SVEUČILIŠTE U ZAGREBU
PMF – MATEMATIČKI ODJEL

Zlatko Drmač Vjeran Hari
Miljenko Marušić Mladen Rogina
Sanja Singer Saša Singer

Numerička matematika

Predavanja i vježbe

Zagreb, 2008.

Sadržaj

1. Mogućnosti današnjih računala

Možda nije najsretnije rješenje početi pričati o mogućnosti današnjih računala, kad se zna kojom se brzinom mijenjaju i ubrzavaju. Ipak, neke osnovne postavke ostat će nepromijenjene, bez obzira poveća li se broj osnovnih aritmetičkih operacija u sekundi koje računalo može izvoditi.

Često, ali pogrešno je mišljenje da se računalom mogu rješavati svi problemi — podaci se "ubace" u računalo, a ono nakon nekog vremena izbaci točan rezultat. Zaboravlja se na činjenicu da današnja računala nisu "inteligentna" (iako se tomu teži) i da su svi procesi u računalu vođeni ljudskom rukom.

Što se onda ipak može riješiti računalom? Mogu se riješiti svi problemi za koje postoji točno definiran, konačan postupak rješavanja — algoritam.

Što je algoritam? Prema definiciji Donalda Knutha, uvaženog stručnjaka za računarstvo, algoritam je konačan niz operacija koji rješava neki problem. Osim toga, algoritam mora zadovoljavati još i:

1. konačnost — za svaki ulaz, algoritam mora završiti nakon konačnog broja koraka;
2. točnu definiranost — u svakom koraku algoritma točno se zna sljedeći korak (nema slučajnosti);
3. algoritam može, ali i ne mora, imati ulazne podatke;
4. algoritam **mora** imati izlazne podatke;
5. efikasnost — svaki algoritam mora završiti u razumnom vremenu.

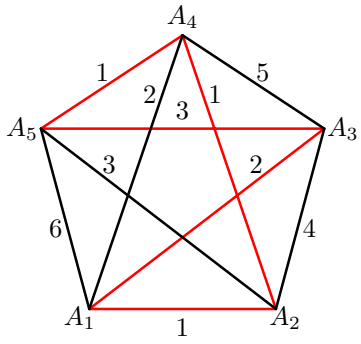
Možda je potrebno prokomentirati samo posljednja tri zahtjeva. Uobičajeno je da algoritmi imaju ulazne podatke, ali to nije nužno. Na primjer, ako želimo izračunati konstantu π nekim algoritmom, takav program vjerojatno neće zahtijevati ulazne podatke. Sasvim je obrnuta situacija s izlaznim podacima. Ako ih nema, algoritam je obavio neki posao za nas, ali nas nije izvjestio o krajnjem rezultatu. A što mi znamo o rješenju? Jednako kao da algoritam nije obavio nikakav posao!

1.1. Efikasni i neefikasni algoritmi

Od svih ovih zahtjeva koji se postavljaju na algoritam, najnerazumljiviji je zahtjev efikasnosti. Zahtjev efikasnosti znači da rješenje problema treba pronaći u razumnom vremenu.

Primjer 1.1.1. (Problem trgovačkog putnika) *Neka je zadano n gradova, tako da su svaka dva vezana cestom. Također, zadane su cijene putovanja. Trgovački putnik kreće iz grada A_1, obilazi sve ostale gradove samo jednom i vraća se ponovno u A_1 (zatvori ciklus). Ako gradovi nemaju cestu koja ih direktno povezuje, za cijenu puta između ta dva grada možemo staviti ∞, odnosno, u praktičnoj realizaciji, neki dovoljno veliki broj. Cilj trgovačkog putnika je naći ciklus najmanje cijene.*

Na primjer, za problem trgovačkog putnika



lako je provjeriti da je ciklus najmanje cijene $(A_1, A_3, A_5, A_4, A_2, A_1)$ ili, naravno, obratan ciklus $(A_1, A_2, A_4, A_5, A_3, A_1)$, i da mu je cijena 8 jedinica.

Algoritam za egzaktno rješavanje ovog problema je ispitivanje svih ciklusa, a njih ima $(n-1)!$. Objašnjenje za broj ciklusa je jednostavno. Ako krećemo iz grada A_1, drugi grad u koji stižemo možemo odabrati na $n-1$ načina, treći grad možemo izabrati na $n-2$ načina (jer se ne smijemo vratiti u početni ili ostati u A_2), ...

Računanje cijene odgovarajućeg ciklusa zahtijeva n zbrajanja. Prema tome, za rješenje problema trgovačkog putnika potrebno je približno $n!$ računskih operacija.

Izračunajmo koliko bi trajalo egzaktno rješavanje problema trgovačkog putnika za 10, 20, 30, 40 i 50 gradova. Pretpostavimo da nam je na raspolaganju najmoder-

nije PC računalo koje izvodi reda veličine 10^8 *računskih operacija u sekundi.*

n	*sekundi*	*sati*	*dana*	*godina*
10	$3.6288 \cdot 10^{-02}$	$1.0080 \cdot 10^{-05}$	$4.2000 \cdot 10^{-07}$	$1.1507 \cdot 10^{-09}$
20	$2.4329 \cdot 10^{+10}$	$6.7581 \cdot 10^{+06}$	$2.8159 \cdot 10^{+05}$	$7.7147 \cdot 10^{+02}$
30	$2.6525 \cdot 10^{+24}$	$7.3681 \cdot 10^{+20}$	$3.0701 \cdot 10^{+19}$	$8.4111 \cdot 10^{+16}$
40	$8.1592 \cdot 10^{+39}$	$2.2664 \cdot 10^{+36}$	$9.4435 \cdot 10^{+34}$	$2.5873 \cdot 10^{+32}$
50	$3.0414 \cdot 10^{+56}$	$8.4484 \cdot 10^{+52}$	$3.5201 \cdot 10^{+51}$	$9.6442 \cdot 10^{+48}$

Uočite da, osim egzaktnog rješenja problema za 10 *gradova, ostali problemi nisu rješivi u razumnom vremenu, jer već za* $n = 20$*, za rješenje problema treba čekati* 771 *godinu!*

Moderna znanost pretpostavlja da je Zemlja stara oko 4.5 *milijarde godina (tj.* $4.5 \cdot 10^9$ *godina), a rješavanje problema za* $n = 30$ *gradova trajalo bi sedam redova veličine dulje.*

Kad bismo na raspolaganju imali neko od danas najbržih, paralelnih računala, koje izvodi približno 10^{13} *računskih operacija u sekundi, brojke u prethodnoj tablici bile bi* 10^5 *puta manje. U tom slučaju, jedino još kako–tako smisleno bilo bi čekati* 2.8 *dana za rješenje problema s* 20 *gradova.*

Što nam prethodni primjer pokazuje? Pokazuje da ne bismo trebali problem trgovačkog putnika rješavati egzaktno, ali nipošto ne kaže da ga uopće ne bismo trebali rješavati! Postoje algoritmi koji dobro (i relativno brzo) približno rješavaju ovaj problem.

Koja je posebnost egzaktnog algoritma za rješavanje problema trgovačkog putnika? Naime, ako imamo problem dimenzije n (tj. n gradova), vrijeme traženja rješenja (ili broj potrebnih aritmetičkih operacija) **eksponencijalno raste** u ovisnosti o n, što slijedi iz nejednakosti

$$n^{n/2} \leq n! \leq n^n.$$

Desna nejednakost je trivijalna. Dakle, ostaje nam pokazati samo prvu. Ako pokažemo da vrijedi

$$k(n-k+1) \geq n, \quad k \in \{1, \ldots, n\}, \tag{1.1.1}$$

onda produkt $1 \cdot 2 \cdots n$ možemo organizirati tako da množimo prvi i posljednji član, drugi i pretposljednji, i tako redom. Takvih parova produkata ima $n/2$, pa je

rezultat očit. Dakle, preostaje pokazati samo relaciju (1.1.1). Prebacivanjem člana n na lijevu stranu slijedi

$$(k-1)n - k^2 + k = (k-1)(n-k) \geq 0,$$

što je istina upravo za $k \in \{1, \ldots, n\}$.

Mnogi problemi koje rješavamo računalom imaju složenost koja ne ovisi eksponencijalno o veličini problema n, nego polinomno, tj. broj aritmetičkih operacija proporcionalan je n^α, gdje je α neka mala konstanta (uobičajeno je $\alpha \leq 3$).

Vrlo je zanimljiv i problem prognoze vremena. On nam na suptilan način pokazuje što efikasnost znači u tom slučaju.

Primjer 1.1.2. (Prognoza vremena) *Najjednostavniji klimatski model funkcija je* 4 *argumenta: zemljopisne širine, dužine, visine (od tla) i vremena. Kao rezultat dobivamo* 6 *vrijednosti: temperaturu, tlak, vlažnost i brzinu vjetra (komponente u* 3 *smjera). Generalniji model mogao bi uključivati, na primjer, koncentraciju različitih plinova u atmosferi. Stvarni model atmosferskih procesa uključivao bi i stvaranje oblaka, količinu padalina, kemiju i sl.*

Primijetimo da se klima neprekidno mijenja (u ovisnosti o u svoje četiri varijable), ali je računalom možemo simulirati samo u ponekim (diskretnim) točkama.

Pretpostavimo da smo površinu Zemlje podijelili u (približne) kvadrate stranice $1\,\mathrm{km}$*. Po visini, također, uzimamo slojeve debljine* $1\,\mathrm{km}$*, do visine* $10\,\mathrm{km}$*. Stanje klime računamo samo u vrhovima kvadrata i točkama na slojevima iznad vrhova. Iz površine Zemlje (*$\approx 5.1 \cdot 10^8\,\mathrm{km}^2$*) slijedi da je ukupan broj takvih točaka približno jednak* $5 \cdot 10^9$*.*

Nadalje, klimu simuliramo u vremenskim trenucima s razmakom 1 *minute. Za svaki vremenski trenutak i svaku prostornu točku moramo pamtiti* 6 *vrijednosti koje opisuju stanje klime. Uzmimo da je svaka od njih realan broj koji se prikazuje s* 4 *byte-a. Onda je za pamćenje svih vrijednosti u jednom trenutku potrebno*

$$6 \cdot 4 \cdot 5 \cdot 10^9 \approx 10^{11}\,\mathrm{B} = 0.1\,\mathrm{TB}$$

memorije.

Simulacija klime napreduje po vremenu, tj. klima u sljedećem trenutku se računa iz stanja klime u nekoliko prethodnih trenutaka. Standardno se stanje klime u određenoj prostornoj točki računa iz stanja u nekoliko susjednih točaka. Pretpostavimo da nam za taj proračun treba samo 100 *osnovnih aritmetičkih operacija (flopova) po svakoj točki, što ukupno daje* $100 \cdot 5 \cdot 10^9 = 5 \cdot 10^{11}$ *flopova.*

Jasno je da za predviđanje vremena za sljedeću minutu ne smijemo potrošiti više od 1 *minute vremena rada računala (inače je brže i jeftinije pogledati kroz prozor). Dakle, računalo mora imati brzinu od najmanje*

$$5 \cdot 10^{11}\,\text{flopova}/60\,\text{sekundi} \approx 8 \cdot 10^9\,\text{flopsa} = 8\,\text{Gflopsa},$$

tj. preko 8 *milijardi aritmetičkih operacija u sekundi.*

Ako želimo dobiti globalnu prognozu vremena za samo 1 *dan unaprijed, uz dozvoljeno vrijeme računanja od* 2 *sata (tako da ostane još* 22 *sata vrijednosti te prognoze), računalo mora biti još barem* 12 *puta brže, tj. brzina mora biti barem* 100 Gflopsa *ili* 0.1 Tflopsa.

2. Greške

2.1. Računalo je izbacilo . . . Neće mi prihvatiti!

Koliko ste puta ove dvije rečenice čuli na TV-u, u banci ili negdje drugdje? U oba slučaja, računalo je personificirano kao nadnaravno sposobna osoba, koja je u prvom slučaju bezgrešna, a u drugom odbija nešto učiniti!

Treba li takvim tužbalicama vjerovati? I tko je krivac? Računalo ili ljudi koji su mu naredili da se upravo tako ponaša? Istina je da smo u 2001. godini, ali vaše računalo nije (svojeglavi) HAL 9000 iz knjige ili filma "2001. odiseja u svemiru" (a i njemu su ljudi pomogli da postane ubojica).

Navedene dvije rečenice najčešće su samo jadno pokriće nečije nesposobnosti. Službeniku za šalterom u banci vjerojatno treba oprostiti, jer on samo izražava svoju bespomoćnost, a pravi krivac je negdje daleko.

Puno je opasnije kad čujete "Računalo je izbacilo . . . " od strane inženjera i znanstvenika kao glavno opravdanje budućih važnih projekata. Tad nas hvata strah! Zašto? Sama rečenica pokazuje da dobivene rezultate nitko nije pogledao, nego da im se slijepo vjeruje. A oni mogu biti pogrešni iz razno-raznih razloga, a najčešći krivac nije računalo.

Iz osobnog iskustva znamo da je provjera dobivenih rezultata najbolnija točka nastave numerike, u koju je slušače najteže uvjeriti. Metode je manje-više lako naučiti. U praksi, brojevi uvijek imaju neko značenje, izvora grešaka je puno — javljaju se na svakom koraku, a analiza grešaka u rezultatima katkad je vrlo sofisticirana. Slijepo vjerovanje rezultatima može biti pogibeljno.

Izvori grešaka su:

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

Sve četiri vrste grešaka lako je razumjeti. Međutim, za posljednju, vrlo je teško vjerovati da ona može biti toliko značajna — dominantna u odnosu na ostale, tako da je rezultat zbog nje besmislen.

No, ipak treba priznati da i računala, iznimno rijetko, ali ipak griješe. Koliko je nama poznato, u novija vremena poznata je samo greška dijeljenja u jednoj seriji Pentium procesora (1994. godine).

2.2. Mjere za grešku

Prije detaljnijeg opisa pojedinih vrsta ili uzroka grešaka, moramo preciznije reći što je greška i kako se ona uobičajeno mjeri.

Neka je x neki realni broj, kojeg smatramo "točnim". Ako je $\widehat{x}$ neka njegova aproksimacija ili "približna vrijednost", onda grešku te aproksimacije definiramo kao

$$\text{greška} = E(x, \widehat{x}) := x - \widehat{x},$$

tako da je $x = \widehat{x} + \text{greška}$. Ovako definirana greška ima predznak i može biti negativna. U praksi nam, obično, predznak nije bitan kad govorimo o *točnosti* ove aproksimacije. Najkorisnije mjere za točnost ili grešku aproksimacije $\widehat{x}$ za x su:

- **apsolutna greška**
$$E_{\text{abs}}(x, \widehat{x}) := |x - \widehat{x}|,$$
koja mjeri stvarnu udaljenost (u smislu metrike na $\mathbb{R}$) brojeva x i $\widehat{x}$, i
- **relativna greška**, definirana za $x \neq 0$,
$$E_{\text{rel}}(x, \widehat{x}) := \frac{|x - \widehat{x}|}{|x|},$$
koja mjeri relativnu točnost obzirom na veličinu broja x, na primjer, u smislu podudaranja "prednjih dijelova", tj. određenog broja vodećih znamenki brojeva x i $\widehat{x}$.

Ako aproksimaciju prikažemo u obliku $\widehat{x} = x(1 + \rho)$, onda dobivamo ekvivalentnu definiciju relativne greške u obliku

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

Baš ovaj oblik se često koristi u analizi grešaka aritmetike računala. Ako želimo da relativna greška ima predznak, možemo ispustiti apsolutne vrijednosti iz definicije.

Na isti način možemo definirati i greške za kompleksne brojeve. Za teorijske potrebe to bi bilo dovoljno. Međutim, kao što ćemo vidjeti, kompleksne brojeve u računalu prikazujemo kao par realnih brojeva, tj. kao vektor s dvije realne komponente. Za mjerenje grešaka u vektorima i matricama, umjesto apsolutne vrijednosti, koristimo pojam norme, kojeg opisujemo u sljedećem poglavlju.

2.3. Greške modela

Najčešće greške modela nastaju zanemarivanjem utjecaja nekih sila (na primjer, zanemarivanje utjecaja otpora zraka). Jednako tako, da bi se dobilo nekakvo rješenje, barem približno, često se kompliciran model zamjenjuje jednostavnijim (na primjer, sistemi nelinearnih parcijalnih diferencijalnih jednadžbi se linearizira ju).

Također, pogreške u modelu mogu nastati kod upotrebe modela u graničnim slučajevima. Na primjer, kod matematičkog njihala se $\sin x$ aproksimira s x, što vrijedi samo za male kuteve, a upotrebljava se, recimo, za kut od 15°.

Primjer 2.3.1. *Među prvim primjenama trenutno jednog od najbržih računala na svijetu bilo je određivanje trodimenzionalne strukture i elektronskog stanja ugljik-*36 *fulerena (engl. carbon-*36 *fullerene) — jednog od najmanjih, ali i najstabilnijih članova iz redova jedne vrste spojeva (engl. buckminsterfullerene). Primjena tog spoja može biti višestruka, od supravodljivosti na visokim temperaturama do preciznog doziranja lijekova u stanice raka.*

Prijašnja istraživanja kvantnih kemičara dala su dvije moguće strukture tog spoja. Eksperimentalna mjerenja pokazivala su da bi jedna struktura trebala biti stabilnija, a teoretičari su tvrdili da bi to trebala biti druga struktura. Naravno, te dvije strukture imaju različita kemijska svojstva. Prijašnja računanja, zbog pojednostavljivanja i interpolacije, kao odgovor davala su prednost "teoretskoj" strukturi. Definitivan odgovor, koji je proveden računanjem bez pojednostavljivanja pokazao je da je prva struktura stabilnija.

Svakako, treba istaknuti da su pogreške modela neuklonjive, ali zato je na inženjerima — korisnicima da procijene da li se primjenom danog modela dobivaju očekivani rezultati.

2.4. Greške u ulaznim podacima

Greške u ulaznim podacima javljaju se zbog nemogućnosti ili besmislenosti točnog mjerenja. Na primjer, tjelesna temperatura se obično mjeri na desetinku stupnja Celziusa točno — pacijentu je jednako loše ako ima tjelesnu temperaturu 39.5° ili 39.513462°.

Naravno, kao što ćemo to kasnije vidjeti, osim tih malih pogrešaka nastalih mjerenjem, dodatne greške nastaju spremanjem tih brojeva u računalo.

Vezano uz pogreške u ulaznim podacima, često se javlja pojam nestabilnog ili loše uvjetovanog problema. U praksi se vrlo često javljaju takvi problemi kod kojih mala perturbacija početnih podataka dovodi do velike promjene u rezultatu. Kao ilustraciju možemo uzeti sljedeći primjer.

Primjer 2.4.1. *Zadana su dva sistema linearnih jednadžbi*

$$\begin{aligned} 2x + 6y &= 8 \\ 2x + 6.0001y &= 8.0001, \end{aligned} \tag{2.4.1}$$

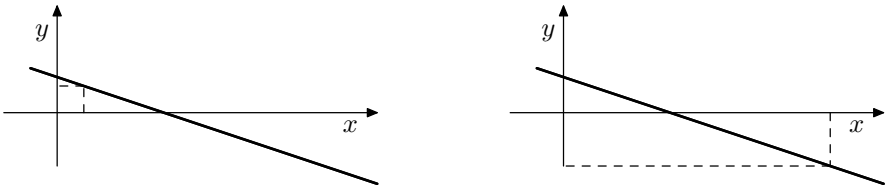
i

$$\begin{aligned} 2x + 6y &= 8 \\ 2x + 5.99999y &= 8.00002. \end{aligned} \tag{2.4.2}$$

Na prvi pogled, malom perturbacijom koeficijenata jednadžbe (2.4.1) *dobivamo jednadžbu* (2.4.2)*. Takva perturbacija mogla bi nastupiti, na primjer, malo pogrešno izmjerenim početnim podacima ili greškom koja je nastala u računu koeficijenata.*

Što očekujemo? Očekujemo da će i rješenje drugog problema biti malo perturbirano rješenje prvog problema. Ali nije tako! Rješenje prvog problema je $x = 1$*,* $y = 1$*, a drugog* $x = 10$*,* $y = -2$*! Zašto?*

U analizi će nam pomoći crtanje odgovarajućih pravaca i njihovih sjecišta.



Ali na prethodnim slikama nacrtan je samo po jedan pravac! Pogrešno! Ako malo bolje pogledamo koeficijente u oba sistema jednadžbi, vidimo da se u oba slučaja radi o dva pravca koja su gotovo jednaka! Sad nije niti čudo da mala perturbacija koeficijenata pravca bitno udaljava presjecište.

2.5. Greške metoda za rješavanje problema

Greške metoda za rješavanje problema često nastaju kad se beskonačni procesi moraju zaustaviti u konačnosti. To vrijedi za sve objekte koji su definirani limesom — poput derivacija i integrala, i za sve postupke u kojima se "pravo" rješenje dobiva na limesu — konvergencijom niza približnih rješenja prema pravom. Velik broj numeričkih metoda za aproksimaciju funkcija i rješavanje jednadžbi upravo je tog oblika. Greške koja nastaju zamjenom beskonačnog nečim konačnim, obično dijelimo u dvije kategorije:

- **greške diskretizacije** (engl. discretization errors), koje nastaju zamjenom kontinuuma konačnim diskretnim skupom točaka, ili "beskonačno" malu veličinu h ili $\varepsilon \to 0$ zamijenjujemo nekim "konačno" malim brojem;

- **greške odbacivanja** (engl. truncation errors), koje nastaju "rezanjem" beskonačnog niza ili reda na konačni niz ili sumu, tj. odbacujemo ostatak niza ili reda.

Grubo rečeno, diskretizacija je vezana za kontinuum, a odbacivanje za diskretnu beskonačnost, poput razlike između skupova $\mathbb{R}$ i $\mathbb{N}$.

Pojam diskretizacije smo već susreli u problemu prognoze vremena. Još jednostavniji, tipični primjer greške diskretizacije je aproksimacija funkcije f na intervalu (segmentu) $[a, b]$, vrijednostima te funkcije na konačnom skupu točaka (tzv. mreži) $\{x_1, \ldots, x_n\} \subset [a, b]$, o čemu će još biti mnogo riječi.

Drugi klasični primjer je 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 neki dovoljno mali $h \neq 0$ i izračunamo kvocijent

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

Postoje i bolje aproksimacije za $f'(x)$, ali o tome kasnije. Uskoro ćemo vidjeti da s ovom vrstom aproksimacija limesa treba vrlo oprezno postupati u aritmetici računala.

Pogledajmo malo i greške odbacivanja. Na primjer, beskonačna suma se mora zamijeniti konačnom, da bi se uopće mogla izračunati u konačnom vremenu.

Primjer 2.5.1. *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 $\sin x$.*

Ako to napravimo računalom, rezultat će biti zanimljiv. Greška metode (tj. greška odbacivanja) lako se računa. Za dovoljno glatku funkciju f, Taylorov red možemo aproksimirati Taylorovim polinomom

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

pri čemu je ξ neki broj između 0 i x. Traženi Taylorovi polinomi s istim brojem članova (ali ne istog stupnja) su

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

Nađimo grešku. Vrijedi

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

pa su pripadne greške odbacivanja

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

Radi jednostavnosti, pretpostavimo da je $x > 0$. *Iz* $\xi \leq x$ *dobivamo ocjene*

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

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*

$$\begin{cases} e^{x}\varepsilon, & za\ e^{x}, \\ \varepsilon, & za\ \sin x. \end{cases} \tag{2.5.1}$$

Izračunajmo $\sin(15\pi)$, $e^{15\pi}$, $\sin(25\pi)$ *i* $e^{25\pi}$ *korištenjem Taylorovog reda oko* 0 *u najvećoj direktno podržanoj preciznosti (tzv.* `extended` *preciznosti). Primjer je izabran tako da je* $\sin(15\pi) = \sin(25\pi) = 0$, *dok su druga dva broja vrlo velika. Prema* (2.5.1), *greška metode za računanje je u slučaju funkcije* e^{x} *relativno mala, a u slučaju funkcije* $\sin x$, *mala po apsolutnoj vrijednosti.*

Izaberemo li $\varepsilon = 10^{-17}$, *dobivamo (napisano je samo prvih par znamenki rezultata)*

$$\begin{aligned}
\sin(15\pi)_{funkcija} &= 9.3241 \cdot 10^{-18} & \exp(15\pi)_{funkcija} &= 2.9218 \cdot 10^{20} \\
\sin(15\pi)_{Taylor} &= -2.8980 \cdot 10^{-1} & \exp(15\pi)_{Taylor} &= 2.9218 \cdot 10^{20} \\
|greška\ odbacivanja| &\leq 2.7310 \cdot 10^{-19} & |greška\ odbacivanja| &\leq 2.7600 \cdot 10^{2} \\
relativna\ greška &= 3.1081 \cdot 10^{16} & relativna\ greška &= 1.4238 \cdot 10^{-18} \\
|maksimalni\ član| &= 1.6969 \cdot 10^{19} & |maksimalni\ član| &= 1.6969 \cdot 10^{19}
\end{aligned}$$

$$\begin{aligned}
\sin(25\pi)_{funkcija} &= 1.6697 \cdot 10^{-17} & \exp(25\pi)_{funkcija} &= 1.2865 \cdot 10^{34} \\
\sin(25\pi)_{Taylor} &= 3.0613 \cdot 10^{13} & \exp(25\pi)_{Taylor} &= 1.2865 \cdot 10^{34} \\
|greška\ odbacivanja| &\leq 5.8309 \cdot 10^{-19} & |greška\ odbacivanja| &\leq 2.3782 \cdot 10^{16} \\
relativna\ greška &= 1.8334 \cdot 10^{30} & relativna\ greška &= 7.0013 \cdot 10^{-19} \\
|maksimalni\ član| &= 5.7605 \cdot 10^{32} & |maksimalni\ član| &= 5.7943 \cdot 10^{32}.
\end{aligned}$$

Ako smo rezultat zbrajanja Taylorovog reda za $\sin(15\pi)$ *spremni prihvatiti kao približno točan, sigurno nije istina da je* $\sin(25\pi) \approx 3 \cdot 10^{13}$. *Što se, zapravo, dogodilo? Objašnjenje leži u aritmetici računala.*

2.6. Greške aritmetike računala

U računalu postoje dva bitno različita tipa brojeva: cijeli brojevi i 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 se brojevi prikazaju korištenjem n bitova — binarnih znamenki, od kojih jedna služi za predznak, a ostalih $n-1$ za znamenke broja. Matematički gledano, aritmetika cijelih brojeva u računalu je modularna aritmetika u prstenu ostataka modulo 2^n, samo je sistem ostataka simetričan oko 0, tj.

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

Brojeve izvan tog raspona uopće ne možemo spremiti u računalo.

Realni brojevi r prikazuju se korištenjem mantise m i eksponenta e u obliku

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

pri čemu je e cijeli broj u određenom rasponu, a m racionalni broj za koji vrijedi $1/2 \leq m < 1$ (tj. mantisa započinje s $0.1\ldots$). Često se i ta vodeća jedinica ne pamti, jer se zna da su brojevi normalizirani, pa se mantisa "umjetno" može produljiti za taj jedan bit. Taj bit se katkad zove "skriveni bit" (engl. hidden bit). Za $r = 0$, mantisa je 0. U računalu se eksponent prikazuje kao s-bitni cijeli broj, a za mantisu pamti se 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

Dakle, skup svih realnih brojeva prikazivih u računalu je omeđen, a možemo ga parametrizirati duljinom mantise i eksponenta i označiti s $\mathbb{R}(t, s)$.

Primijetite da se ne može svaki realni broj egzaktno spremiti u računalo. Pretpostavimo da je broj $x \in \mathbb{R}$ unutar prikazivog raspona i

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

Ako mantisa broja ima više od t znamenki, bit će spremljena aproksimacija tog broja $f\ell(x) \in \mathbb{R}(t, s)$ koja se može prikazati kao

$$f\ell(x) = \pm\Big(\sum_{k=1}^{t} b^*_{-k} 2^{-k}\Big) 2^{e^*}.$$

Slično kao kod decimalne aritmetike (kad je prva odbačena znamenka ≤ 4 zaokružujemo nadolje, inače nagore), ako je prva odbačena znamenka 1, broj zaokružujemo

nagore, a ako je 0 nadolje. Time smo napravili apsolutnu grešku manju ili jednaku 2^{-t-1+e}. 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.

Dakle, ako je $x \in \mathbb{R}$ unutar raspona brojeva prikazivih u računalu, onda se umjesto x sprema zaokruženi broj $f\ell(x) \in \mathbb{R}(t, s)$ i vrijedi

$$f\ell(x) = (1 + \varepsilon)x, \quad |\varepsilon| \leq u, \tag{2.6.1}$$

gdje je ε relativna greška napravljena tim zaokruživanjem.

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

Da bismo stekli osjećaj o veličinama o kojim govorimo, napišimo s i t i veličine koje se iz njih izvode za standardne tipove realnih brojeva u računalu:

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

Za sva tri tipa u ukupnoj duljini rezerviran je još jedan bit za predznak. Kod tipova `single` i `double` dodatni bit u duljini mantise je tzv. "sakriveni bit" (engl. hidden bit), jer je prvi znak iza binarne točke uvijek 1, pa se ne mora pamtiti. To je dogovoreni IEEE standard u kojem je propisano, ne samo kako se brojevi prikazuju u računalu, nego i svojstva aritmetike. Budimo pošteni, taj standard je nešto složeniji nego što smo to ovdje opisali. Međutim, ti dodatni detalji, poput posebnih prikaza za $\pm\infty$ i "rezultat nedozvoljene operacije" (engl. NaN, Not-a-Number), ili "zaštitne znamenke" (engl. guard digit) u aritmetici, nepotrebno zamagljuju bitno.

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

$$f\ell(x \circ y) = (1 + \varepsilon)\,(x \circ y), \quad |\varepsilon| \leq u, \tag{2.6.2}$$

za sve $x, y \in \mathbb{R}(t, s)$ za koje je $x \circ y$ u dozvoljenom rasponu. Naravno, dobiveni rezultat je tada prikaziv, tj. vrijedi $f\ell(x \circ y) \in \mathbb{R}(t, s)$. U protivnom, postoje rezervirani eksponenti koji označavaju "posebno stanje" (overflow, underflow, dijeljenje s 0 i nedozvoljenu operaciju kao što su $0/0$, $\sqrt{-1}$).

Oznaka $fl(\ )$ sad ima značenje rezultata dobivenog računalom za operaciju $x \circ y$. Ovaj model kaže da je izračunata vrijednost za $x \circ y$ "jednako dobra" kao i zaokružen egzaktni rezultat, u smislu da je u oba slučaja jednaka ocjena relativne greške. Model, međutim, ne zahtijeva da za egzaktno prikazivi egzaktni rezultat $x \circ y \in \mathbb{R}(t, s)$ mora vrijediti da je greška $\varepsilon = 0$, tj.

$$x \circ y \in \mathbb{R}(t, s) \not\Longrightarrow fl(x \circ y) = x \circ y.$$

U tom smislu, korištenje oznake $fl(x \circ y)$ za izračunati rezultat nije sasvim korektno, jer to ne mora biti zaokruženi egzaktni rezultat. Preciznije bi bilo uvesti posebne oznake $\oplus$, $\ominus$, $\otimes$ i $\oslash$ za strojne aritmetičke operacije i analizirati njihova svojstva. Takve analize postoje, ali su izuzetno komplicirane, jer većina standardnih svojstava aritmetičkih operacija, poput asocijativnosti zbrajanja ili distributivnosti množenja prema zbrajanju, **ne** vrijedi za aritmetiku realnih brojeva u računalu. Može se pokazati da vrijede neka bitno složenija "zamjenska" svojstva. Međutim, njih je nemoguće praktično iskoristiti za analizu ukupnih grešaka računanja u bilo kojem algoritmu s iole većim brojem aritmetičkih operacija.

Upravo zbog toga, standard za aritmetiku računala propisuje samo da mora vrijediti relacija (2.6.2), a ne neka druga složenija svojstva. Vidimo da greška svake pojedine aritmetičke operacije i njena ocjena u (2.6.2) imaju isti oblik kao i greška zaokruživanja za prikaz brojeva u računalu i njena ocjena iz (2.6.1). Zato obje vrste grešaka (greške prikaza i greške aritmetike) zajedničkim imenom zovemo greškama zaokruživanja (engl. rounding errors).

Objasnimo još točno značenje oznake $fl(\ )$. Jednostavno, fl(izraz) označava **izračunatu** vrijednost tog izraza u floating point aritmetici. U skladu s tim, ako se izraz sastoji samo od jednog broja x, onda se "računanje" vrijednosti tog izraza x svodi na zaokruživanje i spremanje u floating point prikazu, a $fl(x)$ označava spremljenu "izračunatu", tj. zaokruženu vrijednost. Analogno, $fl(x \circ y)$ sad korektno označava izračunati rezultat operacije $x \circ y$. Takva interpretacija znatno olakšava zapis u analizi grešaka zaokruživanja.

2.7. Propagiranje grešaka u aritmetičkim operacijama

Desnu stranu relacije (2.6.2) možemo interpretirati i kao egzaktno izvedenu operaciju $\circ$ na malo perturbiranim podacima. Koje su operacije opasne ako nam je aritmetika egzaktna, a podaci malo perturbirani, tj. ako je $|\varepsilon_x|$, $|\varepsilon_y| \leq u$? Ako je $\circ$ množenje, imamo

$$x\,(1 + \varepsilon_x) * y\,(1 + \varepsilon_y) \approx xy\,(1 + \varepsilon_x + \varepsilon_y) := xy(1 + \varepsilon_*), \quad |\varepsilon_*| \leq 2u.$$

Ako imamo dijeljenje, vrijedi slično

$$\frac{x\,(1+\varepsilon_x)}{y\,(1+\varepsilon_y)} \approx \frac{x}{y}\,(1+\varepsilon_x)\,(1-\varepsilon_y) := \frac{x}{y}\,(1+\varepsilon_/), \quad |\varepsilon_/| \leq 2u.$$

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

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

uz pretpostavku da $x + y \neq 0$. Definiramo

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

Ako su x i y brojevi istog predznaka, onda je

$$\left|\frac{x}{x+y}\right|,\ \left|\frac{y}{x+y}\right| \leq 1, \tag{2.7.1}$$

pa je $|\varepsilon_\pm| \leq 2u$. U suprotnom, ako x i y imaju različite predznake, kvocijenti u (2.7.1) mogu biti proizvoljno veliki kad je $|x+y| \ll |x|, |y|$.

Možemo zaključiti da **opasnost** nastupa ako je rezultat zbrajanja brojeva suprotnog predznaka broj koji je po apsolutnoj vrijednosti mnogo manji od polaznih podataka. Dakle, čak i kad bi aritmetika računala bila egzaktna, zbog početnog zaokruživanja, rezultat može biti (i najčešće je) pogrešan.

Pokažimo to na jednostavnom primjeru računala u bazi 10. Pretpostavimo da je mantisa duga 4 dekadske znamenke, a eksponent dvije. Neka je

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

Umjesto brojeva x i y, spremili smo najbliže prikazive, tj.

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

i napravili malu relativnu grešku. Budući da su im eksponenti jednaki, možemo oduzeti znamenku po znamenku mantise, a zatim normalizirati i dobiti

$$fl(x) - fl(y) = 0.8887 \cdot 10^0 - 0.8884 \cdot 10^0 = 0.0003 \cdot 10^0 = 0.3??? \cdot 10^{-3},$$

pri čemu upitnici predstavljaju znamenke koje više ne možemo restaurirati, pa računalo na ta mjesta upisuje 0. Primijetimo da je pravi rezultat $0.22 \cdot 10^{-3}$, pa je već prva značajna znamenka pogrešna, a relativna greška velika!

Iako je sama operacija oduzimanja bila egzaktna za $fl(x)$ i $fl(y)$, rezultat je pogrešan. Na prvi pogled čini nam se da znamo bar red veličine rezultata i da to nije tako strašno. Prava katastrofa nastupa ako $0.3??? \cdot 10^{-3}$ uđe u naredna zbrajanja i oduzimanja i ako se pritom "skrati" i ta trojka. Tada nemamo nikakve kontrole nad rezultatom.

Primjer 2.7.1. *Objašnjenje pogrešnih rezultata iz primjera* 2.5.1. *sad je sasvim jednostavno. Pogledamo li po apsolutnoj vrijednosti najveće brojeve koji se javljaju u računu, ustanovljavamo da su oni golemi. Za* $\sin x$*, rezultat je malen broj koji je dobiven oduzimanjem velikih brojeva, pa je netočan. Nasuprot tome, kod funkcije* e^x*, uvijek imamo zbrajanje brojeva istog predznaka, pa je rezultat točan.*

Primjer 2.7.2. *U algoritmima se često javlja potreba za izračunavanjem vrijedosti* $y = \sqrt{x+\delta} - \sqrt{x}$ *uz* $|\delta| \ll x$*,* $x > 0$*. Da bismo izbjegli katastrofalno kraćenje,* y *se nikad ne računa po napisanoj formuli. Uvijek se pribjegava deracionalizaciji, tj.*

$$y = (\sqrt{x+\delta} - \sqrt{x}) \cdot \frac{\sqrt{x+\delta} + \sqrt{x}}{\sqrt{x+\delta} + \sqrt{x}} = \frac{\delta}{\sqrt{x+\delta} + \sqrt{x}}.$$

Uočite da je opasno oduzimanje zamijenjeno benignim dijeljenjem i zbrajanjem.

Primjer 2.7.3. *Zbrajanje brojeva računalom nije asocijativno. Izračunajmo*

$$S_1 = \sum_{i=1}^{10000} \frac{1}{i}, \qquad S_2 = \sum_{i=10000}^{1} \frac{1}{i}$$

u tri točnosti. Dobiveni rezultati su:

	`single`	`double`	`extended`
S_1	9.78761291503906250	9.78760603604434465	9.78760603604438230
S_2	9.78760433197021484	9.78760603604438550	9.78760603604438227

Primijetite da nešto točniji rezultat daje zbrajanje S_2*. Objasnite.*

Primjer 2.7.4. *Niti poredak operacija ne mora biti beznačajan. Zadan je linearni sustav*

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

Matrica sustava je regularna, pa postoji jedinstveno rješenje $x_1 = 1.\dot{0}00\dot{1}$*,* $x_2 = 0.\dot{9}99\dot{8}$*. Rješavamo li taj sustav 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*

$$\begin{aligned} 0.1000 \cdot 10^{-3}\, x_1 + 0.1000 \cdot 10^{1}\, x_2 &= 0.1000 \cdot 10^{1} \\ 0.1000 \cdot 10^{1}\ \ x_1 + 0.1000 \cdot 10^{1}\, x_2 &= 0.2000 \cdot 10^{1}. \end{aligned} \tag{2.7.2}$$

Prvo, riješimo sustav (2.7.2) *Gaussovim eliminacijama. Množenjem prve jednadžbe s* 10^4 *i oduzimanjem od druge, dobivamo drugu jednadžbu oblika:*

$$(0.1000 \cdot 10^{1} - 0.1000 \cdot 10^{5})\, x_2 = 0.2000 \cdot 10^{1} - 0.1000 \cdot 10^{5}. \tag{2.7.3}$$

Da bi računalo moglo oduzeti odgovarajuće brojeve, manji eksponent mora postati jednak većem, a mantisa se denormalizira. Dobivamo

$$0.1000 \cdot 10^1 = 0.0100 \cdot 10^2 = 0.0010 \cdot 10^3 = 0.0001 \cdot 10^4 = 0.0000|1 \cdot 10^5,$$

ali za zadnju jedinicu nema mjesta u mantisi, pa je mantisa postala 0*. Slično je i s desnom stranom. Zbog toga jednadžba* (2.7.3) *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:*

$$0.1000 \cdot 10^{-3}\, x_1 = -0.1000 \cdot 10^1 \cdot 0.1000 \cdot 10^1 + 0.1000 \cdot 10^1 = 0.0000,$$

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

Promijenimo li poredak jednadžbi u (2.7.2)*, dobivamo*

$$\begin{aligned} 0.1000 \cdot 10^1 \quad x_1 + 0.1000 \cdot 10^1\, x_2 &= 0.2000 \cdot 10^1 \\ 0.1000 \cdot 10^{-3}\, x_1 + 0.1000 \cdot 10^1\, x_2 &= 0.1000 \cdot 10^1. \end{aligned} \tag{2.7.4}$$

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

$$(0.1000 \cdot 10^1 - 0.1000 \cdot 10^{-3})\, x_2 = 0.1000 \cdot 10^1 - 0.2000 \cdot 10^{-3}, \tag{2.7.5}$$

pa se (2.7.5) *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* (2.7.4) *dobivamo*

$$0.1000 \cdot 10^1\, x_1 = 0.2000 \cdot 10^1 - 0.1000 \cdot 10^1 \cdot 0.1000 \cdot 10^1 = 0.1000 \cdot 10^1,$$

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

Primjer 2.7.5. *Poznato je da studenti kad ne znaju izračunati limese, posežu za kalkulatorom i pokušavaju ih "heuristički" izračunati, tako da se približavaju granici. Pretpostavimo da imaju program koji u* `extended` *točnosti računa sljedeće limese:*

$$\lim_{x \to 0} \frac{1 - \cos x}{x^2} = \frac{1}{2}, \qquad \lim_{x \to 0} \frac{x^2}{1 - \cos x} = 2.$$

Može li se unaprijed reći što će se događati s rezultatima ako stavljamo x*-eve redom*

$10^{-1}, 10^{-2}, \ldots, 10^{-10}$*? Objasnite dobivene rezultate.*

x	`prvi ‘‘limes’’`	`drugi ‘‘limes’’`
10^{-01}	0.4995834722	2.001667500
10^{-02}	0.4999958333	2.000016667
10^{-03}	0.4999999583	2.000000167
10^{-04}	0.4999999996	2.000000002
10^{-05}	0.5000000002	1.999999999
10^{-06}	0.4999999980	2.000000008
10^{-07}	0.5000015159	1.999993936
10^{-08}	0.4998172015	2.000731461
10^{-09}	0.4336808690	2.305843009
10^{-10}	0.0000000000	

2.8. Primjeri iz života

Primjeri koje smo naveli u prethodnom odjeljku su pravi, školski. Nažalost, postoje primjeri kad su zbog grešaka zaokruživanja stradali ljudi ili je počinjena velika materijalna šteta.

Primjer 2.8.1. (Promašaj raketa Patriot.)

U Zaljevskom ratu, 25. veljače 1991. godine, Patriot rakete iznad Dhahrana u Saudijskoj Arabiji nisu uspjele pronaći i oboriti iračku Scud raketu. Projektil je (pukim slučajem) pao na američku vojnu bazu usmrtivši 28 *i ranivši stotinjak ljudi.*

Izvještaj o katastrofi godinu dana poslije, rasvijetlio je okolnosti nesreće. U računalu koje je upravljalo Patriot raketama, vrijeme se brojilo u desetinkama sekunde proteklim od trenutka kad je računalo upaljeno. Kad desetinku prikažemo u binarnom prikazu, dobivamo

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

Realne brojeve u tom računalu prikazivali su korištenjem nenormalizirane mantise duljine 23 *bita. Spremanjem* 0.1 *u registar Patriot računala, napravljena je (apsolutna) greška približno jednaka* $9.5 \cdot 10^{-8}$*.*

Zbog stalne opasnosti od napada Scud raketama, računalo je bilo u pogonu 100 *sati, što je* $100 \cdot 60 \cdot 60 \cdot 10$ *desetinki sekunde. Ukupna greška nastala greškom zaokruživanja je*

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

Ako je poznato da Scud putuje brzinom $1676\,\mathrm{m/s}$ *na predviđenoj visini susreta, onda su ga rakete Patriot pokušale naći više od pola kilometra daleko od njegovog stvarnog položaja.*

Koje je precizno objašnjenje uzroka ove katastrofe? Stvarni uzrok je nedovoljno pažljivo, "mudro" ili "hackersko" pisanje programa. Patriot sustav prati cilj tako da mjeri vrijeme potrebno radarskim signalima da se odbiju od cilja i vrate natrag. Točnost podataka o vremenu je, naravno, ključna za precizno uništenje cilja.

Računalo koje je upravljalo Patriot raketama bazirano je na konstrukciji iz 1970-ih godina i koristi 24*-bitnu aritmetiku. Međutim, realizacija floating point aritmetike u ta "davna" vremena bila je mnogo sporija od cjelobrojne, posebno za množenje i dijeljenje. Zbog toga se u programima često koristila kombinacija cjelobrojne (tzv. fixed point) aritmetike i floating point aritmetike, da se ubrzaju te dvije operacije.*

Tako je sistemski sat mjerio vrijeme u desetinkama sekunde, ali se vrijeme spremalo kao cijeli broj desetinki proteklih od trenutka kad je računalo upaljeno. Za sve ostale proračune, vrijeme se računa tako da se pomnoži broj desetinki n *s osnovnom jedinicom* $t_0 = 0.1\,\mathrm{s}$ *u kvazi-cjelobrojnoj aritmetici, a zatim pretvori u pravi* 24*-bitni floating point broj.*

Kvazi-cjelobrojni ili fixed point prikaz realnog broja odgovara prikazu cijelih brojeva, s tim da se uzima da je binarna točka ispred prve prikazane znamenke. Preciznije rečeno, ako imamo 24 *bita za takav prikaz, realni broj se interpretira kao višekratnik od* 2^{-23}*, a prikaz se dobiva zaokruživanjem višekratnika na cijeli broj i to odbacivanjem racionalnog dijela (u smjeru nule). Dakle, za realni broj* $r \geq 0$ *pamti se cijeli broj* $\lfloor r \cdot 2^{23} \rfloor$*, a za negativni* r *pamti se* $-\lfloor |r| \cdot 2^{23} \rfloor$*. Naravno, tako se mogu prikazati samo realni brojevi iz* $[-1, 1)$*, inače imamo premalo bitova.*

Za analizu grešaka, ignorirajmo da se stvarno sprema cijeli broj, i pogledajmo kojoj aproksimaciji za r *odgovara taj spremljeni broj. Neka* $fi(r)$ *označava tu aproksimaciju za* r*, u smislu da je spremljeni prikaz od* r*, zapravo, egzaktni prikaz broja* $fi(r)$*. Taj broj možemo jednostavno dobiti tako da spremljene bitove interpretiramo kao prikaz s nenormaliziranom mantisom od* 23 *bita, jer se pamte redom znamenke iza binarne točke, a eksponenta nema, tj. jednak je* 0*. Dakle, za* $r \in [-1, 1)$*, broj* $fi(r)$ *ima točno prva* 23 *bita od* r *iza binarne točke, a ostatak zanemarujemo, zbog zaokruživanja odbacivanjem. Za apsolutnu grešku, očito, vrijedi*

$$|x - fi(x)| < 2^{-23},$$

ali relativna greška može biti velika.

Kako to izgleda za osnovnu vremensku jedinicu $t_0 = 0.1$*? Zadržavanjem prva* 23 *bita iza binarne točke i odbacivanjem ostatka u*

$$0.1 = (0.0001\,1001\,1001\,1001\,1001\,100|1\,1001\ldots)_2.$$

dobivamo

$$fi(0.1) = (0.0001\,1001\,1001\,1001\,1001\,100)_2.$$

Učinjena (apsolutna) greška je

$$|0.1 - fi(0.1)| = 0.1 \cdot 2^{-20} = \frac{1}{10485760} = 9.5367431640625 \cdot 10^{-8},$$

dok je relativna greška točno 2^{-20}*, ili* 10 *puta veća, što uopće ne izgleda strašno.*

Nakon n *otkucaja sistemskog sata (u desetinkama), pravo vrijeme u sekundama je* $t = n{\cdot}t_0$*. Umjesto toga, računa se* $n{\cdot}fi(t_0)$*. Pretpostavimo da se to množenje izvodi egzaktno, bez dodatnih grešaka zaokruživanja (kao da smo u cjelobrojnoj aritmetici). Izračunato vrijeme je* $\widehat{t} = n \cdot fi(t_0)$*. Relativna greška ostaje ista*

$$\left|\frac{t - \widehat{t}}{t}\right| = 2^{-20},$$

jer se n *skrati. Međutim, apsolutna greška je* n *puta veća*

$$|t - \widehat{t}| = n \cdot |0.1 - fi(0.1)| \approx n \cdot (9.5 \cdot 10^{-8}).$$

Nažalost, za točno gađanje treba apsolutna, a ne relativna točnost u vremenu. Za dovoljno veliki n*, a nakon* 100 *sati je* $n = 3600000$*, dobivenom* $\widehat{t}$ *nema spasa, čak i prije pretvaranja u floating point prikaz (što još doprinosi ukupnoj pogrešci).*

Što se moglo napraviti? Da su se ista 23 *bita koristila za pravu mantisu u floating point prikazu*

$$0.1 = (0.1100\,1100\,1100\,1100\,1100\,110|0\,1100\ldots)_2 \cdot 2^{-3},$$

bez obzira na vrstu zaokruživanja, dobili bismo

$$f\ell(0.1) = (0.1100\,1100\,1100\,1100\,1100\,110)_2 \cdot 2^{-3},$$

uz točno $2^4 = 16$ *puta manje greške. Na primjer, apsolutna greška je*

$$|0.1 - f\ell(0.1)| = 0.1 \cdot 2^{-24} \approx 5.96 \cdot 10^{-9}.$$

Čak i nakon 100 *sati, posljedica ove greške je promašaj od oko* 40m*, što je (s visokom vjerojatnošću) još uvijek dovoljno točno za uništenje Scuda.*

S druge strane, treba izbjegavati eksplicitno korištenje tzv. apsolutnog vremena od trenutka uključenja. Puno bolje je brojač iznova postaviti na nulu u trenutku prvog radarskog kontakta, ili zapamtiti stanje cjelobrojnog brojača u tom trenutku, a sva ostala vremena računati prvo cjelobrojnim oduzimanjem stanja brojača, pa tek onda dobivenu razliku pretvoriti u sekunde (pomnožiti bitno manji broj s t_0*). Scud ipak leti malo kraće od* 100 *sati!*

Zanimljivo je da je prva indikacija ove pogreške prijavljena punih 14 *dana ranije, 11. veljače. Na jednom sustavu Patriota uočen je pomak u tzv. "prostoru udara" (engl. range gate) za punih* 20% *nakon neprekidnog rada od* 8 *sati. Ti podaci pokazivali su da nakon* 20 *sati neprekidnog rada, sustav neće moći pratiti i presresti nadolazeći Scud. Modificirani program koji kompenzira netočno računanje vremena službeno je izašao 16. veljače. Međutim, u Dhahran je stigao tek 26. veljače, dan nakon nesreće. Ipak, čudno je da posade sustava na terenu nisu dobile barem obavijest o problemu — povremeni "restart" sustava bio bi sasvim dovoljan za prvo vrijeme.*

Primjer 2.8.2. (Eksplozija Ariane 5.)

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

Dva tjedna kasnije, stručnjaci su objasnili događaj. Kontrolna varijabla (koja je služila samo informacije radi) u programu vođenja rakete mjerila je horizontalnu brzinu rakete. 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. Zanimljivo je da je taj isti program bio korišten u prijašnjoj sporijoj verziji Ariane 4, pa do katastrofe nije došlo.*

Primjer 2.8.3. (Potonuće naftne platforme Sleipner A.)

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. Prilikom uronjavanja baze došlo je do pucanja. 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 (netko nije provjerio rezultate programa). Proračun je dao naprezanja 47% *manja od stvarnih. Nakon detaljne analize s točnijim konačnim elementima, izračunato je da su ćelije morale popustiti na dubini od* 62 *metra. Stvarna dubina pucanja bila je* 65 *metara!*

Primjer 2.8.4. (Izabran je pogrešan predsjednik.)

Možda je najbizarniji primjer da greška zaokruživanja može poremetiti izbore za predsjednika SAD. U američkom sustavu izbora predsjednika, svaka od saveznih država ima određen broj predstavnika (ljudi) u tijelu koje se zove Electoral College i koje formalno bira predsjednika. Broj predstavnika svake pojedine države u tom tijelu proporcionalan je broju stanovnika te države u odnosu na ukupan broj stanovnika. Pretpostavimo da u Electoral College-u ima a predstavnika, populacija SAD je p stanovnika, a država i ima p_i stanovnika. Broj predstavnika države i u Electoral

College-u trebao bi biti

$$a_i = \frac{p_i}{p} \cdot a.$$

Ali, predstavnici su ljudi, pa bi a_i morao biti cijeli broj. Zbog toga se a_i mora zaokružiti na cijeli broj $\widehat{a}_i$ po nekom pravilu. Naravno, na kraju mora biti $\sum_i \widehat{a}_i = a$. Razumno i prirodno pravilo je:

- *$\widehat{a}_i$ mora biti jedan od dva cijela broja koji su najbliži a_i (tzv. "uvjet kvote").*

Naime, pravilno zaokruživanje (kao kod prikaza brojeva) je možda najpravednije, ali ne mora dati $\sum_i \widehat{a}_i = a$, pa se mora upotrijebiti slabije pravilo.

Međutim, broj stanovnika p_i se vremenom mijenja (a time i p). Isto tako, ukupni broj predstavnika a u tijelu se može promijeniti od jednih do drugih izbora. Zbog toga se dodaju još dva prirodna "politička" pravila:

- *Ako se poveća ukupan broj predstavnika a, a svi ostali podaci se ne promijene, $\widehat{a}_i$ ne smije opasti (tzv. "monotonost predstavničkog tijela").*
- *Ako je broj stanovnika države p_i porastao, a ostali podaci su nepromijenjeni, $\widehat{a}_i$ ne smije opasti (tzv. "monotonost populacije").*

Svrha je jasna, jer ljudi vole uspoređivati prošle i nove "kvote". Nažalost, ne postoji metoda za određivanje broja predstavnika koja bi zadovoljavala sva tri kriterija.

