezultat će biti zanimljiv.

Greška metode – Taylorov red aproksimiramo Taylorovim polinomom

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

uz grešku

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} x^{n+1},$$

ξ neki broj između 0 i x. Traženi Taylorovi polinomi su

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

Vrijedi

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

pa su pripadne greške odbacivanja

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

Za $x \geq \xi > 0$ dobivamo ocjene

$$\begin{aligned} |R_{n+1}(x)| &\leq \frac{e^x x^{n+1}}{(n+1)!}, \\ |R_{2n+3}(x)| &= \left| \frac{\sin(\xi + \frac{2n+3}{2}\pi) x^{2n+3}}{(2n+3)!} \right| \leq \left| \frac{x^{2n+3}}{(2n+3)!} \right|. \end{aligned}$$

Prvi odbačeni član ispod zadane točnosti $\varepsilon = 10^{-17}$ – greška odbacivanja manja ili jednaka

$$\begin{cases} e^x \varepsilon & \text{za } e^x, \\ \varepsilon & \text{za } \sin x. \end{cases} \tag{4}$$

Izračunajmo $\sin(15\pi)$, $e^{15\pi}$, $\sin(25\pi)$ i $e^{25\pi}$ korištenjem Taylorovog reda oko 0 u `extended` preciznosti.

Primjer je izabran tako da je $\sin(15\pi) = \sin(25\pi) = 0$, dok su druga dva broja vrlo velika.

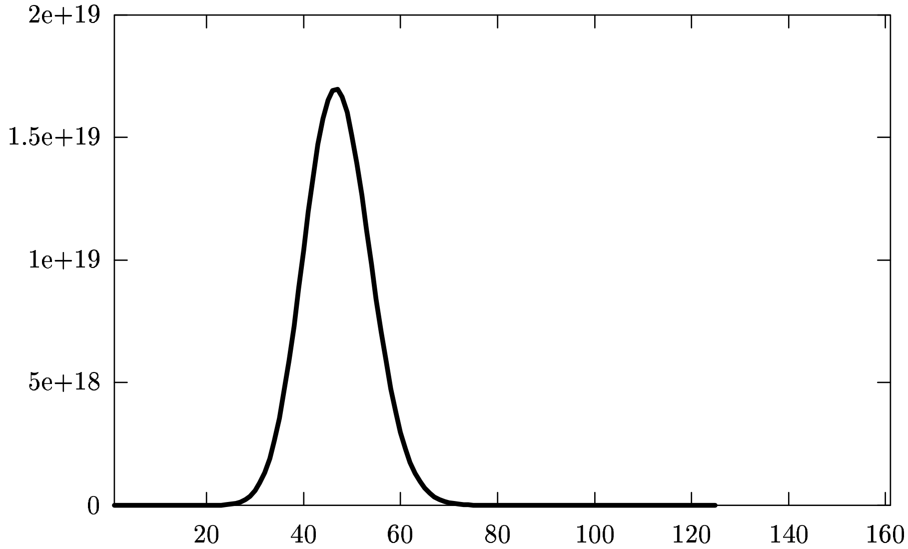
$$\begin{aligned}
\sin(15\pi)_{funkcija} &= 9.3241 \cdot 10^{-18} \\
\sin(15\pi)_{Taylor} &= -2.8980 \cdot 10^{-1} \\
|greška\ odbacivanja| &\leq 2.7310 \cdot 10^{-19} \\
relativna\ greška &= 3.1081 \cdot 10^{16} \\
|maksimalni\ član| &= 1.6969 \cdot 10^{19}
\end{aligned}$$

$$\begin{aligned}
\sin(25\pi)_{funkcija} &= 1.6697 \cdot 10^{-17} \\
\sin(25\pi)_{Taylor} &= 3.0613 \cdot 10^{13} \\
|greška\ odbacivanja| &\leq 5.8309 \cdot 10^{-19} \\
relativna\ greška &= 1.8334 \cdot 10^{30} \\
|maksimalni\ član| &= 5.7605 \cdot 10^{32}
\end{aligned}$$