U američkoj povijesti zaista postoji slučaj da je izabran "pogrešan" predsjednik. Samuel J. Tilden izgubio je izbore 1876. godine u korist Rutherforda B. Hayesa, samo zbog načina dodjele elektorskih glasova u toj prilici. Da stvar bude još zanimljivija, ta metoda dodjele glasova nije bila ona koju je propisivao zakon iz tog vremena.

Matematički gledano, problem izbornih sustava i raznih pravila za računanje broja predstavnika u predstavničkim tijelima, poput Sabora, nije trivijalan, a ozbiljno se proučava već stotinjak godina.

Uzmimo najjednostavniji "izborni sustav" u kojem vrijedi samo "uvjet kvote" i pretpostavimo da se on primjenjuje za svake izbore posebno (tj. vremenski lokalno), uz poznate podatke o broju stanovnika p_i u svim "izbornim jedinicama". Kako biste što pravednije izračunali brojeve $\widehat{a}_i$ predstavnika svake jedinice?

3. Vektorske i matrične norme

3.1. Vektorske norme

Vektorske i matrične norme osnovno su sredstvo koje koristimo kod ocjene grešaka vezanih uz numeričke metode, posebno u numeričkoj linearnoj algebri.

Definicija 3.1.1. (Vektorska norma) *Vektorska norma je svaka funkcija* $\|\cdot\| : \mathbb{C}^n \to \mathbb{R}$ *koja zadovoljava sljedeća svojstva:*

1. $\|x\| \geq 0$, $\forall x \in \mathbb{C}^n$, *a jednakost vrijedi ako i samo ako je* $x = 0$,
2. $\|\alpha x\| = |\alpha| \, \|x\|$, $\forall \alpha \in \mathbb{R}$, $\forall x \in \mathbb{C}^n$,
3. $\|x + y\| \leq \|x\| + \|y\|$, $\forall x, y \in \mathbb{C}^n$. *Ova je nejednakost poznatija pod imenom nejednakost trokuta (zbroj duljina bilo koje dvije stranice trokuta veći je od duljine treće stranice).*

Analogno se definira vektorska norma na bilo kojem vektorskom prostoru V nad poljem $F = \mathbb{R}$ ili $\mathbb{C}$.

Neka je x vektor iz $\mathbb{C}^n$ s komponentama x_i, $i = 1, \ldots, n$, u oznaci $x = (x_1, \ldots, x_n)^T$, ili, skraćeno $x = [x_i]$. U numeričkoj linearnoj algebri najčešće se koriste sljedeće tri norme:

1. 1-norma ili ℓ_1 norma, u engleskom govornom području poznatija kao "Manhattan" ili "taxi-cab" norma
$$\|x\|_1 = \sum_{i=1}^{n} |x_i|,$$
2. 2-norma ili ℓ_2 norma ili euklidska norma
$$\|x\|_2 = (x^* x)^{1/2} = \sqrt{\sum_{i=1}^{n} |x_i|^2},$$
3. ∞-norma ili ℓ_∞ norma
$$\|x\|_\infty = \max_{i=1,\ldots,n} |x_i|.$$

Primijetite da je samo 2-norma izvedena iz skalarnog produkta, dok ostale dvije to nisu.

2-norma ima dva bitna svojstva koje je čine posebno korisnom. Ona je invarijantna na unitarne transformacije vektora x, tj. ako je Q unitarna matrica ($Q^*Q = QQ^* = I$), onda je

$$\|Qx\|_2 = (x^*Q^*Qx)^{1/2} = (x^*x)^{1/2} = \|x\|_2.$$

Također ona je diferencijabilna za sve $x \neq 0$, s gradijentom

$$\nabla\|x\|_2 = \frac{x}{\|x\|_2}.$$

Sve ove tri norme specijalni su slučaj Hölderove p-norme (ℓ_p norme) definirane s:

$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}, \quad p \geq 1.$$

Za Hölderove p-norme vrijedi i poznata Hölderova nejednakost

$$|x^*y| \leq \|x\|_p\|y\|_q, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

Posebni slučaj Hölderove nejednakosti za $p = q = 2$ je Cauchy-Schwarzova nejednakost

$$|x^*y| \leq \|x\|_2\|y\|_2.$$

Koliko se dvije p-norme međusobno razlikuju, pokazuje sljedeća nejednakost, koja se može dostići. Neka su α i β dvije p norme takve da je $\alpha \leq \beta$. Tada vrijedi

$$\|x\|_\beta \leq \|x\|_\alpha \leq n^{(1/\alpha - 1/\beta)}\|x\|_\beta.$$

Ova se nejednakost često proširuje i zapisuje tablicom $\|x\|_\alpha \leq C_M\|x\|_\beta$, gdje su C_M-ovi

$\alpha\backslash\beta$	1	2	∞
1	1	$\sqrt{n}$	n
2	1	1	$\sqrt{n}$
∞	1	1	1

Primijetite da sve p-norme ovise samo o apsolutnoj vrijednosti komponenti x_i, pa je p-norma rastuća funkcija apsolutnih vrijednosti komponenti x_i. Označimo s $|x|$ vektor apsolutnih vrijednosti komponenti vektora x, tj. $|x| = [|x_i|]$. Za vektore apsolutnih vrijednosti (u $\mathbb{R}^n$) možemo uvesti parcijalni uređaj relacijom

$$|x| \leq |y| \iff |x_i| \leq |y_i|, \quad \forall i = 1, \ldots, n.$$

Definicija 3.1.2. (Monotona i apsolutna norma) *Norma na $\mathbb{C}^n$ je monotona ako vrijedi*

$$|x| \leq |y| \implies \|x\| \leq \|y\|, \quad \forall x, y \in \mathbb{C}^n.$$

Norma na $\mathbb{C}^n$ je apsolutna ako vrijedi

$$\| \, |x| \, \| = \|x\|, \quad \forall x \in \mathbb{C}^n.$$

Bauer, Stoer i Witzgall dokazali su netrivijalni teorem koji pokazuje da su ta dva svojstva ekvivelentna.

Teorem 3.1.1. *Norma na $\mathbb{C}^n$ je monotona ako i samo ako je apsolutna.*

Definicija vektorskih normi u sebi ne sadrži zahtjev da je vektorski prostor iz kojeg su vektori konačno dimenzionalan. Na primjer, norme definirane na vektorskom prostoru neprekidnih funkcija na $[a, b]$ (u oznaci $C[a, b]$) definiraju se slično normama na C^n:

1. L_1 norma
$$\|f\|_1 = \int_a^b |f(t)| \, dt,$$
2. L_2 norma
$$\|f\|_2 = \Big(\int_a^b |f(t)|^2 \, dt \Big)^{1/2},$$
3. L_∞ norma
$$\|f\|_\infty = \max\{|f(x)| \mid x \in [a, b]\},$$
4. L_p norma
$$\|f\|_p = \Big(\int_a^b |f(t)|^p \, dt \Big)^{1/p}, \quad p \geq 1.$$

3.2. Matrične norme

Zamijenimo li u definiciji 3.1.1. vektor $x \in \mathbb{C}^n$ matricom $A \in \mathbb{C}^{m \times n}$, dobivamo matričnu normu.

Definicija 3.2.1. (Matrična norma) *Matrična norma je svaka funkcija $\| \cdot \| : \mathbb{C}^{m \times n} \to \mathbb{R}$ koja zadovoljava sljedeća svojstva:*

1. $\|A\| \geq 0$, $\forall A \in \mathbb{C}^{m \times n}$, *a jednakost vrijedi ako i samo ako je* $A = 0$,
2. $\|\alpha A\| = |\alpha| \, \|A\|$, $\forall \alpha \in \mathbb{R}$, $\forall A \in \mathbb{C}^{m \times n}$,
3. $\|A + B\| \leq \|A\| + \|B\|$, $\forall A, B \in \mathbb{C}^{m \times n}$.

Za matričnu normu ćemo reći da je konzistentna ako vrijedi

4. $\|AB\| \leq \|A\| \, \|B\|$

kad god je matrični produkt AB definiran. Oprez, norme od A, B i AB ne moraju biti definirane na istom prostoru (dimenzije)!

Neki autori smatraju da je i ovo posljednje svojstvo sastavni dio definicije matrične norme (tada to svojstvo obično zovu submultiplikativnost). Ako su ispunjena samo prva tri svojstva, onda to zovu generalizirana matrična norma.

Matrične norme mogu nastati na dva različita načina. Ako matricu A promatramo kao vektor s $m \times n$ elemenata, onda, direktna primjena vektorskih normi (uz oznaku a_{ij} matričnog elementa u i-tom retku i j-tom stupcu) daje sljedeće definicije:

1. ℓ_1 norma
$$|||A|||_1 := \|A\|_S = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|,$$

2. ℓ_2 norma (euklidska, Frobeniusova, Hilbert–Schmidtova, Schurova)
$$|||A|||_2 := \|A\|_F = (\mathrm{tr}(A^*A))^{1/2} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2},$$

3. ℓ_∞ norma
$$|||A|||_\infty := \|A\|_M = \max_{\substack{i=1,\ldots,m \\ j=1,\ldots,n}} |a_{ij}|.$$

tr je oznaka za trag matrice – zbroj dijagonalnih elemenata matrice.

Pokažimo da ℓ_1 i ℓ_2 norma zadovoljavaju svojstvo konzistentnosti, a ℓ_∞ norma ga ne zadovoljava. Vrijedi

$$\begin{aligned}
\|AB\|_S &= \sum_{i=1}^{m} \sum_{j=1}^{s} \sum_{k=1}^{n} |a_{ik} b_{kj}| \leq \sum_{i=1}^{m} \sum_{j=1}^{s} \sum_{k=1}^{n} \sum_{\ell=1}^{n} |a_{ik} b_{\ell j}| \\
&\leq \sum_{i=1}^{m} \sum_{k=1}^{n} |a_{ik}| \sum_{\ell=1}^{n} \sum_{j=1}^{s} |b_{\ell j}| = \|A\|_S \, \|B\|_S, \\
\|AB\|_F^2 &= \sum_{i=1}^{m} \sum_{j=1}^{s} \left| \sum_{k=1}^{n} a_{ik} b_{kj} \right|^2 \leq \sum_{i=1}^{m} \sum_{j=1}^{s} \Big(\sum_{k=1}^{n} |a_{ik}|^2 \Big) \Big(\sum_{\ell=1}^{n} |b_{\ell j}|^2 \Big) \\
&= \Big(\sum_{i=1}^{m} \sum_{k=1}^{n} |a_{ik}|^2 \Big) \Big(\sum_{\ell=1}^{n} \sum_{j=1}^{s} |b_{\ell j}|^2 \Big) = \|A\|_F^2 \, \|B\|_F^2.
\end{aligned}$$

Primijetite da se u dokazu da je Frobeniusova norma konzistentna koristila Cauchy-Schwarzova nejednakost.

Pokažimo na jednom primjeru da ℓ_∞ norma ne zadovoljava svojstvo konzistentnosti. Za matrice

$$A = B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \text{je} \quad AB = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix},$$

pa je

$$\|AB\|_M = 2, \quad \|A\|_M \, \|B\|_M = 1.$$

Ipak i od $\|\cdot\|_M$ se može napraviti konzistentna norma. Definiramo li

$$|||A||| = m \, \|A\|_M,$$

vrijedi

$$\begin{aligned} |||AB||| &= m \max_{\substack{i=1,\dots,m \\ j=1,\dots,s}} \left| \sum_{k=1}^{n} a_{ik} b_{kj} \right| \leq m \max_{\substack{i=1,\dots,m \\ j=1,\dots,s}} \sum_{k=1}^{n} |a_{ik} b_{kj}| \\ &\leq m \max_{\substack{i=1,\dots,m \\ j=1,\dots,s}} \sum_{k=1}^{n} \|A\|_M \, \|B\|_M = (m \, \|A\|_M) \, (n \, \|B\|_M) = |||A||| \, |||B|||. \end{aligned}$$

S druge strane, matrične norme možemo dobiti kao **operatorske norme** iz odgovarajućih vektorskih korištenjem definicije

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} (\text{ili } = \max_{\|x\|=1} \|Ax\|). \tag{3.2.1}$$

Kad se uvrste odgovarajuće vektorske norme u (3.2.1), dobivamo

1. matrična 1-norma, "maksimalna stupčana norma"

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{m} |a_{ij}|,$$

2. matrična 2-norma, spektralna norma

$$\|A\|_2 = (\rho(A^* A))^{1/2} = \sigma_{\max}(A),$$

3. matrična ∞-norma, "maksimalna retčana norma"

$$\|A\|_\infty = \max_{i=1,\dots,m} \sum_{j=1}^{n} |a_{ij}|,$$

pri čemu je ρ oznaka za spektralni radijus kvadratne matrice (maksimalna po apsolutnoj vrijednosti svojstvena vrijednost)

$$\rho(B) = \max\{|\lambda| \mid \det(B - \lambda I) = 0\}, \quad (B \text{ kvadratna!}), \tag{3.2.2}$$

a σ je standardna oznaka za tzv. singularnu vrijednost matrice. Detaljnu definiciju što je to singularna vrijednost, dobit ćete u poglavlju koje će se baviti dekompozicijom singularnih vrijednosti.

Matrična 2-norma se teško računa, (trebalo bi naći po apsolutnoj vrijednosti najveću svojstvenu vrijednost), pa je uobičajeno da se ona procjenjuje korištenjem ostalih normi.

Tablica ovisnosti koja vrijedi među matričnim normama je: $\|A\|_\alpha \leq C_M \|A\|_\beta$, gdje su C_M-ovi

$\alpha\backslash\beta$	1	2	∞	F	M	S
1	1	$\sqrt{m}$	m	$\sqrt{m}$	m	1
2	$\sqrt{n}$	1	$\sqrt{m}$	1	$\sqrt{mn}$	1
∞	n	$\sqrt{n}$	1	$\sqrt{n}$	n	1
F	$\sqrt{n}$	$\sqrt{\operatorname{rang}(A)}$	$\sqrt{m}$	1	$\sqrt{mn}$	1
M	1	1	1	1	1	1
S	n	$\sqrt{mn \operatorname{rang}(A)}$	m	$\sqrt{mn}$	mn	1

Posebno su važne **unitarno invarijantne norme**, tj. one za koje vrijedi

$$\|UAV\| = \|A\|, \tag{3.2.3}$$

za sve unitarne matrice U i V.

Dvije najpoznatije unitarno invarijantne norme su Frobeniusova i spektralna norma. Pokažimo to. Kvadrat Frobeniusove norme matrice A možemo promatrati kao zbroj kvadrata normi stupaca a_j:

$$\|A\|_F^2 = \sum_{j=1}^{n} \|a_j\|^2.$$

S druge strane, za svaku unitarnu matricu U vrijedi

$$\|Ua_j\|_2^2 = a_j^* U^* U a_j = a_j^* a_j = \|a_j\|_2^2.$$

Objedinimo li te relacije, dobivamo

$$\|UA\|_F^2 = \sum_{j=1}^{n} \|Ua_j\|^2 = \sum_{j=1}^{n} \|a_j\|^2 = \|A\|_F^2.$$

Konačno, vrijedi

$$\|UAV\|_F^2 = \|AV\|_F^2 = \|V^* A^*\|_F^2 = \|A^*\|_F^2 = \|A\|_F^2.$$

Da bismo dokazali da je matrična 2-norma unitarno ekvivalentna, potrebno je pokazati da transformacije sličnosti čuvaju svojstvene vrijednosti matrice. Ako je S nesingularna (kvadratna) matrica, a B kvadratna, onda je matrica $S^{-1}BS$ slična matrici B. Ako je spektralna faktorizacija matrice $S^{-1}BS$

$$S^{-1}BSX = X\Lambda,$$

pri čemu je X matrica svojstvenih vektora, a Λ svojstvenih vrijednosti. Množenjem sa S slijeva, dobivamo

$$B(SX) = (SX)\Lambda,$$

tj. matrica svojstvenih vektora je SX, dok su svojstvene vrijednosti ostale nepromijenjene. Primijetite da za unitarne matrice vrijedi $V^* = V^{-1}$.

Za matričnu 2-normu, onda vrijedi

$$\|UAV\|_2 = (\rho(V^*A^*U^*UAV))^{1/2} = (\rho(V^*A^*AV))^{1/2}.$$

Budući da je V unitarna, A^*A i V^*A^*AV su unitarno ekvivalentne, pa je

$$\|UAV\|_2 = (\rho(A^*A))^{1/2} = \|A\|_2.$$

4. Stabilnost problema i algoritama

U drugom poglavlju vidjeli smo da se greške javljaju na svakom koraku prilikom rješavanja realnih praktičnih problema. Na nekoliko primjera vidjeli smo da ukupni efekti raznih grešaka mogu biti katastrofalni po konačno rješenje. Prije opisa raznih metoda za numeričko rješavanje određenih vrsta problema, trebamo matematički formalizam i odgovarajući praktični alat za procjenu efekta ili utjecaja raznih vrsta grešaka.

Ponešto od tih tehnika smo već koristili u prethodnim primjerima, bez posebnog formalizma. U nastavku dajemo nešto formalniji i općenitiji pristup analizi grešaka pri rješavanju problema.

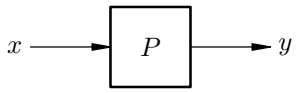
4.1. Jednostavni model problema

Problem P kojeg rješavamo, tipično ima neki ulaz i neki izlaz (slično algoritmu). Ulaz je neki (konačni) skup podataka, a izlaz je opet neki (konačni) skup podataka. U numerici uzimamo da su ti podaci brojevi. Recimo, ulaz su koeficijenti neke jednadžbe, a izlaz su rješenja ili korijeni te jednadžbe u nekom dogovorenom poretku. Pretpostavimo, radi jednostavnosti, da su ulazni i izlazni podaci realni brojevi.

Slična analiza može se provesti i za drugačije podatke brojevnog tipa, ali za numeričku praksu su upravo realni brojevi najvažniji slučaj, posebno kad uzmemo u obzir da računanje provodimo aritmetikom računala koja modelira realne brojeve. Kompleksni brojevi su najmanji problem, njih ionako prikazujemo parom realnih brojeva $z = (\operatorname{Re} z, \operatorname{Im} z)$.

Ako preciznije pogledamo, ulaz i izlaz su uređeni konačni nizovi podataka, a ne skupovi. Poredak podataka ima bitnu ulogu, zbog njihove praktične interpretacije, odnosno, stvarnog značenja. Na primjer, nije svejedno koji koeficijent kvadratne jednadžbe dolazi uz koju potenciju. U tom smislu, ulaz možemo prikazati vektorom $x \in \mathbb{R}^m$, a izlaz vektorom $y \in \mathbb{R}^n$. Naš problem P možemo (pomalo algoritamski)

zamisliti kao crnu kutiju, koja prihvaća neki ulaz x i rješava problem za taj ulaz, producirajući izlaz y.



Na kraju, pretpostavimo da je izlaz jednoznačno određen ulazom, što je razumna pretpostavka za determinističke probleme. Drugim riječima, svakom ulazu x kutija P pridružuje jednoznačni izlaz y. Stoga, problem možemo zamišljati i kao preslikavanje ili funkciju f, zadanu s

$$f : \mathbb{R}^m \to \mathbb{R}^n, \quad y = f(x). \tag{4.1.1}$$

Ono što nas zanima je osjetljivost preslikavanja f u nekoj točki x obzirom na male perturbacije x-a, tj. zanima nas koliko je tada velika perturbacija y-a obzirom na perturbaciju x-a.

Stupanj osjetljivosti želimo mjeriti jednim jedinim brojem — **uvjetovanošću** funkcije f u točki x. Posebno, smatramo da se vrijednost funkcije f u svakoj točki računa egzaktno, u beskonačnoj točnosti. Dakle, uvjetovanost od f je svojstvo funkcije f i ovisi samo o f, a ne ovisi o načinu kako se f računa — u smislu algoritamskih razmatranja ili efekata vezanih uz implementaciju postupka za računanje funkcije f.

Bitno je primijetiti da to ne znači da poznavanje uvjetovanosti problema nema nikakav utjecaj na izbor algoritama za rješenje problema. Vrijedi upravo suprotno! Međutim, za početak, treba znati da li problem po sebi pojačava ili prigušuje male perturbacije u polaznim podacima. Zatim, treba naći dobar algoritam, koji tom inherentnom ponašanju problema ne doprinosi previše dodatnim greškama. Drugim riječima, da bismo razlikovali dobre od loših algoritama (u smislu točnosti), moramo prvo moći razlikovati dobre od loših problema. Uvjetovanost služi kao mjera "kvalitete" problema, a kasnije uvodimo isti pojam i za mjerenje "kvalitete" algoritma.

Potreba za vezom između grešaka u polaznim podacima i grešaka u rezultatima ne dolazi samo zbog grešaka u mjerenju. Do istog problema dolazimo analizom grešaka koje nastaju računanjem nekim algoritmom na računalu.

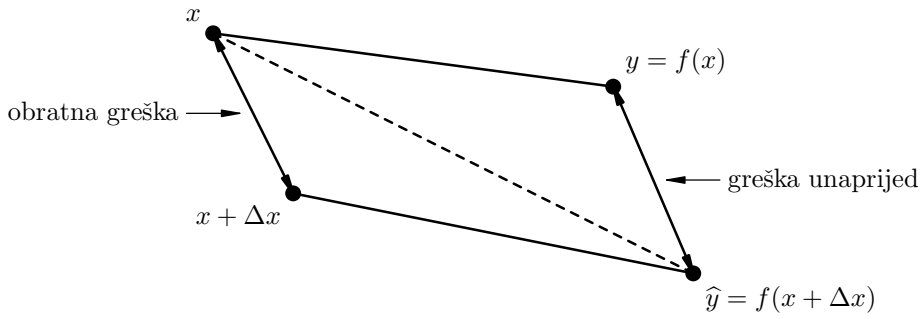
Ilustrirajmo to na najjednostavnijem primjeru funkcije $f : \mathbb{R} \to \mathbb{R}$. Pretpostavimo da je $\widehat{y}$ aproksimacija rješenja nekog problema $y = f(x)$, izračunata nekim algoritmom u realnoj aritmetici računala (tzv. aritmetika pomičnog zareza, ili engl. floating point arithmetic) s preciznošću u. Ono što nas zanima je koliko je izračunato rješenje daleko od pravog.

U većini slučajeva bili bismo zadovoljni s malom relativnom greškom u rezultatu

$$E_{\text{rel}} = \frac{|y - \widehat{y}|}{|y|} \approx u,$$

ali to nije uvijek moguće postići.

Umjesto toga, pitamo se koji bi podaci $x + \Delta x$ zapravo dali rješenje $\widehat{y}$ u egzaktnoj aritmetici, tj. kada je $\widehat{y} = f(x + \Delta x)$. Generalno, može postojati mnogo takvih Δx, pa je zanimljiv samo onaj najmanji. Vrijednost $|\Delta x|$ ili $\min |\Delta x|$ zove se **obratna greška** (engl. backward error). Ako obratnu grešku podijelimo s $|x|$ dobit ćemo relativnu obratnu grešku. Relativnu i apsolutnu grešku od $\widehat{y}$, (relativna E_{rel}, apsolutna $\Delta y = y - \widehat{y}$) zovemo **greškom unaprijed** (engl. forward error). Sljedeća skica pokazuje njihove razlike.



Proces ograđivanja (ili procjene) obratne greške izračunatog rješenja zove se **obratna analiza greške**. Motivacija i opravdanja za primjenu tog postupka ima nekoliko.

1. Analiza propagiranja grešaka zaokruživanja unaprijed, kroz sve operacije algoritma do konačnog rezultata, je ubitačan posao, koji najčešće daje vrlo pesimističke ocjene na točnost rezultata. U prošlom poglavlju smo napravili takvu analizu za jednu jedinu i to egzaktnu operaciju $\circ$, a rezultati već imaju kompliciran oblik. Da smo, uz perturbirane podatke, uzeli u obzir i grešku zaokruživanja rezultata operacije, odnosno operaciju $\circ$ u aritmetici računala, rezultat bi bio još kompliciraniji.
2. Model aritmetike (2.6.2) po sebi kaže da je puno lakše greške zaokruživanja i aritmetike računala interpretirati kao perturbacije početnih podataka, uz egzaktne operacije. Osim toga, vrlo često i ulazni podaci imaju polazne pogreške (zbog mjerenja, prijašnjeg računanja ili zbog grešaka zaokruživanja nastalih spremanjem ulaznih podataka u računalo), pa ih ionako treba uzeti u obzir. Na kraju, teško ćemo kritizirati izračunati rezultat, ako je njegova obratna greška reda veličine grešaka u ulaznim podacima.
3. Velika prednost obratne analize grešaka je da se procjena ili ograda greške unaprijed prepušta teoriji perturbacija, tj. radi se teorija perturbacije za svaki problem, a ne za svaki problem i svaku metodu.

Zbog toga, posebno gledamo stabilnost problema (teorija perturbacije ili uvjetovanosti problema), a posebno analiziramo stabilnost algoritama.

Da bismo imali bolju predodžbu o stabilnosti algoritama možemo odmah, iako ne potpuno formalno, uvesti potrebne pojmove.

Za algoritam ćemo reći da je **obratno stabilan** (engl. backward stable), ako za proizvoljni x, izračunati $\widehat{y}$ ima malu obratnu grešku, tj. vrijedi $\widehat{y} = f(x + \Delta x)$ za neki "mali" Δx. Što znači "mali", naravno, ovisi o kontekstu. Funkcija f ovdje ovisi i o algoritmu, ali se računa točno (bez grešaka zaokruživanja).

Na primjer, pretpostavka o preciznosti osnovnih aritmetičkih operacija (2.6.2) za zbrajanje/oduzimanje je

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

Pogledamo li posljednju jednakost, izlazi da je zbrajanje/oduzimanje obratno stabilna operacija, jer se izračunati rezultat $f\ell(x \pm y)$ može interpretirati kao egzaktni rezultat operacije za malo perturbirane ulazne podatke $(1 + \varepsilon)\,x$ i $(1 + \varepsilon)\,y$.

Slično vrijedi i za mnoge druge numeričke algoritme u praksi. Dobiveni izračunati rezultat može se (i to bez previše napora) interpretirati kao egzaktno rješenje problema koji je "blizak" početnom. Ova tehnika obratne analize grešaka nastala je 1950-ih godina, a njeni začetnici su J. W. Givens, C. Lanczos i, iznad svih, J. H. Wilkinson.

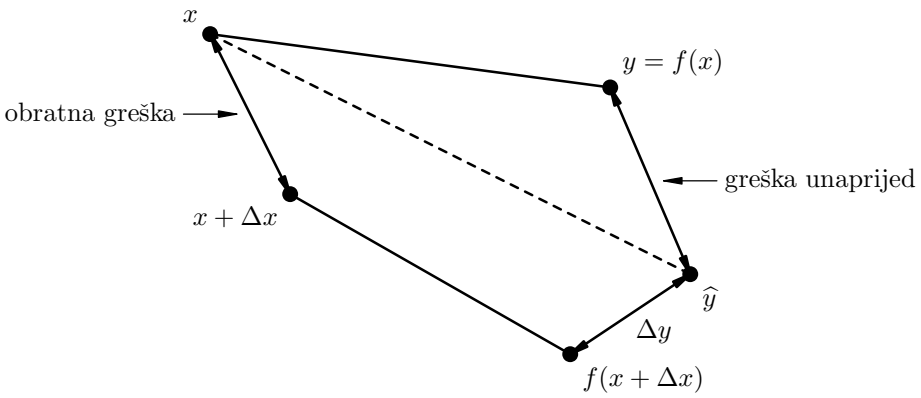
Nažalost, u nekim slučajevima nije moguće postići pravu obratnu stabilnost u relativnom smislu. Odmah je očito da probleme možemo očekivati u blizini nule.

Primjer 4.1.1. *Većina potprograma (u bibliotekama računala) za računanje funkcije* cos *ne poštuje obratnu stabilnost* $\widehat{y} = \cos(x+\Delta x)$ *uz relativno mali* Δx*. Umjesto toga, zadovoljavaju slabiju relaciju*

$$\widehat{y} + \Delta y = f(x + \Delta x), \quad |\Delta y| \leq \varepsilon|y|, \quad |\Delta x| \leq \nu|x|,$$

koja je poznata pod imenom miješana unaprijed-unazad greška (engl. mixed forward-backward error). Pokušajte sami pronaći kad ne vrijedi obratna stabilnost i opravdati da onda vrijedi ovakva stabilnost.

Prethodna relacija, zapravo, kaže da je rezultat računanja skoro točan za skoro točne ulazne podatke. Sljedeća slika ilustrira ovakvo ponašanje grešaka.



U tom smislu, općenito, kažemo da je algoritam **numerički stabilan** ako je stabilan u miješanom unaprijed-unazad smislu. Odavde odmah slijedi da je obratno stabilan algoritam i numerički stabilan!

Uočite da su ove definicije prvenstveno orijentirane na algoritme, tj. uključuju i greške zaokruživanja. Jasno je da ukupno ponašanje grešaka ovisi i o problemu, pa pojam stabilnosti možemo uvesti i za problem po sebi, tj. za egzaktno računanje f. Problem je **stabilan** ako mala perturbacija ulaznih podataka rezultira malom perturbacijom rezultata (u apsolutnom ili relativnom smislu). Veza između greške unaprijed i obratne greške je uvjetovanost problema.

4.2. Uvjetovanost problema

Istražimo uvjetovanost problema za funkciju $f : \mathbb{R} \to \mathbb{R}$. Promatramo ponašanje funkcije f za male perturbacije Δx u okolini točke x. Neka je Δy pripadna perturbacija funkcijske vrijednosti $y = f(x)$, tj. $f(x + \Delta x) = y + \Delta y$. Algoritamski analogon je pretpostavka da aproksimativno rješenje zadovoljava $\widehat{y} = f(x + \Delta x)$. Pretpostavimo još da je f dva puta neprekidno derivabilna. Korištenjem Taylorovog polinoma stupnja 1 dobivamo

$$\Delta y = f(x + \Delta x) - f(x) = f'(x)\Delta x + \frac{f''(x + \theta\Delta x)}{2!}\,(\Delta x)^2, \quad \theta \in (0, 1).$$

Za male perturbacije Δx, apsolutni oblik ove relacije je

$$\Delta y = f'(x)\,\Delta x + O((\Delta x)^2),$$

odakle slijedi da je $f'(x)$ ili $|f'(x)|$ apsolutna uvjetovanost funkcije f.

Pošto nas više zanimaju relativne greške, ako je $x \neq 0$ i $y \neq 0$, prethodnu relaciju možemo napisati u relativnoj formi

$$\frac{\Delta y}{y} = \frac{x f'(x)}{f(x)}\,\frac{\Delta x}{x} + O\left(\left(\frac{\Delta x}{x}\right)^2\right),$$

pa relativnu uvjetovanost funkcije f možemo definirati kao

$$(\operatorname{cond} f)(x) := \left|\frac{x f'(x)}{f(x)}\right|. \tag{4.2.1}$$

Za $x = 0$, $y \neq 0$, umjesto relativne greške u x, razumnije je promatrati apsolutnu grešku u x, a relativnu u y, pa je tada uvjetovanost $(\operatorname{cond} f)(x) = |f'(x)/f(x)|$. Slično vrijedi i za $y = 0$, $x \neq 0$, kad je $(\operatorname{cond} f)(x) = |x f'(x)|$. Ako je $x = y = 0$, onda je uvjetovanost problema jednostavno $|f'(x)|$.

Primjer 4.2.1. *Promotrimo funkciju*

$$f(x) = \ln x.$$

Njena je relativna uvjetovanost

$$(\operatorname{cond} f)(x) = \left|\frac{1}{\ln x}\right|,$$

što je veliko za $x \approx 1$. To znači da mala relativna promjena x-a, kad je $x \approx 1$, uzrokuje mnogo veću relativnu promjenu u $\ln x \approx 0$.

Uočite da mala relativna promjena u x-u, a to je i mala apsolutna promjena za $x \approx 1$, uzrokuje malu apsolutnu promjenu u $\ln x$, jer je

$$\ln(x + \Delta x) \approx \ln x + (\ln x)'\Delta x = \ln x + \frac{\Delta x}{x}.$$

To se vidi i iz odgovarajuće uvjetovanosti. Naime, s obzirom na to da je tada $y \approx 0$, prirodnija mjera uvjetovanosti je "relativno-apsolutna"

$$(\operatorname{cond}_1 f)(x) = |xf'(x)| = 1,$$

za $x = 1$, što potvrđuje prethodni zaključak. Međutim, promjena u $\ln x$ može biti velika u relativnom smislu.

Kad su definirane greška unaprijed, obratna greška i uvjetovanost, možemo reći da je

$$\text{greška unaprijed} \lesssim \text{uvjetovanost} \times \text{obratna greška},$$

što znači da **izračunato rješenje loše uvjetovanog problema može imati veliku grešku unaprijed**. Čak i kad izračunato rješenje ima malu obratnu grešku, ta će se greška "napuhati" unaprijed faktorom veličine uvjetovanosti problema.

Još je jedna definicija korisna. Ako metoda ima grešku unaprijed približno jednakog reda veličine kao obratnu grešku, onda ćemo metodu zvati **stabilnom unaprijed** (engl. forward stable). Takva metoda ne mora biti obratno stabilna, ali obratno vrijedi: obratno stabilna metoda je stabilna i unaprijed (i to, više-manje, po definiciji stabilnosti unaprijed). Na primjer, metoda koja je stabilna unaprijed, a nije obratno stabilna je Cramerovo pravilo za rješavanje 2×2 sistema linearnih jednadžbi. Pokažite to!

Kad je $f : \mathbb{R}^m \to \mathbb{R}^n$, situacija će se malo zakomplicirati. Označimo li

$$x = (x_1, x_2, \ldots, x_m)^T \in \mathbb{R}^m, \quad y = (y_1, y_2, \ldots, y_n)^T \in \mathbb{R}^n,$$

preslikavanje f možemo komponentno zapisati kao

$$y_k = f_k(x_1, x_2, \ldots, x_m), \quad k = 1, 2, \ldots, n.$$

Ponovno, pretpostavljamo da svaka funkcija f_k ima parcijalne derivacije po svim komponentnim varijablama x_ℓ u točki x.

Najdetaljniju analizu dobivamo gledajući promjene svake komponentne funkcije f_k po svakoj pojedinoj varijabli x_ℓ. Promatramo li promjenu koju uzrokuje mala perturbacija varijable x_ℓ u funkciji f_k, dobit ćemo isti rezultat (4.2.1) kao za funkciju jedne varijeble. Relativna uvjetovanost tog problema je

$$\gamma_{k\ell}(x) := (\mathrm{cond}_{k\ell}\, f)(x) := \left| \frac{x_\ell \dfrac{\partial f_k}{\partial x_\ell}}{f_k(x)} \right|.$$

Ako to napravimo za sve varijable x_ℓ i za svaku od funkcija f_k, dobivamo čitavu matricu $\Gamma(x) = [\gamma_{k\ell}(x)] \in \mathbb{R}_+^{n\times m}$ brojeva uvjetovanosti. Da bismo iz te matrice uvjetovanosti dobili samo jedan broj, možemo koristiti bilo koju mjeru "veličine" matrice $\Gamma(x)$, poput neke matrične norme. Definiramo

$$(\mathrm{cond}\, f)(x) := \|\Gamma(x)\|. \tag{4.2.2}$$

Tako definirana uvjetovanost ovisi o izboru norme, ali ne bitno, zbog međusobne ekvivalencije raznih normi.

Ako bilo koja komponenta od x ili y iščezava, problem možemo riješiti na isti način kao što smo to napravili u slučaju funkcije jedne varijable.

Grublju analizu s manje parametara dobivamo po ugledu na jednodimenzionalnu, promatranjem apsolutnih i relativnih perturbacija vektora u smislu norme. Takvu relativnu perturbaciju vektora $x \in \mathbb{R}^m$ definiramo kao

$$\frac{\|\Delta x\|}{\|x\|}, \quad \Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_m)^T,$$

pri čemu je $\|\cdot\|$ bilo koja vektorska norma, a komponente vektora perturbacije Δx su male u odnosu na komponente vektora x. Sada možemo pokušati povezati relativnu perturbaciju od y s relativnom perturbacijom od x.

Po analogiji s (4.2.1), imamo

$$\Delta y_k = f_k(x + \Delta x) - f_k(x) \approx \sum_{\ell=1}^{m} \frac{\partial f_k}{\partial x_\ell}\, \Delta x_\ell.$$

Za male perturbacije, ovu relaciju možemo zapisati u vektorsko-matričnom obliku

$$\Delta y \approx \frac{\partial f}{\partial x} \cdot \Delta x, \tag{4.2.3}$$

gdje je $\partial f/\partial x = J_f(x)$ Jacobijeva matrica preslikavanja f:

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_m} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_m} \end{bmatrix} \in \mathbb{R}^{n\times m}.$$

Zbog toga, barem aproksimativno vrijedi

$$|\Delta y_k| \leq \sum_{\ell=1}^{m} \left|\frac{\partial f_k}{\partial x_\ell}\right| |\Delta x_\ell| \leq \max_{\ell=1,\ldots,m} |\Delta x_\ell| \cdot \sum_{\ell=1}^{m} \left|\frac{\partial f_k}{\partial x_\ell}\right| \leq \max_{\ell=1,\ldots,m} |\Delta x_\ell| \cdot \max_{k=1,\ldots,n} \sum_{\ell=1}^{m} \left|\frac{\partial f_k}{\partial x_\ell}\right|.$$

Budući da prethodna relacija vrijedi za svaki $k = 1, \ldots, n$, onda ona vrijedi i za $\max\limits_{k=1,\ldots,n} |\Delta y_k|$. Korištenjem ∞-norme vektora i matrica dobivamo

$$\|\Delta y\|_\infty \leq \left\|\frac{\partial f}{\partial x}\right\|_\infty \|\Delta x\|_\infty.$$

Konačno, za relativne perturbacije po normi dobivamo

$$\frac{\|\Delta y\|_\infty}{\|y\|_\infty} \leq \frac{\|x\|_\infty \left\|\frac{\partial f}{\partial x}\right\|_\infty}{\|f(x)\|_\infty} \cdot \frac{\|\Delta x\|_\infty}{\|x\|_\infty}.$$

Može se pokazati da je prethodna nejednakost oštra, tj. da postoji perturbacija Δx za koju se ona dostiže. To opravdava definiciju globalne uvjetovanosti u obliku

$$(\operatorname{cond} f)(x) := \frac{\|x\|_\infty \left\|\frac{\partial f}{\partial x}\right\|_\infty}{\|f(x)\|_\infty}. \tag{4.2.4}$$

Primijetite da je ova uvjetovanost mnogo grublja nego ona u (4.2.2), jer norma pokušava "uništiti" detalje o komponentama vektora. Na primjer, ako x ima komponente bitno različitih redova veličina, samo će po apsolutnoj vrijednosti najveća igrati neku ulogu, a ostale će biti zanemarene.

Isti oblik broja uvjetovanosti iz (4.2.4) možemo dobiti u bilo kojoj vektorskoj i njoj pripadnoj operatorskoj normi. To izlazi direktno iz (4.2.3), jer barem približno, za male perturbacije, vrijedi

$$\|\Delta y\| \lesssim \left\|\frac{\partial f}{\partial x}\right\| \|\Delta x\|,$$

a ostatak ide analogno.

Primjer 4.2.2. *Ispitajmo uvjetovanost problema*

$$f(x) = \begin{bmatrix} \dfrac{1}{x_1} + \dfrac{1}{x_2} \\ \dfrac{1}{x_1} - \dfrac{1}{x_2} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Uzmemo li kao uvjetovanost definiciju (4.2.2)*, prvo treba izračunati elemente matrice* $\Gamma(x)$*. Izlazi*

$$\gamma_{11} = \left|\frac{x_2}{x_1 + x_2}\right|, \quad \gamma_{12} = \left|\frac{x_1}{x_1 + x_2}\right|, \quad \gamma_{21} = \left|\frac{x_2}{x_2 - x_1}\right|, \quad \gamma_{22} = \left|\frac{x_1}{x_2 - x_1}\right|.$$

što odmah ukazuje na lošu uvjetovanost (engl. ill–conditioning) za $x_1 \approx \pm x_2$*, uz uvjet da* $|x_1|$ *(a onda i* $|x_2|$*) nisu mali. Za broj uvjetovanosti* $\|\Gamma(x)\|_F$ *dobivamo*

$$\|\Gamma(x)\|_F = \sqrt{2}\,\frac{x_1^2 + x_2^2}{|x_1^2 - x_2^2|},$$

što ponovno pokazuje istu lošu uvjetovanost za $x_1 \approx \pm x_2$*.*

Ako za uvjetovanost uzmemo definiciju (4.2.4) *u* ∞*-normi, onda je*

$$(\operatorname{cond} f)(x) = \frac{\max\{|x_1|, |x_2|\} \cdot (x_1^2 + x_2^2)}{|x_1 x_2| \cdot \max\{|x_1 + x_2|, |x_2 - x_1|\}}.$$

Uvrstimo li $x_1 \approx \pm x_2$*, dobivamo da je* $(\operatorname{cond} f)(x) \approx 2$*, što očito vodi na pogrešan zaključak da je problem dobro uvjetovan i neosjetljiv na perturbacije za* $x_1 \approx \pm x_2$*.*

Primjer 4.2.3. *Ispitajmo uvjetovanost problema računanja integrala*

$$I_n = \int_0^1 \frac{t^n}{t+5}\,dt$$

za fiksni prirodni broj n*. U napisanom obliku ovaj problem zadaje funkciju s* $\mathbb{N}$ *u* $\mathbb{R}$ *i ne odgovara našem pojmu problema iz* (4.1.1)*, jer je* $\mathbb{N}$ *diskretan skup i nema pojma malih perturbacija.*

Međutim, uzmimo da ovaj integral računamo rekurzivno, koristeći vezu između susjednih integrala I_k *i* I_{k-1}*, s tim da početni integral* I_0 *znamo izračunati*

$$I_0 = \int_0^1 \frac{1}{t+5}\,dt = \ln(t+5)\,\Big|_0^1 = \ln\frac{6}{5}. \tag{4.2.5}$$

Za nalaženje rekurzije, primijetimo da je

$$\frac{t}{t+5} = 1 - \frac{5}{t+5},$$

pa moženjem obje strane s t^{k-1} *i integracijom od* 0 *do* 1 *dobivamo željenu rekurzivnu relaciju*

$$I_k = \int_0^1 t^{k-1}\,dt - 5I_{k-1} = \frac{1}{k} - 5I_{k-1}, \quad k = 1, 2, \ldots, n.$$

Vidimo da je niz vrijednosti I_k *rješenje (linearne, nehomogene) diferencijske jednadžbe (prvog reda s konstantnim koeficijentima)*

$$y_k = -5y_{k-1} + \frac{1}{k}, \quad k = 1, 2, \ldots, \tag{4.2.6}$$

s početnim uvjetom $y_0 = I_0$.

Ako pustimo početni uvjet varira, dobivamo željeni oblik problema. Ova rekurzija definira niz funkcija $f_k : \mathbb{R} \to \mathbb{R}$ *koje direktno vežu* y_k *i* y_0, *tj.*

$$y_k = f_k(y_0).$$

Da bismo izračunali I_n, *treba uzeti* $y_0 = I_0$, *primijeniti relaciju* (4.2.6) *redom za* $k = 1, 2, \ldots, n$, *a rezultat je* $y_n = f_n(y_0) = I_n$.

Želimo naći uvjetovanost funkcije f_n *u točki* $y_0 = I_0$ *iz relacije* (4.2.5), *u ovisnosti o parametru* $n \in \mathbb{N}$. *Prvo, primijetimo da početna vrijednost* I_0 *nije egzaktno prikaziva u računalu, i pri spremanju se mora zaokružiti. Umjesto* I_0, *spremi se* $\widehat{I}_0$. *Čak da se u procesu računanja ne dogodi više niti jedna jedina greška, rezultat će biti neka aproksimacija*

$$\widehat{I}_n = f_n(\widehat{I}_0).$$

Iz relacije (4.2.6) *slijedi da je* f_n *linearna (preciznije, afina) funkcija od* y_0. *Na primjer, indukcijom, lako izlazi*

$$y_n = f_n(y_0) = (-5)^n y_0 + p_n,$$

gdje je p_n *neki broj koji ne ovisi o* y_0, *nego samo o nehomogenim članovima rekurzije. Po definiciji broja uvjetovanosti* (4.2.1), *dobivamo*

$$(\operatorname{cond} f_n)(y_0) = \left|\frac{y_0 f_n'(y_0)}{y_n}\right| = \left|\frac{y_0(-5)^n}{y_n}\right|.$$

Za $y_0 = I_0$, *znamo da je* $y_n = I_n$. *Osim toga, iz definicije integrala* I_n *vidimo da* I_n *monotono padaju po* n, *čak vrijedi* $\lim_{n\to\infty} I_n = 0$. *Dakle, zbrajanjima dobivamo sve manje i manje brojeve, što ne sluti na dobro. Zaista, broj uvjetovanosti to i pokazuje*

$$(\operatorname{cond} f_n)(I_0) = \frac{I_0 \cdot 5^n}{I_n} > \frac{I_0 \cdot 5^n}{I_0} = 5^n.$$

Dakle, f_n *je vrlo loše uvjetovana u* $y_0 = I_0$, *i to tim gore kad* n *raste.*

Naravno, to se odmah vidi i iz rekurzije (4.2.6). *Stalno množimo s* (-5), *što povećava vrijednosti, a mi trebamo dobiti sve manje i manje vrijednosti. Stoga, u zbrajanjima neprestano moramo imati sve veća i veća kraćenja, dok se ne izgubi bilo kakva informacija.*

Napomenimo da za približne vrijednosti $\widehat{I}_n$ *i* $\widehat{I}_0$ *egzaktno vrijedi veza relativnih perturbacija*

$$\left|\frac{\widehat{I}_n - I_n}{I_n}\right| = (\operatorname{cond} f_n)(I_0) \cdot \left|\frac{\widehat{I}_0 - I_0}{I_0}\right|,$$

i to za bilo kakve, a ne samo male perturbacije. To slijedi iz linearnosti funkcije f_n, *pa je* $f_n'' = 0$ *u Taylorovoj formuli.*

Ostaje pitanje da li se ova loša uvjetovanost može nekako izbjeći. Rješenje se može naslutiti iz prethodne primjedbe. Umjesto množenja velikim brojem, radije bismo dijelili velikim brojem, posebno ako rezultati rastu, a ne padaju. To se postiže okretanjem rekurzije (4.2.6). *Treba uzeti neki* $\nu > n$ *i računati*

$$y_{k-1} = \frac{1}{5}\left(\frac{1}{k} - y_k\right), \quad k = \nu, \nu-1, \ldots, n+1. \tag{4.2.7}$$

Problem je, naravno, kako izračunati početnu vrijednost y_ν.

Prije toga, uočimo da rekurzija (4.2.7) *definira novi niz funkcija* $g_k : \mathbb{R} \to \mathbb{R}$, *s tim da se naš problem svodi na računanje funkcije* g_n *koja direktno veže* y_n *i* y_ν, *uz* $\nu > n$, *tj.*

$$y_n = g_n(y_\nu).$$

Kao i ranije, g_n *je linearna (afina) funkcija od* y_ν, *pa na isti način dobivamo da je uvjetovanost*

$$(\operatorname{cond} g_n)(y_\nu) = \left|\frac{y_\nu\,(-1/5)^{\nu-n}}{y_n}\right|, \quad \nu > n.$$

Za $y_\nu = I_\nu$, *znamo da je* $y_n = I_n$, *a iz monotonosti* I_n *slijedi*

$$(\operatorname{cond} g_n)(I_\nu) = \frac{I_\nu}{I_n} \cdot \left(\frac{1}{5}\right)^{\nu-n} < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n,$$

što je ispod 1, *tj. greške se prigušuju. Osim toga, faktor prigušenja pada kad* ν *raste, obzirom na* n.

Ako je $\widehat{I}_\nu$ *neka aproksimacija za* I_ν, *onda za relativne perturbacije vrijedi*

$$\left|\frac{\widehat{I}_n - I_n}{I_n}\right| = (\operatorname{cond} g_n)(I_\nu) \cdot \left|\frac{\widehat{I}_\nu - I_\nu}{I_\nu}\right| < \left(\frac{1}{5}\right)^{\nu-n} \cdot \left|\frac{\widehat{I}_\nu - I_\nu}{I_\nu}\right|.$$

Zbog linearnosti funkcije g_n, *ova relacija vrijedi za bilo kakve perturbacije, a ne samo male. Drugim riječima, početna vrijednost* $\widehat{I}_\nu$ *uopće ne mora biti blizu prave*

I_ν. Uzmemo li $\widehat{I}_\nu = 0$, čime smo napravili relativnu grešku od 100% *u početnoj vrijednosti, još uvijek dobivamo $\widehat{I}_n$ s relativnom greškom*

$$\left|\frac{\widehat{I}_n - I_n}{I_n}\right| < \left(\frac{1}{5}\right)^{\nu-n}, \quad \nu > n.$$

Povoljnim izborom ν, ocjenu na desnoj strani možemo napraviti po volji malom, recimo, ispod željene točnosti ε. Dovoljno je uzeti

$$\nu \geq n + \frac{\log(1/\varepsilon)}{\log 5}.$$

Za konačni algoritam uzmemo da je ν najmanji cijeli broj za koji vrijedi prethodna relacija, definiramo $\widehat{I}_\nu = 0$ i računamo vrijednosti

$$\widehat{I}_{k-1} = \frac{1}{5}\left(\frac{1}{k} - \widehat{I}_k\right), \quad k = \nu, \nu-1, \ldots, n+1.$$

U egzaktnoj aritmetici $\widehat{I}_n$ ima relativnu grešku ispod ε. Čak i uz greške zaokruživanja u ovom postupku, još uvijek dobivamo izračunati $\widehat{I}_n$ s relativnom greškom približno jednakom $\max\{\varepsilon, u\}$*. Naime, i sve greške zaokruživanja se stalno prigušuju u ovom postupku.*

Slične ideje okretanja rekurzije imaju ogromnu primjenu kod računanja rješenja linearnih rekurzivnih relacija drugog reda. Na primjer, Besselove funkcije i mnoge druge specijalne funkcije matematičke fizike zadovoljavaju takve rekurzije.

Primjer 4.2.4. (Sustavi linearnih algebarskih jednadžbi)

Promatramo problem rješavanja linearnog sustava jednadžbi oblika

$$Ax = b,$$

gdje je $A \in \mathbb{R}^{n\times n}$ kvadratna regularna matrica reda n, a $b \in \mathbb{R}^n$ zadani vektor. Ulazni podaci su elementi od A i b (njih n^2+n), a rezultat je vektor $x \in \mathbb{R}^n$. Znamo da ovaj problem ima jedinstveno rješenje, tj. imamo korektno definiran problem, a pripadna funkcija je $\mathbb{R}^{n^2+n} \to \mathbb{R}^n$.

Da bismo pojednostavnili stvari, pretpostavimo da je A fiksna zadana matrica koja se ne mijenja (perturbira). Dozvoljene su perturbacije samo vektora b desne strane sustava. Pripadna funkcija ovog problema je $f : \mathbb{R}^n \to \mathbb{R}^n$, zadana s

$$x = f(b) := A^{-1}b.$$

Opet je f linearna funkcija, pa je njena Jacobijeva matrica

$$J_f(b) = \partial f/\partial x = A^{-1}.$$

Pripadni broj uvjetovanosti funkcije f, promatranjem perturbacija po normi je, prema (4.2.4),

$$(\operatorname{cond} f)(b) := \frac{\|b\|\,\|A^{-1}\|}{\|A^{-1}b\|},$$

u bilo kojoj vektorskoj normi na $\mathbb{R}^n$ i pripadnoj operatorskoj normi. Ovaj broj, naravno, ovisi o A i o b. Želimo eliminirati ovisnost o b i dobiti broj koji ovisi samo o A. Prvo uvrstimo $Ax = b$, što daje

$$(\operatorname{cond} f)(b) = \frac{\|Ax\|\,\|A^{-1}\|}{\|x\|},$$

a onda, koristeći bijektivnu vezu x i b, tražimo najgori mogući broj uvjetovanosti po svim b, odnosno, po svim x

$$\max_{\substack{b\in\mathbb{R}^n\\ b\neq 0}}(\operatorname{cond} f)(b) = \max_{\substack{x\in\mathbb{R}^n\\ x\neq 0}} \frac{\|Ax\|}{\|x\|}\cdot\|A^{-1}\| = \|A\|\cdot\|A^{-1}\|,$$

po definiciji operatorske norme od A. Desna strana više ne ovisi o vektoru b i možemo ju interpretirati kao broj uvjetovanosti matrice A linearnog sustava, pa definiramo

$$\operatorname{cond} A := \|A\|\cdot\|A^{-1}\|. \tag{4.2.8}$$

Bitno je reći da ovaj broj mjeri uvjetovanost linearnog sustava s matricom A, a ne uvjetovanost drugih problema ili veličina vezanih uz matricu A, poput svojstvenih vrijednosti (iako i tamo ima ulogu).

Iako je izveden promatranjem perturbacija samo desne strane b, broj uvjetovanosti iz (4.2.8) *ima bitno značenje i kad dozvolimo perturbacije u matrici A. Moramo se ograničiti na dovoljno male perturbacije, tako da sustav ostane regularan — na primjer, takve da je $\|\Delta A\|\cdot\|A^{-1}\| < 1$.*

5. Rješavanje linearnih sustava

5.1. Kako se sustavi rješavaju u praksi

Zadani su matrica $A \in \mathbb{C}^{m\times n}$ i vektor $b \in \mathbb{C}^m$. Teorem Kronecker–Capelli daje odgovor na pitanje kad linearni sustav

$$Ax = b \tag{5.1.1}$$

ima rješenje $x \in \mathbb{C}^n$ i kad je ono jedinstveno.