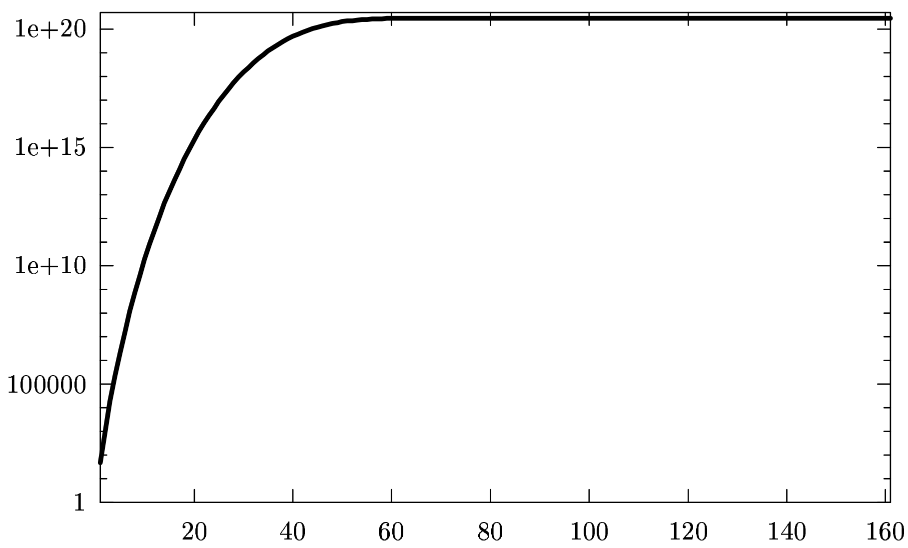
$$\begin{aligned}
\exp(15\pi)_{funkcija} &= 2.9218 \cdot 10^{20} \\
\exp(15\pi)_{Taylor} &= 2.9218 \cdot 10^{20} \\
|greška\ odbacivanja| &\leq 2.7600 \cdot 10^{2} \\
relativna\ greška &= 1.4238 \cdot 10^{-18} \\
|maksimalni\ član| &= 1.6969 \cdot 10^{19}
\end{aligned}$$

$$\begin{aligned}
\exp(25\pi)_{funkcija} &= 1.2865 \cdot 10^{34} \\
\exp(25\pi)_{Taylor} &= 1.2865 \cdot 10^{34} \\
|greška\ odbacivanja| &\leq 2.3782 \cdot 10^{16} \\
relativna\ greška &= 7.0013 \cdot 10^{-19} \\
|maksimalni\ član| &= 5.7943 \cdot 10^{32}.
\end{aligned}$$