U praksi se najčešće rješavaju linearni sustavi kad je matrica sustava A kvadratna i nesingularna. Tada znamo da sustav ima jedinstveno rješenje.

Kako bi se moglo izračunati rješenje takvog sustava? Na primjer, mogao bi se izračunati inverz A^{-1}, pa množenjem relacije (5.1.1) slijeva s A^{-1} dobivamo

$$x = A^{-1}b.$$

Time nam je odmah dana i jedna metoda za rješavanje linearnog sustava (5.1.1) koja je sasvim nepraktična, jer smo rješavanje linearnog sustava preveli u računanje inverza, što je teži problem. Naime, j-ti stupac inverza je rješenje sustava $Ax = e_j$.

Druga metoda, koja se često spominje u linearnoj algebri je Cramerovo pravilo. Prisjetimo se, j-ta komponenta rješenja sustava je

$$x_j = \frac{\det A_j}{\det A},$$

pri čemu je matrica A_j jednaka matrici A, osim što je j-ti stupac u A_j zamijenjen desnom stranom b. Složenost ovog načina rješavanja je eksponencijalna (dokažite to!) i nikad se ne koristi kao metoda numeričkog rješavanja.

Najjednostavnija metoda za rješavanje linearnog sustava (5.1.1) je njegovo svođenje na trokutastu formu $Rx = y$, gdje je R trokutasta matrica (recimo, gornja), iz koje se lako, tzv. povratnom supstitucijom, nalazi rješenje.

Gaussove eliminacije su metoda direktnog transformiranja linearnog sustava $Ax = b$, zajedno s desnom stranom, na trokutastu formu. Gaussove eliminacije

možemo implementirati i tako da se desna strana ne transformira istovremeno kad i matrica A. Tada se formiraju dvije matrice L i R (R je trokutasta matrica iz Gaussovih eliminacija) i koristeći njih lako se dobije rješenje traženog sustava. Kad se Gaussove eliminacije tako implementiraju, metoda se obično zove LR (neki to zovu LU) faktorizacija matrice A. Ovaj pristup je posebno zgodan kad imamo više linearnih sustava s istom matricom A, a desne strane se razlikuju.

Mnogi sustavi linearnih jednadžbi imaju specijalna svojstva i specijalnu strukturu. U nekim od tih slučajeva, koji su za praksu jako bitni, mogu se primijeniti i iterativne metode, koje daju rješenje linearnog sustava (5.1.1) na zadovoljavajuću točnost mnogo brže nego LR faktorizacija.

5.2. Gaussove eliminacije

Elementarne transformacije su one koje ne mijenjaju rješenje linearnog sustava. Takve transformacije su: množenje jednadžbe konstantom različitom od 0, dodavanje jednadžbe (ili linearne kombinacije jednadžbi) drugim jednadžbama i zamjena poretka jednadžbi. Korištenjem elementarnih transformacija, svaki se linearni sustav s kvadratnom nesingularnom matricom može svesti na trokutasti oblik.

Označimo $A^{(1)} := A$, $b^{(1)} := b$. U skraćenoj notaciji, bez pisanja nepoznanica x_i, linearni sustav (5.1.1) možemo zapisati proširenom matricom, kao

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} & b_n^{(1)} \end{array}\right].$$

Počnimo sa svođenjem matrice na trokutastu formu. Za prvi stupac to znači da u tom stupcu moramo poništiti sve elemente, osim prvog. Ako je element $a_{11}^{(1)} \neq 0$, onda redom, možemo od i-te jednadžbe oduzeti prvu jednadžbu pomnoženu s

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad i = 2, \ldots, n.$$

Prva jednadžba se ne mijenja. Time smo dobili ekvivalentni linearni sustav

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} & b_n^{(2)} \end{array}\right].$$

Postupak poništavanja možemo nastaviti s drugim stupcem matrice $A^{(2)}$, od dijagonale nadolje. Ako je $a_{22}^{(2)} \neq 0$, biramo faktore

$$m_{i2} = \frac{a_{i2}^{(2)}}{a_{22}^{(2)}}, \quad i = 3, \ldots, n,$$

tako da poništimo sve elemente drugog stupca ispod dijagonale. I tako redom. Konačno, ako su svi $a_{ii}^{(i)} \neq 0$, za $i = 1, \ldots, n-1$, završni linearni sustav, ekvivalentan polaznom, je

$$\left[\begin{array}{cccc:c} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} & b_1^{(1)} \\ & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} & b_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & a_{nn}^{(n)} & b_n^{(n)} \end{array}\right].$$

Uz pretpostavku da je $a_{nn}^{(n)} \neq 0$, ovaj se linearni sustav lako rješava povratnom supstitucijom

$$\begin{aligned} x_n &= \frac{b_n^{(n)}}{a_{nn}^{(n)}}, \\ x_i &= \frac{1}{a_{ii}^{(i)}} \left(b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j \right), \quad i = n-1, \ldots, 1. \end{aligned}$$

Prvo pitanje koje se nameće je moraju li svi $a_{ii}^{(i)}$ biti različiti od nule, ako je A regularna i kvadratna. Jasno je da ne moraju. Na primjer, matrica linearnog sustava

$$\left[\begin{array}{cc:c} 0 & 1 & 1 \\ 1 & 0 & 1 \end{array}\right]$$

je regularna ($\det A = -1$), sustav ima jedinstveno rješenje $x_1 = x_2 = 1$, a ipak ga ne možemo riješiti Gaussovim eliminacijama ako ne mijenjamo poredak jednadžbi.

Zamjena bilo koje dvije jednadžbe neće promijeniti rješenje sustava. Dakle, ako je $a_{11} = 0$, prije eliminacije elemenata prvog stupca, moramo izabrati ne-nula element u prvom stupcu, zovimo ga a_{r1}, a zatim zamijeniti prvu i r-tu jednadžbu.

Ponovno, nismo sigurni je li to uvijek moguće. No, ako u prvom stupcu ne postoji ne-nula element, matrica A ima nul-stupac za prvi stupac, pa ne može biti regularna. Pokažite da isti argument vrijedi za svaku od matrica u procesu eliminacije, tj. ako su u k-tom koraku eliminacije svi elementi matrice $A^{(k)}$ na ili ispod glavne dijagonale u k-tom stupcu jednaki 0, onda je matrica A singularna.

Dakle, ako je A nesingularna, onda u svakom koraku k ($k = 1, \ldots, n-1$) eliminacije, u matrici $A^{(k)}$ možemo naći element $a_{rk}^{(k)} \neq 0$, uz $r \geq k$, kojeg zamjenom

jednadžbi r i k dovodimo na dijagonalu, tako da je $a_{kk}^{(k)} \neq 0$, a zatim računamo matricu $A^{(k+1)}$. Takve ne-nula elemente koje dovodimo na dijagonalu zovemo pivotnim elementima.

Drugo je pitanje, da li je dovoljno dobro birati pivotne elemente samo tako da budu ne-nula, što je, u principu, dovoljno da provedemo postupak eliminacije. Primjer 2.7.4. pokazuje da biranje veličine pivotnog elementa nije beznačajno.

Uobičajeno **parcijalno pivotiranje** kao pivotni element bira element koji je po apsolutnoj vrijednosti najveći u ostatku tog stupca — na glavnoj dijagonali ili ispod nje. Drugim riječima, ako je u k-tom koraku

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|,$$

onda ćemo zamijeniti r-ti i k-ti redak i početi korak eliminacije elemenata k-tog stupca.

Motivacija za takvo biranje pivotnih elemenata je jednostavna. Elementi "ostatka" linearnog sustava koje treba izračunati u matrici $A^{(k+1)}$ u k-tom koraku transformacije su

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}, \tag{5.2.1}$$

za $i, j = k+1, \ldots, n$, a multiplikatori m_{ik} su

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n. \tag{5.2.2}$$

Ako je multiplikator m_{ik} velik, u aritmetici pomičnog zareza može doći do kraćenja najmanje značajnih znamenki $a_{ij}^{(k)}$, tako da izračunati $a_{ij}^{(k+1)}$ može imati veliku relativnu grešku. Nažalost, to kraćenje može biti ekvivalentno relativno velikoj perturbaciji u originalnoj matrici A.

Na primjer, neka je

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}, \quad \varepsilon \leq u.$$

Eliminacija elementa a_{21} u aritmetici pomičnog zareza, umjesto elementa $a_{22}^{(2)}$, daje

$$fl(a_{22}^{(2)}) = fl\left(1 - \frac{1}{\varepsilon}\right) = -\frac{1}{\varepsilon}, \tag{5.2.3}$$

zbog $1 \ll 1/\varepsilon$. Kad bismo u originalnoj matrici A promijenili a_{22} s 1 u 0, dobili bismo isti rezultat za $fl(a_{22}^{(2)})$, s tim da je on sad i egzaktan. Drugim riječima, greška zaokruživanja napravljena u (5.2.3) ekvivalentna je velikoj relativnoj perturbaciji u originalnoj matrici A. Pogledajte da li bi se isto dogodilo da smo zamijenili jednadžbe prije početka eliminacije.

Sasvim općenito, ideja pivotiranja je minimizirati korekcije elemenata u (5.2.1) pri prijelazu s $A^{(k)}$ na $A^{(k+1)}$. Dakle, multiplikatori u (5.2.2) trebaju biti što manji. To se postiže izborom što je moguće većeg nazivnika (po apsolutnoj vrijednosti), a to je upravo pivotni element. Primijetite da za multiplikatore kod parcijalnog pivotiranja vrijedi

$$|m_{ik}| \leq 1, \quad i = k+1, \ldots, n.$$

U praksi, parcijalno pivotiranje funkcionira izvrsno, ali matematičari su konstruirali primjere kad ono "nije savršeno". Što to točno znači, bit će rečeno u jednom od sljedećih poglavlja.

Osim parcijalnog pivotiranja, može se provoditi i **potpuno pivotiranje**. U k-tom koraku, bira se maksimalni element u cijelom "ostatku" matrice $A^{(k)}$, a ne samo u k-tom stupcu. Ako je u k-tom koraku

$$|a_{rs}^{(k)}| = \max_{k \leq i,j \leq n} |a_{ij}^{(k)}|,$$

onda ćemo zamijeniti r-ti i k-ti redak, s-ti i k-ti stupac i početi korak eliminacije elemenata k-tog stupca. Ipak, trebamo biti malo oprezni. Zamjenom s-tog i k-tog stupca zamijenili smo ulogu varijabli x_s i x_k. Sve takve promjene treba pamtiti u vektoru permutacije varijabli. Osim toga, u usporedbi s parcijalnim pivotiranjem, imamo mnogo više pretraživanja u svakom koraku ($(n-k+1)^2$ elemenata, prema ranijih $n-k+1$), što usporava proces. Ipak, korištenjem potpunog pivotiranja mogu se izvesti bolje ocjene greške nego kod parcijalnog pivotiranja.

Ovo nisu jedine mogućnosti pivotiranja kod rješavanja linearnih sustava. Rutishauser je početkom sedamdesetih godina opisao relativno parcijalno pivotiranje, ali algoritam nije ušao u široku upotrebu.

Napišimo sad algoritam koji korištenjem Gaussovih eliminacija rješava linearni sustav $Ax = b$. Sve transformacije provodimo u istim poljima A i b koja na početku sadrže ulazne podatke.

Algoritam 5.2.1. (Gaussove eliminacije s parcijalnim pivotiranjem)

```
    {Trokutasta redukcija}
  for k := 1 to n - 1 do
    begin
      {Nađi maksimalni element u ostatku stupca}
    max_elt := 0.0;
    ind_max := k;
    for i := k to n do
      if abs(A[i, k]) > max_elt then
        begin
        max_elt := abs(A[i, k]);
```

```
        ind_max := i;
        end;
    if max_elt > 0.0 then
      begin
      if ind_max <> k then
          {Zamijeni k-ti i ind_max-ti redak}
        begin
        for j := k to n do
          begin
          temp := A[ind_max, j];
          A[ind_max, j] := A[k, j];
          A[k, j] := temp;
          end;
        temp := b[ind_max];
        b[ind_max] := b[k];
        b[k] := temp;
        end;
      for i := k + 1 to n do
        begin
        mult := A[i, k]/A[k, k];
        A[i, k] := 0.0;   {Ne treba, ne koristi se kasnije}
        for j := k + 1 to n do
          A[i, j] := A[i, j] − mult ∗ A[k, j];
        b[i] := b[i] − mult ∗ b[k];
        end;
      end
    else
        {Matrica je singularna, stani s algoritmom}
      begin
      error := true;
      exit;
      end;
    end;
    {Povratna supstitucija, rješenje x ostavi u b}
b[n] := b[n]/A[n, n];
for i := n − 1 downto 1 do
    begin
    sum := b[i];
    for j := i + 1 to n do
      sum := sum − A[i, j] ∗ b[j];
    b[i] := sum/A[i, i];
    end;
error := false;
```

Zadatak 5.2.1. *Pokušajte samostalno napisati algoritam koji koristi potpuno pivotiranje. Posebnu pažnju obratite na efikasno pamćenje zamjena varijabli koje su posljedica zamjena stupaca. Može li se isti princip efikasno primijeniti i za pamćenje zamjena redaka, tako da se potpuno izbjegnu eksplicitne zamjene elemenata u matrici A i vektoru b?*

Prebrojimo sve aritmetičke operacije ovog algoritma da bismo dobili jednostavnu mjeru složenosti Gaussovih eliminacija. U prvom koraku trokutaste redukcije obavlja se:

- $n-1$ dijeljenje — računanje *mult*,
- $n(n-1)$ množenje — za svaki od $n-1$ redaka po $n-1$ množenje za računanje elemenata matrice A i jedno množenje za računanje elementa vektora b,
- $n(n-1)$ oduzimanje — javlja se u istoj naredbi gdje i prethodna množenja.

Na sličan način zaključujemo da se u k-tom koraku obavlja:

- $n-k$ dijeljenja,
- $(n-k+1)\,(n-k)$ množenja i $(n-k+1)\,(n-k)$ oduzimanja.

Ukupno, u k-tom koraku imamo

$$n-k+2(n-k+1)\,(n-k) = 2(n-k)^2 + 3(n-k)$$

aritmetičkih operacija.

Broj koraka k varira od 1 do $n-1$, pa je ukupan broj operacija potrebnih za svođenje na trokutastu formu jednak

$$\sum_{k=1}^{n-1} [2(n-k)^2 + 3(n-k)] = \sum_{k=1}^{n-1} (2k^2 + 3k) = \frac{1}{6}(4n^3 + 3n^2 - 7n).$$

Druga suma u prošloj jednakosti dobije se iz prve zamjenom indeksa $n-k \to k$.

Potpuno istim zaključivanjem dobivamo da u povratnoj supstituciji ima:

- $(n-1)\,n/2$ množenja i $(n-1)\,n/2$ zbrajanja,
- n dijeljenja,

što je zajedno točno n^2 operacija.

Dakle, ukupan broj operacija u Gaussovim eliminacijama je

$$OP(n) = \frac{1}{6}(4n^3 + 9n^2 - 7n),$$

što je približno $2n^3/3$, za malo veće n.

Ovaj broj je najjednostavnija mjera efikasnosti ili složenosti Gaussovih eliminacija. Uočimo da ova mjera ignorira pivotiranje, jer tamo nema vidljivih aritmetičkih operacija. Međutim, uspoređivanje dva realna broja u floating point aritmetici se obično radi oduzimanjem ta dva broja i usporedbom rezultata s nulom. U tom smislu, sve takve usporedbe bi, također, trebalo brojati. Nađite njihov broj za parcijalno i potpuno pivotiranje.

Ako se u Gaussovim eliminacijama poništavaju ne samo elementi ispod dijagonale, nego i iznad nje, dobivamo tzv. Gauss–Jordanovu metodu, koja linearni sustav svodi na ekvivalentni dijagonalni sustav. Gauss–Jordanove eliminacije se danas rijetko koriste u praksi, jer zahtijevaju previše računskih operacija.

Zadatak 5.2.2. *Napišite taj algoritam i pokažite da je broj računskih operacija, ne brojeći uspoređivanja, u tom slučaju jednak*

$$OP(n) = n^3 + n^2 - n.$$

To je skoro 50% *više računskih operacija nego u običnim Gaussovim eliminacijama.*

5.3. LR faktorizacija

U praksi se linearni sustavi najčešće rješavaju korištenjem LR faktorizacije. Pretpostavimo da smo dobili matricu A faktoriziranu u obliku

$$A = LR, \tag{5.3.1}$$

pri čemu je L donjetrokutasta matrica s jedinicama na dijagonali, a R gornjetrokutasta. Matrica L je regularna i vrijedi $\det L = 1$, pa regularnost matrice A povlači i regularnost matrice R, jer iz (5.3.1) slijedi

$$\det A = \det L \cdot \det R = \det R.$$

Rješenje linearnog sustava (5.1.1) sad se svodi samo na dva rješavanja trokutastih sustava. Kako? Polazni sustav u faktoriziranoj formi ima oblik

$$LRx = b.$$

Označimo li $y = Rx$, dobivamo dva sustava

$$Ly = b, \quad Rx = y.$$

Oba sustava lako se rješavaju: prvi — supstitucijom unaprijed

$$\begin{aligned} y_1 &= b_1 \\ y_i &= b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j, \quad i = 2, \ldots, n, \end{aligned}$$

a drugi — povratnom supstitucijom

$$x_n = \frac{y_n}{r_{nn}}$$
$$x_i = \frac{1}{r_{ii}}\Big(y_i - \sum_{j=i+1}^{n} r_{ij}\, x_j\Big), \quad i = n-1, \ldots, 1.$$

Zašto je takva faktorizacija korisna? Na primjer, ako se rješavaju linearni sustavi kojima se mijenjaju samo desne strane, onda je dovoljno imati A spremljenu u faktoriziranom obliku, a zatim riješiti već navedena dva trokutasta sustava. Naravno, prvo treba naći LR faktorizaciju matrice A.

Relacije za elemente ℓ_{ij} i r_{ij} matrica L i R dobivamo ako iskoristimo njihovu poznatu strukturu i činjenicu da njihov produkt daje A. Onda je

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} \ell_{ik}\, r_{kj},$$

s tim da je $\ell_{ii} = 1$. Iz ovih relacija računamo redom one elemente koje možemo izraziti preko poznatih veličina. Tako dobivamo rekurziju za elemente matrica L i R

$$r_{1j} = a_{1j}, \quad j = 1, \ldots, n,$$
$$\ell_{j1} = \frac{a_{j1}}{r_{11}}, \quad j = 2, \ldots, n,$$

za $i = 2, \ldots, n$:

$$r_{ij} = a_{ij} - \sum_{k=1}^{i-1} \ell_{ik}\, r_{kj}, \quad j = i, \ldots, n,$$
$$\ell_{ji} = \frac{1}{r_{ii}}\Big(a_{ji} - \sum_{k=1}^{i-1} \ell_{jk}\, r_{ki}\Big), \quad j = i+1, \ldots, n.$$

U zadnjem koraku, za $i = n$, računamo samo r_{nn}. Jedini problem u provedbi ovog algoritma je osigurati da je $r_{ii} \neq 0$. Ako znamo da to vrijedi, onda prethodne relacije daju egzistenciju i jedinstvenost matrica L i R. Sljedeći teorem daje potrebni kriterij u terminima polazne matrice A.

Teorem 5.3.1. *Postoji jedinstvena LR faktorizacija matrice A ako i samo ako su vodeće glavne podmatrice $A_k := A(1:k, 1:k)$, $k = 1, \ldots, n-1$, regularne. Ako je A_k singularna za neki k, faktorizacija može postojati, ali nije jedinstvena.*

Dokaz:

Dokaz se provodi indukcijom po dimenziji matrice. Pretpostavimo da su sve matrice A_k regularne. Za $k = 1$, postoji jedinstvena LR faktorizacija

$$A_1 = [1]\,[a_{11}].$$

Pretpostavimo da A_{k-1} ima jedinstvenu faktorizaciju

$$A_{k-1} = L_{k-1}\, R_{k-1}.$$

Tražimo faktorizaciju matrice A_k, gdje je

$$A_k = \begin{bmatrix} A_{k-1} & b \\ c^T & a_{kk} \end{bmatrix} = \begin{bmatrix} L_{k-1} & 0 \\ \ell^T & 1 \end{bmatrix} \begin{bmatrix} R_{k-1} & r \\ 0 & r_{kk} \end{bmatrix} := L_k\, R_k.$$

Da bi jednadžbe bile zadovoljene, mora vrijediti

$$L_{k-1} r = b, \quad R_{k-1}^T \ell = c, \quad a_{kk} = \ell^T r + r_{kk}.$$

Matrice L_{k-1} i R_{k-1} su regularne, pa postoji jedinstveno rješenje r, ℓ, pa onda i jedinstveni r_{kk}.

Pokažimo obrat, uz pretpostavku da je A nesingularna i da postoji LR faktorizacija od A. Tada je $A_k = L_k\, R_k$, za $k = 1, \ldots, n$. Budući da je A regularna, vrijedi

$$\det A = \det R = r_{11}\, r_{22} \cdots r_{nn} \neq 0.$$

Odatle slijedi

$$\det A_k = r_{11}\, r_{22} \cdots r_{kk} \neq 0,$$

tj. sve matrice A_k su regularne.

Primjer koji ilustrira da LR faktorizacija može postojati u slučaju singularne matrice A, ali da nije jedinstvena, je faktorizacija nul-matrice

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ l & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

S druge strane, matrica

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

nema LR faktorizaciju, iako je regularna. ■

Pokažimo da je matrica R dobivena LR faktorizacijom jednaka gornjetrokutastoj matrici R dobivenoj Gaussovim eliminacijama. Pretpostavimo da je $A^{(k)}$ matrica dobivena u k-tom koraku Gaussovih eliminacija. Njezina blok forma ima oblik

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ 0 & A_{22}^{(k)} \end{bmatrix},$$

pri čemu je $A_{11}^{(k)}$ trokutasta matrica reda $k-1$ (tj. dosad sređena matrica), dok su preostale dvije matrice, generalno, pune. U matričnoj notaciji, sljedeći korak

eliminacija možemo izraziti u obliku produkta

$$A^{(k+1)} = M_k A^{(k)} := \left[\begin{array}{c:ccccc} I_{k-1} & & & & & \\ \hdashline & 1 & & & & \\ & -m_{k+1,k} & 1 & & & \\ & -m_{k+2,k} & & \ddots & & \\ & \vdots & & & \ddots & \\ & -m_{n,k} & & & & 1 \end{array}\right] A^{(k)},$$

gdje su m_{ik} multiplikatori iz relacije (5.2.2). Matricu M_k možemo i kompaktno napisati kao

$$M_k = I - m_k e_k^T,$$

gdje je e_k, k-ti vektor kanonske baze, a m_k vektor s n komponenti,

$$m_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{n,k} \end{bmatrix}.$$

Primijetite da je

$$M_k^{-1} = I + m_k e_k^T,$$

jer je $e_i^T m_k = 0$ za $i \leq k$.

Prema tome je

$$M_{n-1}\, M_{n-2} \cdots M_1\, A = A^{(n)} := \widetilde{R}.$$

S druge strane, možemo dobiti i sam A

$$A = M_1^{-1}\, M_2^{-1} \cdots M_{n-1}^{-1}\, \widetilde{R} = (I + m_1 e_1^T)\,(I + m_2 e_2^T) \cdots (I + m_{n-1} e_{n-1}^T)\, \widetilde{R}$$

$$= \left(I + \sum_{i=1}^{n-1} m_i e_i^T\right) \widetilde{R} = \begin{bmatrix} 1 & & & \\ m_{21} & 1 & & \\ \vdots & m_{32} & \ddots & \\ \vdots & \vdots & & \ddots \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} \quad 1 \end{bmatrix} \widetilde{R} := \widetilde{L}\, \widetilde{R}.$$

Iz jedinstvenosti LR faktorizacije slijedi da je $\widetilde{R} = R$.

Teorem 5.3.1. i činjenica da je R iz LR faktorizacije jednak onom iz Gaussovih eliminacija, upućuju nas da pivotiranje vršimo na isti način kao i kod Gaussovih eliminacija.

Ako vršimo parcijalno pivotiranje, onda se LR faktorizacija tako dobivene matrice (permutiranih redaka) može zapisati kao

$$PA = LR,$$

pri čemu je P matrica permutacije — u svakom retku i stupcu ima točno jednu jedinicu, a ostalo su nule. Ako znamo "permutiranu" faktorizaciju, kako ćemo riješiti linearni sustav $Ax = b$? Najjednostavnije je lijevu i desnu stranu (slijeva) pomnožiti s P (P je uvijek regularna — pokažite to), pa dobivamo

$$PAx = LRx = Pb.$$

Ako vršimo potpuno pivotiranje, na kraju dobivamo LR faktorizaciju matrice koja ima permutirane retke i stupce obzirom na A, tj.

$$PAQ = LR,$$

gdje su P i Q matrice permutacije. U ovom je slučaju rješavanje linearnog sustava malo kompliciranije (skicirajte kako).

5.4. Teorija perturbacije linearnih sustava

U ovom odjeljku prezentirat ćemo rezultate klasične teorije perturbacije po normi linearnih sustava. Pitanje na koje odgovaraju takve teorije perturbacije je koliko se (po normi) rješenje linearnog sustava (5.1.1) promijeni ako se po normi malo promijene A, b ili oba.

Da bismo izbjegli pisanje indeksa normi, sve norme koje ćemo u ovom poglavlju koristiti bit će konzistentne matrične norme i njima odgovarajuće vektorske norme (na primjer, p-norme).

Pretpostavimo da, umjesto sustava (5.1.1), egzaktno rješavamo sustav

$$(A + \Delta A)(x + \Delta x) = b, \tag{5.4.1}$$

tj. samo je matrica sustava malo perturbirana. Možemo pretpostaviti da je norma perturbacije mala prema normi polazne matrice

$$\|\Delta A\| \leq \varepsilon \|A\|.$$

Zbog toga, umjesto x, dobili smo rješenje $x + \Delta x$.

Raspišimo (5.4.1) i iskoristimo (5.1.1). Izlazi

$$A\,\Delta x + \Delta A\,(x + \Delta x) = 0.$$

Množenjem slijeva s A^{-1} i sređivanjem dobivamo

$$\Delta x = -A^{-1}\,\Delta A\,(x + \Delta x).$$

Uzimanjem norme lijeve i desne strane, a zatim ocjenjivanjem odozgo, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\|\,\|\Delta A\|\,\|x + \Delta x\| \leq \varepsilon\,\|A^{-1}\|\,\|A\|\,\|x + \Delta x\| \\ &\leq \varepsilon\,\kappa(A)\,(\|x\| + \|\Delta x\|),\end{aligned}$$

pri čemu je $\kappa(A) = \|A\|\,\|A^{-1}\|$ standardna oznaka za uvjetovanost matrice A. Premještanjem na lijevu stranu svih pribrojnika koji sadrže Δx dobivamo

$$(1 - \varepsilon\,\kappa(A))\,\|\Delta x\| \leq \varepsilon\,\kappa(A)\,\|x\|.$$

Ako je $\varepsilon\,\kappa(A) < 1$, a to znači i $\|\Delta A\|\,\|A^{-1}\| < 1$, onda je

$$\|\Delta x\| \leq \frac{\varepsilon\,\kappa(A)}{1 - \varepsilon\,\kappa(A)}\,\|x\|, \tag{5.4.2}$$

što pokazuje da je pogreška u rješenju približno proporcionalna uvjetovanosti matrice A.

Pretpostavimo sad da, umjesto sustava (5.1.1), egzaktno rješavamo sustav

$$A(x + \Delta x) = b + \Delta b, \tag{5.4.3}$$

tj. samo je desna strana sustava malo perturbirana. Možemo pretpostaviti da je norma perturbacije mala prema normi vektora b

$$\|\Delta b\| \leq \varepsilon\|b\|.$$

Zbog te pertutrbacije, umjesto x, dobili smo rješenje $x + \Delta x$.

Raspišimo (5.4.3) i iskoristimo (5.1.1). Izlazi

$$A\,\Delta x = \Delta b.$$

Množenjem slijeva s A^{-1} dobivamo

$$\Delta x = A^{-1}\,\Delta b.$$

Uzimanjem norme lijeve i desne strane, a zatim ocjenjivanjem odozgo, dobivamo

$$\begin{aligned}\|\Delta x\| &\leq \|A^{-1}\|\,\|\Delta b\| \leq \varepsilon\,\|A^{-1}\|\,\|b\| \leq \varepsilon\,\|A^{-1}\|\,\|Ax\| \\ &\leq \varepsilon\,\|A^{-1}\|\,\|A\|\,\|x\| \leq \varepsilon\,\kappa(A)\,\|x\|,\end{aligned}$$

što pokazuje da je pogreška u rješenju, ponovno, proporcionalna uvjetovanosti matrice A.

Ako se istovremeno perturbiraju A i b, možemo prethodna dva pojedinačna rezultata udružiti u sljedeći teorem.

Teorem 5.4.1. *Neka je* $Ax = b$ *i*

$$(A + \Delta A)(x + \Delta x) = b + \Delta b, \tag{5.4.4}$$

gdje je $\|\Delta A\| \leq \varepsilon \|E\|$, $\|\Delta b\| \leq \varepsilon \|f\|$, *i neka je* $\varepsilon \|A^{-1}\| \|E\| < 1$. *Tada za* $x \neq 0$ *vrijedi*

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\varepsilon}{1 - \varepsilon \|A^{-1}\| \|E\|} \left(\frac{\|A^{-1}\| \|f\|}{\|x\|} + \|A^{-1}\| \|E\| \right). \tag{5.4.5}$$

Ova ocjena se može dostići barem približno, do prvog reda veličine u ε.

Dokaz:

Ocjena (5.4.5) slijedi ako od lijeve i desne strane (5.4.4) oduzmemo (5.1.1) i dobijemo

$$A\,\Delta x = \Delta b - \Delta A\, x - \Delta A\, \Delta x.$$

Množenjem s A^{-1} slijeva, a zatim korištenjem svojstva normi lako pokazujemo da vrijedi (5.4.5). Pokažite, ako je $x = 0$, onda se (5.4.5) svodi na "apsolutni" oblik

$$\|\Delta x\| \leq \frac{\varepsilon \|A^{-1}\| \|f\|}{1 - \varepsilon \|A^{-1}\| \|E\|}.$$

Ocjena se skoro dostiže za $\Delta A = \varepsilon \|E\| \|x\| wv^T$ i $\Delta b = -\varepsilon \|f\| w$, gdje je $\|w\| = 1$, $\|A^{-1}w\| = \|A^{-1}\|$, a v je vektor dualan vektoru x, tj. vrijedi $v^T x = 1$. ■

Primijetite da je u prošlom teoremu oblik ocjene za normu perturbacija polaznih podataka poopćen u sljedećem smislu. U prethodnim ocjenama koristili smo "relativni" oblik perturbacije, poput $\|\Delta A\| \leq \varepsilon \|A\|$, a ovdje smo dozvolili da je norma perturbacije manja ili jednaka normi neke proizvoljne matrice pogreške. Slično vrijedi i za normu perturbacije vektora b. Ako u teorem 5.4.1. ipak uvrstimo prirodne ograde, tj. ako uzmemo $E = A$ i $f = b$, onda se ocjena (5.4.5) može pojednostavniti.

Ovom općenitijem obliku mjerenja perturbacija možemo pridružiti sljedeći broj uvjetovanosti po normi

$$\kappa_{E,f}(A, x) := \limsup_{\varepsilon \to 0} \left\{ \frac{\|\Delta x\|}{\varepsilon \|x\|} \;\middle|\; (A + \Delta A)(x + \Delta x) = b + \Delta b, \; \|\Delta A\| \leq \varepsilon \|E\|, \; \|\Delta b\| \leq \varepsilon \|f\| \right\}.$$

Budući da je ocjena s desne strane u (5.4.5) oštra (ne može se popraviti, jer je skoro dostižna), onda je ova uvjetovanost problema po normi jednaka izrazu u zagradama s desne strane (5.4.5), tj. vrijedi

$$\kappa_{E,f}(A, x) := \frac{\|A^{-1}\| \|f\|}{\|x\|} + \|A^{-1}\| \|E\|.$$

Za izbor $E = A$, $f = b$, vrijedi da je (pokažite to!)

$$\kappa(A) \leq \kappa_{E,f}(A, x) \leq 2\kappa(A).$$

Uvrštavanjem te ocjene u relaciju (5.4.5), dobit ćemo nešto lošiju ocjenu od ranije

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{2\varepsilon\,\kappa(A)}{1 - \varepsilon\,\kappa(A)}.$$

U usporedbi s (5.4.2), ova ocjena je lošija za faktor 2, ali uključuje perturbacije od A i b, a ne samo od A. Sličan rezultat možemo dobiti kombinirajući ranije ocjene, tako da Δx rastavimo u zbroj dva dijela. Jedan je posljedica perturbacije matrice A, a drugi nastaje zbog perturbacije vektora b. Kako izgleda takva ocjena?

5.5. Pivotni rast

Tradicionalno, obratna analiza greške izražava se preko faktora rasta (engl. growth factor)

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|}.$$

U procesu Gaussovih eliminacija, očito vrijedi da je

$$|r_{ij}| = |a_{ij}^{(i)}| \leq \rho_n \max_{i,j} |a_{ij}|.$$

Sada možemo izreći klasični teorem koji govori o obratnoj grešci u terminu rasta elemenata u Gaussovim eliminacijama.

Teorem 5.5.1. (Wilkinson) *Neka je A regularna kvadratna matrica reda n i neka je $\widehat{x}$ izračunato rješenje sustava $Ax = b$ Gaussovim eliminacijama s parcijalnim pivotiranjem u aritmetici pomičnog zareza. Tada vrijedi*

$$(A + \Delta A)\,\widehat{x} = b, \quad \|\Delta A\|_\infty \leq n^2\,\frac{3nu}{1 - 3nu}\,\rho_n \|A\|_\infty,$$

uz uvjet da je $3nu < 1$, gdje je u jedinična greša zaokruživanja. ■

Pretpostavka da koristimo parcijalno pivotiranje u prethodnom teoremu, nije nužna. Naime, isto vrijedi i za Gaussove eliminacije bez pivotiranja, samo s malo drugačijom konstantom.

Korištenjem relacija za poništavanje elemenata

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)},$$

za parcijalno pivotiranje vrijedi da je

$$|a_{ij}^{(k+1)}| \leq |a_{ij}^{(k)}| + |a_{kj}^{(k)}| \leq 2 \max_{i,j} |a_{ij}^{(k)}|.$$

Prethodna ocjena, zajedno s definicijom faktora rasta daje jednostavnu ocjenu da je za parcijalno pivotiranje

$$\rho_n \leq 2^{n-1}.$$

Već je J. Wilkinson primijetio da se taj pivotni rast može dostići za sve matrice oblika

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & \ddots & & 1 \\ -1 & -1 & \ddots & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix}.$$

Za te matrice, parcijalno pivotiranje nije potrebno, a eksponencijalni rast elemenata primjećuje se u posljednjem stupcu. Ove su matrice samo jedna od klasa matrica koje dostižu takav maksimalni rast. Kasnije su N. Higham i D. Higham okarakterizirali oblik svih realnih matrica kod kojih se dostiže maksimalan pivotni rast (kod parcijalnog pivotiranja).

Ipak, ovo je samo "umjetno" konstruirani primjer, a u praksi je takvih matrica izrazito malo, pa se parcijalno pivotiranje ponaša mnogo bolje. I to je primijetio Wilkinson. Danas se tim problemom bavi N. L. Trefethen, koji je pokazao da je statistički, za razne vrste slučajnih matrica pivotni rast u prosjeku oko $n^{2/3}$.

Za potpuno pivotiranje, situacija je bitno drugačija. Označimo s ρ_n^c pivotni rast za potpuno pivotiranje. Početkom šezdesetih Wilkinson je dokazao da vrijedi

$$\rho_n^c \leq n^{1/2}(2 \cdot 3^{1/2} \cdots n^{1/(n-1)})^{1/2} \approx c\, n^{1/2}\, n^{(\log n)/4},$$

ali ta ograda nije dostižna. Ograda je bitno sporije rastuća funkcija nego što je to 2^{n-1}, ali još uvijek može biti dosta velika. Dugo se smatralo da je $\rho_n^c \leq n$, a tek je 1991. ta slutnja oborena na matrici reda 13, kad je nađen faktor rasta 13.0205. Kasnije je pokazano da, na primjer, za matricu reda 25, ρ_n^c može doseći 32.986341. Ako promatramo

$$g(n) = \sup_{A \in \mathbb{R}^{n \times n}} \rho_n^c(A),$$

poznati su još i sljedeći rezultati

n	2	3	4	5
$g(n)$	2	2.25	4	< 5.005.

5.6. Posebni tipovi matrica

Za posebne tipove matrica, katkad je moguće reći nešto više o ponašanju Gaussovih eliminacija, naročito o potrebi za pivotiranjem i veličini faktora rasta.

Za kompleksnu matricu $A \in \mathbb{C}^{n\times n}$ reći ćemo da je dijagonalno dominantna po recima ako vrijedi

$$\sum_{\substack{j=1\\ i\neq j}}^{n} |a_{ij}| \leq |a_{ii}|, \quad i = 1, \ldots, n.$$

Ako vrijedi stroga nejednakost za sve $i = 1, \ldots, n$, onda kažemo da je A strogo dijagonalno dominantna po recima. Matrica A je (strogo) dijagonalno dominantna po stupcima, ako je A^* (strogo) dijagonalno dominantna po recima.

U oba su slučaja Gaussove eliminacije savršeno stabilne i bez pivotiranja.

Teorem 5.6.1. (Wilkinson) *Neka je $A \in \mathbb{C}^{n\times n}$ regularna matrica. Ako je A dijagonalno dominantna po recima ili stupcima, tada A ima LR faktorizaciju (bez pivotiranja!) i za faktor rasta vrijedi $\rho_n \leq 2$. Ako je A dijagonalno dominantna po stupcima, tada je $|\ell_{ij}| \leq 1$ za sve i, j u LR faktorizaciji bez pivotiranja (pa parcijalno pivotiranje ne radi nikakve zamjene redaka).*

Dokaz:

Prvo uočimo da pretpostavka regularnosti matrice A osigurava da dijagonalni elementi nisu nula, tj. vrijedi $a_{ii} \neq 0$ za sve i. U suprotnom, da je $a_{ii} = 0$ za neki i, zbog dijagonalne dominantnosti i svi ostali elementi u tom retku ili stupcu morali bi biti jednaki nula, pa bi A očito bila singularna, što je protivno pretpostavci.

Pretpostavimo da je matrica A dijagonalno dominantna po stupcima. Dokaz za dijagonalno dominantne matrice po recima bit će analogan.

Na početku je $a_{11} \neq 0$, pa sigurno možemo napraviti prvi korak eliminacija (bez pivotiranja) i dobiti matricu $A^{(2)}$ oblika

$$A^{(2)} = \begin{bmatrix} r_{11} & r_1 \\ 0 & S \end{bmatrix}.$$

Prvi redak u $A^{(2)}$ je isti kao u A, a eliminacije nastavljamo na matrici S. Očito je da S mora biti regularna, na primjer, preko determinanti, zbog $r_{11} = a_{11}$ i $\det(A) = r_{11}\det(S) \neq 0$. Moramo još pokazati da je matrica S ponovno dijago-

nalno dominantna po stupcima. Za $j = 2, \ldots, n$ vrijedi

$$\begin{aligned}
\sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij}^{(2)}| &= \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij} - a_{i1}a_{11}^{-1}a_{1j}| \leq \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij}| + |a_{11}^{-1}|\,|a_{1j}| \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{i1}| \\
&\qquad \text{(koristimo dijagonalnu dominantnost u obje sume)} \\
&\leq (|a_{jj}| - |a_{1j}|) + |a_{11}^{-1}|\,|a_{1j}|\,(|a_{11}| - |a_{j1}|) \\
&= |a_{jj}| - |a_{1j}a_{11}^{-1}a_{j1}| \qquad \text{(koristimo } |a| - |b| \leq |a - b|) \\
&\leq |a_{jj} - a_{1j}a_{11}^{-1}a_{j1}| = |a_{jj}^{(2)}|,
\end{aligned}$$

što pokazuje da je i $A^{(2)}$ dijagonalno dominantna po stupcima.

Dakle, indukcijom zaljučujemo da je u svakom koraku algoritma matrica dijagonalno dominantna po stupcima. To znači da je u svakom stupcu maksimalni element na dijagonali, pa su pripadni $|\ell_{ij}| \leq 1$.

Dokažimo sad tvrdnju o faktoru rasta. Neka je A dijagonalno dominantna po stupcima i $A^{(k)}$ matrica dobivena nakon $k-1$ koraka eliminacija. Dokaz za dijagonalno dominantne matrice po recima bit će analogan. Tvrdimo da je

$$\max_{k\leq i,j\leq n} |a_{ij}^{(k)}| \leq 2 \max_{1\leq i,j\leq n} |a_{ij}|.$$

U prvom koraku, za $k = 2$, vrijedi

$$\begin{aligned}
\sum_{i=2}^{n} |a_{ij}^{(2)}| &= \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij} - a_{i1}a_{11}^{-1}a_{1j}| \leq \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij}| + |a_{11}^{-1}|\,|a_{1j}| \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{i1}| \\
&\leq \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij}| + |a_{11}^{-1}|\,|a_{1j}|\,(|a_{11}| - |a_{j1}|) \leq \sum_{\substack{i=2\\ i\neq j}}^{n} |a_{ij}| + |a_{1j}| - |a_{11}^{-1}|\,|a_{1j}|\,|a_{j1}| \\
&\leq \sum_{\substack{i=1\\ i\neq j}}^{n} |a_{ij}| - |a_{11}^{-1}|\,|a_{1j}|\,|a_{j1}| \leq \sum_{i=1}^{n} |a_{ij}|.
\end{aligned}$$

Analogno, u matrici $A^{(k)}$ mora vrijediti (dokaz indukcijom) da je

$$\sum_{i=k}^{n} |a_{ij}^{(k)}| \leq \sum_{i=1}^{n} |a_{ij}|.$$

Sada imamo

$$\begin{aligned}
\max_{k\leq i,j\leq n} |a_{ij}^{(k)}| &\leq \max_{k\leq j\leq n} \sum_{i=k}^{n} |a_{ij}^{(k)}| \leq \max_{k\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| \\
&\qquad \text{(koristimo dijagonalnu dominantnost po stupcima)} \\
&\leq 2 \max_{k\leq j\leq n} |a_{jj}| \leq 2 \max_{1\leq j\leq n} |a_{jj}| \\
&\qquad \text{(koristimo dijagonalnu dominantnost po stupcima)} \\
&\leq 2 \max_{1\leq i,j\leq n} |a_{ij}|,
\end{aligned}$$

što pokazuje da faktor rasta ne prelazi 2. ■

Prethodni teorem može se dokazati i u općenitijoj formi za blok LR faktorizaciju i blok dijagonalno dominantne matrice.

Posebnoj vrsti matrica pripadaju i trodijagonalne matrice oblika

$$A = \begin{bmatrix} d_1 & e_1 & & & \\ c_2 & d_2 & e_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1} & d_{n-1} & e_{n-1} \\ & & & c_n & d_n \end{bmatrix}.$$

Pretpostavimo da postoji LR faktorizacija bez pivotiranja za matricu A. Tada nije teško pokazati da su matrice L i R oblika

$$L = \begin{bmatrix} 1 & & & & \\ \ell_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ell_{n-1} & 1 & \\ & & & \ell_n & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} r_1 & e_1 & & & \\ & r_2 & e_2 & & \\ & & \ddots & \ddots & \\ & & & r_{n-1} & e_{n-1} \\ & & & & r_n \end{bmatrix}. \tag{5.6.1}$$

Primijetite da je dijagonala iznad glavne jednaka u matricama A i R. Ostale elemente matrica L i R računamo po sljedećim rekurzijama

$$\begin{aligned} &r_1 = d_1, \\ &\text{za } i = 2, \ldots, n: \\ &\quad \ell_i = c_i / r_{i-1}, \\ &\quad r_i = d_i - \ell_i \, e_{i-1}. \end{aligned} \tag{5.6.2}$$

Može se pokazati da za izračunato rješenje linearnog sustava $Ax = b$, nakon ovakve LR faktorizacije i supstitucija unaprijed i unatrag, u aritmetici pomičnog zareza, vrijedi

$$(A + \Delta A)\,\widehat{x} = b, \quad |\Delta A| \leq (4u + 3u^2 + u^3)|\widehat{L}|\,|\widehat{R}|.$$

Naravno, ova ocjena obratne greške vrijedi za bilo koju nesingularnu trodijagonalnu matricu za koju postoji LR faktorizacija bez pivotiranja. Zainteresirani smo za nalaženje onih klasa matrica za koje će vrijediti ocjena oblika

$$|\Delta A| \leq g(u)\,|A|.$$

Takva će ocjena sigurno vrijediti ako je $|\widehat{L}|\,|\widehat{R}| = |\widehat{L}\widehat{R}|$. Koje su to matrice? Odgovor na to pitanje za egzaktne faktore L i R daje sljedeći teorem.

Teorem 5.6.2. *Neka je $A \in \mathbb{R}^{n\times n}$ nesingularna trodijagonalna matrica. Ako vrijedi bilo koji od uvjeta (a)–(d), onda A ima LR faktorizaciju i vrijedi $|L|\,|R| = |LR|$:*

(a) A je simetrična pozitivno definitna,

(b) A je totalno nenegativna, ili, ekvivalentno, $L \geq 0$ i $R \geq 0$,

(c) A je M-matrica, ili, ekvivalentno, L i R imaju pozitivne dijagonalne elemente i nepozitivne vandijagonalne elemente,

(d) A je po predznacima ekvivalentna matrici B koja je tipa (a)–(c), tj. A se može prikazati u obliku $A = D_1 B D_2$, gdje su $|D_1| = |D_2| = I$.

■

U praksi se često pojavljuju i dijagonalno dominantne trodijagonalne matrice, koje ne pripadaju nekom od tiova (a)–(d) iz prethodnog teorema. Za njih, općenito, ne vrijedi da je $|L|\,|R|$ jednako $|LR| = |A|$, ali ne može biti ni mnogo veći.

Teorem 5.6.3. *Neka je $A \in \mathbb{R}^{n\times n}$ nesingularna trodijagonalna matrica, dijagonalno dominantna po recima ili stupcima, i neka A ima LR faktorizaciju $A = LR$. Tada vrijedi*

$$|L|\,|R| \leq 3\,|A|.$$

■

Korištenjem prethodna dva teorema, dobivamo i ocjenu obratne greške za izračunato rješenje ovakvih specijalnih trodijagonalnih sustava.

Teorem 5.6.4. *Ako je zadana nesingularna trodijagonalna matrica A tipa* (a)–(d) *iz teorema* 5.6.2. *i ako je jedinična greška zaokruživanja u dovoljno mala, tada Gaussove eliminacije za rješavanje sustava $Ax = b$ uspješno završavaju i nalaze rješenje $\widehat{x}$ za koje vrijedi*

$$(A + \Delta A)\,\widehat{x} = b, \quad |\Delta A| \leq \frac{4u + 3u^2 + u^3}{1 - u}\,|A|.$$

Isti zaključak vrijedi i za A dijagonalno dominantnu po recima ili stupcima, ali bez ograde na u, s tim da se ocjena množi faktorom 3. ■

Posljedica ovog teorema je da pivotiranje **nije** potrebno za tipove matrica na koje se odnosi tvrdnja. Tu činjenicu ćemo kasnije više puta iskoristiti u raznim primjenama (na primjer, kod kubične spline interpolacije). Čak i više od toga, korištenje pivotiranja može pokvariti i poništiti ove rezultate o stabilnosti.

6. Faktorizacija Choleskog

6.1. Faktorizacija Choleskog

Na kraju prethodnog poglavlja vidjeli smo da simetrične pozitivno definitne matrice imaju neka dobra svojstva vezana uz LR faktorizaciju. Na primjer, njihova LR faktorizacija se može "simetrizirati", tj. napisati u obliku LDL^T, gdje je L jedinična donja trokutasta, a D dijagonalna matrica.

U nastavku, ukratko analiziramo takve simetrične faktorizacije simetričnih, a posebno, pozitivno definitnih matrica, i njima pripadne tzv. ortogonalne ili implicitne faktorizacije. Ove faktorizacije imaju ogromnu primjenu, ne samo kod rješavanja linearnih sustava, već i kod rješavanja problema svojstvenih i singularnih vrijednosti.

Simetrija i pozitivna definitnost nisu samo zgodna matematička svojstva, već imaju i svoj dublji "fizikalni" značaj. Zbog toga se simetrične pozitivno definitne matrice prirodno javljaju u numeričkom rješavanju različitih problema, poput diskretizacije diferencijalnih jednadžbi i raznih vrsta aproksimacija.

Podsjetimo, kvadratna realna matrica A je **simetrična** ako je $A^T = A$. Simetrična matrica $A \in \mathbb{R}^{n\times n}$ je **pozitivno definitna** ako je $x^T Ax > 0$ za svaki nenula vektor $x \in \mathbb{R}^n$. Poznati ekvivalentni uvjeti za pozitivnu definitnost simetrične matrice A su:

- sve vodeće glavne minore od A su pozitivne, tj. vrijedi $\det(A_k) > 0$, za $k = 1, \ldots, n$, gdje je $A_k = A(1:k, 1:k)$ vodeća glavna podmatrica od A reda k;
- sve svojstvene vrijednosti od A su pozitivne, tj. vrijedi $\lambda_k(A) > 0$, za $k = 1, \ldots, n$, gdje λ_k označava k-tu najveću svojstvenu vrijednost (silazni poredak po k). Znamo da simetrična matrica ima realne svojstvene vrijednosti, pa ima smisla govoriti o poretku. Uz ove oznake, dovoljan je zahtjev $\lambda_n(A) > 0$, za najmanju svojstvenu vrijednost.

Iz prve karakterizacije, po teoremu 5.3.1., odmah slijedi da simetrična pozitivno definitna matrica A ima LR faktorizaciju $A = LR$. Promatranjem dijagonalnih elemenata matrice R dobivamo još jednu karakterizaciju pozitivne definitnosti, koja glasi:

- matrica R ima pozitivnu dijagonalu, tj. vrijedi $r_{kk} > 0$, za $k = 1, \ldots, n$, što slijedi iz
$$r_{kk} = \frac{\det(A_k)}{\det(A_{k-1})}.$$
Ako se sjetimo da su dijagonalni elementi od R ujedno i pivotni elementi, ako koristimo pivotiranje, karakterizaciju možemo izreći i ovako: svi pivotni elementi u LR faktorizaciji od A su pozitivni. Pri tome treba biti malo oprezan, jer pivotiranje može uništiti i simetričnost i pozitivnu definitnost od A. Zbog toga se koristi tzv. simetrično pivotiranje, tj. istovremene zamjene redaka i stupaca u A, o čemu će još biti riječi malo niže.

Zbog toga što R ima pozitivnu dijagonalu, možemo tu dijagonalu $D = \text{diag}(r_{ii})$ izlučiti kao skaliranje redaka od R, što daje jediničnu gornjetrokutastu matricu, a zatim izvući drugi korijen iz dijagonale i vratiti takvu skalu na oba faktora

$$A = LR = LDR_0 = L(\sqrt{D}\sqrt{D})R_0 = (L\sqrt{D})(\sqrt{D}R_0) = L_1L_1^T = R_1^TR_1.$$

Već smo dokazali da je $R_0 = L^T$, što daje zadnje dvije jednakosti. Time dobivamo faktorizaciju oblika $A = R^TR$, gdje je R gornjetrokutasta matrica s pozitivnom dijagonalom, koja se zove **faktorizacija Choleskog**. Ova faktorizacija je toliko važna da zaslužuje i direktan dokaz.

Teorem 6.1.1. *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica. Onda postoji jedinstvena gornja trokutasta matrica $R \in \mathbb{R}^{n\times n}$ s pozitivnim dijagonalnim elementima takva da je $A = R^TR$. Drugim riječima, A ima jedinstvenu faktorizaciju Choleskog.*

Dokaz:

Dokaz se provodi indukcijom po redu n matrice. Za $n = 1$, $A = [a_{11}]$ je sigurno simetrična, a pozitivna definitnost je ekvivalentna s $a_{11} > 0$. Tada je $R = [\sqrt{a_{11}}]$ dobro definirana i očito vrijedi

$$A = [\sqrt{a_{11}}]\,[\sqrt{a_{11}}] = R^TR.$$

Pretpostavimo da tvrdnja vrijedi za matrice reda $n-1$. Neka je A bilo koja simetrična pozitivno definitna matrica reda n. Onda je vodeća glavna podmatrica $A_{n-1} = A(1:n-1, 1:n-1)$ pozitivno definitna, pa ima jedinstvenu faktorizaciju Choleskog $A_{n-1} = R_{n-1}^TR_{n-1}$. Tražimo faktorizaciju matrice A u blok zapisu oblika

$$A = \begin{bmatrix} A_{n-1} & c \\ c^T & a_{nn} \end{bmatrix} = \begin{bmatrix} R_{n-1}^T & 0 \\ r^T & r_{nn} \end{bmatrix} \begin{bmatrix} R_{n-1} & r \\ 0 & r_{nn} \end{bmatrix} := R^TR. \tag{6.1.1}$$

Množenjem faktorizacije dobivamo jednadžbe koje moraju zadovoljavati nepoznati vektor $r \in \mathbb{R}^{n-1}$ i skalar r_{nn}

$$R_{n-1}^Tr = c, \quad r^Tr + r_{nn}^2 = a_{nn}.$$

Matrica R_{n-1}^T je regularna, pa postoji jedinstveno rješenje r prvog linearnog sustava. Iz druge jednadžbe slijedi

$$r_{nn}^2 = a_{nn} - r^T r. \tag{6.1.2}$$

Da bismo dobili jedinstveno realno pozitivno rješenje za r_{nn}, treba pokazati da je lijeva ili desna strana pozitivna. Primjenom Binet–Cauchyjevog teorema u (6.1.1) dobivamo

$$0 < \det(A) = \det(R^T)\det(R) = (\det(R))^2 = (\det(R_{n-1})\, r_{nn})^2 = (\det(R_{n-1}))^2 r_{nn}^2,$$

odakle, zbog regularnosti matrice R_{n-1}, slijedi $r_{nn}^2 > 0$, pa (6.1.2) daje jedinstveni realni $r_{nn} > 0$. To ujedno dokazuje da R ima pozitivnu dijagonalu. ■

Ovaj dokaz je konstruktivan i daje jedan način za računanje faktorizacije Choleskog — matrica R se gradi stupac po stupac, od prvog prema zadnjem. Kad rješavanje donjetrokutastog sustava $R_{n-1}^T r = c$ zapišemo u obliku supstitucije unaprijed, dobivamo potrebne relacije za elemente r_{ij} matrice R.

Do tih relacija možemo doći i analognim putem kao kod LR faktorizacije. Iskoristimo poznatu strukturu od R i činjenicu da mora vrijediti $A = R^T R$. Zbog simetrije, dovoljno je gledati, recimo, gornji trokut matrice A, tj. elemente a_{ij} za $i \le j$. Množenjem izlazi

$$a_{ij} = \sum_{k=1}^{i} r_{ki}\, r_{kj}, \quad i \le j. \tag{6.1.3}$$

Ove jednadžbe rješavamo tako da računamo redom one elemente koje možemo izraziti preko već poznatih veličina. Jedan od mogućih redoslijeda je $(1,1)$, $(1,2)$, $(2,2)$, $(1,3)$, $(2,3)$, $(3,3)$, $\ldots$, (n,n), tj. stupac po stupac, od vrha stupca prema dnu. Dobivamo sljedeću rekurziju za elemente matrice R

$$\begin{aligned}
&\text{za } j = 1, \ldots, n: \\
&\qquad r_{ij} = \Big(a_{ij} - \sum_{k=1}^{i-1} r_{ki}\, r_{kj}\Big) \Big/ r_{ii}, \quad i = 1, \ldots, j-1, \\
&\qquad r_{jj} = \Big(a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2\Big)^{1/2}.
\end{aligned}$$

U prvom koraku, za $j = 1$, računamo samo r_{11}. Jedini problem u provedbi ovog algoritma je pozitivnost izraza pod korijenom, a to slijedi iz pozitivne definitnosti, pa nema opasnosti, barem u egzaktnoj aritmetici.

Međutim, u aritmetici računala treba biti oprezan. Zbog mogućih grešaka zaokruživanja, korisno je dodati barem kontrolu pozitivnosti prije vađenja drugog korijena.

Algoritam 6.1.1. (Faktorizacija Choleskog)

```
for j := 1 to n do
  begin
    {Nađi j-ti stupac od R}
    {Supstitucija unaprijed iznad dijagonale}
  for i := 1 to j − 1 do
    begin
    sum := A[i, j];
    for k := 1 to i − 1 do
      sum := sum − R[k, i] ∗ R[k, j];
    R[i, j] := sum/R[i, i];
    end;
    {Dijagonalni element}
  sum := A[j, j];
  for k := 1 to j − 1 do
    sum := sum − sqr(R[k, j]);
  if sum > 0.0 then
    R[j, j] := sqrt(sum)
  else
      {Matrica nije pozitivno definitna, stani s algoritmom}
    begin
    error := true;
    exit;
    end;
  end;
error := false;
```

Ovdje pretpostavljamo da strojna realizacija funkcije *sqrt* za drugi korijen zadovoljava

$$x > 0 \Longrightarrow fl(sqrt(x)) > 0.$$

To je razumna pretpostavka, jer *sqrt* "smanjuje" raspon brojeva. U tom slučaju dobivamo pozitivne dijagonalne elemente i nema opasnosti od dijeljenja s nulom.

Napomenimo još jednom da se po prethodnoj rekurziji matrica R generira stupac po stupac, za razliku od standardnog zapisa algoritma za LR faktorizaciju, gdje se R generira redak po redak, a L stupac po stupac.

Ovo je tzv. *jik* varijanta algoritma, a naziv dolazi od poretka petlji izvana prema unutra, uz prirodno imenovanje indeksa — i za retke, j za stupce i k za sumu kod produkta. Pažljivijim pogledom vidimo da "najdublje" petlje po k odgovaraju skalarnim produktima komada stupaca od R, pa se ova varijanta katkad zove "skalarna" (engl. "dot" ili "inner product") varijanta.

To nipošto nije jedina varijanta za realizaciju algoritma. Ovu smo dobili tako da redoslijed rješavanja jednadžbi (6.1.3) odgovara supstituciji unaprijed za stupce

matrice R. Pokažite da možemo koristiti i redoslijed $(1,1)$, $(1,2)$, $\ldots$, $(1,n)$, $(2,2)$, $\ldots$, $(2,n)$, $(3,3)$, $\ldots$, (n,n), tj. redak po redak, od dijagonale prema kraju retka. Time dobivamo ijk varijantu algoritma, koja odgovara zamjeni poretka indeksa i, j. U njoj se R računa na isti način kao i u LR faktorizaciji. Pokušajte napraviti kji varijantu i njenu interpretaciju.

Složenost ovog algoritma opet mjerimo brojem aritmetičkih operacija (flopova) u floating-point aritmetici. Prebrajanjem dobivamo da približno (asimptotski proporcionalno) vrijedi

$$OP(n) \sim \frac{1}{3}\, n^3,$$

s tim da pišemo samo vodeći član, a ignoriraramo sve ostale članove nižeg reda. Vidimo da je složenost ili cijena faktorizacije Choleskog približno **polovina** složenosti (cijene) LR faktorizacije. To je dodatna motivacija za korištenje ove faktorizacije za simetrične pozitivno definitne matrice.

Kad imamo faktorizaciju Choleskog $A = R^T R$, onda se rješenje linearnog sustava $Ax = b$ svodi na dva rješavanja trokutastih sustava

$$R^T y = b, \quad Rx = y,$$

koje lako rješavamo supstitucijom unaprijed

$$\begin{aligned} y_1 &= b_1 / r_{11} \\ y_i &= \Big(b_i - \sum_{j=1}^{i-1} r_{ji}\, y_j \Big) \Big/ r_{ii}, \quad i = 2, \ldots, n, \end{aligned}$$

odnosno, unatrag

$$\begin{aligned} x_n &= y_n / r_{nn} \\ x_i &= \Big(y_i - \sum_{j=i+1}^{n} r_{ij}\, x_j \Big) \Big/ r_{ii}, \quad i = n-1, \ldots, 1. \end{aligned}$$

Za razliku od LR faktorizacije, ovdje u obje supstitucije imamo dijeljenja.

Zbog toga se, barem za rješavanje linearnih sustava, dosta često koristi LDL^T oblik faktorizacije. Neka je $A = R^T R$ faktorizacija Choleskog. Definiramo dijagonalnu matricu $D = \operatorname{diag}(r_{ii}^2)$ i $L = R^T \operatorname{diag}(r_{ii}^{-1}) = R^T D^{-1/2}$. Onda A možemo napisati u obliku

$$A = LDL^T, \tag{6.1.4}$$

gdje je L jedinična donjetrokutasta matrica. Upravo zato se ova faktorizacija i piše u ovom obliku, da asocira na isto značenje matrice L kao u LR faktorizaciji. Naravno, mogli bismo koristiti i zapis oblika $A = R^T D R$, gdje je R jedinična gornjetrokutasta.

Algoritam dobivamo na isti način kao i algoritam za faktorizaciju Choleskog, a možemo ga organizirati tako da računa L ili L^T, po želji. U tom algoritmu nema

računanja n drugih korijena, jer spremamo kvadrate r_{ii}^2 koji su dijagonalni elementi matrice D. Faktorizacijom $A = LDL^T$, rješenje linearnog sustava $Ax = b$ dobivamo rješavanjem 3 linearna sustava

$$Lz = b, \quad Dy = z, \quad L^T x = y.$$

Prvi i zadnji sustav su trokutasti s jediničnom dijagonalom, pa u supstitucijama nema dijeljenja. Srednji sustav je dijagonalan i trivijalno se rješava sa samo n dijeljenja. Time dobivamo uštedu od n dijeljenja obzirom na trokutaste sustave iz faktorizacije Choleskog. Ova ušteda možda nije velika za pune matrice, jer imamo oko n^2 operacija po supstituciji. Međutim, za vrpčaste matrice s malom širnom vrpce, a posebno za trodijagonalne matrice, ovo je velika ušteda. Preciznije, za trodijagonalne simetrične pozitivno definitne matrice, faza supstitucije iz faktorizacije Choleskog treba oko $6n$ operacija, a ovdje samo oko $5n$ operacija.

Na prvi pogled izgleda da bismo faktorizaciju (6.1.4) mogli provesti za bilo koju simetričnu matricu A, bez zahtjeva pozitivne definitnosti, s tim da dozvolimo da D ima i negativne elemente. Međutim, to ne vrijedi. Trivijalan kontraprimjer je matrica

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

koja je simetrična, ali indefinitna. Pokažite da za ovu matricu ne postoji faktorizacija oblika (6.1.4) s dijagonalnom matricom D. Poopćenje na indefinitne matrice dobivamo tako da dozvolimo dijagonalne blokove reda 2 u matrici D.

Svi rezultati koje ćemo napraviti za faktorizaciju Choleskog mogu se poopćiti i na LDL^T faktorizaciju, ali ih nećemo posebno formulirati. Slično vrijedi i za blok–faktorizacije.

6.2. Pivotiranje u faktorizaciji Choleskog

Dodatnu potvrdu stabilnosti za simetrične pozitivno definitne matrice dobivamo promatranjem LR faktorizacije ili Gaussovih eliminacija. Nije teško pokazati da, i bez pivotiranja, nakon svakog koraka eliminacije dobivamo reducirani, još nesređeni, dio matrice koji je opet simetrična i pozitivno definitna matrica reda manjeg za 1. Osim toga, može se pokazati da dijagonalni elementi padaju iz koraka u korak, tj. vrijedi

$$a_{kk} = a_{kk}^{(1)} \geq a_{kk}^{(2)} \geq \cdots \geq a_{kk}^{(k)} > 0.$$

Na kraju, trivijalno se vidi da elementi simetrične pozitivno definitne matrice A zadovoljavaju nejednakost

$$|a_{ij}| \leq \sqrt{a_{ii}\, a_{jj}}, \quad \text{za svaki } i \neq j, \tag{6.2.1}$$

jer bilo determinanta bilo koje glavne podmatrice reda 2 mora, također, biti pozitivna. To znači da je barem jedan od dijagonalnih elemenata iz (6.2.1) veći ili jednak $|a_{ij}|$, tj. apsolutno najveći element u A se nalazi na dijagonali

$$\|A\|_M = \max_{1\le i\le n} a_{ii}.$$

Odavde odmah slijedi da za pivotni rast u Gaussovim eliminacijama vrijedi $\rho_n = 1$ i to bez ikakvog pivotiranja. Važno je uočiti da to **ne** znači da su multiplikatori ograničeni na bilo koji način. Kontraprimjer je matrica

$$A = \begin{bmatrix} \varepsilon^2 & \varepsilon \\ \varepsilon & 2 \end{bmatrix},$$

kad $\varepsilon \to 0$. Ali, ono što je bitno, za simetrične pozitivno definitne matrice veličina multiplikatora nema utjecaja na stabilnost.

Međutim, **pogrešan** bi bio zaključak da pivotiranje u faktorizaciji Choleskog nije potrebno ili korisno. Sjetimo se samo rezultata za trokutaste sustave.

Kako se vrši pivotiranje? Za početak, da bismo očuvali simetriju radne matrice, pivotiranje mora biti "simetrično", tj. kad radimo zamjene, transformacija mora imati oblik

$$A \to P^T A P,$$

gdje je P matrica permutacije koja opisuje pripadnu zamjenu (stupaca). To znači da radimo istovremene zamjene redaka i stupaca u A. Kod takve zamjene, dijagonalni elementi prelaze opet u dijagonalne, a vandijagonalni ostaju izvan dijagonale. Dakle, ne možemo vandijagonalni element dovesti na dijagonalu, pa parcijalno pivotiranje nema analogon u faktorizaciji Choleskog. Srećom, iz (6.2.1) slijedi da to ionako ne bi imalo smisla.

Nesređeni radni dio matrice je simetričan i pozitivno definitan u svakom koraku, pa se najveći element u cijelom tom dijelu matrice mora nalaziti na dijagonali. Standardni izbor pivotnog elementa u k-tom koraku je

$$a_{rr}^{(k)} = \max_{k\le i\le n} a_{ii}^{(k)},$$

s tim da se obično uzima najmanji indeks r za koji se ovaj maksimum dostiže. To je ekvivalentno potpunom pivotiranju u Gaussovim eliminacijama.

Ovim postupkom dobivamo faktorizaciju Choleskog

$$P^T A P = R^T R,$$

a za elemente matrice R vrijedi

$$r_{kk}^2 \ge \sum_{i=k}^{j} r_{ij}^2, \quad j = k+1, \ldots, n, \quad k = 1, \ldots, n.$$

Posebno, to znači da R ima nerastuću dijagonalu $r_{11} \ge \cdots \ge r_{nn} > 0$.

7. Aproksimacija i interpolacija

7.1. Opći problem aproksimacije

Što je problem aproksimacije? Ako su poznate neke informacije o funkciji f, definiranoj na nekom skupu $X \subseteq \mathbb{R}$, na osnovu tih informacija želimo f zamijeniti nekom drugom funkcijom φ na skupu X, tako da su f i φ bliske u nekom smislu.

Skup X je najčešće interval oblika $[a, b]$ (može i neograničen), ili diskretni skup točaka.

Problem aproksimacije javlja se u dvije bitno različite formulacije.

(a) **Znamo** funkciju f (analitički ili slično), ali je njena forma prekomplicirana za računanje. U tom slučaju **odaberemo** neke informacije o f i po nekom kriteriju odredimo aproksimacionu funkciju φ. U ovom slučaju možemo birati informacije o f koje ćemo koristiti. Jednako tako, možemo ocijeniti grešku dobivene aproksimacije, obzirom na pravu vrijednost funkcije f.

(b) **Ne znamo** funkciju f, nego samo neke informacije o njoj, na primjer, vrijednosti na nekom skupu točaka. Zamjenska funkcija φ određuje se iz raspoloživih informacija, koje, osim samih podataka, mogu uključivati i očekivani oblik ponašanja podataka, tj. funkcije φ. U ovom se slučaju **ne može** napraviti ocjena pogreške bez dodatnih informacija o nepoznatoj funkciji f.

Varijanta (b) je puno češća u praksi. Najčešće se javlja kod mjerenja nekih veličina, jer, osim izmjerenih podataka, pokušavamo aproksimirati i podatke koji se nalaze "između" izmjerenih točaka. Primijetite da se kod mjerenja javljaju i pogreške mjerenja, pa postoje posebne tehnike za ublažavanje tako nastalih grešaka.

Funkcija φ bira se prema prirodi modela, ali tako da bude relativno jednostavna za računanje. Ona obično ovisi o parametrima a_k, $k = 0, \ldots, m$, koje treba odrediti po nekom kriteriju,

$$\varphi(x) = \varphi(x; a_0, a_1, \ldots, a_m).$$

Kad smo funkciju φ zapisali u ovom obliku, kao funkciju koja ovisi o parametrima a_k, onda kažemo da smo odabrali opći oblik aproksimacione funkcije.

Oblike aproksimacionih funkcija možemo (grubo) podijeliti na:

(a) linearne aproksimacione funkcije,

(b) nelinearne aproksimacione funkcije.

Bitne razlike između ove dvije grupe aproksimacionih funkcija opisujemo u nastavku.

7.1.1. Linearne aproksimacione funkcije

Opći oblik linearne aproksimacione funkcije je

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_m\varphi_m(x),$$

gdje su $\varphi_0, \ldots, \varphi_m$ poznate funkcije koje znamo računati. Primijetite da se linearnost **ne** odnosi na oblik funkcije φ, već na ovisnost o parametrima a_k koje treba odrediti. Prednost ovog oblika aproksimacione funkcije je da određivanje parametara a_k obično vodi na **sustave linearnih jednadžbi**.

Najčešće korišteni oblici linearnih aproksimacionih funkcija su:

1. algebarski polinomi, $\varphi_k(x) = x^k$, $k = 0, \ldots, m$, tj.

   $$\varphi(x) = a_0 + a_1x + \cdots + a_mx^m.$$

   Nije nužno da $\varphi(x)$ zapisujemo u bazi $\{1, x, \ldots, x^m\}$. Vrlo često je neka druga baza bitno pogodnija, na primjer, $\{1, (x - x_0), (x - x_0)(x - x_1), \ldots\}$, gdje su $x_0, x_1, \ldots$ zadane točke;

2. trigonometrijski polinomi, pogodni za aproksimaciju periodičkih funkcija, recimo, u modeliranju signala. Za funkcije φ_k uzima se $m + 1$ funkcija iz skupa

   $$\{1, \cos x, \sin x, \cos 2x, \sin 2x, \ldots\}.$$

   Katkad se koristi i faktor u argumentu sinusa i kosinusa koji služi za kontrolu perioda, a ponekad se biraju samo parne ili samo neparne funkcije iz ovog skupa;

3. po dijelovima polinomi (splajn funkcije). Ako su zadane točke $x_0, \ldots, x_n$, onda se splajn funkcija na svakom podintervalu svodi na polinom određenog fiksnog (niskog) stupnja, tj.

   $$\varphi\,\big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, 2, \ldots, n,$$

   a p_k su polinomi najčešće stupnjeva 1, 2, 3 ili 5. U točkama x_i obično se zahtijeva da funkcija φ zadovoljava još i tzv. "uvjete ljepljenja" vrijednosti funkcije i nekih njenih derivacija ili nekih aproksimacija za te derivacije. Splajnovi se danas često koriste zbog dobrih svojstava obzirom na grešku aproksimacije i kontrolu oblika aproksimacione funkcije.

7.1.2. Nelinearne aproksimacione funkcije

Najčešće korišteni oblici nelinearnih aproksimacionih funkcija su:

4. eksponencijalne aproksimacije

$$\varphi(x) = c_0 e^{b_0 x} + c_1 e^{b_1 x} + \cdots + c_r e^{b_r x},$$

koje imaju $n = 2r + 2$ nezavisna parametra, a opisuju, na primjer, procese rasta i odumiranja u raznim populacijama, s primjenom u biologiji, ekonomiji i medicini;

5. racionalne aproksimacije

$$\varphi(x) = \frac{b_0 + b_1 x + \cdots + b_r x^r}{c_0 + c_1 x + \cdots + c_s x^s},$$

koje imaju $n = r+s+1$ nezavisni parametar, a ne $r+s+2$, kako formalno piše. Naime, razlomci se mogu proširivati (ili skalirati), pa ako su b_i, c_i parametri, onda su to i tb_i, tc_i, za $t \neq 0$. Zbog toga se uvijek fiksira jedan od koeficijenata b_i ili c_i, a koji je to — obično slijedi iz prirode modela.

Ovako definirane racionalne funkcije imaju mnogo bolja svojstva aproksimacije nego polinomi, a pripadna teorija je relativno nova.

7.1.3. Kriteriji aproksimacije

Interpolacija

Interpolacija je zahtjev da se funkcije f i φ podudaraju na nekom konačnom skupu točaka. Te točke obično nazivamo **čvorovima** interpolacije. Ovom zahtjevu se može, ali i ne mora dodati zahtjev da se u čvorovima, osim funkcijskih vrijednosti, poklapaju i vrijednosti nekih derivacija.

Drugim riječima, u najjednostavnijem obliku interpolacije, kad tražimo samo podudaranje funkcijskih vrijednosti, od podataka o funkciji f koristi se samo informacija o njenoj vrijednosti na skupu od $(n + 1)$ točaka, tj. podaci oblika (x_k, f_k), gdje je $f_k := f(x_k)$, za $k = 0, \ldots, n$.

Parametri $a_0, \ldots, a_n$ (primijetite da parametara mora biti točno onoliko koliko i podataka!) određuju se iz uvjeta

$$\varphi(x_k; a_0, a_1, \ldots, a_n) = f_k, \quad k = 0, \ldots, n,$$

što je, općenito, nelinearni sustav jednadžbi. Ako je aproksimaciona funkcija φ linearna, onda za parametre a_k dobivamo sustav linearnih jednadžbi koji ima točno $n+1$ jednadžbi za $n+1$ nepoznanica. Matrica tog sustava je **kvadratna**, što bitno olakšava analizu egzistencije i jedinstvenosti rješenja za parametre interpolacije.

Minimizacija pogreške

Minimizacija pogreške je drugi kriterij određivanja parametara aproksimacije. Funkcija φ bira se tako da se minimizira neka odabrana norma pogreške

$$e(x) = f(x) - \varphi(x)$$

u nekom odabranom prostoru funkcija za φ na nekoj domeni X. Ove aproksimacije, često zvane i najbolje aproksimacije po normi, dijele se na diskretne i kontinuirane, prema tome minimizira li se norma pogreške e na diskretnom ili kontinuiranom skupu podataka X.

Standardno se kao norme pogreške koriste 2-norma i ∞-norma. Za 2-normu pripadna se aproksimacija zove **srednjekvadratna**, a metoda za njeno nalaženje zove se metoda najmanjih kvadrata. Funkcija φ, odnosno njeni parametri, se traže tako da bude

$$\min_{\varphi} \|e(x)\|_2.$$

U diskretnom slučaju $X = \{x_0, \ldots, x_n\}$, kad raspišemo prethodnu relaciju, dobivamo

$$\sqrt{\sum_{k=0}^{n} (f(x_k) - \varphi(x_k))^2} \to \min,$$

a u kontinuiranom

$$\sqrt{\int_a^b (f(x) - \varphi(x))^2 \, dx} \to \min .$$

Preciznije, minimizira se samo ono pod korijenom, jer to daje jednako rješenje kao da se minimizira i korijen! Zašto se baš minimiziraju kvadrati grešaka? To ima veze sa statistikom, jer se izmjereni podaci obično ponašaju kao normalna slučajna varijabla, s očekivanjem koje je točna vrijednost podatka. Odgovarajući kvadrati su varijanca i nju treba minimizirati.

U slučaju ∞-norme pripadna se aproksimacija zove **minimaks** aproksimacija, a parametri se biraju tako da se nađe

$$\min_{\varphi} \|e(x)\|_\infty.$$

U diskretnom slučaju traži se

$$\max_{k=0,\ldots,n} |f(x_k) - \varphi(x_k)| \to \min,$$

a u kontinuiranom

$$\max_{x \in [a,b]} |f(x) - \varphi(x)| \to \min .$$

U nekim problemima ovaj je tip aproksimacija poželjniji od srednjekvadratnih, jer se traži da maksimalna greška bude minimalna, tj. najmanja moguća, ali ih je općenito mnogo teže izračunati (na primjer, dobivamo problem minimizacije nederivabilne funkcije!).

Napomenimo još da smo u prethodnim primjerima koristili uobičajene (diskretne i kontinuirane) norme na odgovarajućim prostorima funkcija, ovisno o domeni X. Naravno, normirani prostor u kojem tražimo aproksimacionu funkciju ovisi o problemu kojeg rješavamo. Nerijetko u praksi, norme, pored funkcije uključuju i neke njene derivacije. Primjer takve norme je norma

$$\|f\| = \sqrt{\int_a^b (f(x))^2 + (f'(x))^2\, dx},$$

recimo, na prostoru $C^1[a,b]$ svih funkcija koje imaju neprekidnu prvu derivaciju na segmentu $[a,b]$, ili na nekom još "većem" prostoru.

Pri kraju ovog uvoda u opći problem aproksimacije funkcija postaje jasno koji su ključni matematički problemi koje treba riješiti:

- egzistencija i jedinstvenost rješenja problema aproksimacije, što ovisi o tome koje funkcije f aproksimiramo kojim funkcijama φ (dva prostora) i kako mjerimo grešku,
- analiza kvalitete dobivene aproksimacije — vrijednost "najmanje" pogreške i ponašenje funkcije greške e (jer norma je ipak samo broj),
- konstrukcija algoritama za računanje najbolje aproksimacije.

Objasnimo još koja je uloga "parametrizacije" aproksimacionih funkcija. Očito, riječ je o izboru prikaza ili "baze" u prostoru aproksimacionih funkcija ili načinu zadavanja tog prostora. Dok prva dva problema uglavnom ne ovise o "parametrizaciji", kao što ćemo vidjeti, dobar izbor "baze" je ključan korak u konstrukciji točnih i efikasnih algoritama.

Lako se vidi da problem interpolacije možemo smatrati specijalnim, ali posebno važnim slučajem aproksimacije po normi na diskretnom skupu X čvorova interpolacije uz neku od standardnih normi na konačnodimenzionalnim prostorima. Posebnost se ogleda u činjenici da se dodatno traži da je minimum norme pogreške jednak nuli, što je onda ekvivalentno odgovarajućim uvjetima interpolacije.

Na primjer, uzmimo da je $X = \{x_0, \ldots, x_n\}$ i tražimo aproksimacionu funkciju φ u prostoru $\mathcal{P}_n$ svih polinoma stupnja najviše n. Kao kriterij aproksimacije uzmimo neku p-normu ($1 \le p \le \infty$) vektora e pogreške funkcijskih vrijednosti na skupu X, tj. zahtjev je

$$\|e\|_p = \|f - \varphi\|_p = \Big(\sum_{k=0}^{n} |f(x_k) - \varphi(x_k)|^p \Big)^{1/p} \to \min, \quad 1 \le p < \infty,$$

odnosno

$$\|e\|_\infty = \|f - \varphi\|_\infty = \max_{k=0,\dots,n} |f(x_k) - \varphi(x_k)| \to \min .$$

Očito je $\|e\|_p = 0$ ekvivalentno uvjetima interpolacije

$$f(x_k) = \varphi(x_k), \quad k = 0, \dots, n,$$

samo nije jasno da li se to može postići, tj. da li postoji takva aproksimaciona funkcija $\varphi \in \mathcal{P}_n$ za koju je minimum greške jednak nuli, tako da je φ i interpolaciona funkcija. U sljedećem odjeljku pokazat ćemo da je odgovor potvrdan za ovaj primjer.

7.2. Interpolacija polinomima

Pretpostavimo da imamo funkciju f zadanu na diskretnom skupu različitih točaka x_k, $k = 0, \dots, n$, tj. $x_i \neq x_j$ za $i \neq j$. Poznate funkcijske vrijednosti u tim točkama skraćeno označavamo s $f_k = f(x_k)$.

Primijetite da pretpostavka o različitosti točaka nije bitno ograničenje. Naime, kad bismo dozvolili da je $x_i = x_j$ uz $i \neq j$, ili f ne bi bila funkcija (ako je $f_i \neq f_j$) ili bismo imali redundantan podatak, koji možemo ispustiti (ako je $f_i = f_j$).

Ako je $[a, b]$ segment na kojem koristimo interpolaciju (i promatramo grešku), u praksi su točke obično numerirane tako da vrijedi $a \leq x_0 < x_1 < \cdots < x_n \leq b$.

7.2.1. Egzistencija i jedinstvenost interpolacionog polinoma

Za polinomnu interpolaciju vrijedi sljedeći teorem.

Teorem 7.2.1. *Neka je $n \in \mathbb{N}_0$. Za zadane točke (x_k, f_k), $k = 0, \dots, n$, gdje je $x_i \neq x_j$ za $i \neq j$, postoji jedinstveni (interpolacioni) polinom stupnja najviše n*

$$\varphi(x) := p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

za koji vrijedi

$$p_n(x_k) = f_k, \quad k = 0, \dots, n.$$

Dokaz:

Neka je $p_n = a_0 + a_1 x + \cdots + a_n x^n$ polinom stupnja najviše n. Uvjete interpolacije možemo napisati u obliku

$$\begin{aligned}
p_n(x_0) &= a_0 + a_1 x_0 + \cdots + a_n x_0^n = f_0 \\
p_n(x_1) &= a_0 + a_1 x_1 + \cdots + a_n x_1^n = f_1 \\
&\cdots\cdots\cdots \\
p_n(x_n) &= a_0 + a_1 x_n + \cdots + a_n x_n^n = f_n.
\end{aligned}$$

Drugim riječima, treba provjeriti ima li ovaj sustav od $(n+1)$-e linearne jednadžbe s $(n+1)$-om nepoznanicom $a_0, \ldots, a_n$ jedinstveno rješenje. Dovoljno je provjeriti da li je (kvadratna) matrica tog linearnog sustava regularna. To možemo napraviti računanjem vrijednosti determinante te matrice. Ta determinanta je tzv. Vandermondeova determinanta

$$D_n = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}.$$

Definiramo determinantu koja naliči na D_n, samo umjesto x_n, posljednji je redak funkcija od x:

$$V_n(x) = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \\ 1 & x & x^2 & \cdots & x^n \end{vmatrix}.$$

Primijetimo da je $D_n = V_n(x_n)$. Gledamo li $V_n(x)$ kao funkciju od x, lako se vidi — razvojem po posljednjem retku, da je to polinom stupnja najviše n u varijabli x, s vodećim koeficijentom D_{n-1} uz x^n.

S druge strane, ako za x redom uvrštavamo $x_0, \ldots, x_{n-1}$, determinanta $V_n(x)$ će imati dva jednaka retka pa će biti

$$V_n(x_0) = V_n(x_1) = \cdots = V_n(x_{n-1}) = 0,$$

tj. točke $x_0, \ldots, x_{n-1}$ su nultočke polinoma $V_n(x)$ stupnja n. Da bismo točno odredili polinom stupnja n, ako su poznate njegove nultočke, potrebno je samo znati njegov vodeći koeficijent. U ovom slučaju, pokazali smo da je to D_{n-1}. Odatle odmah slijedi da je

$$V_n(x) = D_{n-1}\,(x - x_0)\,(x - x_1) \cdots (x - x_{n-1}).$$

Kad uvrstimo $x = x_n$, dobivamo rekurzivnu relaciju za D_n

$$D_n = D_{n-1}\,(x_n - x_0)\,(x_n - x_1) \cdots (x_n - x_{n-1}).$$

Ako znamo da je $D_0 = 1$, što je trivijalno, dobivamo da je

$$D_n = \prod_{0 \le j < i \le n} (x_i - x_j).$$

Budući da je $x_i \neq x_j$ za $i \neq j$, onda je $D_n \neq 0$, tj. matrica linearnog sustava je regularna, pa postoji jedinstveno rješenje $a_0, \ldots, a_n$ za koeficijente polinoma p_n, odnosno jedinstven interpolacioni polinom. ■

Ovaj teorem u potpunosti rješava prvo ključno pitanje egzistencije i jedinstvenosti rješenja problema polinomne interpolacije u njegovom najjednostavnijem obliku, kad su zadane funkcijske vrijednosti u međusobno različitim točkama.

Takav oblik interpolacije, kad tražena funkcija (u ovom slučaju polinom) mora interpolirati samo funkcijske vrijednosti zadane funkcije, obično zovemo **Lagrangeova interpolacija**. U općenitijem slučaju, možemo zahtijevati interpolaciju zadanih vrijednosti funkcije i njezinih uzastopnih derivacija. Takvu interpolaciju zovemo **Hermiteova interpolacija**. Nešto kasnije ćemo pokazati da problem Hermiteove interpolacije možemo riješiti kao granični slučaj Lagrangeove, kad dozvolimo višestruko "ponavljanje" istih čvorova, tj. otpustimo ograničenje na međusobnu različitost čvorova.

Za početak, moramo riješiti preostala dva problema vezana uz polinomnu Lagrangeovu interpolaciju, a to su: konstrukcija algoritama i analiza greške.

7.2.2. Potrebni algoritmi

Koje algoritme trebamo? Odgovor, naravno, ovisi o tome što želimo postići interpolacijom. Kao i kod svih aproksimacija, očita izravna primjena je zamjena funkcijskih vrijednosti $f(x)$ vrijednostima interpolacionog polinoma $p_n(x)$, i to u točkama x koje u principu **nisu** čvorovi interpolacije, posebno ako vrijednosti od f ne znamo u ostalim točkama, ili se f teško računa, pa smo jedva izračunali i ove vrijednosti od f koje smo iskoristili za interpolaciju.

Dakle, sigurno trebamo algoritam za računanje vrijednosti interpolacionog polinoma u nekoj zadanoj točki x koja nije čvor. Naime, zato što interpoliramo, u čvorovima je lako — vrijednosti od f su poznate i jednake onima od p_n, pa ih je dovoljno potražiti u tablici.

Točaka x u kojima želimo izračunati $p_n(x)$ može biti vrlo mnogo, a gotovo nikad nije samo jedna. Zbog toga se problem računanja vrijednosti $p_n(x)$ uvijek rješava u dvije faze:

1. prvo nađemo polinom p_n, jer on ne ovisi o točki x, već samo o zadanim podacima (x_k, f_k), $k = 0, \ldots, n$,
2. zatim, za svaku zadanu točku x izračunamo $p_n(x)$.

Prvu fazu je dovoljno napraviti samo jednom i zato svaku od ovih faza treba realizirati posebnim algoritmom. Dodatno, želimo što brži algoritam, posebno u drugoj fazi, jer se on tamo puno puta izvršava. Međutim, nećemo zahtijevati brzinu na uštrb stabilnosti, ako se to može izbjeći, bez većeg gubitka brzine.

Pogledajmo detaljnije prvu fazu. Što znači "naći polinom p_n"? Broj podataka $n + 1$ u potpunosti određuje vektorski prostor polinoma $\mathcal{P}_n$ u kojem, prema teo-

remu 7.2.1, postoji jedinstveni polinom p_n koji interpolira zadane podatke. Izaberimo neku bazu $\{b_0, b_1, \ldots, b_n\}$ u tom prostoru $\mathcal{P}_n$. Polinom p_n se može jednoznačno prikazati kao linearna kombinacija polinoma b_i iz te baze. Dakle, da bismo našli p_n, treba (i dovoljno je) naći koeficijente a_i u prikazu

$$p_n = \sum_{i=0}^{n} a_i b_i.$$

Njih možemo naći tako da u ovu relaciju uvrstimo sve uvjete interpolacije

$$p_n(x_k) = \sum_{i=0}^{n} a_i b_i(x_k) = f_k, \quad k = 0, \ldots, n,$$

i tako dobijemo linearni sustav reda $n+1$ za nepoznate koeficijente. Matrica tog linearnog sustava je sigurno regularna (dokažite!), a njezini elementi imaju oblik $B_{i+1,k+1} = b_i(x_k)$, za $i, k = 0, \ldots, n$.

U pripadnom algoritmu, prvo treba izračunati sve elemente matrice linearnog sustava, a zatim ga riješiti. Ako pretpostavimo da znamo prikaze svih polinoma b_i u standardnoj bazi i koristimo Hornerovu shemu za izvrednjavanje u svim točkama, onda svako izvrednjavanje traje najviše $O(n)$ operacija. Takvih izvrednjavanja ima najviše $(n+1)^2$, pa sve elemente matrice sustava možemo izračunati s najviše $O(n^3)$ operacija. Za posebne izbore baza i čvorova, ovaj broj operacija može biti i bitno manji.

Gaussovim eliminacijama ili LR faktorizacijom možemo riješiti dobiveni linearni sustav za najviše $O(n^3)$ operacija. Dakle, ukupan broj operacija u algoritamu za prvu fazu je najviše reda veličine $O(n^3)$. To, samo po sebi i nije tako loše, jer se izvršava samo jednom. Međutim, u nastavku ćemo pokazati da pažljivim izborom baze to možemo napraviti i bitno brže.

Algoritam za izvrednjavanje $p_n(x)$ u drugoj fazi, također, fundamentalno ovisi o izboru baze u $\mathcal{P}_n$. Naravno, iz prve faze treba zapamtiti izračunati vektor koeficijenata a_i. Tada se računanje $p_n(x)$ u zadanoj točki x svodi na računanje sume

$$p_n(x) = \sum_{i=0}^{n} a_i b_i(x).$$

U najopćenitijem obliku, točno po ovoj relaciji, imamo $n+1$-u Hornerovu shemu za izvrednjavanje $b_i(x)$ i još jedan skalarni produkt (ili linearnu kombinaciju). Ukupno trajanje je $O(n^2)$, što je vrlo skupo, kad usporedimo s običnom Hornerovom shemom.

Uočite da ova dva opća algoritma za interpolaciju možemo sažeto prikazati u obliku:

1. izaberi bazu u $\mathcal{P}_n$ i nađi koeficijente od p_n u toj bazi,

2. u zadanoj točki x izračunaj linearnu kombinaciju polinoma baze s poznatim koeficijentima u linearnoj kombinaciji.

Iz prethodne analize slijedi da bi bilo vrlo poželjno odabrati bazu tako da druga faza ima najviše $O(n)$ operacija, tj. da traje linearno, a ne kvadratno, u funkciji od n.

Kad u ovom kontekstu pogledamo tvrdnju i dokaz teorema 7.2.1., odmah možemo zaključiti da to odgovara izboru standardne baze $b_i(x) = x^i$, $i = 0, \ldots, n$, u prostoru $\mathcal{P}_n$. U prvoj fazi za nalaženje koeficijenata interpolacionog polinoma u standardnoj bazi ne moramo koristiti samo već spomenute numeričke metode. Osim njih, uz malo pažnje, možemo koristiti čak i Cramerovo pravilo. Determinanta D_n sustava je Vandermondeova, a sve ostale potrebne determinante se jednostavnim razvojem po stupcu svode na linearne kombinacije Vandermondeovih (za 1 manjeg reda). Ako njih izrazimo preko D_n, dobivamo opet algoritam koji treba $O(n^3)$ operacija.

Nadalje, vidimo da se druga faza svodi upravo na Hornerovu shemu, tj. ima linearno trajanje. Čak jače od toga, što se brzine tiče, ovim izborom baze dobivamo optimalan — najbrži mogući algoritam za izvrednjavanje u drugoj fazi.

Nažalost, u pogledu stabilnosti, situacija je mnogo manje "ružičasta", posebno u prvoj fazi. Matrica sustava može imati skoro linearno zavisne retke, a da čvorovi uopće nisu "patološki" raspoređeni. Dovoljno je samo da su razumno bliski i relativno daleko od nule (što je "centar" baze). Na primjer

$$x_k = k + 10^p, \quad k = 0, \ldots, n,$$

gdje je p "iole veći" pozitivni eksponent, recimo $p = 5$. Zbog toga se ovaj izbor baze ne koristi u praksi, već samo za dokazivanje u teoriji, jer baza ne ovisi o čvorovima.

Problemu izbora baze za prikaz interpolacionog polinoma možemo, sasvim općenito, pristupiti na 3 načina.

1. "Jednostavna baza, komplicirani koeficijenti". Fiksiramo jednostavnu bazu u $\mathcal{P}_n$, neovisno o zadanim podacima, ali tako da dobijemo brzo izvrednjavanje. Zatim nađemo koeficijente od p_n u toj bazi. Sva ovisnost o zadanim podacima ulazi u koeficijente, pa je prva faza spora.
2. "Jednostavni koeficijenti, komplicirana baza". Podijelimo ovisnost o zadanim podacima tako da koeficijenti jednostavno ovise o zadanim podacima i lako se računaju (na primjer, jednaki su zadanim funkcijskim vrijednostima f_k). Tada je prva faza brza, ali zato baza komplicirano ovisi o čvorovima, pa je druga faza spora, jer u svakoj točki x izvrednjavamo sve funkcije baze.
3. "Kompromis između baze i koeficijenata". Pustimo da baza jednostavno ovisi o čvorovima, a koeficijenti mogu ovisiti o svim zadanim podacima, ali tako da dobijemo jednostavne algoritme u obje faze.

Ove pristupe je najlakše ilustrirati preko složenosti rješavanja linearnog sustava za koeficijente.

Prvim pristupom dobivamo puni linearni sustav za čije rješavanje treba $\Theta(n^3)$ operacija. Ako baza ne ovisi o čvorovima, taj sustav može biti vrlo nestabilan, kao u ranijem primjeru standardne baze.

Drugi pristup vodi na dijagonalni linearni sustav u kojem se rješenje "čita" ili traje najviše $O(n)$ operacija. No, tada je izvrednjavanje u svakoj točki sporo, jer svi polinomi baze imaju puni stupanj n. Primjer takve baze je tzv. Lagrangeova baza.

U zadnjem pristupu bazu izaberemo tako da dobijemo (donje)trokutasti linearni sustav. Za nalaženje koeficijenata tada trebamo "samo" $O(n^2)$ operacija. Tako dobivamo tzv. Newtonovu bazu u kojoj stupnjevi polinoma b_i rastu, tj. vrijedi $\deg b_i = i$, kao i u standardnoj bazi. Osim toga, za b_i vrijedi jednostavna rekurzija koja vodi na brzi linearni algoritam izvrednjavanja.

Ova 3 pristupa možemo vrlo lijepo ilustrirati na jednostavnom primjeru linearne interpolacije, tj. kad je $n = 1$. Problem linearne interpolacije se svodi na nalaženje jednadžbe pravca p koji prolazi kroz dvije zadane točke (x_0, f_0) i (x_1, f_1).

Standardni oblik jednadžbe pravca je $p(x) = a_0 + a_1 x$. Iz uvjeta interpolacije dobivamo linearni sustav za koeficijente a_0 i a_1

$$\begin{aligned} p(x_0) &= a_0 + a_1 x_0 = f_0 \\ p(x_1) &= a_0 + a_1 x_1 = f_1, \end{aligned}$$

odakle slijedi

$$a_0 = \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0}, \quad a_1 = \frac{f_1 - f_0}{x_1 - x_0},$$

ili

$$p(x) = \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0} + \frac{f_1 - f_0}{x_1 - x_0}\, x.$$

Pravac p možemo napisati i kao težinsku sredinu zadanih funkcijskih vrijednosti f_0 i f_1, u obliku

$$p(x) = f_0 b_0(x) + f_1 b_1(x),$$

gdje su $b_0(x)$ i $b_1(x)$ funkcije koje treba naći. Iz uvjeta interpolacije sada dobivamo jednadžbe

$$\begin{aligned} p(x_0) &= f_0 b_0(x_0) + f_1 b_1(x_0) = f_0 \\ p(x_1) &= f_0 b_0(x_1) + f_1 b_1(x_1) = f_1. \end{aligned}$$

Bez dodatnih pretpostavki, iz ovih jednadžbi ne možemo odrediti $b_0(x)$ i $b_1(x)$, jer takvih funkcija ima puno. Pretpostavimo stoga da su obje funkcije, također, polinomi prvog stupnja i to specijalnog oblika, tako da ovaj linearni sustav postane dijagonalan. Tada iz vandijagonalnih elemenata dobivamo uvjete

$$b_1(x_0) = 0, \quad b_0(x_1) = 0,$$

a onda za dijagonalne elemente dobivamo

$$b_0(x_0) = 1, \quad b_1(x_1) = 1.$$

Vidmo da su polinomi b_0 i b_1 rješenja specijalnih problema interpolacije

$$b_i(x_k) = \delta_{ik}, \quad i, k = 0, 1,$$

tj. b_i mora biti nula u svim čvorovima osim i-tog, a u i-tom mora imati vrijednost 1. To znači da znamo sve nultočke od b_i, a vrijednost vodećeg koeficijenta izlazi iz $b_i(x_i) = 1$. Odmah možemo napisati te dvije funkcije baze u obliku

$$b_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad b_1(x) = \frac{x - x_0}{x_1 - x_0},$$

pa je

$$p(x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0}$$

što odgovara jednadžbi pravca "kroz dvije točke". Ovo je Lagrangeov oblik interpolacionog polinoma. Vidimo da funkcije baze b_0 i b_1 ovise o oba čvora interpolacije.

Jednadžbu pravca možemo napisati i tako da pravac prolazi "kroz jednu točku" (x_0, f_0) i ima zadani koeficijent smjera

$$p(x) = f_0 + k(x - x_0).$$

Ovaj oblik automatski zadovoljava prvi uvjet interpolacije $p(x_0) = f_0$, a iz drugog uvjeta

$$p(x_1) = f_0 + k(x_1 - x_0) = f_1$$

se lako izračuna k

$$k = \frac{f_1 - f_0}{x_1 - x_0},$$

što je poznata formula za koeficijent smjera pravca kroz dvije točke. Dobiveni oblik za p

$$p(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0)$$

je Newtonov oblik interpolacionog polinoma. Njega možemo interpretirati na još nekoliko načina. Prvo, to je i Taylorov oblik za p napisan oko točke x_0, s tim da je "podijeljena razlika" k baš derivacija od p u točki x_0 (i, naravno, svakoj drugoj točki).

Nadalje, prvi član ovog oblika za p, u ovom slučaju konstanta f_0, je interpolacioni polinom stupnja 0 za zadanu prvu točku (x_0, f_0). Dakle, ovaj oblik za p odgovara korekciji interpolacionog polinoma kroz prethodne točke, kad dodamo još jednu novu točku (x_1, f_1). To isto vrijedi i u općem slučaju.

Na kraju, ovaj oblik pravca možemo dobiti tako da u prostoru $\mathcal{P}_1$ izaberemo bazu b_0, b_1, koja daje donjetrokutasti linearni sustav za koeficijente c_0 i c_1 u prikazu

$$p(x) = c_0 b_0(x) + c_1 b_1(x).$$

Uvjeti interpolacije daju jednadžbe

$$\begin{aligned} p(x_0) &= c_0 b_0(x_0) + c_1 b_1(x_0) = f_0 \\ p(x_1) &= c_0 b_0(x_1) + c_1 b_1(x_1) = f_1. \end{aligned}$$

Kako ćemo dobiti donjetrokutasti linearni sustav? Postavljamo redom uvjete na polinome baze, stupac po stupac, i još imamo na umu prethodnu interpretaciju "dopunjavanja" ranijeg interpolacionog polinoma.

Za polinom b_0 u prvom stupcu nemamo nikavih uvjeta, pa uzmemo najjednostavniju oblik, koja odgovara interpolaciji stupnja 0 u prvom čvoru, a to je $b_0(x) = 1$. Iz prve jednadžbe (supstitucija unaprijed) odmah dobivamo i $c_0 = f_0$.

Za polinom b_1 u drugom stupcu dobivamo točno jedan uvjet $b_1(x_0) = 0$. Opet uzmemo najjednostavniji oblik polinoma koji zadovoljava taj uvjet, a to je

$$b_1(x) = (x - x_0).$$

To, usput, odgovara i "dizanju" stupnja interpolacije kod dodavanja novog čvora. Supstitucijom unaprijed izlazi i koeficijent c_1

$$c_1 = \frac{f_1 - f_0}{x_1 - x_0}.$$

Kao što ćemo vidjeti, ovaj postupak se može nastaviti. Općenito, iz uvjeta da stupac s polinomom b_i ima donjetrokutasti oblik dobivamo da b_i mora imati nultočke u svim prethodnim čvorovima $x_0, \ldots, x_{i-1}$, pa možemo uzeti

$$b_i(x) = (x - x_0) \cdots (x - x_{i-1}),$$

što opet odgovara dizanju stupnja. Kako općenito izgledaju koeficijenti c_i, opisat ćemo malo kasnije.

7.2.3. Lagrangeov oblik interpolacionog polinoma

Da bismo našli koeficijente interpolacionog polinoma, nije nužno rješavati linearni sustav za koeficijente. Interpolacioni polinom p_n možemo odmah napisati korištenjem tzv. Lagrangeove baze $\{\ell_k, k = 0, \ldots, n\}$ prostora polinoma $\mathcal{P}_n$

$$p_n(x) = \sum_{k=0}^{n} f_k \ell_k(x), \tag{7.2.1}$$

pri čemu je

$$\begin{aligned}\ell_k(x) &= \frac{(x-x_0)\cdots(x-x_{k-1})\,(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\ &= \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i} := \frac{\omega_k(x)}{\omega_k(x_k)}, \quad k=0,\ldots,n.\end{aligned} \tag{7.2.2}$$

Polinomi ℓ_k su stupnja n, pa je p_n polinom stupnja najviše n. Osim toga, vrijedi

$$\ell_k(x_i) = \begin{cases} 0, & \text{za } i \neq k, \\ 1, & \text{za } i = k. \end{cases}$$

Uvrstimo li to u (7.2.1), odmah slijedi da se suma u (7.2.1) svodi na jedan jedini član za $i = k$, tj. da vrijedi

$$p_n(x_k) = f_k.$$

Oblik (7.2.1)–(7.2.2) zove se Lagrangeov oblik interpolacionog polinoma. Taj polinom možemo napisati u još jednom, zgodnijem obliku. Definiramo

$$\omega(x) = \prod_{k=0}^{n}(x-x_k),$$

pa je

$$\ell_k(x) = \frac{\omega(x)}{(x-x_k)\,\omega_k(x_k)}.$$

Uvrštavanjem u (7.2.1) dobivamo da je

$$p_n(x) = \omega(x)\sum_{k=0}^{n}\frac{f_k}{(x-x_k)\,\omega_k(x_k)}. \tag{7.2.3}$$

Uočimo da je

$$\omega_k(x_k) = \omega'(x_k),$$

pa (7.2.3) možemo pisati kao

$$p_n(x) = \omega(x)\sum_{k=0}^{n}\frac{f_k}{(x-x_k)\,\omega'(x_k)}. \tag{7.2.4}$$

Ova se forma može koristiti za računanje vrijednosti polinoma u točki $x \neq x_k$, $k = 0,\ldots,n$. Prednost je što se za svaki novi x računa samo $\omega(x)$ i $(x - x_k)$, dok se $\omega_k(x_k) = \omega'(x_k)$ izračuna samo jednom za svaki k i čuva u tablici, jer ne ovisi o x.

Ukupan broj operacija je proporcionalan s n^2, a za računanje u svakoj novoj točki x, trebamo još reda veličine n operacija. Ipak, u praksi se ne koristi ovaj oblik interpolacionog polinoma, već nešto bolji Newtonov oblik. Lagrangeov oblik interpolacionog polinoma uglavnom se koristi u teoretske svrhe (za dokaze).

7.2.4. Ocjena greške interpolacionog polinoma

Ako znamo još neke informacije o funkciji f, možemo napraviti i ocjenu greške interpolacionog polinoma.

Teorem 7.2.2. *Pretpostavimo da funkcija f ima $(n+1)$-u derivaciju na segmentu $[a,b]$ za neki $n \in \mathbb{N}_0$. Neka su $x_k \in [a,b]$, $k = 0, \ldots, n$, međusobno različiti čvorovi interpolacije, tj. $x_i \neq x_j$ za $i \neq j$, i neka je p_n interpolacioni polinom za funkciju f u tim čvorovima. Za bilo koju točku $x \in [a,b]$ postoji točka ξ iz otvorenog intervala*

$$x_{\min} := \min\{x_0, \ldots, x_n, x\} < \xi < \max\{x_0, \ldots, x_n, x\} =: x_{\max}$$

takva da za grešku interpolacionog polinoma vrijedi

$$e(x) := f(x) - p_n(x) = \frac{\omega(x)}{(n+1)!} f^{(n+1)}(\xi), \tag{7.2.5}$$

pri čemu je $\omega(x) := \prod_{k=0}^{n} (x - x_k)$.

Dokaz:

Ako je $x = x_k$, za neki $k \in \{0, \ldots, n\}$, iz uvjeta interpolacije i definicije polinoma ω dobivamo da su obje strane u (7.2.5) jednake 0, pa teorem očito vrijedi (ξ može biti bilo koji).

Pretpostavimo stoga da x nije čvor interpolacije. Tada je $\omega(x) \neq 0$ i grešku interpolacije možemo prikazati u obliku

$$e(x) = f(x) - p_n(x) = \omega(x) s(x),$$

s time da je $s(x)$ korektno definiran čim x nije čvor. Uzmimo sad da je x fiksan i definiramo funkciju

$$g(t) = e(t) - \omega(t) s(x) = e(t) - \omega(t) \frac{e(x)}{\omega(x)}, \quad t \in [a,b]. \tag{7.2.6}$$

Funkcija pogreške e ima točno onoliko derivacija (po t) koliko i f, i one su neprekidne kad su to i odgovarajuće derivacije od f. Budući da x nije čvor, to isto vrijedi i za funkciju g, tj. $g^{(n+1)}$ je korektno definirana na $[a,b]$. Nađimo koliko nultočaka ima funkcija g. Ako za t uvrstimo x_k, dobivamo

$$g(x_k) = e(x_k) - \omega(x_k) \frac{e(x)}{\omega(x)} = 0, \quad k = 0, \ldots, n.$$

Jednako tako je i

$$g(x) = e(x) - e(x) = 0.$$

Drugim riječima, g ima barem $n+2$ nultočke na $[a,b]$. Čak i jače, sve te nultočke su na segmentu $[x_{\min}, x_{\max}]$. Budući da je g derivabilna na tom segmentu, po Rolleovom teoremu slijedi da g' ima barem $n+1$ nultočku na otvorenom intervalu $(x_{\min}, x_{\max})$. Induktivnom primjenom Rolleovog teorema zaključujemo da $g^{(j)}$ ima bar $n+2-j$ nultočaka na $(x_{\min}, x_{\max})$, za $j = 0, \ldots, n+1$. Dakle, za $j = n+1$ dobivamo da $g^{(n+1)}$ ima bar jednu nultočku $\xi \in (x_{\min}, x_{\max})$.