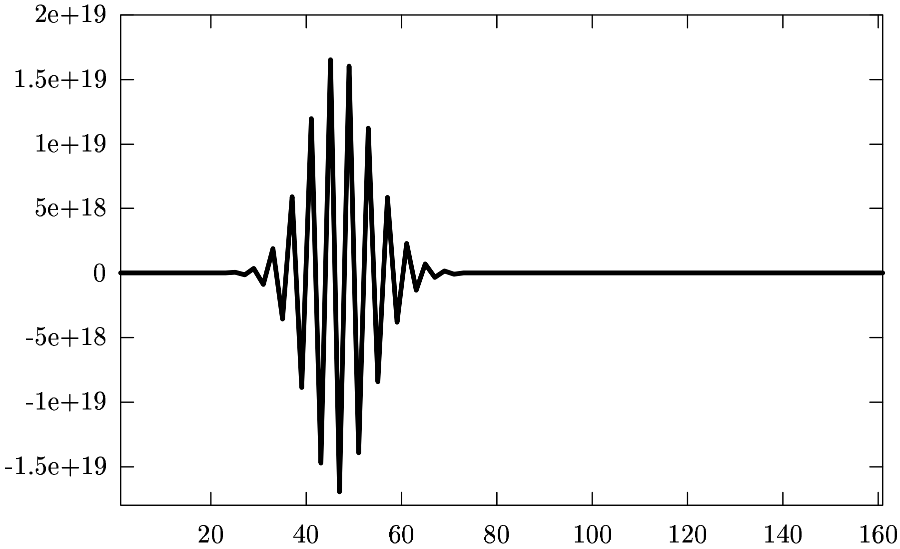
članovi reda $e^{15\pi}$



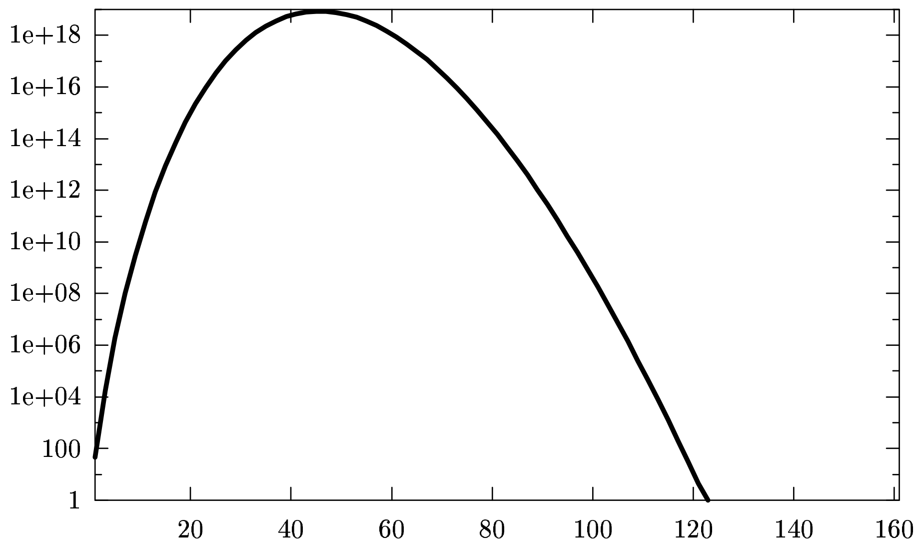
|zbroj prvih članova reda za $e^{15\pi}$|



članovi reda $\sin(15\pi)$



$|$zbroj prvih članova reda za $\sin(15\pi)|$



Objašnjenje — očito. Za $\sin x$, rezultat je malen broj koji je dobiven – oduzimanjem velikih brojeva, pa je netočan. Kod e^x, uvijek imamo zbrajanje brojeva istog predznaka, pa je rezultat točan. $\square$

Primjer. Poredak operacija nije beznačajan. Zadan je linearni sustav

$$\begin{aligned} 0.0001\,x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2. \end{aligned}$$

Jedinstveno rješenje $x_1 = 1.\dot{0}00\dot{1}$, $x_2 = 0.\dot{9}99\dot{8}$.

Rješavanje računalom koje ima 4 decimalne znamenke mantise i 2 znamenke eksponenta, onda njegovo rješenje ovisi o poretku jednadžbi. Sustav zapisan u takvom računalu pamti se kao

$$\begin{aligned} 0.1000 \cdot 10^{-3}\,x_1 + 0.1000 \cdot 10^{1}\,x_2 &= 0.1000 \cdot 10^{1} \\ 0.1000 \cdot 10^{1}\quad x_1 + 0.1000 \cdot 10^{1}\,x_2 &= 0.2000 \cdot 10^{1}. \end{aligned} \tag{5}$$

Množenjem prve jednadžbe s 10^4 i oduzimanjem od druge, dobivamo drugu jednadžbu:

$$(0.1000\cdot 10^{1} - 0.1000\cdot 10^{5})\,x_2 = 0.2000\cdot 10^{1} - 0.1000\cdot 10^{5}. \tag{6}$$