Iskoristimo još da je p_n polinom stupnja najviše n, a ω polinom stupnja $n+1$, pa je

$$e^{(n+1)}(t) = f^{(n+1)}(t), \quad \omega^{(n+1)}(t) = (n+1)!.$$

Uvrštavanjem u $n+1$-u derivaciju definicione formule (7.2.6) za g dobivamo

$$g^{(n+1)}(t) = e^{(n+1)}(t) - \omega^{(n+1)}(t)\,\frac{e(x)}{\omega(x)} = f^{(n+1)}(t) - (n+1)!\,\frac{e(x)}{\omega(x)}.$$

Konačno, ako uvažimo da je $g^{(n+1)}(\xi) = 0$, onda je

$$0 = g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)!\,\frac{e(x)}{\omega(x)},$$

odnosno

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

što je upravo (7.2.5). ■

Ako je $f^{(n+1)}$ ograničena na $[a,b]$ ili, jače, ako je $f \in C^{n+1}[a,b]$, onda se iz prethodnog teorema može dobiti sljedeća ocjena greške interpolacionog polinoma za funkciju f u točki $x \in [a,b]$

$$|f(x) - p_n(x)| \le \frac{|\omega(x)|}{(n+1)!}\,M_{n+1}, \quad M_{n+1} := \max_{x\in[a,b]} |f^{(n+1)}(x)|.$$

Ova ocjena direktno slijedi iz (7.2.5), a korisna je ako relativno jednostavno možemo izračunati ili odozgo ocijeniti M_{n+1}.

7.2.5. Newtonov oblik interpolacionog polinoma

Lagrangeov oblik interpolacionog polinoma nije dobar kad želimo dizati stupanj interpolacionog polinoma da bismo eventualno poboljšali aproksimaciju i smanjili grešku, zbog toga što interpolacioni polinom moramo računati od početka.

Postoji forma interpolacionog polinoma kod koje je mnogo lakše dodavati točke interpolacije, tj. dizati stupanj interpolacionog polinoma. Neka je p_{n-1} interpolacioni polinom koji interpolira funkciju f u točkama x_k, $k = 0, \ldots, n-1$. Neka je p_n

interpolacioni polinom koji interpolira funkciju f još i u točki x_n. Polinom p_n tada možemo napisati u obliku

$$p_n(x) = p_{n-1}(x) + c(x), \tag{7.2.7}$$

gdje je c korekcija, polinom stupnja n. Također, mora vrijediti

$$c(x_k) = p_n(x_k) - p_{n-1}(x_k) = f(x_k) - f(x_k) = 0, \quad k = 0, \ldots, n-1.$$

Vidimo da su x_k nultočke od c, pa ga možemo napisati u obliku

$$c(x) = a_n\,(x - x_0)\cdots(x - x_{n-1}).$$

Nadalje, iz zadnjeg uvjeta interpolacije $p_n(x_n) = f(x_n)$, dobivamo

$$\begin{aligned} f(x_n) &= p_n(x_n) = p_{n-1}(x_n) + c(x_n) \\ &= p_{n-1}(x_n) + a_n\,(x_n - x_0)\cdots(x_n - x_{n-1}), \end{aligned}$$

odakle lako izračunavamo vodeći koeficijent a_n polinoma c

$$a_n = \frac{f(x_n) - p_{n-1}(x_n)}{(x_n - x_0)\cdots(x_n - x_{n-1})} = \frac{f(x_n) - p_{n-1}(x_n)}{\omega(x_n)}.$$

Nakon ovog, imamo sve elemente za računanje $p_n(x)$ u bilo kojoj točki x, korištenjem relacije (7.2.7). Taj koeficijent bit će n-ta podijeljena razlika, u oznaci

$$a_n = f[x_0, x_1, \ldots, x_n].$$

Drugim riječima, dobivamo rekurzivnu formulu za podizanje stupnja interpolacionog polinoma

$$p_n(x) = p_{n-1}(x) + (x - x_0)\cdots(x - x_{n-1})f[x_0, \ldots, x_n]. \tag{7.2.8}$$

Da bismo bolje opisali a_n, vratimo se na Lagrangeov oblik interpolacionog polinoma. Primijetimo da je a_n koeficijent uz vodeću potenciju x^n u p_n.

Iskoristimo relaciju (7.2.4), tj. nađimo koeficijent uz x^n u toj relaciji. Dobivamo

$$f[x_0, x_1, \ldots, x_n] = \sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)}. \tag{7.2.9}$$

Iz formule (7.2.9) slijede neka svojstva podijeljenih razlika. Ako permutiramo čvorove, opet dobijemo istu podijeljenu razliku. Druga korisna formula je rekurzivna definicija podijeljenih razlika

$$f[x_0, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0}.$$

Izvedimo tu formulu. Vrijedi

$$\begin{aligned}
f[x_1,\dots,x_n] &= \sum_{k=1}^{n} \frac{f(x_k)}{(x_k-x_1)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\
&= \sum_{k=1}^{n-1} \frac{f(x_k)(x_k-x_0)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\
&\quad + \frac{f(x_n)(x_n-x_0)}{(x_n-x_0)\cdots(x_n-x_{n-1})} \\
f[x_0,\dots,x_{n-1}] &= \sum_{k=0}^{n-1} \frac{f(x_k)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_{n-1})} \\
&= \sum_{k=1}^{n-1} \frac{f(x_k)(x_k-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\
&\quad - \frac{f(x_0)(x_n-x_0)}{(x_0-x_1)\cdots(x_0-x_n)}.
\end{aligned}$$

Oduzimanjem dobivamo

$$\begin{aligned}
&f[x_1,\dots,x_n] - f[x_0,\dots,x_{n-1}] \\
&\qquad = \sum_{k=1}^{n-1} \frac{f(x_k)(x_n-x_0)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\
&\qquad\quad + \frac{f(x_n)(x_n-x_0)}{(x_n-x_0)\cdots(x_n-x_{n-1})} + \frac{f(x_0)(x_n-x_0)}{(x_0-x_1)\cdots(x_0-x_n)} \\
&\qquad = (x_n-x_0)\sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)} = (x_n-x_0)\,f[x_0,\dots,x_n],
\end{aligned}$$

čime je dokazana tražena formula.

Ostaje još vidjeti što je start rekurzije za podijeljenje razlike. Ako znamo da je konstanta koja prolazi točkom $(x_0, f(x_0))$, interpolacioni polinom stupnja 0, onda je $a_0 = f[x_0] = f(x_0)$. Jednako tako vrijedi

$$f[x_k] = f(x_k),$$

pa tablicu podijeljenih razlika lako sastavljamo.

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$\cdots$	$f[x_0, \ldots, x_n]$
x_0	$f[x_0]$				
		$f[x_0, x_1]$			
x_1	$f[x_1]$		$f[x_0, x_1, x_2]$		
		$f[x_1, x_2]$		$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$f[x_0, \ldots, x_n]$
		$f[x_{n-2}, x_{n-1}]$		$\iddots$	
x_{n-1}	$f[x_{n-1}]$		$f[x_{n-2}, x_{n-1}, x_n]$		
		$f[x_{n-1}, x_n]$			
x_n	$f[x_n]$				

Dakle, kad uvažimo rekurziju i oblik polinoma $c(x)$ u (7.2.8), dobivamo da je oblik Newtonovog interpolacionog polinoma

$$\begin{aligned} p_n(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)\,(x - x_1)f[x_0, x_1, x_2] \\ + \cdots + (x - x_0)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n]. \end{aligned}$$

Primijetite da nam od tablica podijeljenih razlika treba samo "gornja dijagonala", pa ćemo se u računanju podijeljenih razlika moći služiti jednodimenzionalnim poljem. Pretpostavimo da je na početku algoritma u i-tom elementu polja f spremljena funkcijska vrijednost $f(x_i)$. Na kraju algoritma u polju f ostavit ćemo redom $f[x_0], f[x_0, x_1], \ldots, f[x_0, \ldots, x_n]$.

Algoritam 7.2.1. (Algoritam računanja podijeljenih razlika)

for $i := 1$ **to** n **do**
  for $j := n$ **downto** i **do**
    $f[j] := (f[j] - f[j-1])/(x[j] - x[j-i])$;

I grešku interpolacionog polinoma (koja je jednaka onoj kod Lagrangeovog), možemo pisati korištenjem podijeljenih razlika. Neka je $x_{n+1} \in (a, b)$ realan broj koji nije čvor. Konstruirajmo interpolacioni polinom koji prolazi točkama $x_0, \ldots, x_n$ i x_{n+1}. Dobivamo

$$\begin{aligned} p_{n+1}(x) &= f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)\,(x - x_1)f[x_0, x_1, x_2] \\ &\quad + \cdots + (x - x_0)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n] \\ &\quad + (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}] \\ &= p_n(x) + (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}]. \end{aligned} \tag{7.2.10}$$

Budući da je

$$p_{n+1}(x_{n+1}) = f(x_{n+1}),$$

onda iz relacije (7.2.10) slijedi

$$f(x_{n+1}) = p_n(x_{n+1}) + (x_{n+1} - x_0) \cdots (x_{n+1} - x_n) f[x_0, x_1, \ldots, x_n, x_{n+1}].$$

Usporedimo li tu formulu s ocjenom greške iz Teorema 7.2.2. (napisanu u točki x_{n+1}, a ne x)

$$f(x_{n+1}) - p_n(x_{n+1}) = \frac{\omega(x_{n+1})}{(n+1)!} f^{(n+1)}(\xi),$$

odmah se čita da je

$$f[x_0, x_1, \ldots, x_n, x_{n+1}] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

za neki $\xi \in I$. Prethodna se formula uobičajeno piše u ovisnosti o varijabli x, tj. x_{n+1} se zamijeni s x (Prije nam to nije odgovaralo zbog pisanja interpolacionog polinoma u varijabli x)

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{7.2.11}$$

Zajedno s (7.2.10), Newtonov interpolacioni polinom tada poprima oblik Taylorovog polinoma (s greškom nastalom zanemarivanjem viših članova), samo razvijenog oko točaka $x_0, \ldots, x_n$. To nas motivira da interpolacioni polinom u točki x izvrednjavamo na sličan način kao što se Hornerovom shemom izvrednjava vrijednost polinoma. Pretpostavimo da u polju f na mjestu i piše $f[x_0, x_1, \ldots, x_i]$.

Algoritam 7.2.2. (Algoritam izvrednjavanja interpolacionog polinoma)

$sum := f[n]$;
for $i := n - 1$ **downto** 0 **do**
  $sum := sum * (x - x_i) + f[i]$;
{ Na kraju je $p_n(x) = sum$. }

7.2.6. Koliko je dobar interpolacioni polinom?

U praksi se obično koriste interpolacioni polinomi niskih stupnjeva – do 5. Zašto? Kod nekih funkcija za neki izbor točaka interpolacije, povećavanje stupnja interpolacionog polinoma može dovesti do povećanja grešaka. Zbog toga se umjesto visokog stupnja interpolacionog polinoma u praksi koristi po dijelovima polinomna interpolacija.

Njemački matematičar Runge prvi je uočio probleme koji nastupaju kod interpolacije na ekvidistantnoj mreži i konstruirao funkciju (poznatu kao funkcija Runge), koja ima svojstvo da niz Newtonovih interpolacionih polinoma na ekvidistantnoj mreži ne konvergira prema toj funkciji.

Primjer 7.2.1. (Runge, 1901.) *Promotrimo funkciju*

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5].$$

i izaberimo ekvidistantne čvorove interpolacije x_k, $k = 0, \ldots, n$

$$x_k = -5 + kh, \quad h = \frac{10}{n}, \quad k = 0, \ldots, n.$$

Zanima nas ponašanje grešaka koje nastaju dizanjem stupnja n *interpolacionog polinoma. Po Teoremu* 7.2.2*, uvažavanjem relacije* (7.2.11)*, dobivamo*

$$e_n(x) = f(x) - p_n(x) = \omega(x)\, f[x_0, x_1, \ldots, x_n, x].$$

Tvrdimo da vrijedi

$$f[x_0, x_1, \ldots, x_n, x] = f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r}(1+x_k^2)} \cdot \begin{cases} 1, & \text{ako je } n = 2r+1, \\ x, & \text{ako je } n = 2r. \end{cases} \tag{7.2.12}$$

Prvo pokažimo za $n = 2r+1$*, indukcijom po* r*. U tom slučaju imamo paran broj interpolacijskih točaka, koje su simetrične obzirom na ishodište, tj. zadovoljavaju*

$$x_k = -x_{n-k}.$$

Ako je $r = 0$*, onda je* $n = 1$ *i* $x_1 = -x_0$*, a zbog parnosti funkcije* f *i* $f(-x_0) = f(x_0)$*. Izračunajmo podijeljenu razliku*

$$\begin{aligned} f[x_0, x_1, x] = f[x_0, -x_0, x] &= \frac{f[-x_0, x] - f[x_0, -x_0]}{x - x_0} \\ &= \frac{f(x) - f(x_0)}{x^2 - x_0^2} = f(x)\, \frac{-1}{1+x_0^2}. \end{aligned}$$

Time je pokazana baza indukcije. Provedimo korak indukcije ali s r *u* $r+1$*, tj. "skačemo" za* 2 *u* n*. Neka vrijedi* (7.2.12) *za* $n = 2r+1$ *i* **bilo koji** *skup od* $r+1$ *parova simetričnih točaka* ($x_k = -x_{n-k}$)*. Neka je* $m = n+2 = 2(r+1)+1$*. Definiramo funkciju*

$$g(x) = f[x_1, \ldots, x_{m-1}, x].$$

Zbog definicije g*, po pretpostavci indukcije, vrijedi*

$$g(x) = f(x) \cdot a_r, \quad a_r = \frac{(-1)^{r+1}}{\prod_{k=1}^{r}(1+x_k^2)},$$

Po definiciji podijeljenjih razlika, lako je pokazati da vrijedi

$$g[x_0, x_m, x] = f[x_0, \ldots, x_m, x].$$

Osim toga je

$$g[x_0, x_m, x] = a_r f[x_0, x_m, x] = a_r\, f(x) \frac{-1}{1 + x_0^2},$$

što zaključuje korak indukcije. Za paran n, dokaz je vrlo sličan.

Budući da je

$$(x - x_k)\,(x - x_{n-k}) = (x - x_k)\,(x + x_k) = x^2 - x_k^2,$$

onda je za $n = 2r + 1$

$$\prod_{k=0}^{n} (x - x_k) = \prod_{k=0}^{r} (x^2 - x_k^2).$$

U parnom je slučaju $n = 2r$, $x_r = 0$, pa izdvajanjem srednje točke dobivamo

$$\prod_{k=0}^{n} (x - x_k) = x \cdot \prod_{k=0}^{r-1} (x^2 - x_k^2) = \frac{1}{x} \cdot \prod_{k=0}^{r} (x^2 - x_k^2),$$

ili zajedno

$$\omega(x) = \prod_{k=0}^{n} (x - x_k) = \prod_{k=0}^{r} (x^2 - x_k^2) \cdot \begin{cases} 1, & \text{ako je } n = 2r + 1, \\ 1/x, & \text{ako je } n = 2r. \end{cases}$$

Time smo pokazali željeni oblik formule za podijeljene razlike. Ako tu formulu uvrstimo u grešku, dobivamo

$$\begin{aligned} e_n(x) = f(x) - p_n(x) &= \omega(x)\, f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r} (1 + x_k^2)} \cdot \begin{cases} 1, & \text{ako je } n = 2r + 1, \\ x, & \text{ako je } n = 2r. \end{cases} \\ &= (-1)^{r+1}\, f(x)\, g_n(x), \end{aligned}$$

gdje je

$$g_n(x) = \prod_{k=0}^{r} \frac{x^2 - x_k^2}{1 + x_k^2}. \tag{7.2.13}$$

Funkcija f pada od 0 do 5, pa se zbog simetrije, njena najveća vrijednost nalazi u 0, a najmanja u ± 5, pa imamo

$$\frac{1}{26} \le f(x) \le 1.$$

Zbog toga, konvergencija Newtonovih polinoma ovisi samo o $g_n(x)$. Osim toga je i g_n parna, tj. $g_n(x) = g_n(-x)$, pa možemo sve gledati na intervalu $[0, 5]$.

I apsolutnu vrijednost funkcije g_n možemo napisati na malo neobičan način

$$|g_n(x)| = \left(e^{h \ln |g_n(x)|}\right)^{1/h}.$$

Prema (7.2.13)*, za eksponent eksponencijalne funkcije imamo*

$$h \ln |g_n(x)| = h \cdot \sum_{k=0}^{r} \ln \left| \frac{x^2 - x_k^2}{1 + x_k^2} \right|.$$

Tvrdimo da je

$$\begin{aligned} \lim_{n\to\infty} h \ln |g_n(x)| &= \lim_{r\to\infty} h \cdot \sum_{k=0}^{r} \ln \left| \frac{x^2 - x_k^2}{1 + x_k^2} \right| \\ &= \int_{-5}^{0} \ln \left| \frac{x^2 - \xi^2}{1 + \xi^2} \right| d\xi =: q(x). \end{aligned}$$

Ostavimo li zasad jednakost posljednje sume i integrala po strani (treba naći malo složeniji limes), primijetimo da se integral može izračunati analitički

$$\int_{-5}^{0} \ln \left| \frac{x^2 - \xi^2}{1 + \xi^2} \right| d\xi = (5 + x) \ln(5 + x) + (5 - x) \ln(5 - x) - 5 \ln 26 - 2 \operatorname{arctg} 5.$$

Analizom toka funkcije vidimo da $q(x)$ ima jednu nultočku u intervalu $[0, 5]$, prilično jednaku 3.63 *(možemo ju i točnije odrediti). Preciznije, zbog parnosti funkcije q, na $[-5, 5]$ vrijedi*

$$\begin{aligned} &q(x) = 0 \ za \ |x| = 3.63, \\ &q(x) < 0 \ za \ |x| < 3.63, \\ &q(x) > 0 \ za \ 3.63 < |x| \leq 5. \end{aligned}$$

Za $|x| > 3.63$ i $h = 10/n$ slijedi dakle da je

$$\lim_{n\to\infty} |g_n(x)| = \infty,$$

pa i

$$e_n(x) \to \infty,$$

tj. niz interpolacijskih polinoma divergira za $|x| > 3.63$!

Zanimljivo je da, ako umjesto ekvidistantnih točaka interpolacije u primjeru Runge uzmemo neekvidistantne, točnije tzv. Čebiševljeve točke, onda će porastom stupnja niz interpolacionih polinoma konvergirati prema funkciji f. Na intervalu $[a, b]$, uzlazno poredane Čebiševljeve točke su

$$x_k = \frac{1}{2}\left(a + b + (a - b) \cos \frac{(2k + 1)\pi}{2n + 2} \right), \quad k = 0, \ldots, n.$$

Zadatak 7.2.1. *Dokažite da vrijedi*

$$\lim_{n\to\infty} h \ln |g_n(x)| = \int_{-5}^{0} \ln \left| \frac{x^2 - \xi^2}{1 + \xi^2} \right| d\xi.$$

Uputa: Očito je

$$\ln\left|\frac{x^2-x_k^2}{1+x_k^2}\right| = \ln|x+x_k| + \ln|x-x_k| - \ln|1+x_k^2|$$

i lako se vidi da je

$$\lim_{r\to\infty}\sum_{k=0}^{r} h\,\ln|1+x_k^2| = \int_{-5}^{0}\ln|1+\xi^2|\,d\xi$$
$$\lim_{r\to\infty}\sum_{k=0}^{r} h\,\ln|x-x_k| = \int_{-5}^{0}\ln|x-\xi|\,d\xi,$$

zbog neprekidnosti podintegralnih funkcija i definicije Riemannovog integrala, budući je riječ o specijalnim Darbouxovim sumama. Za dokaz da je

$$\lim_{r\to\infty}\sum_{k=0}^{r} h\,\ln|x+x_k| = \int_{-5}^{0}\ln|x+\xi|\,d\xi,$$

potrebno je napraviti "finu analizu" i posebno razmatrati situacije $|x+x_k|<\delta$, $|x+x_k|>\delta$, *za neki mali* $0<\delta<1$ *(ili se pozvati na jače teoreme iz teorije mjere).*

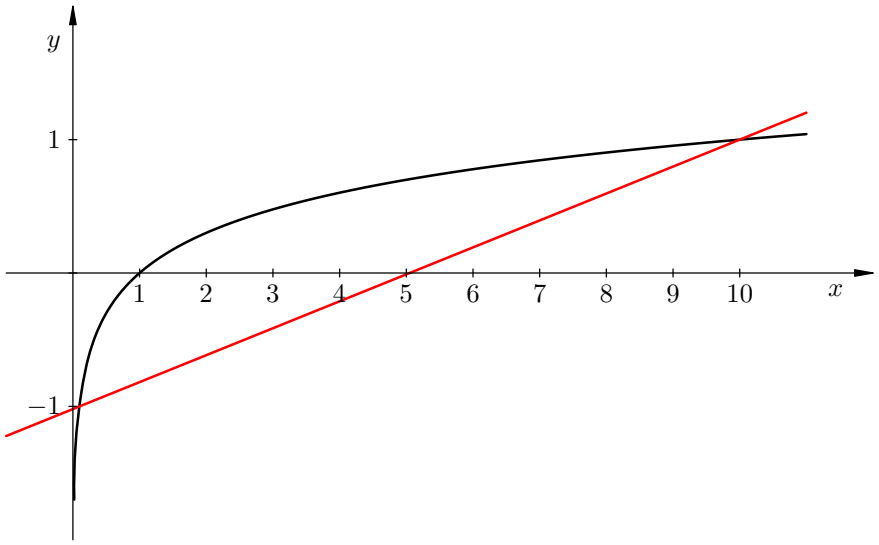
Primjer 7.2.2. *Promotrimo grafove interpolacionih polinoma stupnjeva 1–6 koji interpoliraju funkciju*

$$f(x)=\log(x) \quad za\ x\in[0.1,10]$$

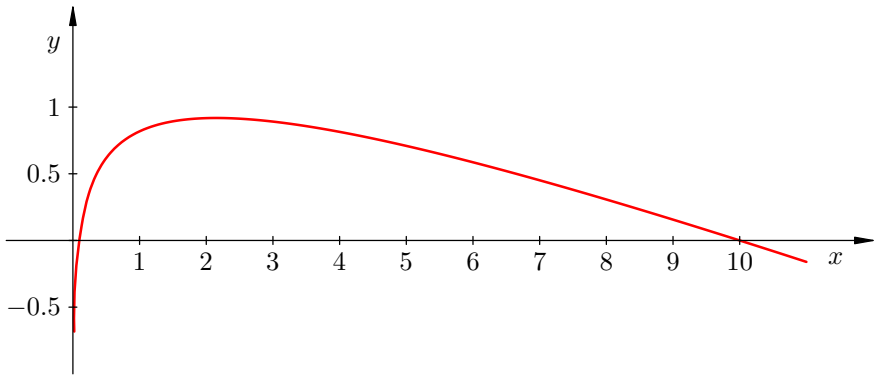
na ekvidistantnoj i Čebiševljevoj mreži.

Primijetit ćete da je greška interpolacije najveća na prvom podintervalu bez obzira na stupanj interpolacionog polinoma. Razlog leži u činjenici da funkcija $\log(x)$ *ima singularitet u* 0*, a početna točka interpolacije je blizu.*

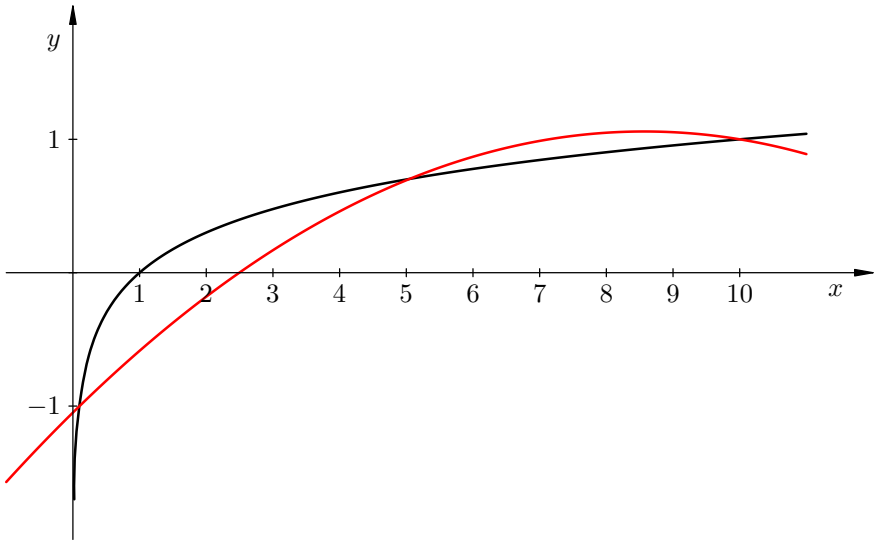
Prva grupa slika su redom funkcija (crno) i interpolacioni polinom (crveno) za ekvidistantnu mrežu, te pripadna greška, a zatim to isto za Čebiševljevu mrežu.



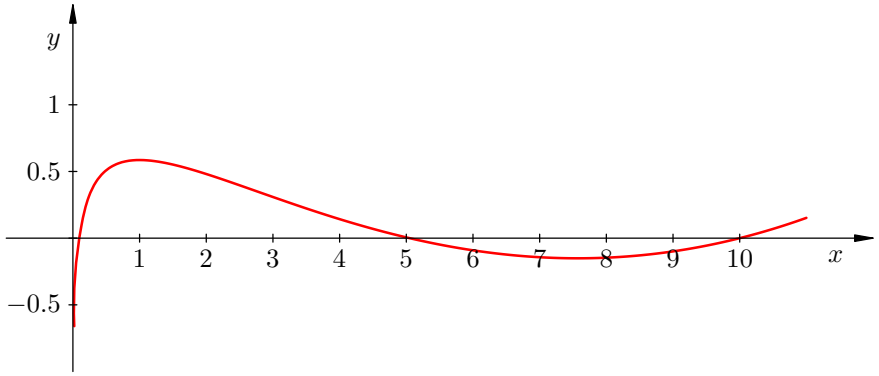
Ekvidistantna mreža, interpolacioni polinom stupnja 1.



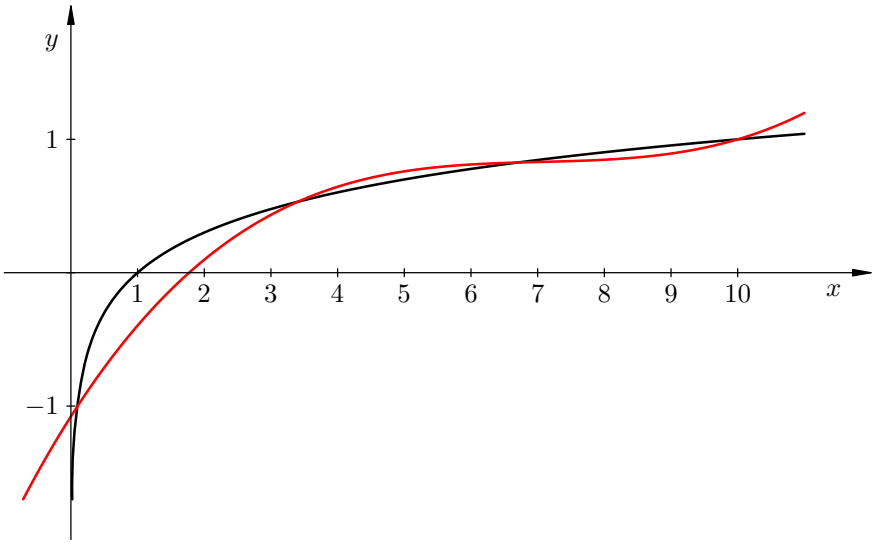
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 1.



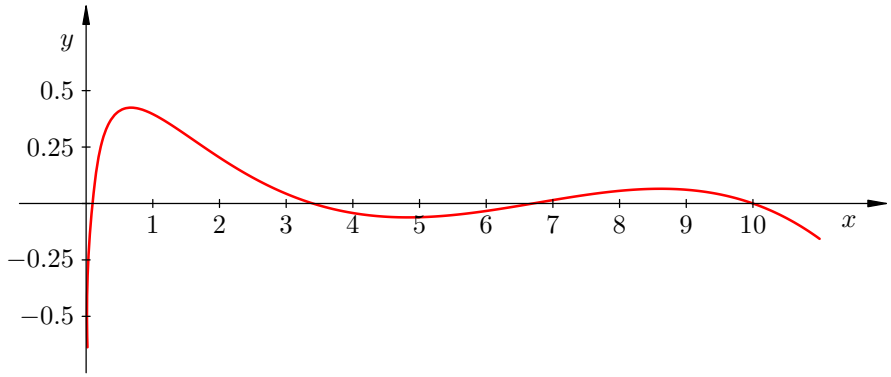
Ekvidistantna mreža, interpolacioni polinom stupnja 2.



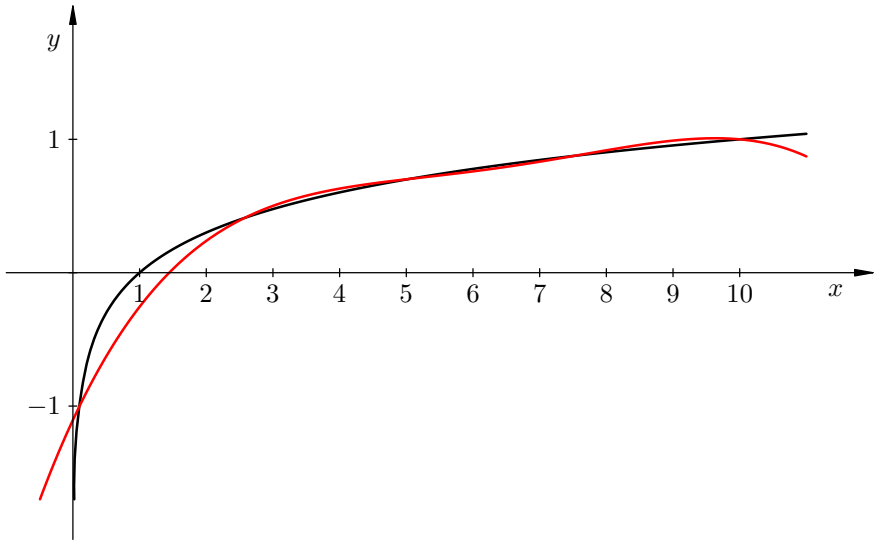
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 2.



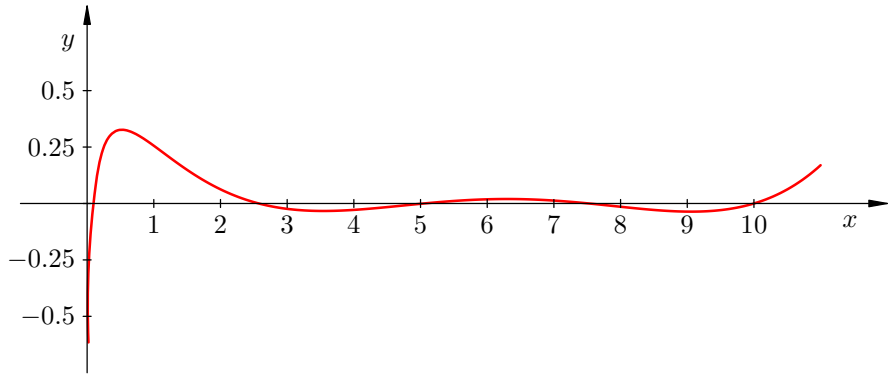
Ekvidistantna mreža, interpolacioni polinom stupnja 3.



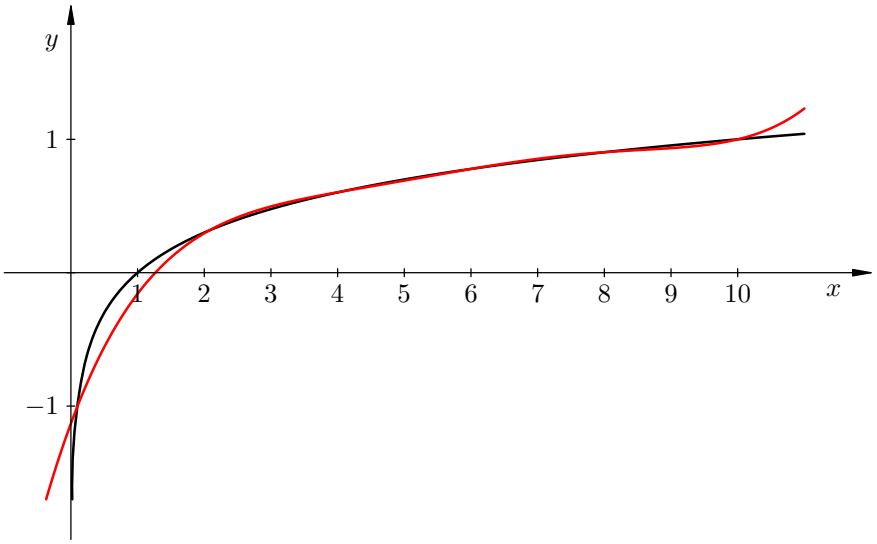
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 3.



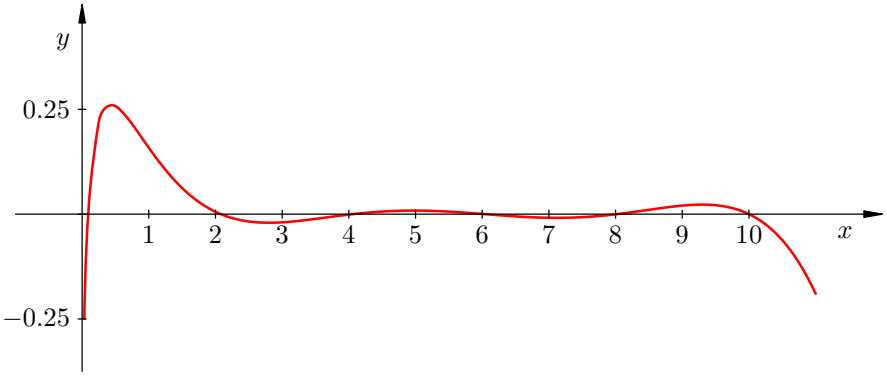
Ekvidistantna mreža, interpolacioni polinom stupnja 4.



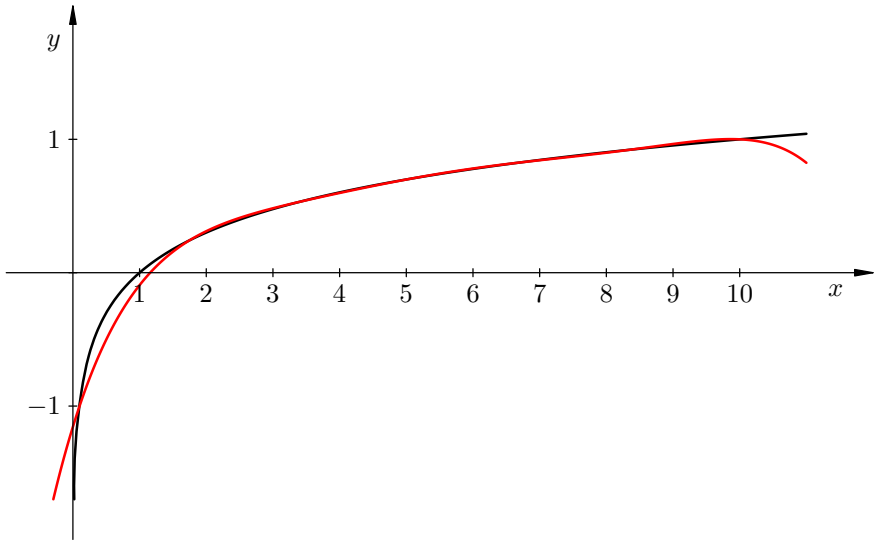
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 4.



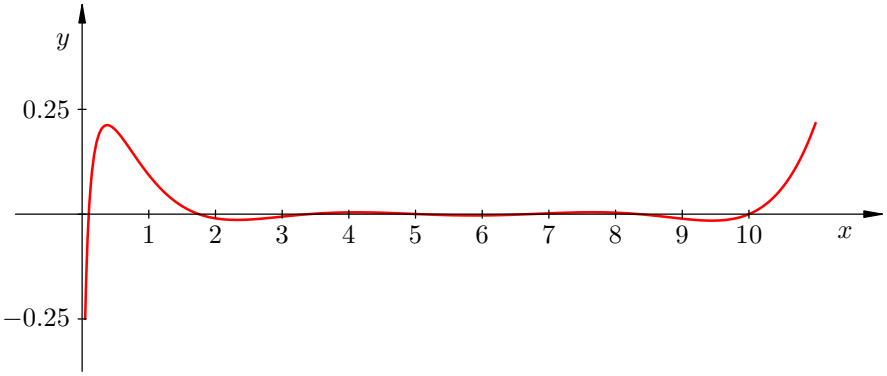
Ekvidistantna mreža, interpolacioni polinom stupnja 5.



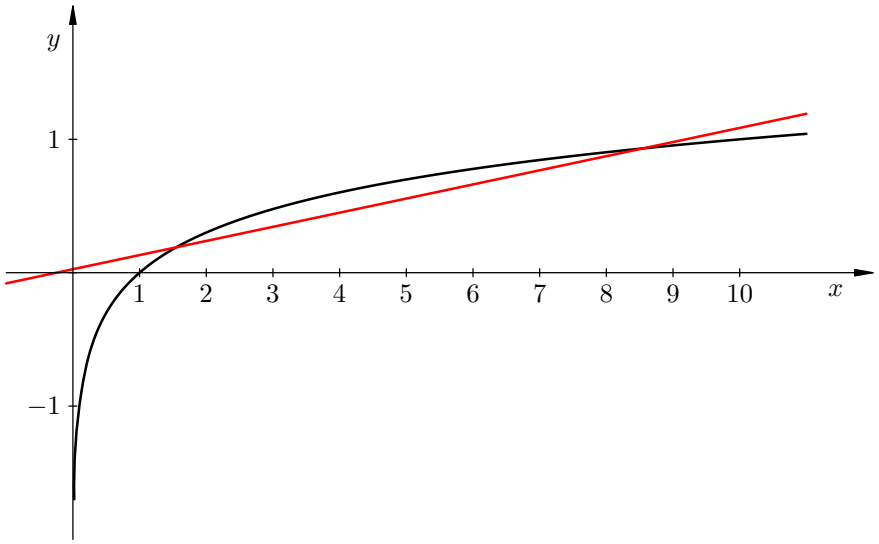
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 5.



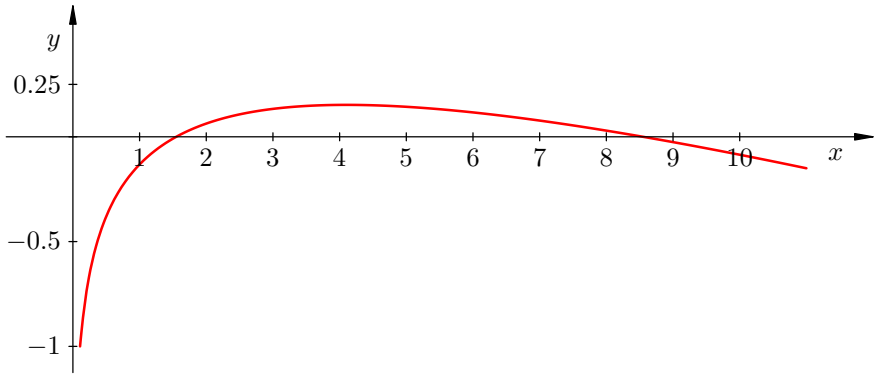
Ekvidistantna mreža, interpolacioni polinom stupnja 6.



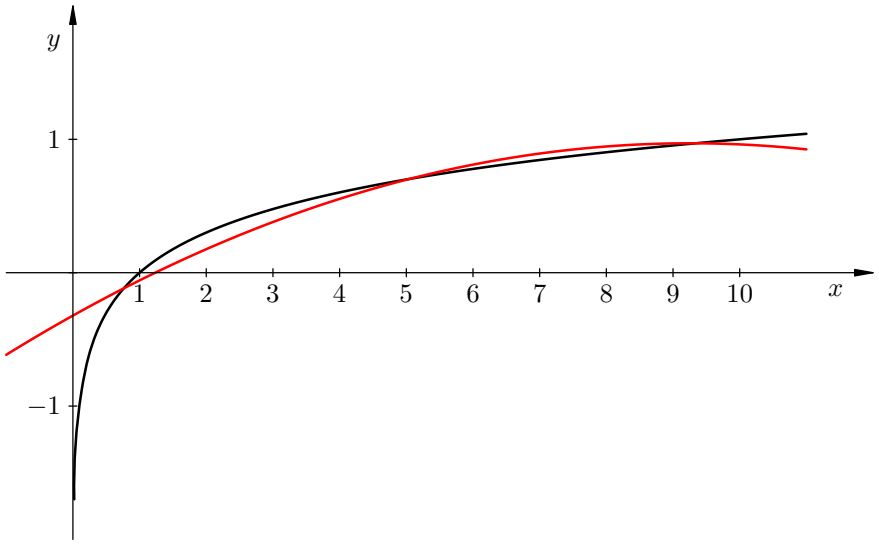
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 6.



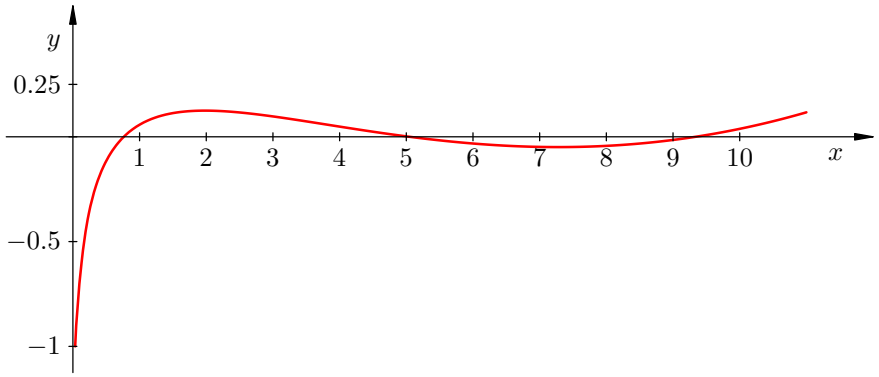
Čebiševljeva mreža, interpolacioni polinom stupnja 1.



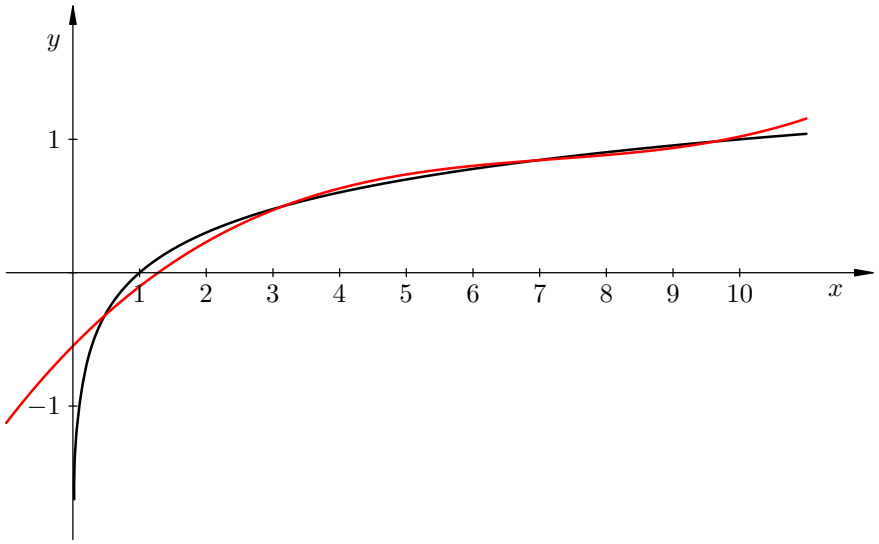
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 1.



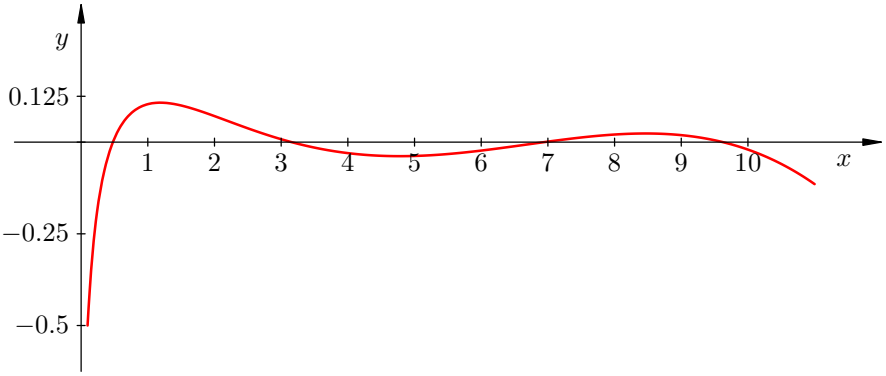
Čebiševljeva mreža, interpolacioni polinom stupnja 2.



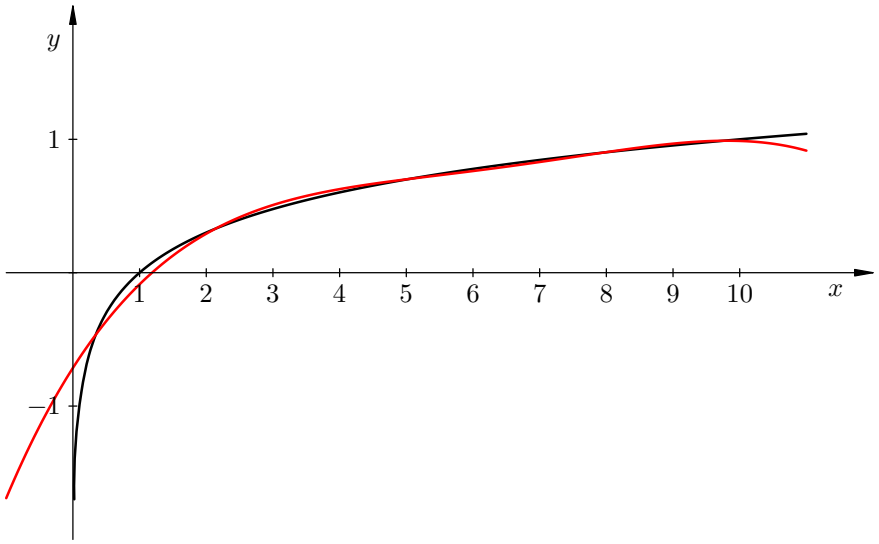
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 2.



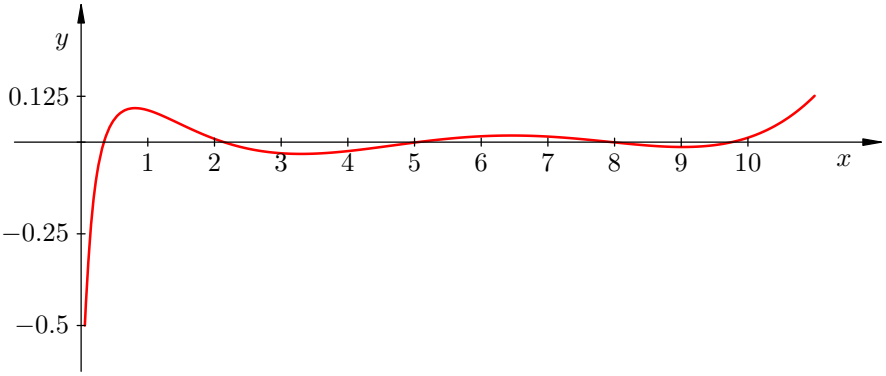
Čebiševljeva mreža, interpolacioni polinom stupnja 3.



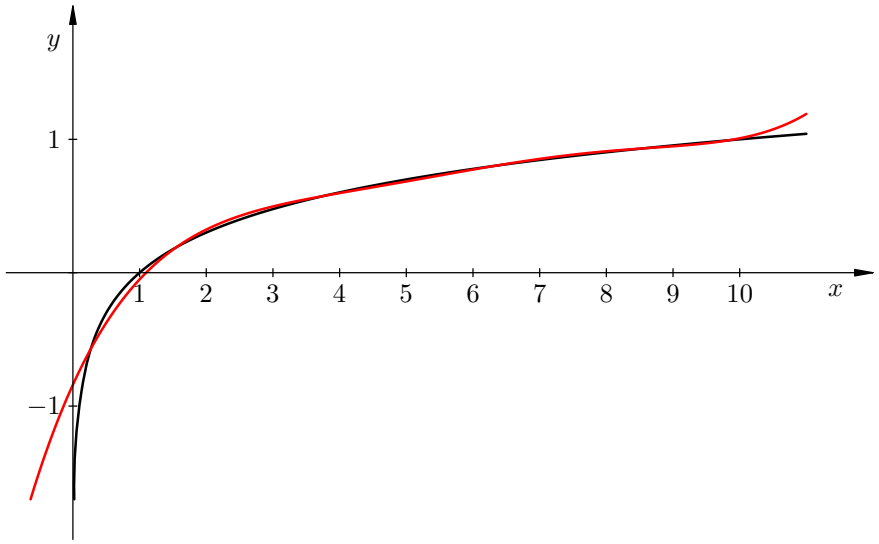
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 3.



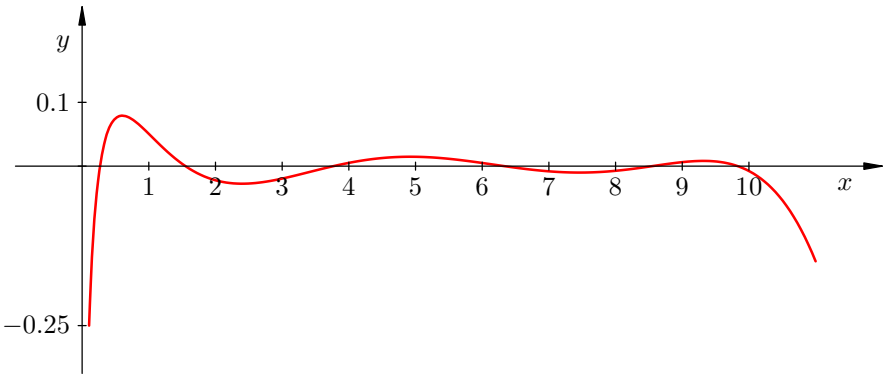
Čebiševljeva mreža, interpolacioni polinom stupnja 4.



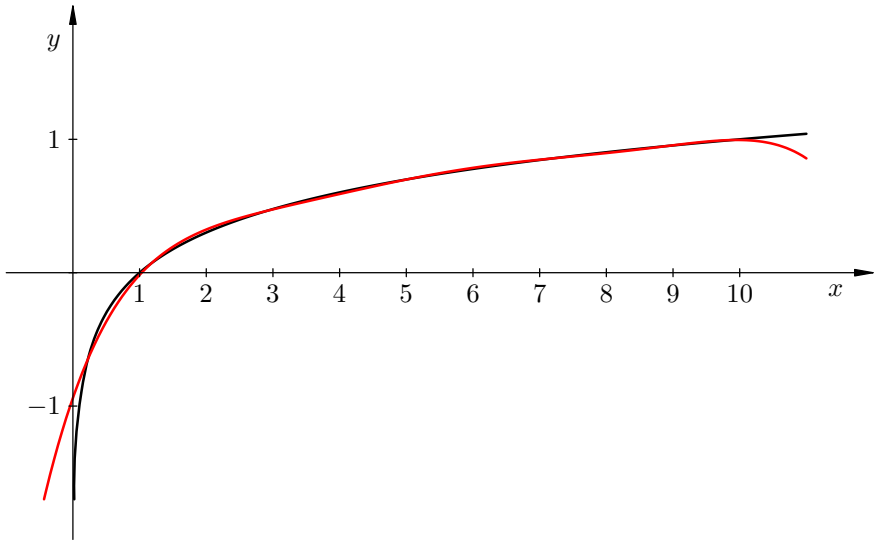
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 4.



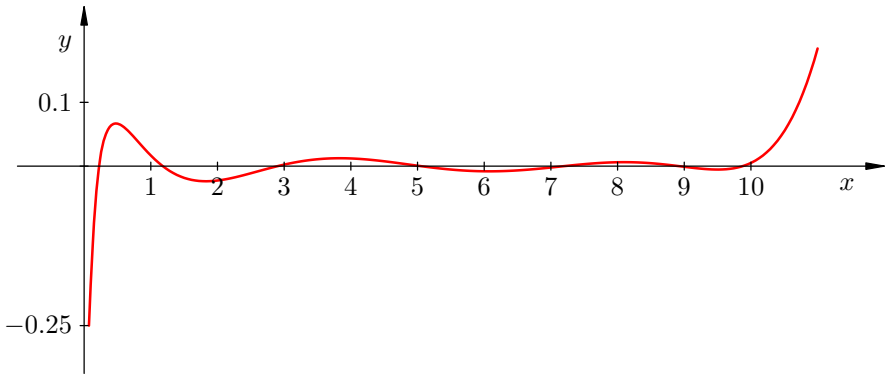
Čebiševljeva mreža, interpolacioni polinom stupnja 5.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 5.



Čebiševljeva mreža, interpolacioni polinom stupnja 6.



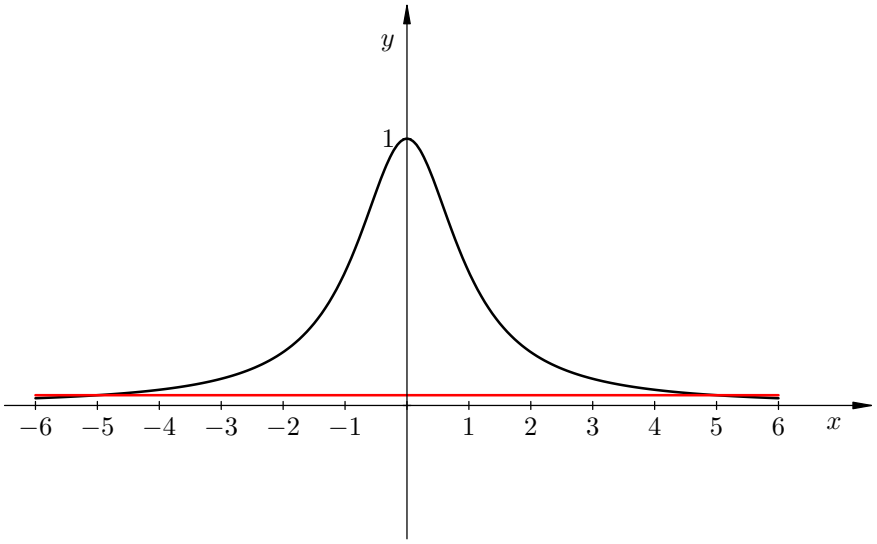
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 6.

Primjer 7.2.3. *Već smo pokazali na primjeru Runge da interpolacioni polinomi koji interpoliraju funkciju*

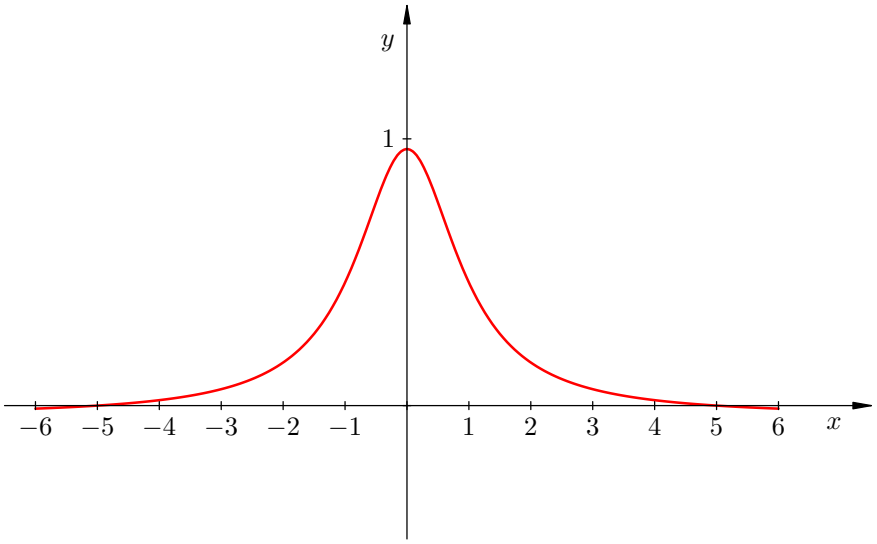
$$f(x) = \frac{1}{1+x^2} \quad za\ x \in [-5, 5]$$

na ekvidistantnoj mreži ne konvergiraju. S druge strane, pogledajmo što se događa s polinomima koji interpoliraju tu funkciju u Čebiševljevim točkama. Interpolacioni polinomi su stupnjeva 1–6, 8, 10, 12, 14, 16 (parnost funkcije!).

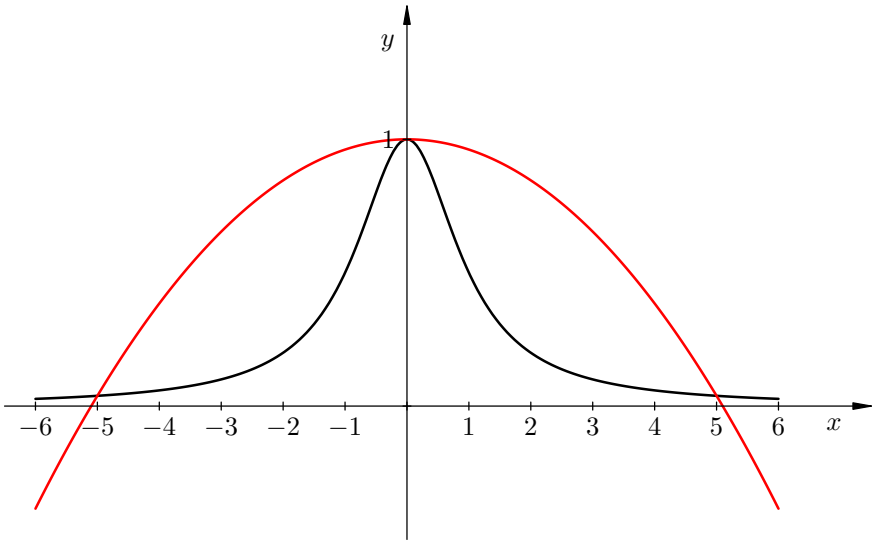
Ponovno, kao i u prošlom primjeru, grafovi su u parovima.



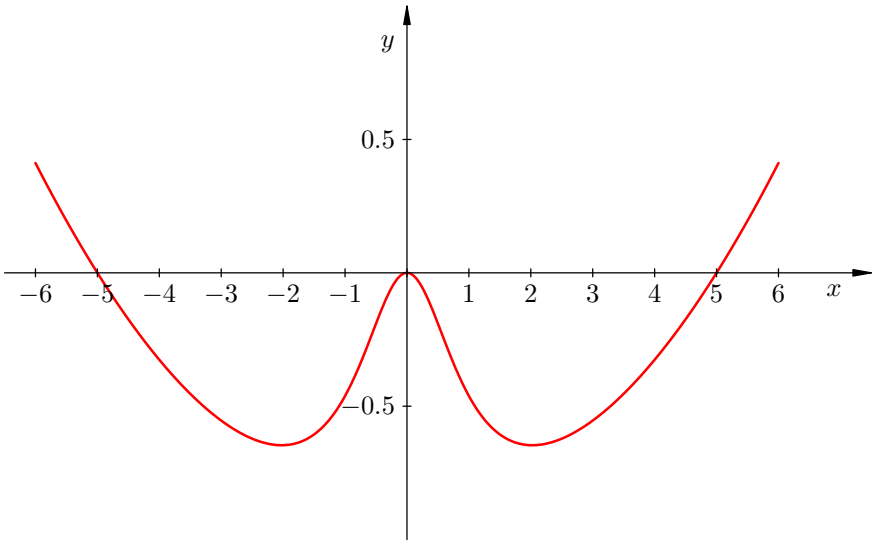
Ekvidistantna mreža, interpolacioni polinom stupnja 1.



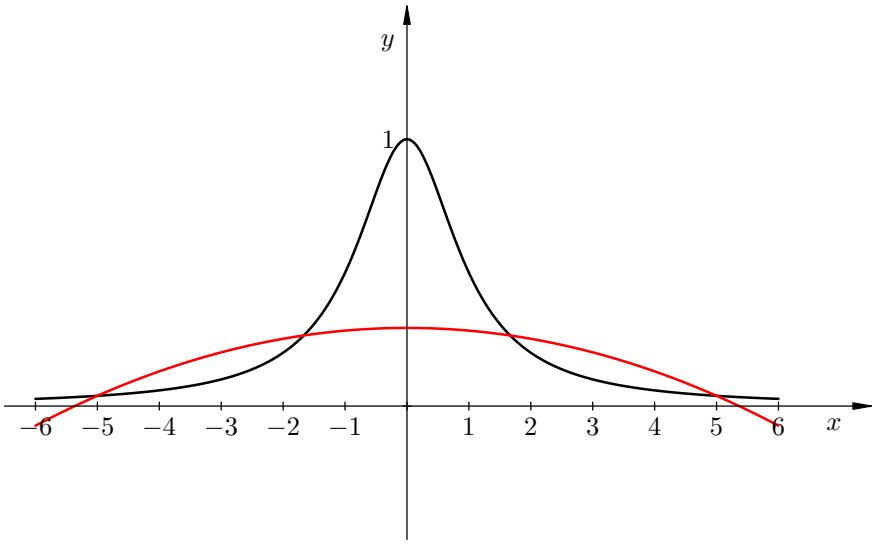
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 1.



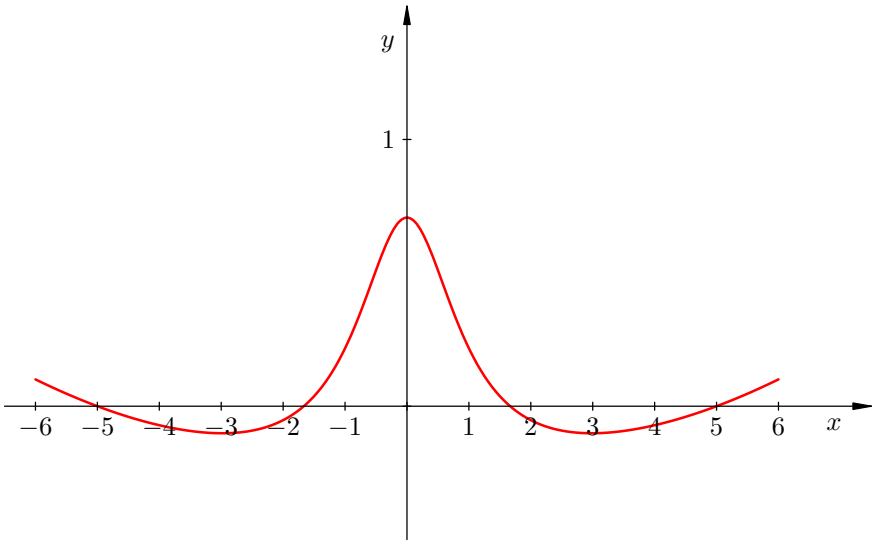
Ekvidistantna mreža, interpolacioni polinom stupnja 2.



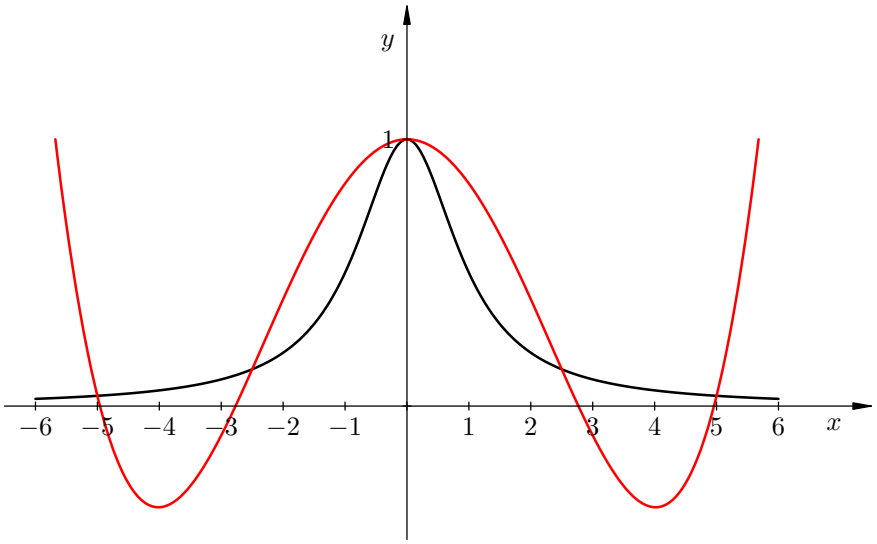
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 2.



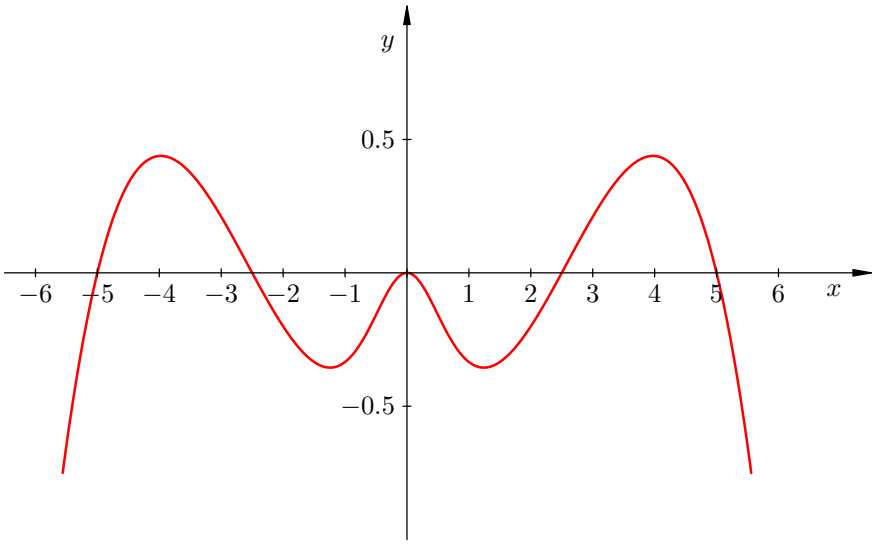
Ekvidistantna mreža, interpolacioni polinom stupnja 3.



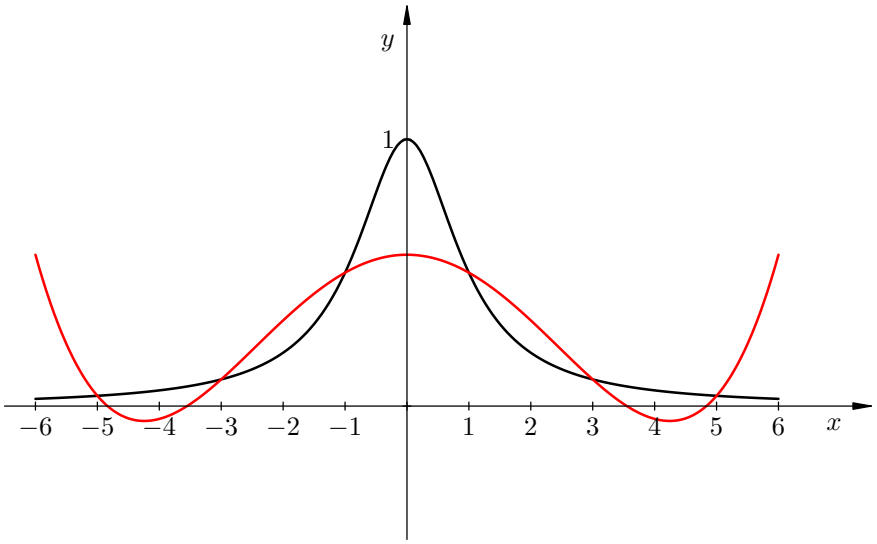
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 3.



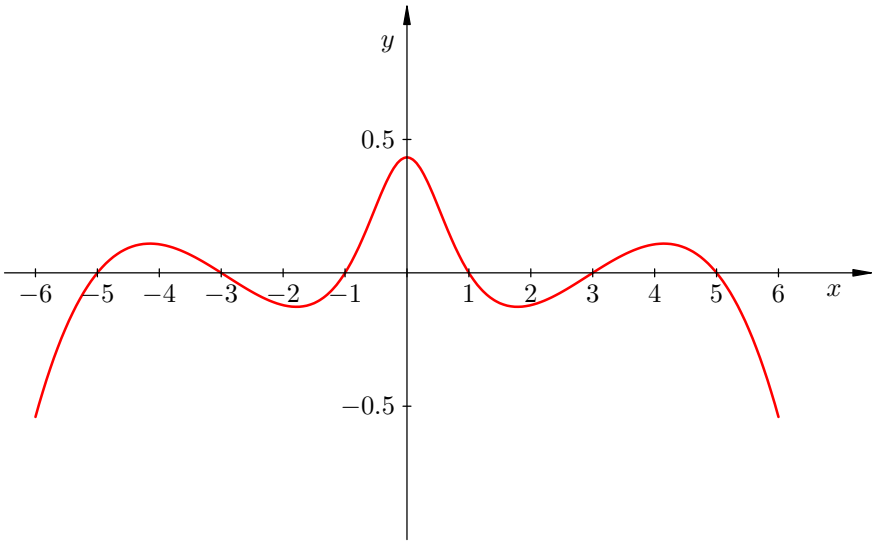
Ekvidistantna mreža, interpolacioni polinom stupnja 4.



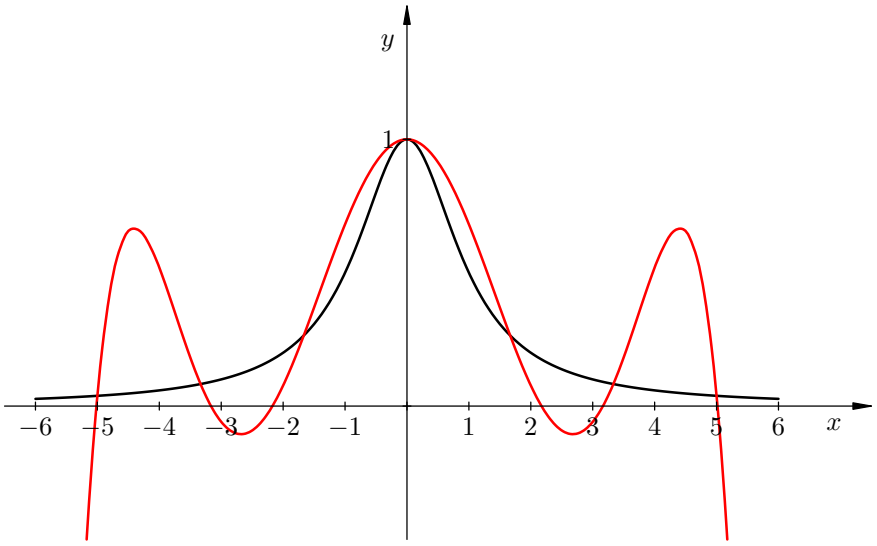
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 4.



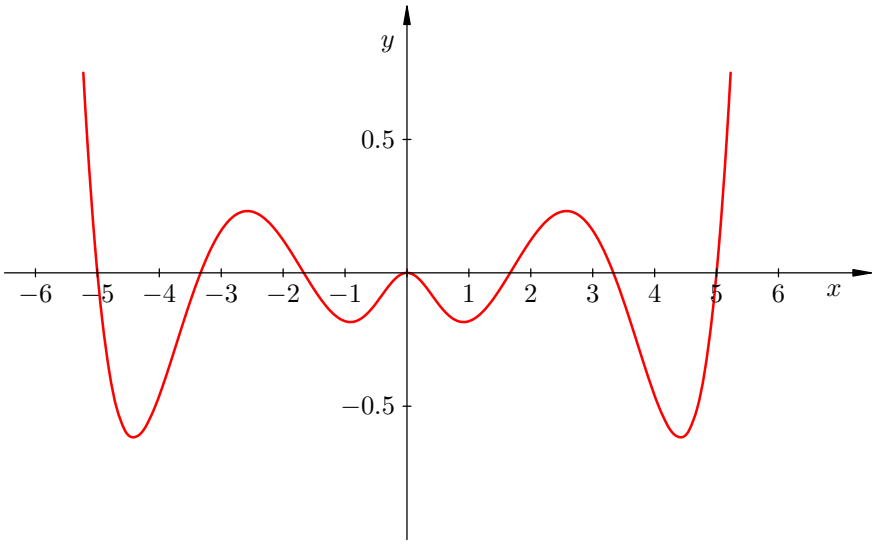
Ekvidistantna mreža, interpolacioni polinom stupnja 5.



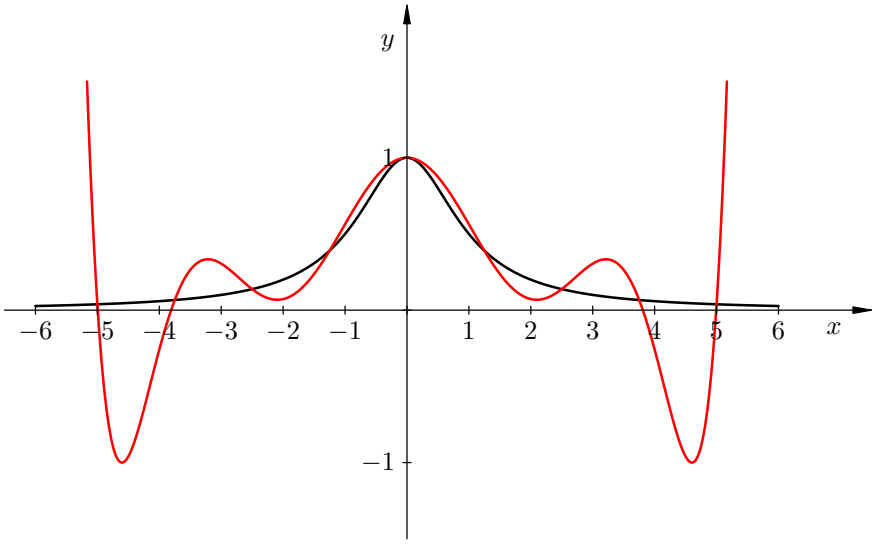
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 5.



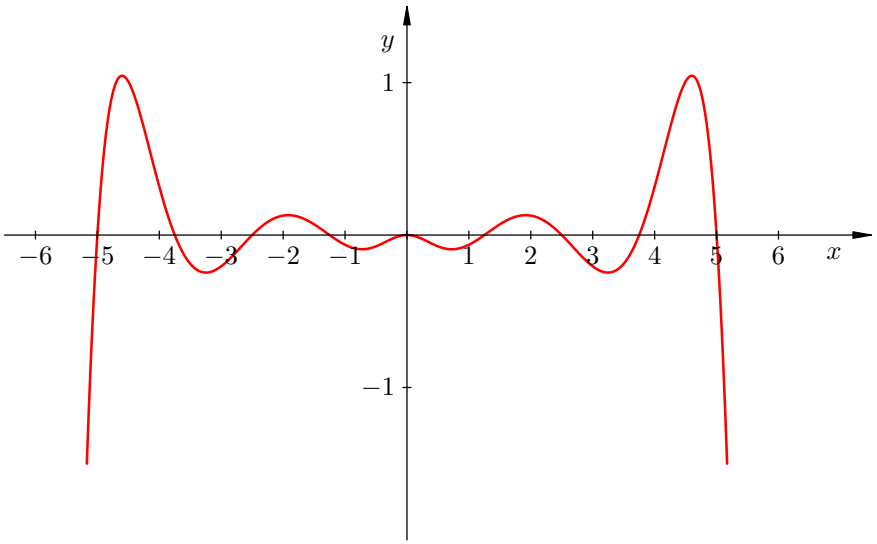
Ekvidistantna mreža, interpolacioni polinom stupnja 6.



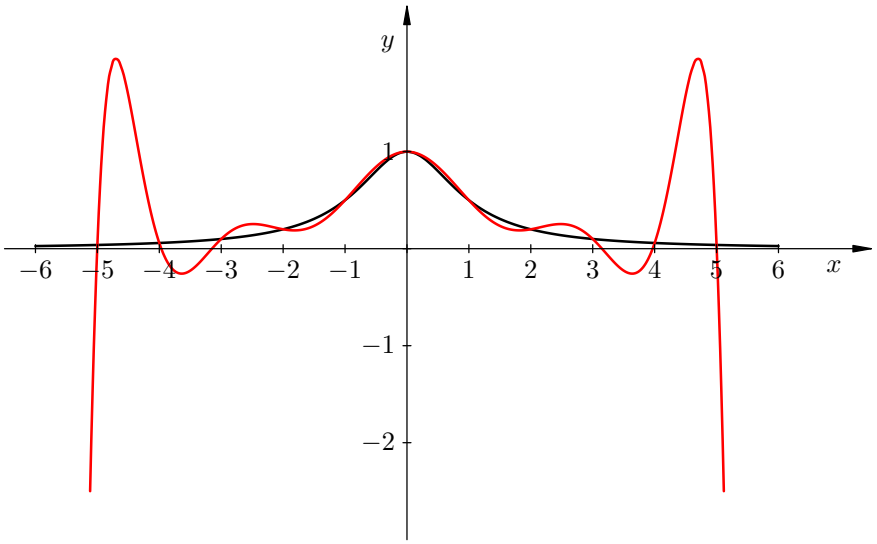
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 6.



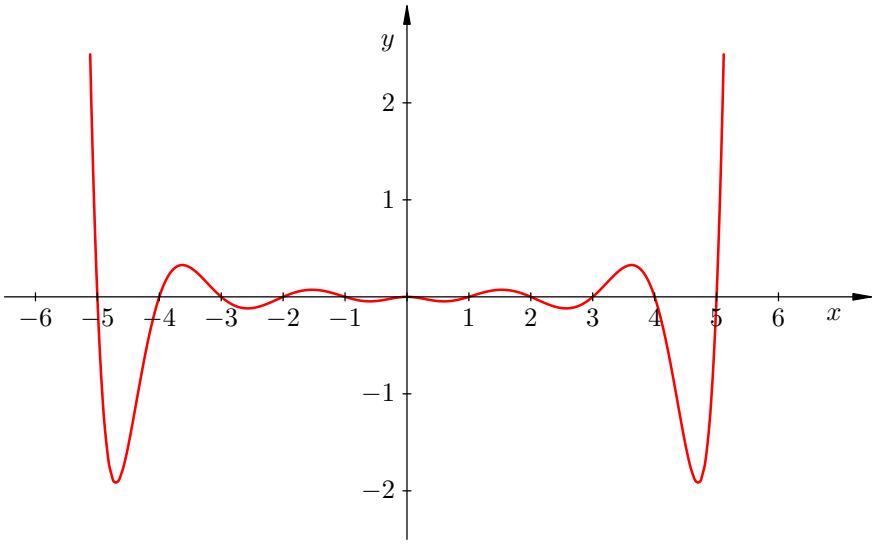
Ekvidistantna mreža, interpolacioni polinom stupnja 8.



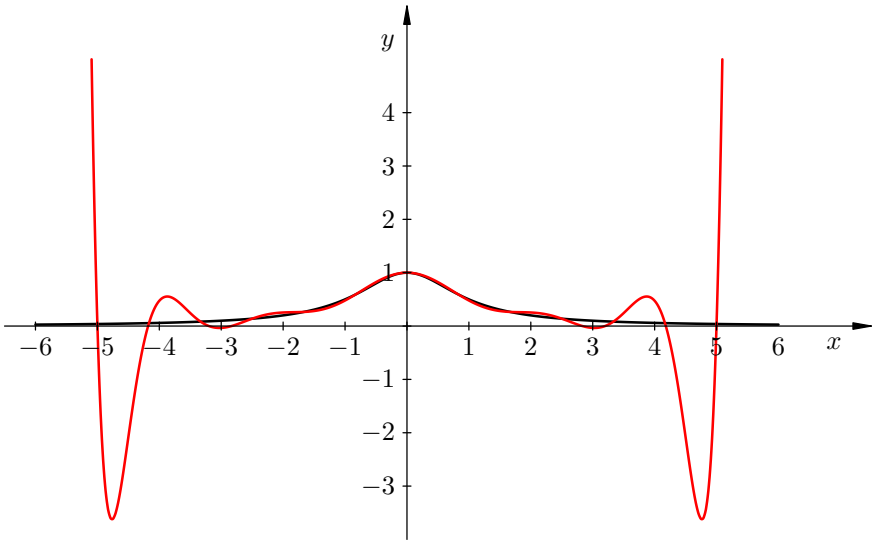
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 8.



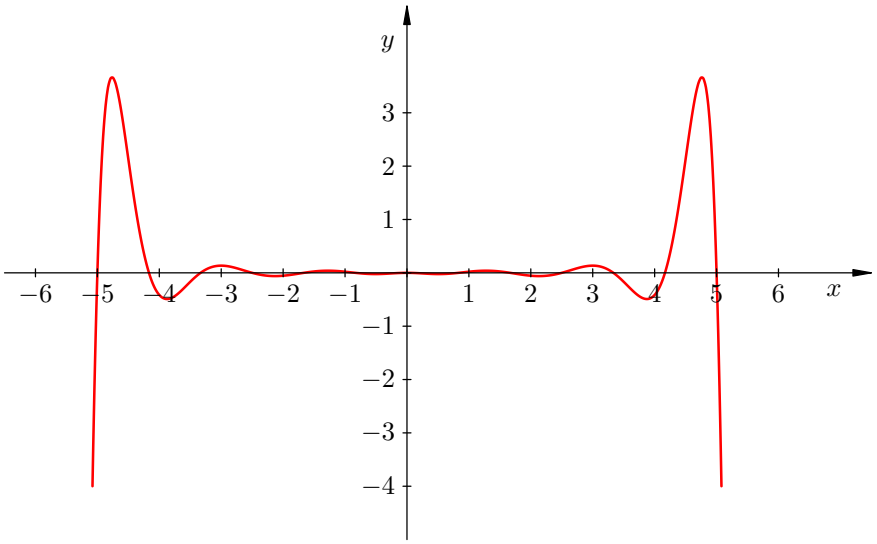
Ekvidistantna mreža, interpolacioni polinom stupnja 10.



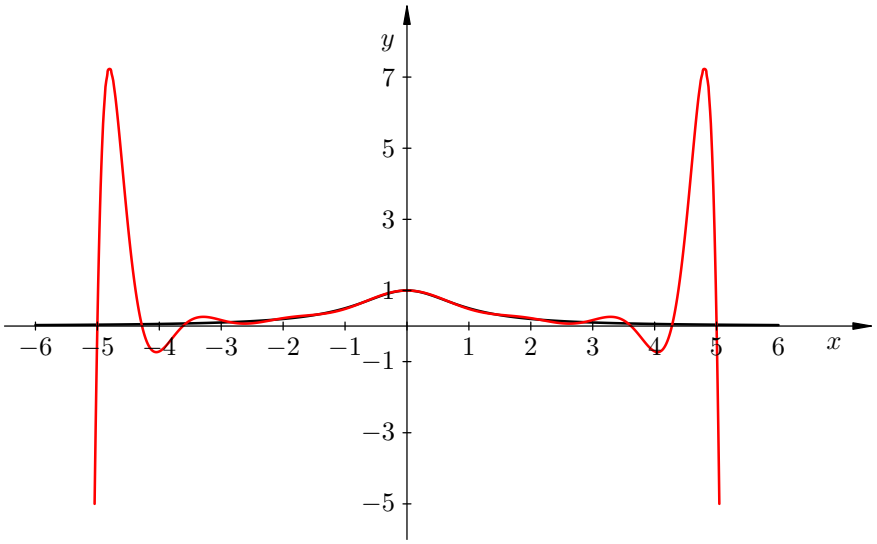
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 10.



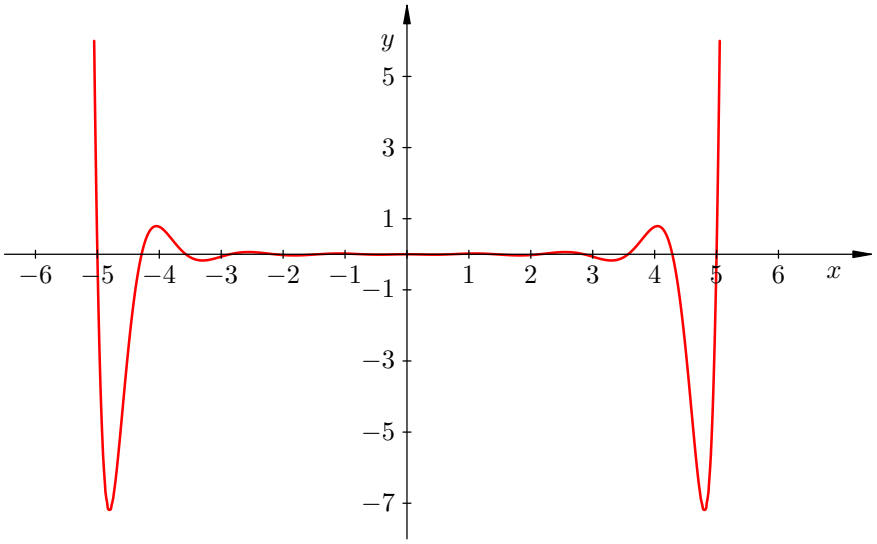
Ekvidistantna mreža, interpolacioni polinom stupnja 12.



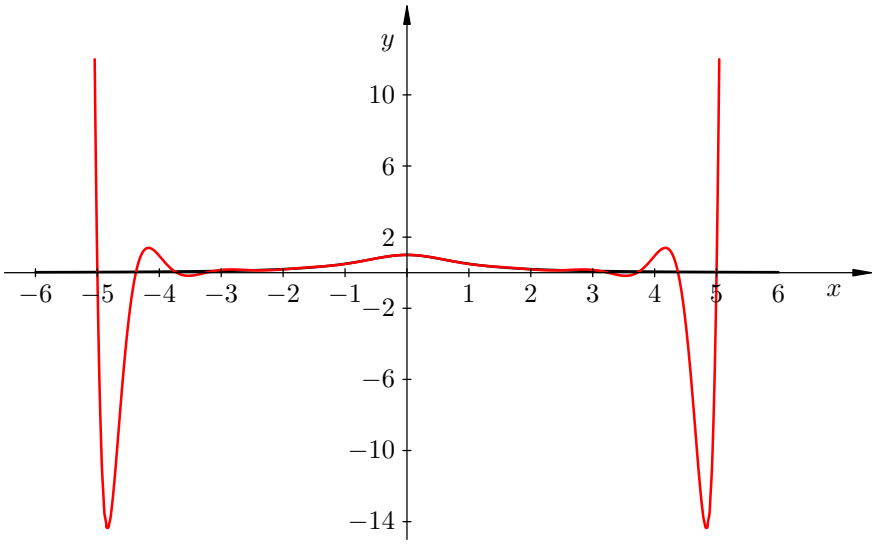
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 12.



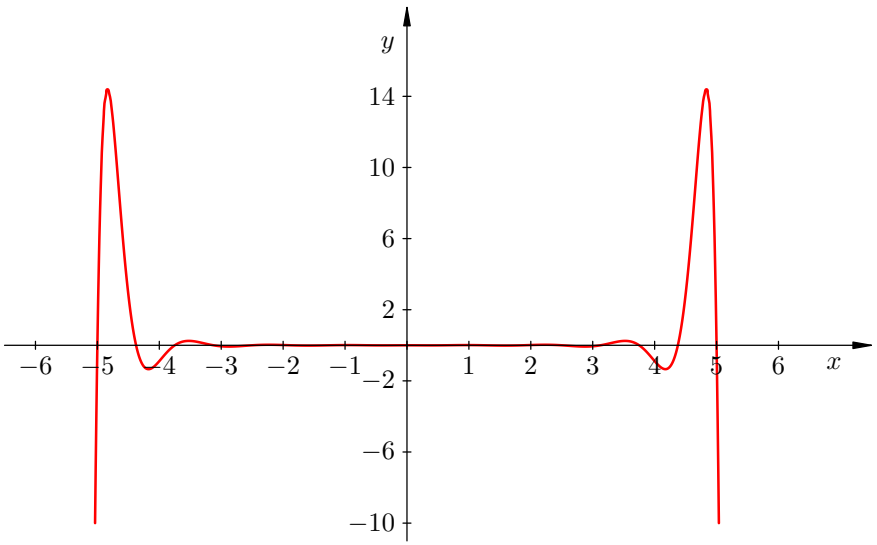
Ekvidistantna mreža, interpolacioni polinom stupnja 14.



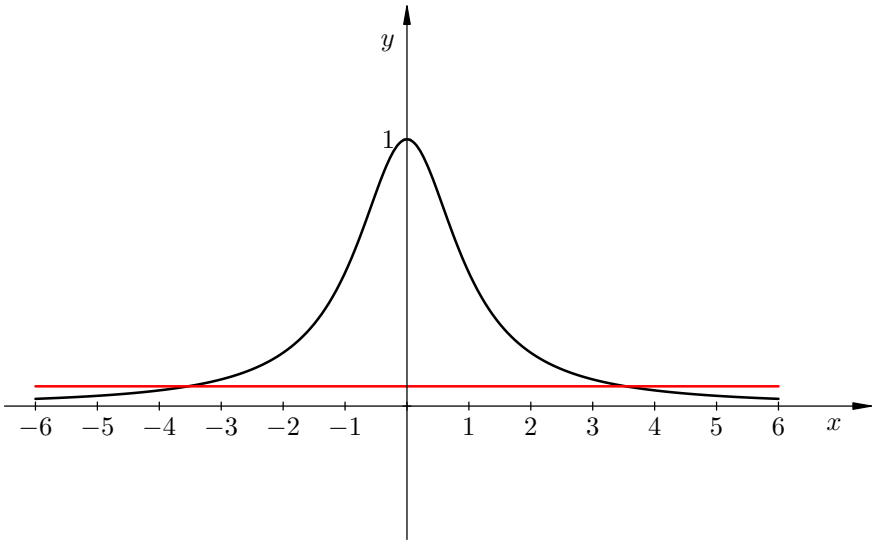
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 14.



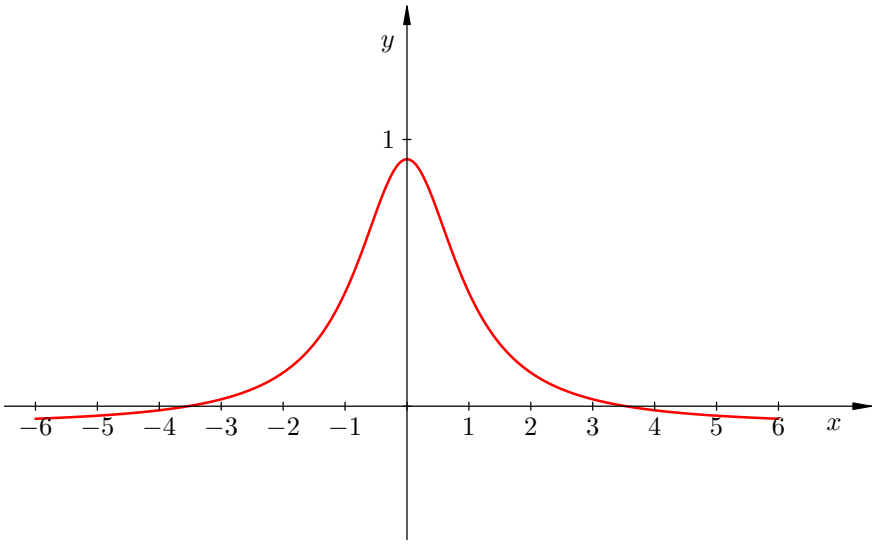
Ekvidistantna mreža, interpolacioni polinom stupnja 16.



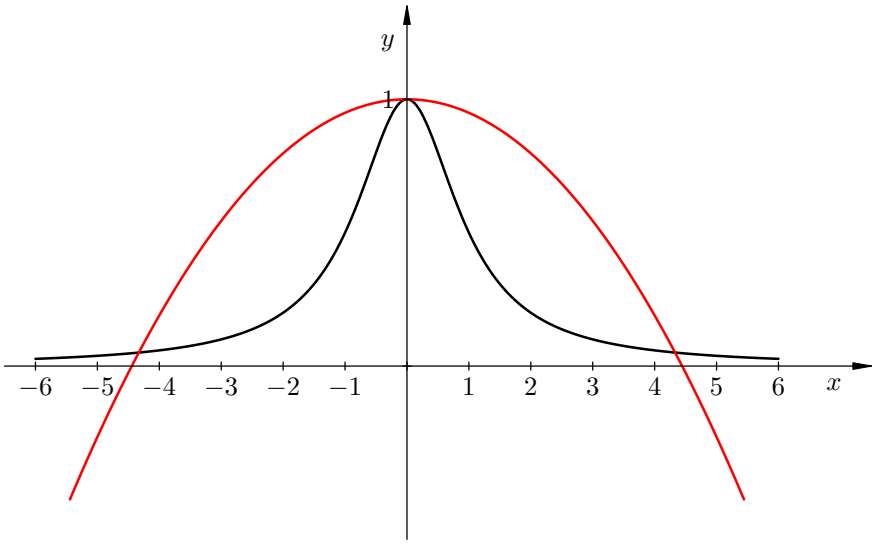
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 16.



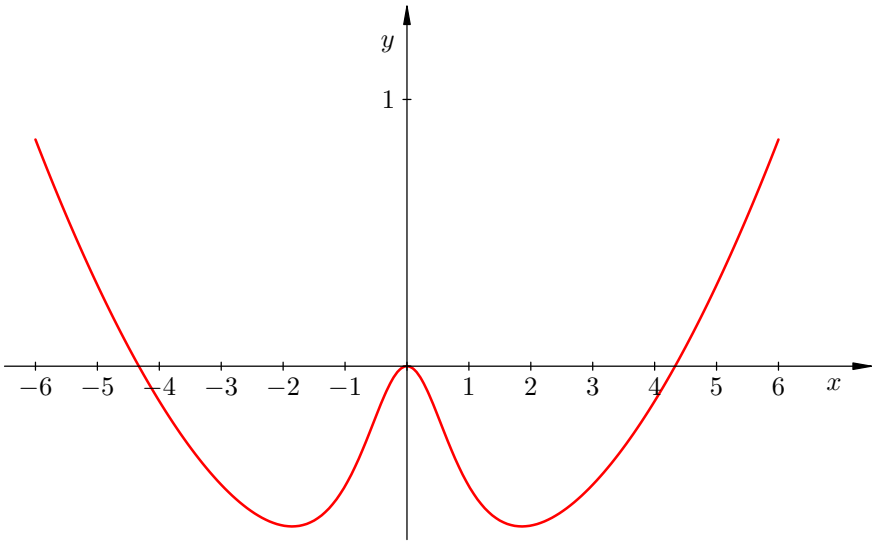
Čebiševljeva mreža, interpolacioni polinom stupnja 1.



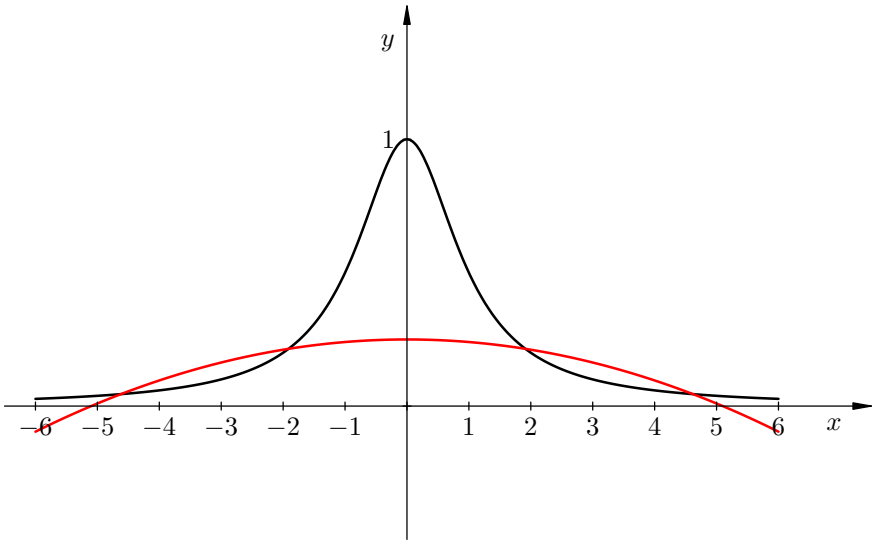
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 1.



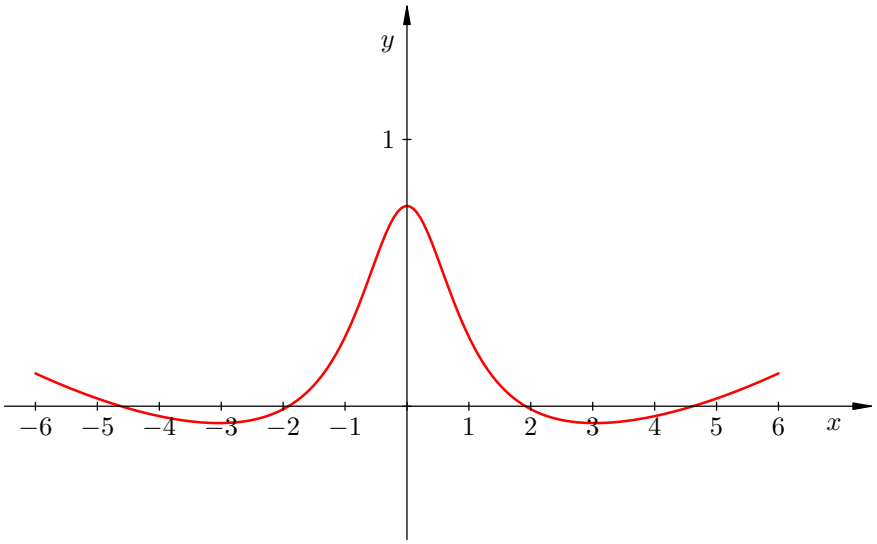
Čebiševljeva mreža, interpolacioni polinom stupnja 2.



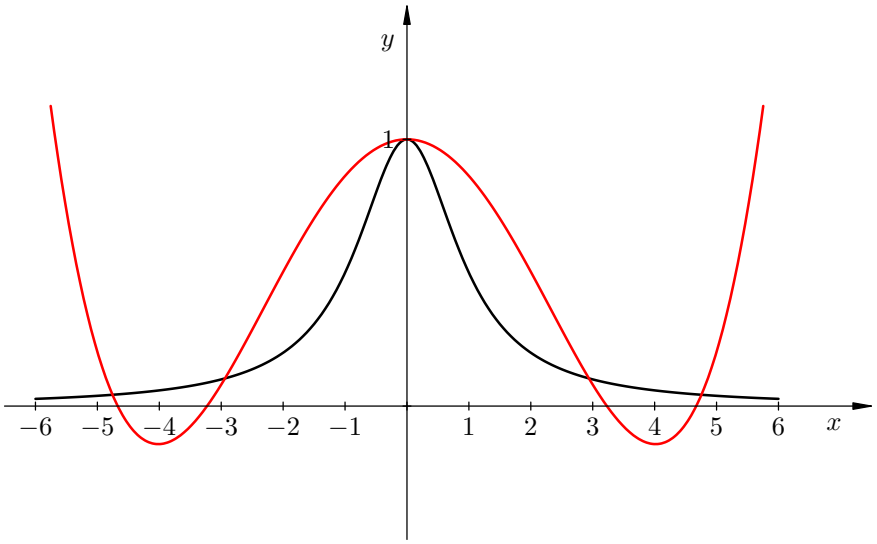
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 2.



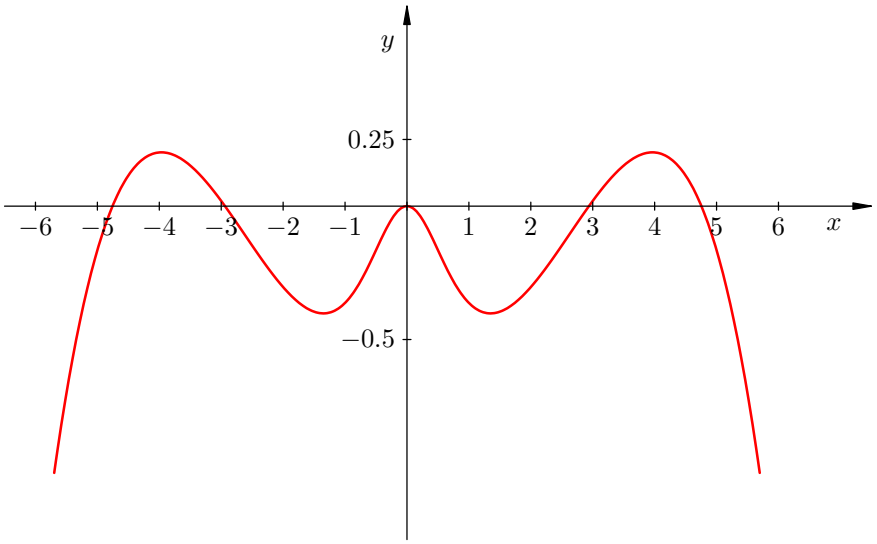
Čebiševljeva mreža, interpolacioni polinom stupnja 3.



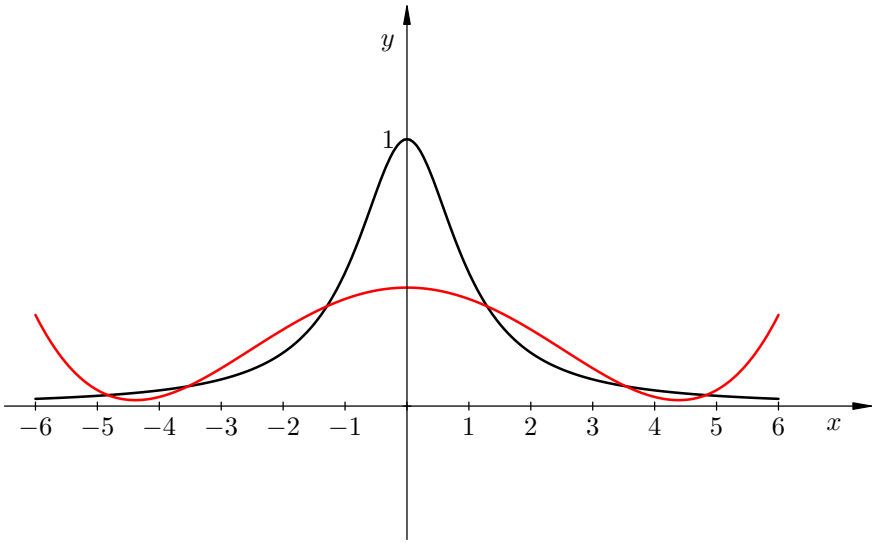
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 3.



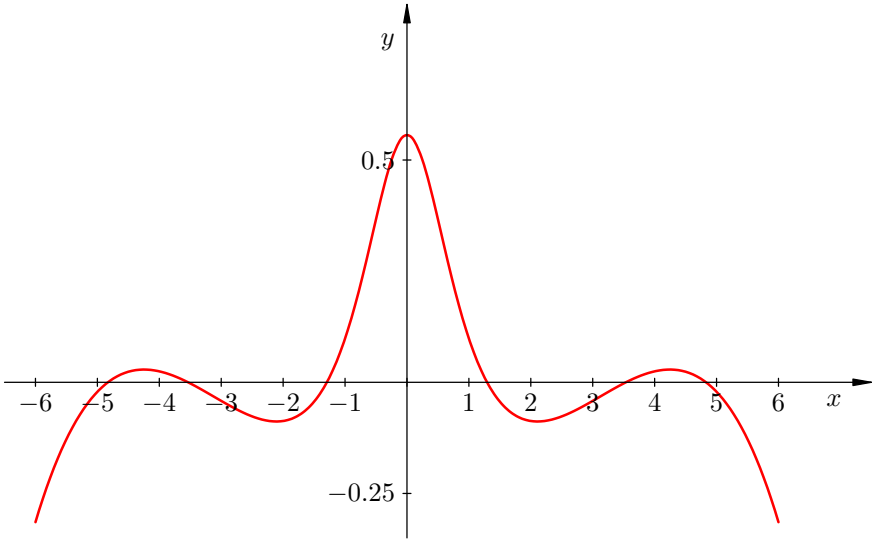
Čebiševljeva mreža, interpolacioni polinom stupnja 4.



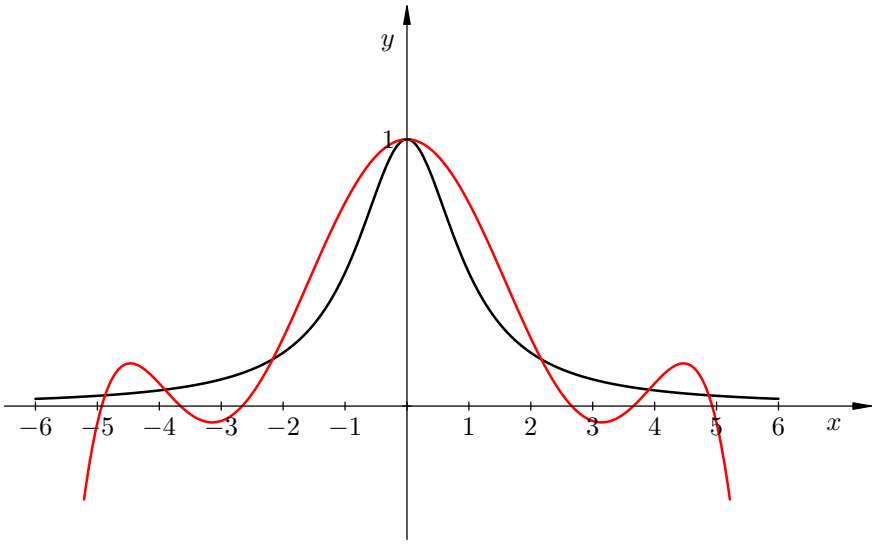
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 4.