Računalo mora oduzimati: manji eksponent postaje jednak većem, a mantisa se denormalizira. Dobivamo

$$0.1000 \cdot 10^{1} = \cdots = 0.0001 \cdot 10^{4} = 0.0000|1 \cdot 10^{5}.$$

Zadnju znamenku nemamo gdje spremiti – mantisa postala 0. Slično je i s desnom stranom. Zbog toga (6) postaje

$$-0.1000 \cdot 10^{5}\,x_2 = -0.1000 \cdot 10^{5},$$

pa joj je rješenje $x_2 = 0.1000 \cdot 10^{1}$. Uvrštavanjem u prvu jednadžbu, dobivamo:

$$0.1000 \cdot 10^{-3}\,x_1 = 0.0000,$$

pa je $x_1 = 0$, što nije niti približno točan rezultat.

Promijenimo li poredak jednadžbi u (5), dobivamo

$$\begin{aligned} 0.1000 \cdot 10^1 \quad x_1 + 0.1000 \cdot 10^1\, x_2 &= 0.2000 \cdot 10^1 \\ 0.1000 \cdot 10^{-3}\, x_1 + 0.1000 \cdot 10^1\, x_2 &= 0.1000 \cdot 10^1. \end{aligned} \tag{7}$$

Množenjem prve jednadžbe s 10^{-4} i oduzimanjem od druge, dobivamo drugu jednadžbu

$$(0.1000 \cdot 10^1 - 0.1000 \cdot 10^{-3})\, x_2 = 0.1000 \cdot 10^1 - 0.2000 \cdot 10^{-3}, \tag{8}$$

pa se (8) svede na $0.1000 \cdot 10^1\, x_2 = 0.1000 \cdot 10^1$, tj. $x_2 = 0.1000 \cdot 10^1$. Uvrštavanjem u prvu jednadžbu u (7) dobivamo

$$0.1000 \cdot 10^1\, x_1 = 0.1000 \cdot 10^1,$$

pa je $x_1 = 0.1000 \cdot 10^1$, što točan rezultat korektno zaokružen na četiri decimalne znamenke. □

Promašaj raketa Patriot. Patriot rakete nisu uspjele pronaći i oboriti Scud.

Razlog katastrofe: vrijeme u računalu brojilo se desetinkama sekunde proteklim od trenutka od kad je računalo upaljeno. Desetinka u binarnom prikazu:

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

Realne brojeve u tom računalu prikazivali su korištenjem mantise (nenormalizirane) duljine 24 bita. Spremanjem 0.1 u registar Patriot računala, napravljena je greška približno jednaka $9.5 \cdot 10^{-8}$.

Zbog stalne opasnosti od napada Scud raketama, računalo je bilo u pogonu 100 sati – ukupna greška nastala greškom zaokruživanja je

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

Scud putuje brzinom 1676 m/s, pogreška u položaju veća od pola kilometra. □

Eksplozija Ariane 5. 37 sekundi nakon lansiranja izvršila je samouništenje. Razlog katastrofe: program je pokušao pretvoriti preveliki 64-bitni realni broj u 16-bitni cijeli broj. Računalo je javilo grešku, što je izazvalo samouništenje. □

Potonuće naftne platforme Sleipner A. Naftna platforma Sleipner A potonula je prilikom sidrenja. Razlog katastrofe: kod projektiranja platforme upotrijebljena metoda konačnih elemenata s nedovoljnom točnošću. □

Zaključak

Ponašanje rezultata se može predvidjeti — strah je nepotreban, a oprez nužan. Imamo li sumnju da je oduzimanjem brojeva došlo do kraćenja, svakako treba pokušati problem preformulirati ili provjeriti drugom metodom i/ili u višoj točnosti. Jasno je da treba poznavati i greške metode.