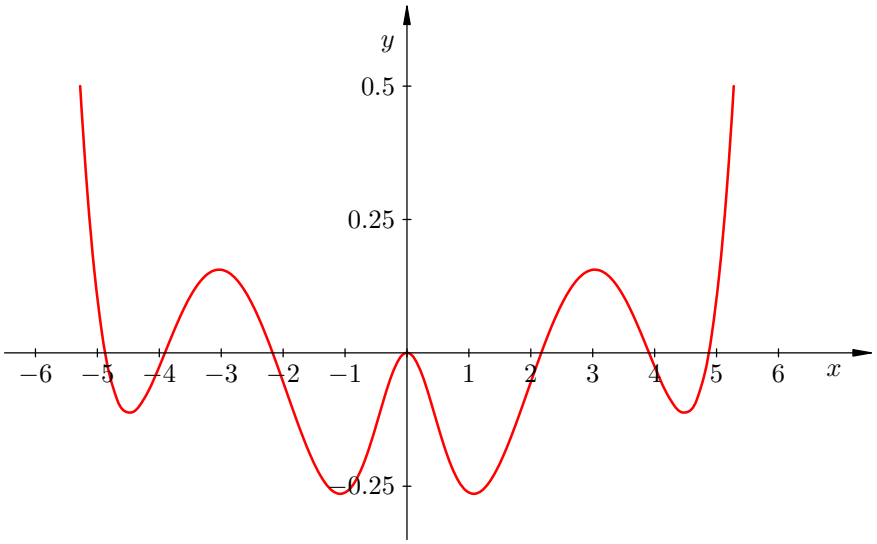
Čebiševljeva mreža, interpolacioni polinom stupnja 5.



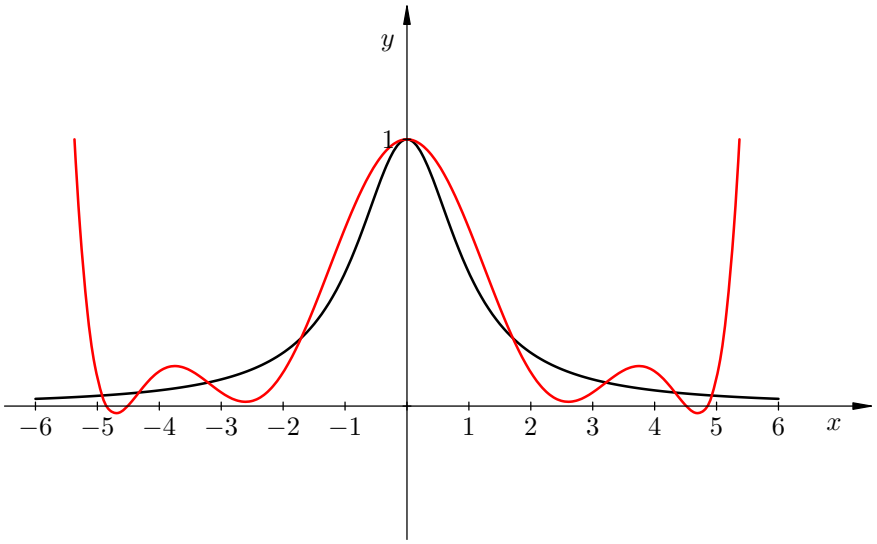
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 5.



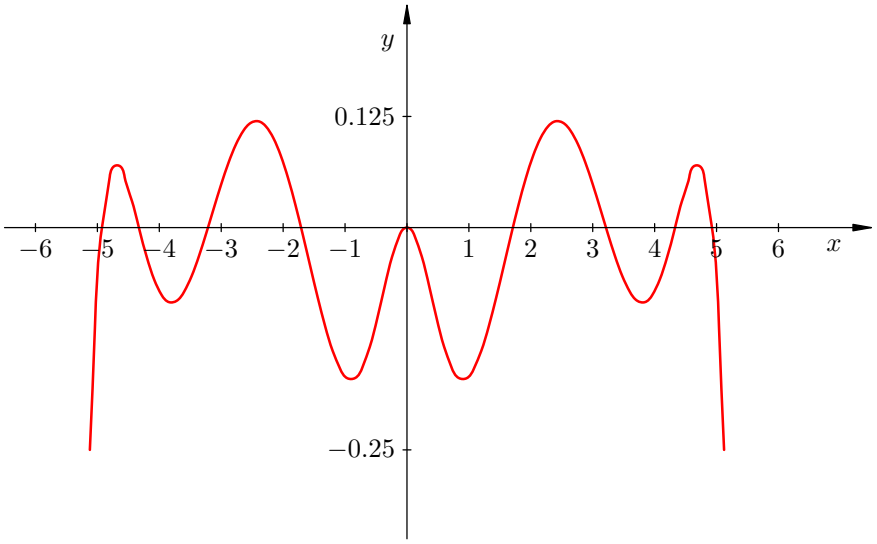
Čebiševljeva mreža, interpolacioni polinom stupnja 6.



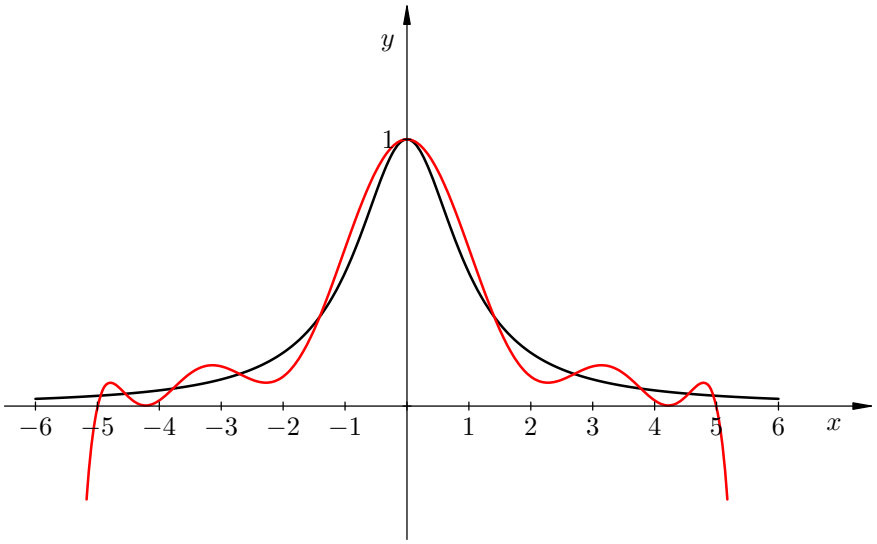
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 6.



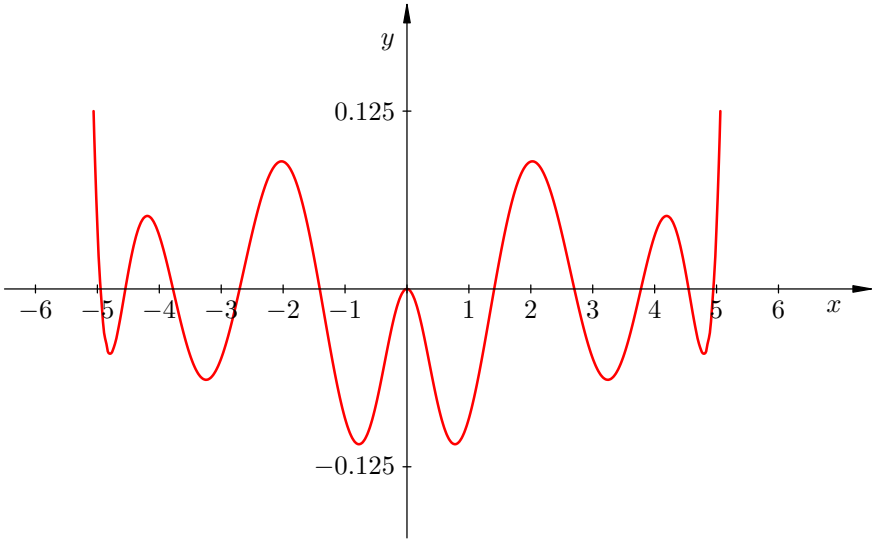
Čebiševljeva mreža, interpolacioni polinom stupnja 8.



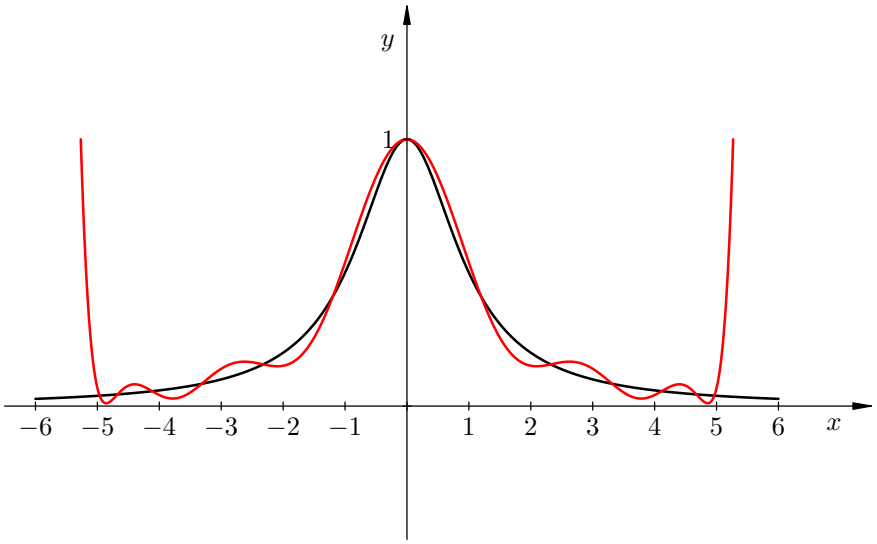
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 8.



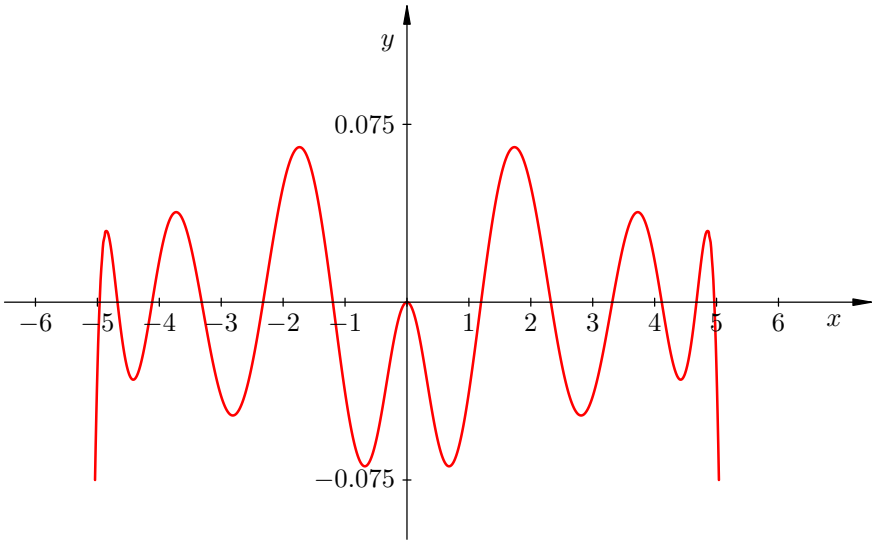
Čebiševljeva mreža, interpolacioni polinom stupnja 10.



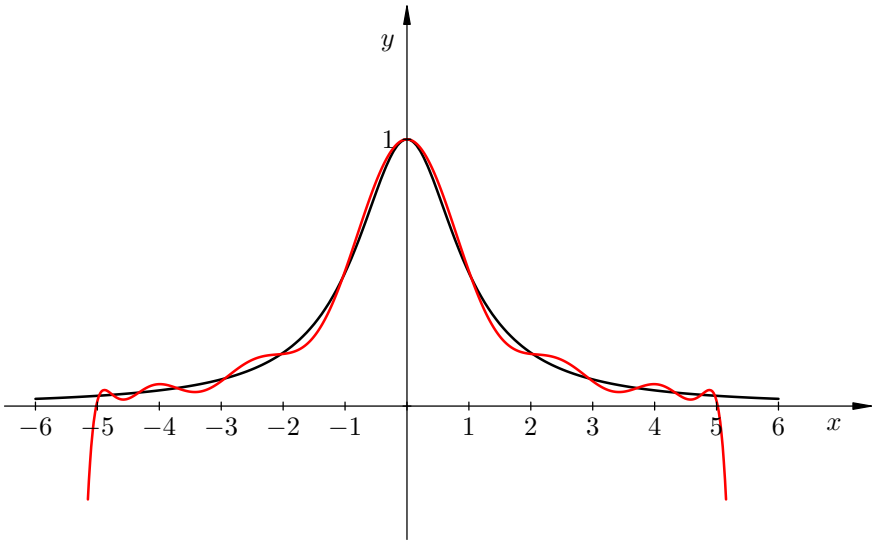
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 10.



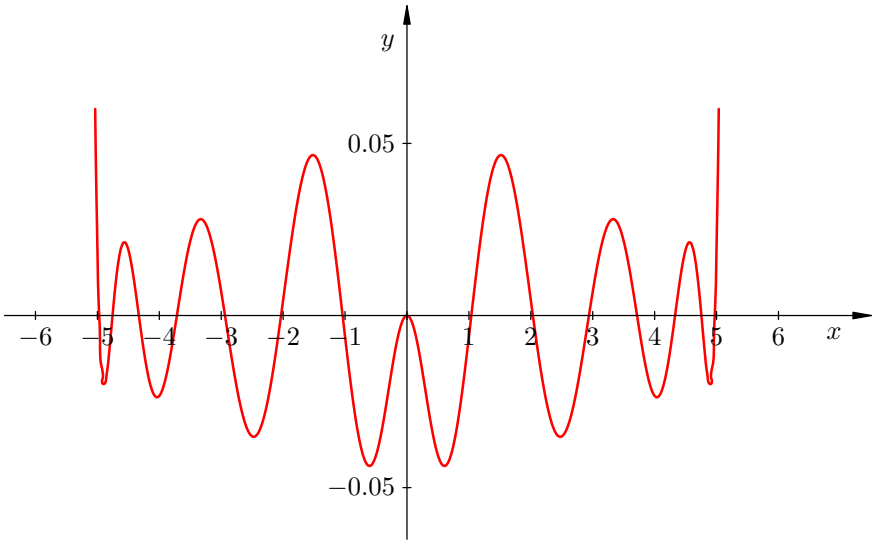
Čebiševljeva mreža, interpolacioni polinom stupnja 12.



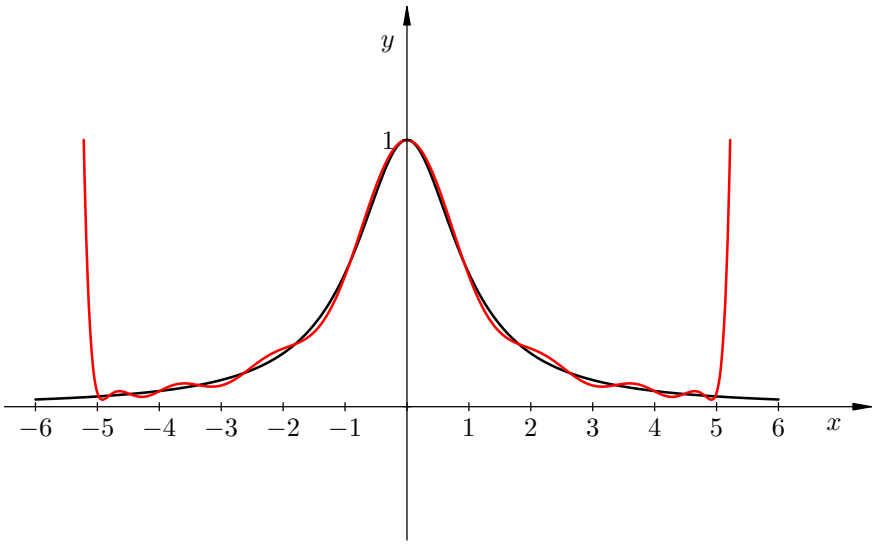
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 12.



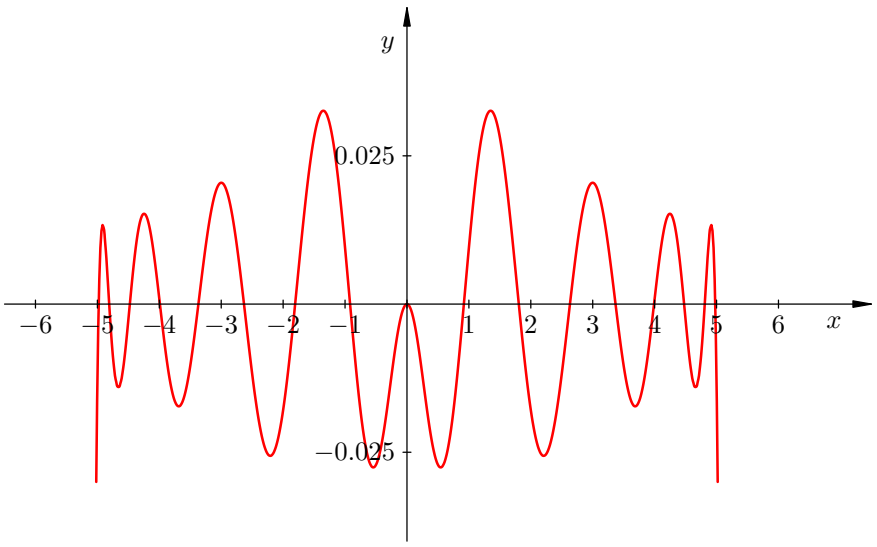
Čebiševljeva mreža, interpolacioni polinom stupnja 14.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 14.



Čebiševljeva mreža, interpolacioni polinom stupnja 16.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 16.

7.2.7. Konvergencija interpolacionih polinoma

Interpolacija polinomima vrlo je značajna zbog upotrebe u raznim postupcima u numeričkoj analizi, kao što su numerička integracija, deriviranje, rješavanje diferencijalnih jednadžbi i još mnogo toga.

Međutim, sa stanovišta teorije aproksimacije, interpolacija se ne pokazuje kao sredstvo kojim možemo doći do dobrih aproksimacija funkcija. Istina, poznati Weierstraßov teorem tvrdi da za svaku neprekidnu funkciju $f(x)$ postoji niz polinoma stupnja n, nazovimo ih $B_n(x)$, tako da

$$\|f(x) - B_n(x)\|_\infty \to 0 \qquad \text{za} \quad n \to \infty.$$

Nažalost, primjer funkcije Runge pokazuje da ovakav rezultat općenito ne vrijedi za Lagrangeove interpolacijske polinome — niz polinoma generiran ekvidistantnim mrežama ne konvergira prema toj funkciji ni po točkama (za x dovoljno blizu ruba intervala), a kamo li uniformno.

Postoje i još "gori" primjeri divergencije. Dovoljno je uzeti manje glatku funkciju od funkcije Runge.

Primjer 7.2.4. (Bernstein, 1912.) *Neka je*

$$f(x) = |x|$$

i neka je $p_n(x)$ interpolacijski polinom u $n+1$ ekvidistantnih točaka u $[-1, 1]$. Tada $|f(x) - p_n(x)| \to 0$, kad $n \to \infty$, samo u tri točke: $x = -1, 0, 1$.

Na prvi pogled se čini da to što interpolacija ne mora biti dobra aproksimacija funkcije ovisi o izboru čvorova interpolacije. To je samo djelimično točno, tj. izborom točaka interpolacije možemo poboljšati aproksimativna svojstva interpolacionih polinoma. Drugi bitni faktor kvalitete je glatkoća funkcije.

Iz primjera funkcije Runge vidi se da je Lagrangeova interpolacija dobrih svojstava aproksimacije u sredini intervala, ali ne i na rubovima. Pitanje je, da li neki izbor neekvidistantne mreže, s čvorovima koji su bliže rubovima intervala, može popraviti konvergenciju. Odgovor nije potpuno jednostavan. Iako se mogu konstruirati mreže (poput Čebiševljeve) na kojima se funkcija Runge bolje aproksimira interpolacionim polinomima, to je nemoguće napraviti za svaku neprekidnu funkciju.

Sljedeći teorem je egzistencijalnog tipa, ali ukazuje na to da je nemoguće naći dobar izbor točaka interpolacije za svaku funkciju.

Teorem 7.2.3. (Faber, 1914.) *Za svaki mogući izbor točaka interpolacije postoji neprekidna funkcija f, za čiji interpolacijski polinom $p_n(x)$ stupnja n vrijedi*

$$\|f(x) - p_n(x)\|_\infty \not\to 0.$$

7.2.8. Hermiteova i druge interpolacije polinomima

Do sada smo promatrali problem interpolacije polinomima u kojem su zadane samo funkcijske vrijednosti f_i u čvorovima interpolacije x_i. Takva interpolacija

funkcijskih vrijednosti se obično zove Lagrangeova interpolacija (čak i kad ne koristimo samo polinome kao aproksimacione ili interpolacione funkcije).

Lagrangeova interpolacija nikako ne iscrpljuje sve moguće slučajeve interpolacije polinomima. Moguće su razne generalizacije ovog problema za funkcije f koje imaju dodatna svojstva, recimo, veći broj derivacija (globalno, ili barem, u okolini svakog čvora).

Da bismo jednostavno došli do tih generalizacija, ponovimo ukratko "izvod" i konstrukciju Lagrangeove interpolacije polinomom. Traženi polinom p_n mora zadovoljavati interpolacione jednadžbe

$$p_n(x_i) = f_i = f(x_i), \quad i = 0, \ldots, n. \tag{7.2.14}$$

Zapis polinoma p_n u standardnoj bazi potencija $1, x, \ldots, x^n$ vodi na linearni sustav s Vandermondeovom matricom, a za pripadnu Vandermondeovu determinantu (vidjeti teorem 7.2.1.) pokazali smo da vrijedi

$$V(x_0, \ldots, x_n) := \det \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} = \prod_{0 \le i < j \le n} (x_j - x_i). \tag{7.2.15}$$

Iz pretpostavke o međusobnoj različitosti čvorova x_k slijedi regularnost sustava i egzistencija i jedinstvenost polinoma p_n.

Lagrangeov interpolacijski polinom p_n može se napisati i eksplicitno u tzv. **Lagrangeovoj formi**, koja se često zove i **Lagrangeova interpolacijska formula**. Ako definiramo $n+1$ polinom $\{\ell_i(x)\}_{i=0}^n$ specijalnim interpolacijskim uvjetima

$$\ell_i(x_j) := \delta_{ij}, \tag{7.2.16}$$

gdje je δ_{ij} Kroneckerov simbol, tada Lagrangeov interpolacijski polinom koji udovoljava uvjetima (7.2.14) možemo zapisati kao

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x). \tag{7.2.17}$$

Tražene polinome ℓ_i stupnja n, koji su jednoznačno određeni interpolacijskim uvjetima (7.2.16), možemo "pogoditi"

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \ne i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, \ldots, n. \tag{7.2.18}$$

Funkcije ℓ_i zovu se funkcije **Lagrangeove baze**.

Zadatak 7.2.2. *Dokažite da su funkcije Lagrangeove baze linearno nezavisne i čine skup izvodnica za prostor polinoma stupnja n, što opravdava naziv baza.*

Postoji još jedan slučaj koji se može riješiti jednostavnom formulom, a posebno ga tretiramo zbog važnosti za teoriju numeričke integracije (preciznije, Gaussovih integracionih formula). U svakom čvoru x_i, osim funkcijske vrijednosti $f_i = f(x_i)$, interpoliramo i vrijednost derivacije $f_i' = f'(x_i)$.

Teorem 7.2.4. *Postoji jedinstveni polinom h_{2n+1} stupnja najviše $2n+1$, koji zadovoljava interpolacijske uvjete*

$$h_{2n+1}(x_i) = f_i, \quad h'_{2n+1}(x_i) = f_i', \quad i = 0, \ldots, n,$$

gdje su x_i međusobno različite točke i f_i, f_i' zadani realni brojevi.

Dokaz:

Egzistenciju polinoma $h_{2n+1}(x)$ možemo dokazati konstruktivnim metodama — konstrukcijom eksplicitne baze, slično kao i za Lagrangeov polinom. Neka su

$$\begin{aligned} h_{i,0}(x) &= [1 - 2(x - x_i)\ell_i'(x_i)]\, \ell_i^2(x) \\ h_{i,1}(x) &= (x - x_i)\, \ell_i^2(x), \end{aligned} \tag{7.2.19}$$

gdje su ℓ_i funkcije Lagrangeove baze (7.2.18). Direktno možemo provjeriti da su $h_{i,0}(x)$ i $h_{i,1}(x)$ polinomi stupnja $2n+1$ koji zadovoljavaju sljedeće relacije

$$\begin{aligned} h_{i,0}(x_j) &= \delta_{ij}, \quad & h_{i,1}(x_j) &= 0, \\ h'_{i,0}(x_j) &= 0, & h'_{i,1}(x_j) &= \delta_{ij}, \end{aligned} \quad \text{za} \quad i, j = 0, \ldots, n.$$

Ako definiramo polinom formulom

$$h_{2n+1}(x) = \sum_{i=0}^{n} \Big(f_i\, h_{i,0}(x) + f_i'\, h_{i,1}(x) \Big), \tag{7.2.20}$$

lagano provjerimo da h_{2n+1} zadovoljava uvjete teorema.

Obzirom da iz gornjeg ne slijedi jedinstvenost, moramo ju dokazati posebno. Neka je $q_{2n+1}(x)$ bilo koji drugi polinom koji ispunjava interpolacijske uvjete teorema. Tada je $h_{2n+1}(x) - q_{2n+1}(x)$ polinom stupnja ne većeg od $2n+1$, koji ima nultočke multipliciteta barem 2 u svakom čvoru interpolacije x_i, tj. barem $2n+2$ nultočke, što je moguće samo ako je identički jednak nuli. ■

Polinomi $h_{i,0}$, $h_{i,1}$, zovu se funkcije **Hermiteove baze**, a polinom h_{2n+1} obično se zove **Hermiteov interpolacijski polinom**.

Zadatak 7.2.3. *Pokažite da za funkcije Lagrangeove, odnosno Hermiteove baze, vrijedi*

$$\sum_{i=0}^{n} \ell_i(x) = 1, \qquad \sum_{i=0}^{n} h_{i,0}(x) = 1.$$

Zadatak 7.2.4. *Pokažite da za funkcije Lagrangeove, odnosno Hermiteove baze, vrijedi*

$$\sum_{i=0}^{n} x_i h_{i,0}(x) + h_{i,1}(x) = x, \qquad \sum_{i=0}^{n} (x - x_i)\, \ell_i^2(x)\, \ell_i'(x_i) = 0.$$

Za ocjenu greške Hermiteove interpolacije vrijedi vrlo sličan rezultat kao i za običnu Lagrangeovu interpolaciju (teorem 7.2.2.).

Teorem 7.2.5. *Greška kod interpolacije Hermiteovim polinomom $h_{2n+1}(x)$ (v. teorem 7.2.4.) funkcije $f \in C^{(2n+2)}[x_{\min}, x_{\max}]$ u $n+1$ čvorova $x_0, \ldots, x_n$ je oblika*

$$e(x) := f(x) - h_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\, \omega^2(x),$$

gdje su ξ i ω kao u teoremu 7.2.2.

Dokaz:

Iz uvjeta interpolacije znamo da je $f(x) = h_{2n+1}(x)$ i $f'(x) = h'_{2n+1}(x)$ za $x = x_0, \ldots, x_n$, pa očekujemo da je

$$f(x) - h_{2n+1}(x) \approx C\omega^2(x)$$

za neku konstantu C. Definiramo li

$$F(x) = f(x) - h_{2n+1}(x) - C\omega^2(x),$$

vidimo da F ima nultočke multipliciteta 2 u $x_0, \ldots, x_n$, tj. $F(x_k) = F'(x_k) = 0$ za $k = 0, \ldots, n$. Izaberemo li neki $x_{n+1} \in [x_{\min}, x_{\max}]$ različit od postojećih čvorova, možemo odrediti konstantu C tako da vrijedi $F(x_{n+1}) = 0$. Kako $F(x)$ sada ima (barem) $n+2$ nule, F' ima $n+1$ nulu u nekim točkama između njih. Ona također ima nule u $x_0, \ldots, x_n$, pa ukupno ima (barem) $2n+2$ nula. No onda F'' ima bar $2n+1$ nula, F''' $2n$ nula, itd., na osnovu Rolleovog teorema. Na kraju, $F^{(2n+2)}$ ima barem jednu nulu u promatranom intervalu, označimo ju s ξ. Deriviranjem izraza za $F(x)$ dobijemo

$$F^{(2n+2)}(\xi) = f^{(2n+2)}(\xi) - C(2n+2)! = 0,$$

odakle izračunamo C. Uvrstimo li taj rezultat u izraz za grešku, dobijemo

$$F(x_{n+1}) - h_{2n+1}(x_{n+1}) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\, \omega^2(x_{n+1}).$$

Ali kako je x_{n+1} proizvoljan, različit samo od čvorova $x_0, \ldots, x_n$, možemo ga zamijeniti s proizvoljnim x. Na kraju primijetimo da je gornji rezultat točan i za $x \in \{x_0, \ldots, x_n\}$, jer su obje strane nula, pa dokaz slijedi. ■

Hermiteov interpolacioni polinom, naravno, osim u "Lagrangeovom" obliku, možemo zapisati i u "Newtonovom" obliku — koristeći podijeljene razlike, ali sada i s dvostrukim čvorovima. Što to znači? Pokušajte ga sami izvesti! (Taj oblik ćemo kasnije uvesti i iskoristiti za zapis po dijelovima polinomne interpolacije.)

Ponekad se naziv "Hermiteova interpolacija" koristi i za općenitiji slučaj **proširene Hermiteove interpolacije** koji uključuje i više derivacije od prvih. Bitno je samo da u određenom čvoru x_i interpoliramo **redom** funkcijsku vrijednost i prvih nekoliko (uzastopnih) derivacija.

Pretpostavimo da u čvoru x_i koristimo $l_i > 0$ podataka (funkcija i prvih $l_i - 1$ derivacija). Tada je zgodno gledati x_i kao čvor multipliciteta $l_i \geq 1$ i uvesti posebne oznake t_j za međusobno različite čvorove (uzmimo da ih je $d+1$):

$$x_0 \leq \cdots \leq x_n = \underbrace{t_0, \ldots, t_0}_{l_0}, \ldots, \underbrace{t_d, \ldots, t_d}_{l_d},$$

uz $t_i \neq t_j$ za $i \neq j$, s tim da je $l_0 + \cdots + l_d = n+1$. Problem proširene Hermiteove interpolacije, također, ima jedinstveno rješenje.

Zadatak 7.2.5. *Neka su $t_0, t_1, \ldots, t_d$ zadani međusobno različiti čvorovi i neka su $l_0, l_1, \ldots, l_d$ zadani prirodni brojevi koji zadovoljavaju $\sum_{i=0}^{d} l_i = n+1$. Pokažite da za svaki skup realnih brojeva*

$$\{f_{ij} \mid j = 1, \ldots, l_i,\ i = 0, \ldots, d\}$$

postoji jedinstveni polinom h_n, stupnja ne većeg od n, za koji vrijedi

$$h_n^{(j-1)}(t_i) = f_{ij}, \quad j = 1, \ldots, l_i, \quad i = 0, \ldots, d.$$

Uputa: Konstrukcija Hermiteove baze postaje vrlo komplicirana (pokušajte!). Zato zapišite h_n kao linearnu kombinaciju potencija, formulirajte problem interpolacije matrično i analizirajte determinantu dobivenog linearnog sustava. Ta determinanta je generalizacija Vandermondeove determinante iz (7.2.15)*, bez pretpostavke da su čvorovi različiti, pa ju, također, označavamo s $V(x_0, \ldots, x_n)$. Dokažite da vrijedi*

$$V(x_0, \ldots, x_n) = \prod_{0 \leq i < j \leq d} (t_j - t_i)^{l_i l_j} \cdot \prod_{i=0}^{d} \prod_{\nu=1}^{l_i - 1} \nu!\,,$$

odakle slijedi egzistenicija i jedinstvenost polinoma h_n.

Općeniti slučaj interpolacije funkcije i derivacija, koji obuhvaća gornje interpolacije kao specijalni slučaj, može se zapisati na sljedeći način. Neka je E matrica tipa $(m+1) \times (n+1)$ s elementima E_{ij} koji su svi 0, osim $n+1$ njih, koji su jednaki 1, i neka je zadan skup od $m+1$ točaka $x_0 < x_1 < \cdots < x_m$. Tada problem nalaženja polinoma $P(x)$ stupnja n koji zadovoljava

$$E_{ij}(P^{(j-1)}(x_i) - c_{ij}) = 0, \quad i = 0, \ldots, m, \quad j = 1, \ldots, n+1,$$

za neki izbor brojeva c_{ij}, zovemo **Hermite–Birkhoffovim** interpolacijskim problemom. U punoj općenitosti, kako je formuliran, problem može i nemati rješenje. Identifikacija matrica E koje vode na regularne sisteme jednadžbi već je dosta izučena. I na kraju, spomenimo da i time problem nije do kraja iscrpljen. Moguće je umjesto derivacija zadavati razne linearne funkcionale u čvorovima. Jedan specijalni problem u kojem su ovi linearni funkcionali linearne kombinacije derivacija, donekle je proučen. Taj se problem često naziva **proširena Hermite–Birkhoffova interpolacija**, a u vezi je s numeričkim metodama za rješavanje diferencijalnih jednadžbi.

Zadatak 7.2.6. *Zapisom polinoma P u standardnoj bazi potencija formulirajte matrično problem proširene Hermite–Birkhoffove interpolacije.*

7.3. Optimalni izbor čvorova interpolcaije

Ako se prisjetimo problema interpolacije, onda znamo da je greška interpolacionog polinoma stupnja n jednaka

$$f(x) - p_n(x) = \frac{(x - x_0) \cdots (x - x_n)}{(n+1)!} f^{(n+1)}(\xi).$$

Vrijednost $(n+1)$-ve derivacije ovisi o točkama interpolacije, ali nije jednostavno reći kako. Ipak, ono što možemo kontrolirati je izbor točaka interpolacije. Pretpostavimo da interpoliramo funkciju na intervalu $[-1, 1]$. Ako naš interval nije $[-1, 1]$, nego $[a, b]$, onda ga linearnom transformacijom

$$y = cx + d$$

možemo svesti na zadani interval.

Ideja za optimalni izbor čvorova interpolacije je **minimizirati** faktor koji ovisi o samo čvorovima, a to je

$$(x - x_0) \cdots (x - x_n).$$

Minimizaciju radimo globalno, preko cijelog intervala $[-1, 1]$, i to uniformno — u max normi. Želimo izaberati točke interpolacije $x_j \in [-1, 1]$ tako da minimiziraju

$$\max_{-1 \le x \le 1} |(x - x_0) \cdots (x - x_n)|.$$

Dobivamo problem minimaks aproksimacije. Da bismo ga riješili, trebamo pojam Čebiševljevih polinoma (prve vrste) i jedno njihovo svojstvo.

7.3.1. Čebiševljevi polinomi prve vrste

Čebiševljevi polinomi prve vrste obično se označavaju s T_n. Najjednostavnija definicija ovih polinoma je preko veze između cosinusa nekog kuta i cosinusa n-terostrukog kuta

$$T_n(x) = \cos n\vartheta, \quad x = \cos\vartheta,$$

za bilo koji $n \geq 0$. Ovu relaciju možemo napisati i u eksplicitnom obliku

$$T_n(x) = \cos(n \arccos x), \quad x \in [-1, 1].$$

Očito $T_0(x) = 1$ i $T_1(x) = x$, ali se ne vidi odmah da je T_n polinom stupnja n, za svaki $n \geq 0$.

Međutim, iz adicionog teorema za $\cos(n+1)\vartheta$ i $\cos(n-1)\vartheta$, lako se dokaže da funkcije T_n zadovoljavaju tročlanu rekurzivnu relaciju

$$T_{n+1}(x) - 2xT_n(x) + T_{n-1}(x) = 0,$$

uz start

$$T_0(x) = 1, \quad T_1(x) = x.$$

Iz ove rekurzivne relacije odmah slijedi da je T_n polinom stupnja n.

Čebiševljevi polinomi T_n su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = \frac{1}{\sqrt{1-x^2}}.$$

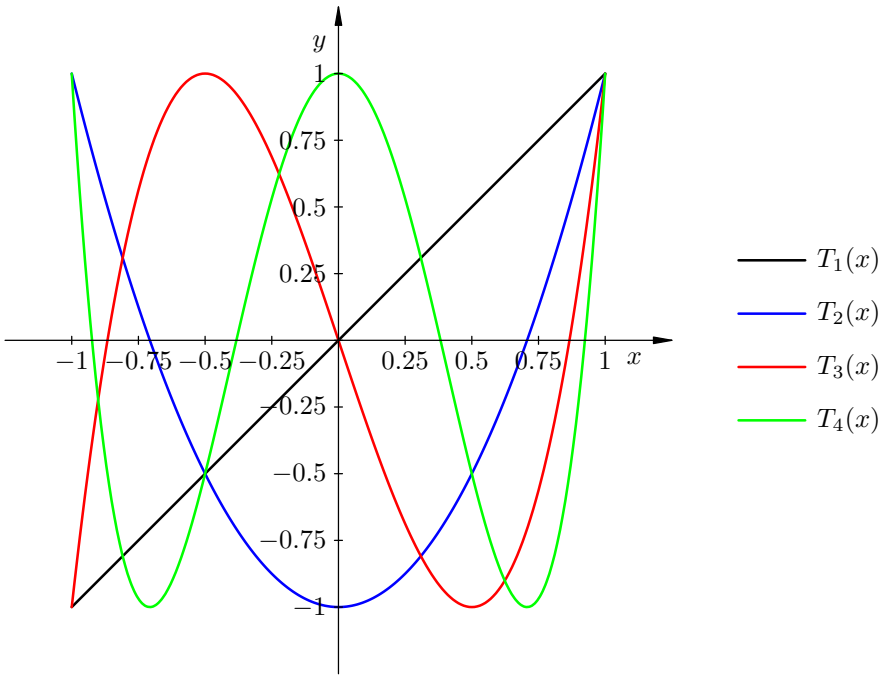
Pripadna relacija ortogonalnosti je

$$\int_{-1}^{1} \frac{T_m(x)\, T_n(x)}{\sqrt{1-x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0. \end{cases}$$

Osim toga, n-ti Čebiševljev polinom prve vrste T_n zadovoljava diferencijalnu jednadžbu

$$(1-x^2)y'' - xy' + n^2 y = 0.$$

Graf prvih par polinoma izgleda ovako.



Nultočke i ekstreme polinoma T_{n+1} nije teško izračunati. Nultočke pripadnog kosinusa su na odgovarajućem intervalu su

$$(n+1)\vartheta_j = \frac{(2j+1)\pi}{2}, \quad j = 0, \ldots, n,$$

pa su nultočke T_{n+1} jednake (silazno poredane)

$$x_j = \cos\left(\frac{(2j+1)\pi}{2(n+1)}\right), \quad j = 0, \ldots, n.$$

S druge strane, lokalni ekstremi se postižu kad je

$$(n+1)\vartheta'_k = k\pi, \quad k = 0, \ldots, n+1,$$

pa su ekstremi T_{n+1} jednaki

$$x'_k = \cos\left(\frac{k\pi}{(n+1)}\right), \quad k = 0, \ldots, n+1.$$

Drugim riječima, vrijedi

$$T_{n+1}(x_k) = (-1)^k, \quad k = 0, \ldots, n+1.$$

Primijetite da tih ekstrema ima točno $n+2$ i da alterniraju po znaku.

7.3.2. Minimaks svojstvo Čebiševljevih polinoma

Čebiševljevi polinomi T_n imaju važno svojstvo minimizacije "uniformnog otklona polinoma od nule". Vrijedi sljedeći teorem.

Teorem 7.3.1. *Za fiksni prirodni broj n, promatrajmo minimizacijski problem*

$$\tau_n = \inf_{\deg(P)\leq n-1} (\max_{-1\leq x\leq 1} |x^n + P(x)|),$$

gdje je P polinom. Minimum τ_n se dostiže samo za

$$x^n + P(x) = \frac{1}{2^{n-1}} T_n(x).$$

Pripadna pogreška je

$$\tau_n = \frac{1}{2^{n-1}}.$$

Dokaz:

Iz tročlane rekurzije, nije teško induktivno dokazati da je vodeći koeficijent T_n jednak

$$T_n(x) = 2^{n-1}x^n + \text{članovi nižeg stupnja}, \quad n \geq 1.$$

Zbog toga vrijedi da je

$$\frac{1}{2^{n-1}} T_n(x) = x^n + \text{članovi nižeg stupnja}.$$

Znamo da su točke

$$x'_k = \cos\left(\frac{k\pi}{n}\right), \quad k = 0, \ldots, n,$$

lokalni ekstremi od T_n, u kojima je

$$T_n(x'_k) = (-1)^k, \quad k = 0, \ldots, n$$

i

$$-1 = x'_n < x'_{n-1} < \cdots < x'_1 < x'_0 = 1.$$

Polinom

$$\frac{1}{2^{n-1}} T_n$$

ima vodeći koeficijent 1 i vrijedi

$$\max_{-1\leq x\leq 1} \left|\frac{1}{2^{n-1}} T_n\right| = \frac{1}{2^{n-1}}.$$

Zbog toga je

$$\tau_n \leq \frac{1}{2^{n-1}}.$$

Pokažimo da je τ_n baš jednak desnoj strani. Pretpostavimo suprotno, tj. da je

$$\tau_n < \frac{1}{2^{n-1}}.$$

Pokazat ćemo da to vodi na kontradikciju. Definicija τ_n i prethodna pretpostavka pokazuju da postoji polinom M takav da je

$$M(x) = x^n + P(x), \quad \deg(P) \leq n-1,$$

gdje je

$$\tau_n \leq \max_{-1\leq x\leq 1} |M(x)| < \frac{1}{2^{n-1}}. \tag{7.3.1}$$

Definiramo

$$R(x) = \frac{1}{2^{n-1}} T_n(x) - M(x).$$

Tvrdimo da će se vodeći koeficijenti funkcija s desne strane skratiti, pa je $\deg(R) \leq n-1$. Ispitajmo vrijednosti funkcije R u lokalnim ekstremima funkcije T_n. Iz (7.3.1) redom, izlazi

$$R(x_0') = R(1) = \frac{1}{2^{n-1}} - M(1) > 0$$

$$R(x_1') = -\frac{1}{2^{n-1}} - M(x_1) < 0, \ldots$$

Tj. za polinom R vrijedi

$$\operatorname{sign}(R(x_k')) = (-1)^k.$$

Budući da ima bar $n+1$ različiti predznak, to mora postojati bar n nultočaka, što je moguće damo ako je $R = 0$. Odatle odmah izlazi da je

$$M(x) = \frac{1}{2^{n-1}} T_n(x).$$

Sad bi još trebalo pokazati da je to jedini polinom s takvim svojstvom. Taj dio dokaza vrlo je sličan ovom što je već dokazano. ■

7.3.3. Interpolacija u Čebiševljevim točkama

Vratimo se sad polaznom problemu optimalnog izbora čvorova interpolacije. Želimo izaberati točke interpolacije $x_j \in [-1, 1]$ tako da minimiziraju

$$\max_{-1\leq x\leq 1} |(x - x_0) \cdots (x - x_n)|.$$

Polinom u prethodnoj relaciji je stupnja $n+1$ i ima vodeći koeficijent 1. Po Teoremu 7.3.1., minimum ćemo dobiti ako stavimo

$$(x - x_0) \cdots (x - x_n) = \frac{1}{2^n} T_{n+1}(x),$$

a minimalna će vrijednost biti $1/2^n$. Odatle odmah čitamo da su čvorovi $x_0, \ldots, x_n$ nultočke polinoma T_{n+1}, a njih smo već izračunali da su jednake

$$x_j = \cos\Big(\frac{(2j+1)\pi}{2n+2}\Big), \quad j = 0, \ldots, n.$$

Ako želimo uzlazni poredak čvorova, onda je

$$x_j = \cos\Big(\frac{(2(n-j)+1)\pi}{2n+2}\Big), \quad j = 0, \ldots, n.$$

7.4. Interpolacija po dijelovima polinomima

U prošlom smo poglavlju pokazali da polinomna interpolacija visokog stupnja može imati vrlo loša svojstva, pa se u praksi **ne smije** koristiti. Umjesto toga, koristi se po dijelovima polinomna interpolacija, tj. na svakom podintervalu je

$$\varphi\,\Big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, 2, \ldots, n,$$

a p_k su polinomi niskog (fiksnog) stupnja. Za razliku od polinomne interpolacije funkcijskih vrijednosti, gdje je bilo dovoljno da su čvorovi interpolacije međusobno različiti, ovdje pretpostavljamo da su rubovi podintervala interpolacije uzlazno numerirani, tj. da vrijedi $a = x_0 < x_1 < \cdots < x_n = b$. To još ne osigurava da je φ funkcija (moguća dvoznačnost u dodirnim točkama podintervala), ali o tome ćemo voditi računa kod zadavanja uvjeta interpolacije.

Preciznije, pretpostavimo da na svakom podintervalu $[x_{k-1}, x_k]$ koristimo polinom stupnja m, tj. da je

$$\varphi\,\Big|_{[x_{k-1},x_k]} = p_k, \quad k = 1, \ldots, n.$$

Svaki polinom p_k (stupnja m) je određen s $(m+1)$-im koeficijentom, odnosno, ukupno moramo odrediti koeficijente polinoma za n podintervala, tj. ukupno

$$(m+1) \cdot n \tag{7.4.1}$$

koeficijenata.

Interpolacioni uvjeti su

$$\varphi(x_k) = f_k, \quad k = 0, \ldots, n,$$

što za svaki polinom daje po 2 uvjeta

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1} \\ p_k(x_k) &= f_k, \end{aligned} \quad k = 1, \ldots, n, \tag{7.4.2}$$

odnosno, ukupno imamo $2n$ uvjeta interpolacije. Uočimo da smo postavljenjem prethodnih uvjeta interpolacije osigurali neprekidnost funkcije φ, jer je

$$p_{k-1}(x_{k-1}) = p_k(x_{k-1}), \quad k = 2, \ldots, n.$$

Primijetimo da uvjeta interpolacije ima $2n$, a moramo naći $(m+1) \cdot n$ koeficijenata. Bez dodatnih uvjeta to je moguće napraviti samo za $m = 1$, tj. za po dijelovima linearnu interpolaciju.

Za $m > 1$ moraju se dodati uvjeti na glatkoću interpolacione funkcije φ u čvorovima interpolacije.

7.4.1. Po dijelovima linearna interpolacija

Osnovna ideja po dijelovima linearne interpolacije je umjesto jednog polinoma visokog stupnja koristiti više polinoma, ali stupnja 1.

Na svakom podintervalu p_k je jedinstveno određen. Obično ga zapisujemo relativno obzirom na početnu točku intervala (stabilnost) u obliku

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) \quad \text{za} \quad x \in [x_{k-1}, x_k], \quad k = 1, \ldots, n.$$

Taj interpolacioni polinom možemo zapisati u Newtonovoj formi

$$p_k(x) = f[x_{k-1}] + f[x_{k-1}, x_k] \cdot (x - x_{k-1}),$$

pa se odmah vidi da vrijedi

$$\begin{aligned} c_{0,k} &= f[x_{k-1}] = f_{k-1} \\ c_{1,k} &= f[x_{k-1}, x_k] = \frac{f_k - f_{k-1}}{x_k - x_{k-1}}, \end{aligned} \quad k = 1, \ldots, n.$$

Ako želimo aproksimirati vrijednost funkcije f u točki $x \in [a, b]$, prvo treba pronaći između kojih se čvorova točka x nalazi, tj za koji k vrijedi $x_{k-1} \leq x \leq x_k$. Tek tada možemo računati koeficijente pripadnog linearnog polinoma.

Za traženje tog intervala koristimo algoritam binarnog pretraživanja.

Algoritam 7.4.1. (Binarno pretraživanje)

```
low := 0;
high := n;
while (high − low) > 1 do
  begin
  mid := (low + high) div 2;
  if x < x_mid then
    high := mid
  else
    low := mid
  end;
```

Trajanje ovog algoritma je proporcionalno s $\log_2(n)$.

Ako je funkcija f klase $C^2[a,b]$ (na intervalu na kojem aproksimiramo), onda je pogreška takve interpolacije zapravo maksimalna pogreška od n linearnih interpolacija. Na podintervalu $[x_{k-1}, x_k]$ ocjena greške linearne interpolacije je

$$|f(x) - p_k(x)| \leq \frac{M_2^k}{2!} |\omega(x)|,$$

pri čemu je

$$\omega(x) = (x - x_{k-1})(x - x_k), \quad M_2^k = \max_{x \in [x_{k-1}, x_k]} |f''(x)|.$$

Ocijenimo $\omega(x)$ na $[x_{k-1}, x_k]$, tj. nađimo po apsolutnoj vrijednosti njen maksimum. Funkcija ω može imati maksimum samo na otvorenom intervalu (x_{k-1}, x_k), a nikako na rubu (čvorovi interpolacije — greška je 0). Nađimo lokalni ekstrem funkcije

$$\omega(x) = (x - x_{k-1})(x - x_k).$$

Deriviranjem izalazi

$$\omega'(x) = 2x - (x_{k-1} + x_k),$$

pa je kandidat za lokalni ekstrem točka

$$x_e = \frac{(x_{k-1} + x_k)}{2}.$$

Tvrdimo da je to baš i lokalni minimum, jer se radi o paraboli (a ona nema infleksiju). Nevjerni Tome mogu to provjeriti, recimo, deriviranjem

$$\omega''(x_e) = 2 > 0.$$

Vrijednost funkcije ω u lokalnom ekstremu je

$$\omega(x_e) = (x_e - x_{k-1})(x_e - x_k) = \frac{x_k - x_{k-1}}{2} \cdot \frac{x_{k-1} - x_k}{2} = -\frac{(x_k - x_{k-1})^2}{4}.$$

Osim toga, za bilo koji $x \in (x_{k-1}, x_k)$ vrijedi $\omega(x) < 0$. Odatle, prijelazom na apsolutnu vrijednost, odmah slijedi da je x_e točka lokalnog maksimuma za $|\omega|$ i

$$|\omega(x)| \leq |\omega(x_e)| \leq \frac{(x_k - x_{k-1})^2}{4}, \quad \forall x \in [x_{k-1}, x_k].$$

Definiramo li maksimalni razmak čvorova

$$h = \max_{1 \leq k \leq n} \{h_k = x_k - x_{k-1}\},$$

onda, na čitavom $[a, b]$, možemo pisati

$$|f(x) - \varphi(x)| \leq \frac{M_2}{2!} \frac{h^2}{4} = \frac{1}{8} M_2 \cdot h^2.$$

Drugim riječima ako ravnomjerno povećavamo broj čvorova, tako da $h \to 0$, onda i maksimalna greška teži u 0.

Na primjer, za ekvidistantne mreže, tj. za mreže za koje vrijedi

$$x_k = a + kh, \quad h = \frac{b - a}{n}$$

je pogreška reda veličine h^2, odnosno n^{-2} i potrebno je dosta podintervala da se dobije sasvim umjerena točnost aproksimacije. Na primjer, za $h = 0.01$, tj. za $n = 100$, greška aproksimacije je reda veličine 10^{-4}.

Druga je mana da aproksimaciona funkcija φ nije dovoljno glatka, tj. ona je samo neprekidna. Zbog ta dva razloga (dosta točaka za umjerenu točnost i pomanjkanje glatkoće), obično se na svakom podintervalu koriste polinomi viših stupnjeva.

Ako stavimo $m = 2$, tj. na svakom podintervalu postavimo kvadratni polinom, moramo naći $3n$ koeficijenata, a imamo $2n$ uvjeta interpolacije. Ako zahtijevamo da aproksimaciona funkcija φ ima u unutarnjim čvorovima interpolacije $x_1, \ldots, x_{n-1}$ neprekidnu derivaciju, onda smo dodali još $n-1$ uvjet. A treba nam još jedan! Ako i njega postavimo (a to ne možemo lijepo, simetrično), onda bismo mogli naći i takvu aproksimaciju. Ona se uobičajeno ne koristi, jer kontrolu derivacije možemo napraviti samo na jednom rubu (to bi odgovaralo inicijalnim problemima). Preciznije rečeno, po dijelovima kvadratna interpolacija nema pravu fizikalnu podlogu, pa se vrlo rijetko koristi (katkad kod računarske grafike). Za razliku od po dijelovima parabolne interpolacije, po dijelovima kubna interpolacija ima vrlo važnu fizikalnu podlogu i vjerojatno je jedna od najčešće korištenih metoda interpolacije uopće.

7.4.2. Po dijelovima kubna interpolacija

Kod po dijelovima kubne interpolacije, restrikcija aproksimacione funkcije φ na svaki interval je kubični polinom kojeg obično zapisujemo relativno obzirom na

početnu točku intervala u obliku

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) + c_{2,k}(x - x_{k-1})^2 + c_{3,k}(x - x_{k-1})^3 \quad \text{za} \quad x \in [x_{k-1}, x_k], \quad k = 1, \ldots, n. \tag{7.4.3}$$

Budući da ukupno imamo n kubnih polinoma, od kojih svakome treba odrediti 4 koeficijenta, ukupno moramo odrediti $4n$ koeficijenata. Uvjeta interpolacije je $2n$, jer svaki kubni polinom p_k mora interpolirati rubove svog podintervala $[x_{k-1}, x_k]$, tj. mora vrijediti

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1} \\ p_k(x_k) &= f_k, \end{aligned} \quad k = 1, \ldots, n.$$

Ovi uvjeti automatski osiguravaju neprekidnost funkcije φ. Obično želimo da interpolaciona funkcija bude glađa — barem klase $C^1[a, b]$, tj. da je i derivacija funkcije φ neprekidna i u čvorovima. Dodavanjem tih uvjeta za svaki kubni polinom, dobivamo još $2n$ uvjeta

$$\begin{aligned} p_k'(x_{k-1}) &= s_{k-1} \\ p_k'(x_k) &= s_k, \end{aligned} \quad k = 1, \ldots, n,$$

pri čemu su s_k neki brojevi. Njihova uloga može biti višeznačna, pa ćemo je detaljno opisati kasnije. Zasad, možemo zamišljati da su brojevi s_k neke aproksimacije derivacije u čvorovima.

Primijetite da je takvim izborom dodatnih uvjeta osigurana neprekidnost prve derivacije, jer je

$$p_{k-1}'(x_{k-1}) = p_k'(x_{k-1}) = s_{k-1}, \quad k = 2, \ldots, n.$$

Ako pretpostavimo da su s_k nekako zadani brojevi, nađimo koeficijente interpolacionog polinoma p_k.

Ponovno, najzgodnije je koristiti Newtonov oblik interpolacionog polinoma, ali sada s tzv. dvostrukim čvorovima, jer su u x_{k-1} i x_k dani i funkcijska vrijednost i derivacija.

Što, zapravo, znači dvostruki čvor? Pretpostavimo li da se u podijeljenoj razlici dva čvora približavaju jedan drugom, onda je podijeljena razlika na limesu

$$\lim_{h_k \to 0} f[x_k, x_k + h_k] = \lim_{h_k \to 0} \frac{f(x_k + h_k) - f(x_k)}{h_k} = f'(x_k),$$

naravno, pod uvjetom da f ima derivaciju u točki x_k. Drugim riječima, vrijedi

$$f[x_k, x_k] = f'(x_k).$$

U našem slučaju, ako u točki x_k derivaciju $f'(x_k)$ zadajemo ili aproksimiramo s s_k, onda je

$$f[x_k, x_k] = s_k.$$

Sada možemo napisati tablicu podijeljenih razlika za kubni interpolacioni polinom koji ima dva dvostruka čvora x_{k-1} i x_k. To je najjednostavnije predočiti si kao kubni interpolacioni polinom koji prolazi kroz četiri točke: x_{k-1}, točkom koja je "jako blizu" x_{k-1}, točkom koja je "jako blizu" x_k i točkom x_k. Kad se te dvije točke koje su "jako blizu" stope sa svojim parom, dobivamo dva dvostruka čvora, pa tablica podijeljenih razlika izgleda ovako:

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$	$f[x_k, x_{k+1}, x_{k+2}, x_{k+3}]$
x_{k-1}	f_{k-1}			
		s_{k-1}		
x_{k-1}	f_{k-1}		$\dfrac{f[x_{k-1}, x_k] - s_{k-1}}{h_k}$	
		$f[x_{k-1}, x_k]$		$\dfrac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}$
x_k	f_k		$\dfrac{s_k - f[x_{k-1}, x_k]}{h_k}$	
		s_k		
x_k	f_k			

Forma Newtonovog interpolacionog polinoma ostat će po obliku jednaka kao u slučaju da su sve četiri točke različite, pa imamo

$$\begin{aligned} p_k(x) = f[x_{k-1}] &+ f[x_{k-1}, x_{k-1}] \cdot (x - x_{k-1}) \\ &+ f[x_{k-1}, x_{k-1}, x_k] \cdot (x - x_{k-1})^2 \\ &+ f[x_{k-1}, x_{k-1}, x_k, x_k] \cdot (x - x_{k-1})^2 (x - x_k) \end{aligned} \tag{7.4.4}$$

uz uvažavanje da je

$$\begin{aligned} f[x_{k-1}, x_{k-1}] &= s_{k-1} \\ f[x_{k-1}, x_{k-1}, x_k] &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} \\ f[x_{k-1}, x_{k-1}, x_k, x_k] &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}. \end{aligned}$$

Uvrštavanjem x_{k-1} i x_k u prethodnu formulu, odmah možemo provjeriti da je

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1}, & p_k'(x_{k-1}) &= s_{k-1}, \\ p_k(x_k) &= f_k, & p_k'(x_k) &= s_k. \end{aligned}$$

Drugim riječima, našli smo traženi p_k. Usporedimo li forme (7.4.3) i (7.4.4), dobit ćemo koeficijente $c_{i,k}$. Jednadžbu (7.4.4) možemo malo drugačije zapisati, tako da polinom bude napisan po potencijama od $(x - x_{k-1})$. Posljednji član tog polinoma možemo napisati kao

$$\begin{aligned}(x - x_{k-1})^2(x - x_k) &= (x - x_{k-1})^2(x - x_{k-1} + x_{k-1} - x_k)\\ &= (x - x_{k-1})^2(x - x_{k-1} - h_k)\\ &= (x - x_{k-1})^3 - h_k(x - x_{k-1})^2.\end{aligned}$$

Sada (7.4.4) glasi

$$\begin{aligned}p_k(x) = f[x_{k-1}] &+ f[x_{k-1}, x_{k-1}] \cdot (x - x_{k-1})\\ &+ (f[x_{k-1}, x_{k-1}, x_k] - h_k\, f[x_{k-1}, x_{k-1}, x_k, x_k]) \cdot (x - x_{k-1})^2\\ &+ f[x_{k-1}, x_{k-1}, x_k, x_k] \cdot (x - x_{k-1})^3.\end{aligned}$$

Uspoređivanjem koeficijenata uz odgovarajuće potencije prethodne relacije i relacije (7.4.3), za sve $k = 1, \ldots, n$, dobivamo

$$\begin{aligned}c_{0,k} &= p_k(x_{k-1}) = f_{k-1},\\ c_{1,k} &= p_k'(x_{k-1}) = s_{k-1},\\ c_{2,k} &= \frac{p_k''(x_{k-1})}{2} = f[x_{k-1}, x_{k-1}, x_k] - h_k\, f[x_{k-1}, x_{k-1}, x_k, x_k],\\ c_{3,k} &= \frac{p_k'''(x_{k-1})}{6} = f[x_{k-1}, x_{k-1}, x_k, x_k].\end{aligned}$$

Promotrimo li bolje posljednje dvije relacije, otkrivamo da se isplati prvo izračunati koeficijent $c_{3,k}$, a zatim ga upotrijebiti za računanje $c_{2,k}$. Dobivamo

$$\begin{aligned}c_{3,k} &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2},\\ c_{2,k} &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} - h_k c_{3,k}.\end{aligned}$$

Drugim riječima, ako znamo s_k, onda nije problem naći koeficijente po dijelovima kubne interpolacije. Ostaje nam samo pokazati kako bismo mogli birati s_k-ove. Ponovno, postoje dva bitno različita načina.

7.4.3. Po dijelovima kubna Hermiteova interpolacija

Vrijednosti s_k možemo izabrati tako da su one baš jednake derivaciji zadane funkcije u odgovarajućoj točki, tj. da vrijedi

$$s_k = f'(x_k).$$

U tom slučaju je kubni polinom određen **lokalno**, tj. ne ovisi o drugim kubnim polinomima, jer su mu na rubovima zadane funkcijske vrijednosti i vrijednosti derivacija. Takva se interpolacija zove po dijelovima kubna Hermiteova interpolacija.

Nađimo grešku takve interpolacije, uz pretpostavku da je funkcija $f \in C^4[a,b]$. Prvo, pronađimo grešku na intervalu $[x_{k-1}, x_k]$. Interpolacioni polinom s dvostrukim čvorovima na rubu ponaša se kao polinom koji ima četiri različita čvora, takva da se parovi čvorova u rubu "stope". Zbog toga, možemo promatrati i grešku interpolacionog polinoma reda 3 koji interpolira funkciju f u točkama x_{k-1}, x_k i još dvijema točkama koje su blizu x_{k-1} i x_k. Grešku takvog interpolacionog polinoma možemo ocijeniti s

$$|f(x) - p_k(x)| \leq \frac{M_4^k}{4!} |\omega(x)|,$$

pri čemu je

$$\omega(x) = (x - x_{k-1})^2 (x - x_k)^2, \quad M_4^k = \max_{x \in [x_{k-1}, x_k]} |f^{(4)}(x)|.$$

Ostaje samo još pronaći u kojoj je točki intervala $[x_{k-1}, x_k]$ maksimum funkcije $|\omega|$.

Dovoljno je naći sve lokalne ekstreme funkcije ω i u njima provjeriti vrijednost. Derivirajmo

$$\begin{aligned} \omega'(x) &= 2(x - x_{k-1})(x - x_k)^2 + 2(x - x_{k-1})^2 (x - x_k) \\ &= 2(x - x_{k-1})(x - x_k)(2x - x_{k-1} - x_k). \end{aligned}$$

Budući da maksimum greške ne može biti u rubovima intervala, jer su tamo točke interpolacije (tj. minimumi greške i $|\omega|$), onda je jedino još moguće da se ekstrem dostiže u nultočki od ω' jednakoj

$$x_e = \frac{(x_{k-1} + x_k)}{2}.$$

Lako se provjerava da je to lokalni maksimum. Vrijednost u x_e je kvadrat vrijednosti greške za po dijelovima linearnu interpolaciju na istoj mreži čvorova

$$\omega(x_e) = (x_e - x_{k-1})^2 (x_e - x_k)^2 = \frac{(x_k - x_{k-1})^4}{16}.$$

Odatle, prijelazom na apsolutnu vrijednost, odmah slijedi da je x_e točka lokalnog maksimuma za $|\omega|$ i

$$|\omega(x)| \leq |\omega(x_e)| \leq \frac{(x_k - x_{k-1})^4}{16}, \quad \forall x \in [x_{k-1}, x_k].$$

Definiramo li, ponovno, maksimalni razmak čvorova

$$h = \max_{1 \leq k \leq n} \{h_k = x_k - x_{k-1}\},$$

onda, na čitavom $[a, b]$, možemo pisati

$$|f(x) - \varphi(x)| \leq \frac{M_4}{4!} \frac{h^4}{16} = \frac{1}{384} M_4 \cdot h^4.$$

Drugim riječima, ako ravnomjerno povećavamo broj čvorova, tako da $h \to 0$, onda i maksimalna greška teži u 0.

Ipak, u cijelom ovom pristupu ima jedan problem. Vrlo često derivacije funkcije u točkama interpolacije nisu poznate. Zamislite, recimo, točke dobivene mjerenjem. No, tada možemo aproksimirati prave vrijednosti derivacije korištenjem vrijednosti funkcije u susjednim točkama. Ostaje još samo pokazati kako.

7.4.4. Numeričko deriviranje

Problem koji trebamo riješiti je kako aproksimirati derivaciju diferencijabilne funkcije f u nekoj točki, recimo x_0 i susjednim točkama $x_1, \ldots, x_n$, korištenjem samo vrijednosti funkcije f u zadanim točkama.

Taj problem možemo riješiti korištenjem interpolacionog polinoma. Tada, uz pretpostavku da je f klase $C^{n+1}[a, b]$, funkciju f možemo napisati (vidjeti relaciju (7.2.5)) kao

$$f(x) = p_n(x) + e_n(x),$$

gdje je $p_n(x)$ interpolacioni polinom napisan, recimo, u Newotnovoj formi

$$p_n(x) = f[x_0] + (x - x_0) f[x_0, x_1] + \cdots + (x - x_0) \cdots (x - x_{n-1}) f[x_0, x_1, \ldots, x_n],$$

a $e_n(x)$ greška interpolacionog polinoma

$$e_n(x) = \frac{\omega(x)}{(n+1)!} f^{(n+1)}(\xi).$$

Deriviranjem interpolacionog polinoma, a zatim uvrštavanjem $x = x_0$ dobivamo

$$\begin{aligned} p_n'(x_0) = f[x_0, x_1] &+ (x_0 - x_1) f[x_0, x_1, x_2] \\ &+ \cdots + (x_0 - x_1) \cdots (x_0 - x_{n-1}) f[x_0, x_1, \ldots, x_n]. \end{aligned}$$

Ako pretpostavimo da f ima još jednu neprekidnu derivaciju, tj. da je f klase $C^{n+2}[a, b]$, onda dobivamo i da je

$$e_n'(x_0) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x_0 - x_1) \cdots (x_0 - x_n).$$

Dakle, $p_n'(x_0)$ je aproksimacija derivacije funkcije f u točki x_0 i vrijedi

$$f'(x_0) = p_n'(x_0) + e_n'(x_0).$$

Ako označimo s

$$H = \max_k |x_0 - x_k|,$$

onda je, za $H \to 0$, greška $e_n'(x_0)$ reda veličine

$$e_n'(x_0) \leq O(H^n).$$

To nam pokazuje da aproksimaciona formula za derivaciju može biti proizvoljno visokog reda n, ali takve formule s velikim n imaju ograničenu praktičnu vrijednost.

Pokažimo kako se ta formula ponaša za niske n. Za $n = 1$ imamo

$$p_1'(x_0) = f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{f_1 - f_0}{h},$$

pri čemu smo napravili grešku

$$e_1'(x_0) = \frac{f^{(2)}(\xi)}{2!}\,(x_0 - x_1) = -\frac{f^{(2)}(\xi)}{2}\,h,$$

uz pretpostavku da je $f \in C^3[x_0, x_1]$. Greška je reda veličine $O(h)$ za $h \to 0$.

Za $n = 2$, uzmimo točke x_1 i x_2 koje se nalaze simetrično oko x_0 (to je poseban slučaj!), tj.

$$x_1 = x_0 + h, \quad x_2 = x_0 - h.$$

Puno sugestivnija notacija točaka u tom slučaju je da s x_{-1} označimo x_2, jer onda točke pišemo u prirodnom redosljedu: x_{-1}, x_0, x_1. U tom slučaju je

$$p_2'(x_0) = f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_{-1}].$$

Izračunajmo potrebne podijeljene razlike.

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$
x_{-1}	f_{-1}		
		$\dfrac{f_0 - f_{-1}}{h}$	
x_0	f_0		$\dfrac{f_1 - 2f_0 + f_{-1}}{2h^2}$
		$\dfrac{f_1 - f_0}{h}$	
x_1	f_1		

Uvrštavanjem dobivamo

$$p_2'(x_0) = \frac{f_1 - f_0}{h} - h\,\frac{f_1 - 2f_0 + f_{-1}}{2h^2} = \frac{f_1 - f_{-1}}{2h}.$$

Ovu posljednju formulu često zovemo simetrična (centralna) razlika, jer su točke x_1 i x_{-1} simetrične obzirom na x_0. Takva aproksimacija derivacije ima bolju ocjenu greške nego obične podijeljene razlike, tj. vrijedi

$$e_2'(x_0) = \frac{f^{(3)}(\xi)}{6}\,(x_0 - x_1)\,(x_0 - x_{-1}) = -h^2\,\frac{f^{(3)}(\xi)}{6}.$$

Pokažimo što bi se zbivalo kad točke x_1 i x_{-1} (odnosno x_2) ne bismo simetrično rasporedili oko x_0. Na primjer, uzmimo

$$x_1 = x_0 + h, \quad x_2 = x_0 + 2h.$$

Iako su i u ovom slučaju točke ekvidistantne, deriviramo u najljevijoj, a ne u srednjoj točki. Pripadna tablica podijeljenih razlika je

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, x_{k+1}, x_{k+2}]$
x_0	f_0		
		$\dfrac{f_1 - f_0}{h}$	
x_1	f_1		$\dfrac{f_2 - 2f_1 + f_0}{2h^2}$
		$\dfrac{f_2 - f_1}{h}$	
x_2	f_2		

Konačno, aproksimacija derivacije u x_0 je

$$p_2'(x_0) = f[x_0, x_1] + (x_0 - x_1) f[x_0, x_1, x_{-1}] = \frac{f_1 - f_0}{h} - h\frac{f_2 - 2f_1 + f_0}{2h^2}$$
$$= \frac{-f_2 + 4f_1 - 3f_0}{2h},$$

dok je greška jednaka

$$e_2'(x_0) = \frac{f^{(3)}(\xi)}{6}\,(x_0 - x_1)\,(x_0 - x_2) = h^2\,\frac{f^{(3)}(\xi)}{3},$$

tj. greška je istog reda veličine $O(h^2)$, međutim konstanta je dvostruko veća nego u prethodnom (simetričnom) slučaju.

Primijetite da formula za derivaciju postaje sve točnija što su bliže točke iz kojih se derivacija aproksimira, tj. što je h manji, naravno, uz pretpostavku da je funkcija f dovoljno glatka. Međutim, to vrijedi samo u teoriji. U praksi, mnogi podaci su mjereni, pa nose neku pogrešku, u najmanju ruku zbog grešaka zaokruživanja.

Kao što ste vidjeli u prethodnim primjerima, osnovu numeričkog deriviranja čine podijeljene razlike, pa ako su točke bliske, dolazi do kraćenja. To nije slučajno.

Do kraćenja **mora** doći, zbog neprekidnosti funkcije f. Problem je to izrazitiji, što su točke bliže, tj. što je h manji. Dakle, za numeričko deriviranje imamo dva oprečna zahtjeva na veličinu h. Manji h daje bolju ocjenu greške, ali veću grešku zaokruživanja.

Ilustrirajmo to analizom simetrične razlike,

$$f'(x_0) = \frac{f_1 - f_{-1}}{2h} + e_2'(x_0), \quad e_2'(x_0) = -h^2\,\frac{f^{(3)}(\xi)}{6}.$$

Pretpostavimo da smo, umjesto vrijednosti f_{-1} i f_1, uzeli malo perturbirane vrijednosti

$$\widehat{f}_1 = f_1 + \varepsilon_1, \quad \widehat{f}_{-1} = f_{-1} + \varepsilon_{-1}, \quad |\varepsilon_1|, |\varepsilon_{-1}| \leq \varepsilon.$$

Ako odatle izrazimo f_1 i f_{-1} i uvrstimo ih u formulu za derivaciju, dobivamo

$$f'(x_0) = \frac{\widehat{f}_1 - \widehat{f}_{-1}}{2h} - \frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2'(x_0).$$

Prvi član s desne strane je ono što smo mi zaista izračunali kao aproksimaciju derivacije, a ostalo je greška. Da bismo analizu napravili jednostavnijom, pretpostavimo da je h prikaziv u računalu i da je greška pri računanju kvocijenta u podijeljenoj razlici zanemariva. U tom je slučaju napravljena ukupna greška

$$err_2 = f'(x_0) - \frac{\widehat{f}_1 - \widehat{f}_{-1}}{2h} = -\frac{\varepsilon_1 - \varepsilon_{-1}}{2h} + e_2'(x_0).$$

Ogradimo err_2 po apsolutnoj vrijednosti. Greška u prvom članu je najveća ako su ε_1 i ε_{-1} suprotnih predznaka, maksimalne apsolutne vrijednosti ε. Za drugi član koristimo ocjenu za $e_2'(x_0)$, pa zajedno dobivamo

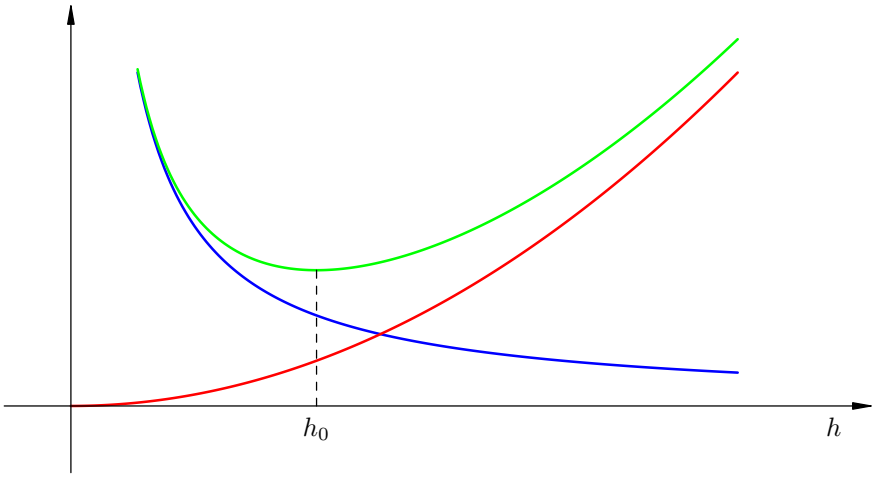
$$|err_2| \leq \frac{\varepsilon}{h} + \frac{M_3}{6}h^2, \quad M_3 = \max_{x\in[x_{-1},x_1]} |f^{(3)}(x)|.$$

Lako se vidi da je ocjena na desnoj strani najbolja moguća, tj. da se može dostići. Označimo tu ocjenu s $e(h)$

$$e(h) := \frac{\varepsilon}{h} + \frac{M_3}{6}h^2.$$

Ponašanje ove ocjene i njezina dva člana u ovisnosti od h možemo prikazati sljedećim grafom. Plavom bojom označen je prvi član ε/h oblika hiperbole, koji dolazi od greške u podacima, a crvenom bojom drugi član oblika parabole, koji predstavlja maksimalnu grešku odbacivanja kod aproksimacije derivacije podijeljenom razlikom.

Zelena boja označava njihov zbroj $e(h)$.



Odmah vidimo da $e(h)$ ima minimum po h. Taj minimum se lako računa, jer iz

$$e'(h) = -\frac{\varepsilon}{h^2} + \frac{M_3}{3}h = 0$$

izlazi da se lokalni, a onda (zbog $e''(h) > 0$ za $h > 0$) i globalni minimum postiže za

$$h_0 = \left(\frac{3\varepsilon}{M_3}\right)^{1/3}.$$

Najmanja vrijednost funkcije je

$$e(h_0) = \frac{3}{2}\left(\frac{M_3}{3}\right)^{1/3}\varepsilon^{2/3}.$$

To pokazuje da čak i u najboljem slučaju, kad je ukupna greška najmanja, dobivamo da je ona reda veličine $O(\varepsilon^{2/3})$, a ne $O(\varepsilon)$, kao što bismo željeli. To predstavlja značajni gubitak točnosti. Posebno, daljnje smanjivanje koraka h samo povećava grešku!

Isti problem se javlja, i to u još ozbiljnijem obliku, u formulama višeg reda za aproksimaciju derivacija. Kako tada izgleda prethodni graf? Što se događa kad aproksimiramo više derivacije?

7.4.5. Po dijelovima kubna kvazihermiteova interpolacija

Sad se možemo vratiti problemu kako napraviti po dijelovima kubnu Hermiteovu interpolaciju, ako nemamo zadane derivacije. U tom slučaju derivacije možemo aproksimirati na različite načine, a samu interpolaciju ćemo zvati kvazihermiteova po dijelovima kubna interpolacija.

Primijetite da u slučaju aproksimacije derivacije, greška po dijelovima kubne interpolacije ovisi o tome koliko je "dobra" aproksimacija derivacije.

Najjednostavnije je uzeti podijeljene razlike kao aproksimacije derivacija u čvorovima. One mogu biti unaprijed (do na posljednju) ili unazad (do na prvu). Ako koristimo podijeljene razlike unaprijed, onda je

$$s_k = \begin{cases} \dfrac{f_{k+1} - f_k}{x_{k+1} - x_k}, & \text{za } k = 0, \ldots, n-1, \\ \dfrac{f_n - f_{n-1}}{x_n - x_{n-1}}, & \text{za } k = n. \end{cases}$$

a ako koristimo podijeljene razlike unazad, onda je

$$s_k = \begin{cases} \dfrac{f_1 - f_0}{x_1 - x_0}, & \text{za } k = 0, \\ \dfrac{f_k - f_{k-1}}{x_k - x_{k-1}}, & \text{za } k = 1, \ldots, n. \end{cases}$$

Međutim, prema prethodnom odjeljku, greška koju smo napravili takvom aproksimacijom derivacije je reda veličine $O(h)$ u derivaciji, odnosno $O(h^2)$ u funkcijskoj vrijednosti, što je dosta loše.

Prethodnu aproksimaciju možemo ponešto popraviti ako su točke x_k ekvidistantne, a koristimo simetričnu razliku (osim na lijevom i desnom rubu gdje to nije moguće). Uz oznaku $h = x_k - x_{k-1}$, u tom slučaju možemo staviti

$$s_k = \begin{cases} \dfrac{f_1 - f_0}{h}, & \text{za } k = 0, \\ \dfrac{f_{k+1} - f_{k-1}}{2h}, & \text{za } k = 1, \ldots, n-1, \\ \dfrac{f_n - f_{n-1}}{h}, & \text{za } k = n. \end{cases}$$

U ovom će se slučaju greška obzirom na obične podijeljene razlike popraviti tamo gdje se koristi simetrična razlika. Nažalost, najveće greške ostat će u prvom i posljednjem podintervalu, gdje nije moguće koristiti simetričnu razliku.

Kao što smo vidjeli, postoje i bolje aproksimacije derivacija, a pripadni kvazihermiteovi kubni polinomi obično dobivaju ime po načinu aproksimacije derivacija.

Ako derivaciju u točki x_k aproksimiramo tako da povučemo kvadratni interpolacioni polinom kroz x_{k-1}, x_k i x_{k+1}, a zatim ga deriviramo, pripadna kvazihermiteova interpolacija zove se Besselova po dijelovima kubična interpolacija. Naravno, u prvoj i posljednjoj točki ne možemo postupiti na jednak način (jer nema lijeve, odnosno desne točke). Zbog toga derivaciju u x_0 aproksimiramo tako da povučemo

kvadratni interpolacioni polinom kroz x_0, x_1 i x_2, i njega deriviramo u x_0. Slično, derivaciju u x_n aproksimiramo tako da povučemo kvadratni interpolacioni polinom kroz x_{n-2}, x_{n-1} i x_n, i njega deriviramo u x_n.

U unutrašnjim čvorovima x_k, za $k = 1, \ldots, n-1$, dobivamo

$$p_{2,k}(x) = f_{k-1} + f[x_{k-1}, x_k](x - x_{k-1}) + f[x_{k-1}, x_k, x_{k+1}](x - x_{k-1})\,(x - x_k),$$

a zatim, deriviranjem i uvrštavanjem x_k

$$s_k = p'_{2,k}(x_k) = f[x_{k-1}, x_k] + f[x_{k-1}, x_k, x_{k+1}](x_k - x_{k-1}).$$

Uz oznaku

$$h_k = x_k - x_{k-1}, \quad k = 1, \ldots, n,$$

prethodna se formula može napisati i kao

$$s_k = f[x_{k-1}, x_k] + h_k\,\frac{f[x_k, x_{k+1}] - f[x_{k-1}, x_k]}{h_k + h_{k+1}} = \frac{h_{k+1}\,f[x_{k-1}, x_k] + h_k\,f[x_k, x_{k+1}]}{h_k + h_{k+1}},$$

tj. s_k je težinska srednja vrijednost podijeljene razlike unaprijed i unatrag.

Za $k = 0$ pripadni polinom je

$$p_{2,1}(x) = f_0 + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)\,(x - x_1).$$

Deriviranjem, pa uvrštavanjem x_0 dobivamo

$$s_0 = p'_{2,1}(x_0) = f[x_0, x_1] + f[x_0, x_1, x_2](x_0 - x_1) = \frac{(2h_1 + h_2)\,f[x_0, x_1] - h_1\,f[x_1, x_2]}{h_1 + h_2}.$$

Za $k = n$ pripadni polinom je

$$p_{2,n-1}(x) = f_{n-2} + f[x_{n-2}, x_{n-1}](x - x_{n-2}) + f[x_{n-2}, x_{n-1}, x_n](x - x_{n-2})\,(x - x_{n-1}).$$

Deriviranjem, pa uvrštavanjem x_n dobivamo

$$\begin{aligned} s_n = p'_{2,n-1}(x_n) &= f[x_{n-2}, x_{n-1}] + f[x_{n-2}, x_{n-1}, x_n](x_n - x_{n-2}) \\ &\quad + f[x_{n-2}, x_{n-1}, x_n](x_n - x_{n-1}) \\ &= \frac{-h_n\,f[x_{n-2}, x_{n-1}] + (h_{n-1} + 2h_n)\,f[x_{n-1}, x_n]}{h_{n-1} + h_n}. \end{aligned}$$

Dakle, za Besselovu po dijelovima kubičnu interpolaciju stavljamo

$$s_k = \begin{cases} \dfrac{(2h_1 + h_2)\,f[x_0, x_1] - h_1\,f[x_1, x_2]}{h_1 + h_2}, & \text{za } k = 0, \\ \dfrac{h_{k+1}\,f[x_{k-1}, x_k] + h_k\,f[x_k, x_{k+1}]}{h_k + h_{k+1}}, & \text{za } k = 1, \ldots, n-1, \\ \dfrac{-h_n\,f[x_{n-2}, x_{n-1}] + (h_{n-1} + 2h_n)\,f[x_{n-1}, x_n]}{h_{n-1} + h_n}, & \text{za } k = n. \end{cases}$$

Greška u derivaciji (vidjeti prethodni odjeljak) je reda veličine $O(h^2)$, što znači da je greška u funkciji reda veličine $O(h^3)$.

Postoji još jedna varijanta aproksimacije derivacija "s imenom". Akima je 1970. godine dao sljedeću aproksimaciju koja usrednjava podijeljene razlike, s ciljem da se spriječe oscilacije interpolacione funkcije φ:

$$s_k = \frac{w_{k+1} f[x_{k-1}, x_k] + w_{k-1} f[x_k, x_{k+1}]}{w_{k+1} + w_{k-1}}, \quad k = 0, 1, \ldots, n-1, n,$$

uz

$$w_k = |f[x_k, x_{k+1}] - f[x_{k-1}, x_k]|$$

i $w_{-1} = w_0 = w_1$, $w_{n-1} = w_n = w_{n+1}$.

Za $k = 0$ i $k = n$, ove formule se ne mogu odmah iskoristiti, bez dodatnih definicija. Naime, kraćenjem svih težina w_k u formuli za $k = 0$ dobivamo da je

$$s_0 = \frac{f[x_{-1}, x_0] + f[x_0, x_1]}{2}.$$

Ostaje nam samo još definirati što je $f[x_{-1}, x_0]$. Podijeljenu razliku $f[x_0, x_1]$ možemo interpretirati kao sredinu dvije susjedne podijeljene razlike, tj. možemo staviti

$$f[x_0, x_1] = \frac{f[x_{-1}, x_0] + f[x_1, x_2]}{2}.$$

Odatle slijedi da je

$$f[x_{-1}, x_0] = 2f[x_0, x_1] - f[x_1, x_2],$$

odnosno

$$s_0 = \frac{3f[x_0, x_1] - f[x_1, x_2]}{2}$$

i to je praktična formula za s_0. Na sličan način, možemo dobiti i relaciju za s_n

$$s_n = \frac{3f[x_{n-1}, x_n] - f[x_{n-2}, x_{n-1}]}{2}.$$

Akimin je algoritam dosta popularan u praksi i nalazi se u standardnim numeričkim paketima, poput IMSL-a, iako je točnost ovih formula za aproksimaciju derivacije relativno slaba. Općenito, za neekvidistantne točke, greška u derivaciji je reda veličine samo $O(h)$, a to znači samo $O(h^2)$ za funkcijske vrijednosti. Ako su točke ekvidistantne, onda je greška reda veličine $O(h^2)$ za derivaciju, a $O(h^3)$ za funkciju, tj. kao i kod Besselove po dijelovima kvazihermitske interpolacije.

Međutim, ova slabija točnost je potpuno u skladu s osnovnim ciljem Akimine aproksimacije derivacija. U mnogim primjenama, aproksimacijom želimo dobiti geometrijski ili vizuelno poželjan, "lijepo izgledajući" oblik aproksimacione funkcije φ, pa makar i na uštrb točnosti. Tipičan primjer je (približno) crtanje grafova

funkcija, gdje iz nekog relativno malog broja zadanih podataka (točaka) treba, i to brzo, dobiti veliki broj točaka za crtanje vizuelno glatkog grafa. Iako nije nužno da nacrtani graf baš interpolira zadane podatke (male, za oko nevidljive greške sigurno možemo tolerirati), interpolacija obično daje najbrži algoritam.

Ostaje još pitanje kako postići vizuelnu "glatkoću"? Očita heuristika je izbjegavanje naglih promjena u derivaciji. Drugim riječima, želimo "izgladiti" dobivene podatke za derivaciju, a to su izračunate podijeljene razlike. Problem izglađivanja podataka je klasični problem numeričke analize. Jedan od najjednostavnijih i najbržih pristupa je zamjena podatka srednjom vrijednošću podataka preko nekoliko susjednih točaka. Ova ideja je vrlo bliska numeričkoj integraciji, jer integracija "izglađuje" funkciju, pa ćemo tamo dati precizniji opis i opravdanje numeričkog izglađivanja podataka.

Ako bolje pogledamo Akimine formule za aproksimaciju derivacije, one se svode na težinsko usrednjavanje podijeljenih razlika preko nekoliko susjednih točaka s ciljem izglađivanja derivacije (pa onda i funkcije). Vidimo da na s_k utječu točke $x_{k-2}, \ldots, x_{k+2}$, tj. usrednjavanje ide preko 5 susjednih točaka, osim na rubovima. Slično možemo interpretirati i Besselove formule. Tamo usrednjavanje ide preko 3 susjedne točke.

Aproksimacija derivacije mogla bi se napraviti još i bolje, ako povučemo interpolacioni polinom stupnja 3 koji prolazi točkama x_k, x_{k-1}, x_{k+1} i jednom od točaka x_{k-2} ili x_{k+2} (nesimetričnost, jer za kubni polinom trebamo 4 točke, pa s jedne strane od x_k uzimamo dvije, a s druge samo jednu točku) i njega deriviramo u x_k (uz pažljivo deriviranje na rubovima). Takvim postupkom možemo dobiti grešku u funkcijskoj vrijednosti $O(h^4)$. Primijetite da bolja aproksimacija derivacija nije potrebna, jer je greška kod po dijelovima Hermiteove kubične interpolacije također reda veličine $O(h^4)$.

Kvazihermiteova po dijelovima kubična interpolacija je također lokalna, tj. promjenom jedne točke promijenit će se samo nekoliko susjednih kubičnih polinoma. Točno koliko, ovisi o tome koju smo aproksimaciju derivacije izabrali.

7.4.6. Kubična splajn interpolacija

Brojeve $s_0, \ldots, s_n$ možemo odrediti na još jedan način. Umjesto da su s_k neke aproksimacije derivacije funkcije f u čvorovima, možemo zahtijevati da se s_k biraju tako da funkcija φ bude još glađa — da joj je i druga derivacija neprekidna, tj. da je klase $C^2[a, b]$.

Nagibe $s_1, \ldots, s_{n-1}$ određujemo iz uvjeta neprekidnosti druge derivacije u unutarnjim čvorovima $x_1, \ldots x_{n-1}$. Takva se interpolacija zove (kubična) splajn interpolacija.

Možemo li iz tih uvjeta jednoznačno izračunati splajn? Prisjetimo se, imamo $4n$ koeficijenata kubičnih polinoma. Uvjeta interpolacije (svaki polinom mora interpolirati rubne točke svog podintervala) ima $2n$. Uvjeta ljepljenja prve derivacije u unutarnjim točkama ima $n-1$ (toliko je unutarnjih točaka) i jednako je toliko uvjeta ljepljenja druge derivacije.

Dakle, imamo ukupno $4n-2$ uvjeta, a moramo odrediti $4n$ koeficijenata. Odmah vidimo da nam nedostaju 2 uvjeta da bismo te koeficijente mogli odrediti. Kako se oni biraju, to ostavimo za kasnije. Za početak, prva derivacija se lijepi u unutarnjim točkama čim postavimo zahtjev da je $\varphi'(x_k) = s_k$ u tim točkama, bez obzira na to koliki je s_k i ima li on značenje aproksimacije derivacije (vidjeti početak odjeljka o po dijelovima kubičnoj interpolaciji). To nam omogućava da s_k-ove odredimo i na neki drugi način. Zbog toga, ostaje nam samo postaviti uvjete ljepljenja druge derivacije u unutarnjim čvorovima. Zahtjev je

$$p_k''(x_k) = p_{k+1}''(x_k), \quad k = 1, \ldots, n-1.$$

Ako polinome p_k pišemo u formi (7.4.3) relativno obzirom na početnu točku podintervala, tj. ako je

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) + c_{2,k}(x - x_{k-1})^2 + c_{3,k}(x - x_{k-1})^3,$$

onda je

$$\begin{aligned} p_k''(x) &= 2c_{2,k} + 6c_{3,k}(x - x_{k-1}) \\ p_{k+1}''(x) &= 2c_{2,k+1} + 6c_{3,k+1}(x - x_k), \end{aligned}$$

pa je

$$\begin{aligned} p_k''(x_k) &= 2c_{2,k} + 6c_{3,k}(x_k - x_{k-1}) \\ p_{k+1}''(x_k) &= 2c_{2,k+1}. \end{aligned}$$

Drugim riječima, podijelimo li prethodne jednadžbe s 2, uvjet ljepljenja glasi

$$c_{2,k} + 3c_{3,k}(x_k - x_{k-1}) = c_{2,k+1}. \tag{7.4.5}$$

Ostaje samo ispisati koeficijente $c_{i,k}$ iz već ranije dobivenih relacija, u terminima f_k i s_k. Ponovimo

$$\begin{aligned} c_{3,k} &= \frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k^2}, \\ c_{2,k} &= \frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} - h_k c_{3,k}. \end{aligned}$$

Uvrštavanjem u (7.4.5), dobivamo

$$\begin{aligned} &\frac{f[x_{k-1}, x_k] - s_{k-1}}{h_k} + 2\,\frac{s_k + s_{k-1} - 2f[x_{k-1}, x_k]}{h_k} \\ &\qquad = \frac{f[x_k, x_{k+1}] - s_k}{h_{k+1}} - \frac{s_{k+1} + s_k - 2f[x_k, x_{k+1}]}{h_{k+1}}. \end{aligned}$$

Sređivanjem dobivamo

$$\frac{-3f[x_{k-1},x_k]+s_{k-1}+2s_k}{h_k}=\frac{3f[x_k,x_{k+1}]-2s_k-s_{k+1}}{h_{k+1}}.$$

Pomnožimo li prethodnu relaciju s h_kh_{k+1} i prebacimo li sve s_k na lijevu stranu, a članove koji nemaju s_k na desnu stranu, za $k=1,\dots,n-1$, dobivamo

$$h_{k+1}s_{k-1}+2(h_k+h_{k+1})s_k+h_ks_{k+1}=3(h_{k+1}f[x_{k-1},x_k]+h_kf[x_k,x_{k+1}]).$$

Ovo je linearni sustav s $(n+1)$-om nepoznanicom i $(n-1)$-om jednadžbom. Ako na neki način zadamo rubne nagibe s_0 i s_n, onda ostaje točno $n-1$ nepoznanica.

Matrica tako dobivenog linearnog sustava je trodijagonalna

$$\begin{bmatrix} 2(h_1+h_2) & h_1 & & & & \\ h_3 & 2(h_2+h_3) & h_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{n-1} & 2(h_{n-2}+h_{n-1}) & h_{n-2} & \\ & & & h_n & 2(h_{n-1}+h_n) & \end{bmatrix}$$

i strogo dijagonalno dominantna po recima, jer je

$$2(h_k+h_{k+1})>h_k+h_{k+1},$$

pa je i regularna (Geršgorin). Dakle, ovaj linearni sustav sigurno ima jedinstveno rješenje $s_1,\dots s_{n-1}$. Osim toga, za rješavanje možemo koristiti Gaussove eliminacije ili LR faktorizaciju **bez** pivotiranja, a pripadni algoritam smo već napravili.

Primijetite, sada s_k nisu nezavisni, nego ovise jedan o drugom. To znači da aproksimacija više **nije lokalna**, jer se promjenom jedne točke mijenjaju **svi** polinomi. Preciznije, promjena jedne vrijednosti f_{k_0} mijenja desne strane u 3 jednadžbe (za k_0-1, k_0 i k_0+1), ali se zbog toga promijeni cijeli vektor rješenja sustava, tj. svi s_k-ovi. I tada se može pokazati da su promjene lokalizirane — najviše se promjene s_k-ovi za k blizu k_0, a promjene padaju prema rubovima.

Posljednje otvoreno pitanje je kako možemo izabrati s_0 i s_n. Oni se ne moraju direktno zadati, već se uobičajeno zadaju rubni uvjeti na funkciju φ iz kojih se određuju s_0 i s_n ili još dvije jednadžbe linearnog sustava (prva i zadnja).

Postoji nekoliko tradicionalnih načina zadavanja rubnih uvjeta, odnosno jednadžbi koje nedostaju.

(a) Potpuni splajn

Ako je poznata derivacija funkcije f u rubovima, a to je, recimo slučaj kod rješavanja rubnih problema za običnu diferencijalnu jednadžbu, onda je prirodno zadati

$$s_0 = f'(x_0), \quad s_n = f'(x_n).$$

Takav oblik splajna se katkad zove potpuni ili kompletni splajn. Greška aproksimacije u funkcijskoj vrijednosti je $O(h^4)$.

(b) Zadana druga derivacija u rubovima

Ako je poznata druga derivacija funkcije f u rubovima, onda treba staviti

$$f''(x_0) = \varphi''(x_0) = p_1''(x_0), \quad f''(x_n) = \varphi''(x_n) = p_n''(x_n).$$

Ostaje još samo izraziti $p_1''(x_0)$ preko s_0, s_1, a $p_n''(x_n)$ preko s_{n-1} i s_n. Znamo da je

$$c_{2,1} = \frac{p_1''(x_0)}{2} = \frac{f''(x_0)}{2},$$

pa iz izraza za $c_{2,1}$ izlazi

$$\frac{3f[x_0, x_1] - 2s_0 - s_1}{h_1} = \frac{f''(x_0)}{2},$$

ili, ako sredimo, dobivamo jednadžbu

$$2s_0 + s_1 = 3f[x_0, x_1] - \frac{h_1}{2} f''(x_0).$$

Ovu jednadžbu treba dodati kao prvu u linearni sustav. Slično, korištenjem da je

$$p_n''(x_n) = 2c_{2,n} + 6c_{3,n}h_n,$$

te uvrštavanjem izraza za $c_{2,n}$ i $c_{3,n}$, izlazi i

$$s_{n-1} + 2s_n = 3f[x_{n-1}, x_n] + \frac{h_n}{2} f''(x_n).$$

Tu jednadžbu dodajemo kao zadnju u linearni sustav. Dobiveni linearni sustav ima $(n+1)$-u jednadžbu i isto toliko nepoznanica, a može se pokazati da ima i jedinstveno rješenje. Ponovno, greška aproksimacije u funkcijskoj vrijednosti je $O(h^4)$.

(c) Prirodni splajn

Ako zadamo tzv. slobodne krajeve, tj ako je

$$\varphi''(x_0) = \varphi''(x_n) = 0$$

dobivamo prirodnu splajn interpolaciju. Na isti način kao u (b), dobivamo dvije dodatne jednadžbe

$$2s_0 + s_1 = 3f[x_0, x_1], \quad s_{n-1} + 2s_n = 3f[x_{n-1}, x_n].$$

Ako aproksimirana funkcija f nema na rubu druge derivacije jednake 0, onda je greška aproksimacije u funkcijskoj vrijednosti $O(h^2)$, a ako ih ima, onda je (kao u (b) slučaju) greška $O(h^4)$.

(d) Numerička aproksimacija derivacija

Ako ništa ne znamo o ponašanju derivacije funkcije f na rubovima, bolje je ne zadavati njeno ponašanje.

Preostala dva parametra mogu se odrediti tako da numerički aproksimiramo φ' ili φ'' ili φ''' u rubovima, koristeći kao aproksimaciju odgovarajuću derivaciju kubnog interpolacionog polinoma koji prolazi točkama $x_0, \ldots, x_3$, odnosno $x_{n-3}, \ldots, x_n$. Bilo koja od ovih varijanti daje pogrešku reda $O(h^4)$.

(e) Not-a-knot splajn

Moguć je i drugačiji pristup. Umjesto neke aproksimacije derivacije, koristimo tzv. "not-a-knot" (nije čvor) uvjet. Parametre s_0 i s_n biramo tako da su prva dva i posljednja dva kubna polinoma jednaka, tj. da je

$$p_1 = p_2, \quad p_{n-1} = p_n.$$

Ekvivalentno, to znači da se u čvoru x_1 zalijepi i treća derivacija polinoma p_1 i p_2, odnosno da se u čvoru x_{n-1} zalijepi treća derivacija polinoma p_{n-1} i p_n. Te zahtjeve možemo pisati kao

$$p_1'''(x_1) = p_2'''(x_1), \quad p_{n-1}'''(x_{n-1}) = p_n'''(x_{n-1}).$$

Zahtjev $p_1'''(x_1) = p_2'''(x_1)$ znači da su vodeći koeficijenti polinoma p_1 i p_2 jednaki, tj. da je

$$c_{3,1} = c_{3,2}.$$

Pridružimo li taj zahtjev zahtjevu ljepljenja druge derivacije,

$$c_{2,1} + 3c_{3,1}h_k = c_{2,2},$$

dobivamo

$$\frac{f[x_0, x_1] - s_0}{h_1} + 2\,\frac{s_1 + s_0 - 2f[x_0, x_1]}{h_1} = \frac{f[x_1, x_2] - s_1}{h_2} - h_2\,\frac{s_1 + s_0 - 2f[x_0, x_1]}{h_1^2}.$$

Sređivanjem, izlazi

$$h_2 s_0 + (h_1 + h_2)s_1 = \frac{(h_1 + 2(h_1 + h_2))h_2\, f[x_0, x_1] + h_1^2\, f[x_1, x_2]}{h_1 + h_2}.$$

Na sličan način dobivamo i zadnju jednadžbu

$$(h_{n-1}+h_n)s_{n-1}+h_{n-1}s_n=\frac{(h_n+2(h_{n-1}+h_n))h_{n-1}\,f[x_{n-1},x_n]+h_n^2\,f[x_{n-2},x_{n-1}]}{h_{n-1}+h_n}.$$

Kao i dosad, greška aproksimacije za funkcijske vrijednosti je $O(h^4)$.

Objasnimo još porijeklo naziva "not-a-knot" za ovaj tip određivanja dodatnih jednadžbi. Standardno, kubični splajn je klase $C^2[a,b]$, tj. funkcija φ ima neprekidne druge derivacije u unutarnjim čvorovima $x_1,\ldots,x_{n-1}$. Treća derivacija funkcije φ općenito "puca" u tim čvorovima, jer se treće derivacije polinoma p_k i p_{k+1} ne moraju zalijepiti u x_k, za $k=1,\ldots,n-1$. Kad uzmemo u obzir da su svi polinomi p_k kubni, onda je njihova treća derivacija ujedno i zadnja netrivijalna derivacija (sve više derivacije su nula). Dakle, zadnja netrivijalna derivacija splajna puca u unutarnjim čvorovima.

Ova činjenica, u terminologiji teorije splajn funkcija, odgovara tome da svi unutarnji čvorovi splajna imaju multiplicitet 1, jer je multiplicitet čvora jednak broju zadnjih derivacija koje pucaju ili mogu pucati u tom čvoru (derivacije se broje unatrag, počev od zadnje netrivijalne, koja odgovara stupnju polinoma). U tom smislu, povećanje glatkoće splajna u (unutarnjem) čvoru smanjuje multiplicitet tog čvora.

Prethodni zahtjev da se i zadnje netrivijalne derivacije splajna zalijepe u čvorovima x_1 i x_{n-1} odgovara tome da njihov multiplicitet više nije 1, nego 0. Čvorovi multipliciteta 0, naravno, nisu "pravi" čvorovi splajna, jer u njima nema pucanja derivacija (jednako kao i u svim ostalim točkama iz $[a,b]$ koje nisu čvorovi). Međutim, to **ne** znači da čvorove x_1 i x_{n-1} možemo izbaciti, jer u njima i dalje moraju biti zadovoljeni uvjeti interpolacije $\varphi(x_1)=f_1$ i $\varphi(x_{n-1})=f_{n-1}$. Dakle, te točke **ostaju** čvorovi interpolacije, iako nisu čvorovi splajna u smislu pucanja derivacija.

(f) Ostali rubni uvjeti

Svi dosad opisani načini zadavanja rubnih uvjeta "čuvaju" trodijagonalnu strukturu linearnog sustava za parametre s_k, pod uvjetom da eventualne dodatne jednadžbe prirodno dodamo kao prvu i zadnju.

Za aproksimaciju periodičkih funkcija na intervalu koji odgovara periodu, ovakvi oblici zadavanja rubnih uvjeta nisu pogodni. Da bismo očuvali periodičnost, prirodno je postaviti tzv. periodičke rubne uvjete. U praksi se najčešće koristi zahtjev periodičnosti prve i druge derivacije na rubovima

$$\varphi'(x_0)=\varphi'(x_n),\quad \varphi''(x_0)=\varphi''(x_n),$$

što vodi na jednadžbe

$$p_1'(x_0)=p_n'(x_n),\quad p_1''(x_0)=p_n''(x_n).$$

Dobiveni linearni sustav više nije trodijagonalan. Probajte napraviti efikasan algoritam za njegovo rješavanje, tako da složenost ostane linearna u n.

U slučaju potrebe, dozvoljeno je i kombinirati razne oblike rubnih uvjeta u jednom i drugom rubu.

Primjer 7.4.1. *Nađite po dijelovima kubičnu Hermiteovu interpolaciju za podatke*

x_k	0	1	2
f_k	1	2	0
f'_k	0	1	1

.

Očito, treba povući dva kubna polinoma p_1 i p_2. Polinom p_1 "vrijedi" na $[0,1]$, a p_2 na $[1,2]$. Prije računanja ovih polinoma, uvedimo još skraćenu oznaku za podijeljene razlike reda j, po ugledu na oznaku za derivacije višeg reda,

$$f^{[j]}[x_k] := f[x_k, \ldots, x_{k+j}], \quad j \geq 0,$$

tako da tablice imaju kraće "naslove" stupaca.

Za prvi polinom imamo sljedeću tablicu podijeljenih razlika

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
0	1			
		0		
0	1		1	
		1		-1
1	2		0	
		1		
1	2			

Iz nje dobivamo

$$p_1(x) = 1 + (1+1)(x-0)^2 - 1(x-0)^3 = 1 + 2x^2 - x^3.$$

Na sličan način, za p_2 dobivamo tablicu podijeljenih razlika

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
1	2			
		1		
1	2		-3	
		-2		6
2	0		3	
		1		
2	0			

pa je

$$\begin{aligned} p_2(x) &= 2 + (x-1) + (-3-6)(x-1)^2 + 6(x-1)^3 \\ &= 2 + (x-1) - 9(x-1)^2 + 6(x-1)^3. \end{aligned}$$

Primjer 7.4.2. *Neka je*

$$f(x) = \sin(\pi x).$$

Nađite prirodni splajn koji aproksimira funkciju f na $[0,1]$ s čvorovima interpolacije $x_k = 0.2k$, za $k = 0, \ldots, 5$. Izračunajte vrijednost tog splajna u točki 0.55.

Budući da su točke ekvidistantne s razmakom $h = 0.2$, "srednje" jednadžbe linearnog sustava za splajn su

$$hs_{k-1} + 4hs_k + hs_{k+1} = 3(hf[x_{k-1}, x_k] + hf[x_k, x_{k+1}]), \quad k = 1, \ldots, 4.$$

Dodatne jednadžbe (prva i zadnja) za prirodni splajn su

$$\begin{aligned} 2s_0 + s_1 &= 3f[x_0, x_1] \\ s_4 + 2s_5 &= 3f[x_4, x_5]. \end{aligned}$$

Za desnu stranu sustava trebamo izračunati prve podijeljene razlike

x_k	f_k	$f[x_k, x_{k+1}]$
0.0	0.0000000000	
		2.9389262615
0.2	0.5877852523	
		1.8163563200
0.4	0.9510565163	
		0.0000000000
0.6	0.9510565163	
		−1.8163563200
0.8	0.5877852523	
		−2.9389262615
1.0	0.0000000000	

Iz svih ovih podataka dobivamo linearni sustav

$$\begin{bmatrix} 0.4 & 0.2 & & & & \\ 0.2 & 0.8 & 0.2 & & & \\ & 0.2 & 0.8 & 0.2 & & \\ & & 0.2 & 0.8 & 0.2 & \\ & & & 0.2 & 0.8 & 0.2 \\ & & & & 0.2 & 0.4 \end{bmatrix} \cdot \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{bmatrix} = \begin{bmatrix} 1.7633557569 \\ 2.8531695489 \\ 1.0898137920 \\ -1.0898137920 \\ -2.8531695489 \\ -1.7633557569 \end{bmatrix}$$

Rješenje tog linearnog sustava za "nagibe" je

$$\begin{aligned} s_0 &= -s_5 = 3.1387417029, \\ s_1 &= -s_4 = 2.5392953786, \\ s_2 &= -s_3 = 0.9699245271. \end{aligned}$$

Budući da se točka $x = 0.55$ *nalazi u intervalu* $[x_2, x_3] = [0.4, 0.6]$, *restrikcija splajna na taj interval je polinom* p_3, *kojeg nalazimo iz tablice podijeljenih razlika*

x_k	f_k	$f^{[1]}[x_k]$	$f^{[2]}[x_k]$	$f^{[3]}[x_k]$
0.4	0.9510565163			
		0.9699245271		
0.4	0.9510565163		−4.8496226357	
		0.0000000000		0.0000000000
0.6	0.9510565163		−4.8496226357	
		−0.9699245271		
0.6	0.9510565163			

Odavde odmah slijedi da je taj kubični polinom jednak

$$p_3(x) = 0.9510565163 + 0.9699245271(x - 0.4) - 4.8496226357(x - 0.4)^2,$$

tj. p_3 *je zapravo kvadratni polinom.*

Pogledajmo još aproksimacije za funkciju, prvu i drugu derivaciju u točki 0.55.

	funkcija $j = 0$	prva derivacija $j = 1$	druga derivacija $j = 2$
$f^{(j)}(0.55)$	0.9876883406	−0.4914533661	−9.7480931932
$\varphi^{(j)}(0.55)$	0.9874286861	−0.4849622636	−9.6992452715
greška	0.0002596545	−0.0064911026	−0.0488479218

Vidimo da su aproksimacije vrlo točne, iako je h *relativno velik. To je zato što funkcija* $f(x) = \sin(\pi x)$ *zadovoljava prirodne rubne uvjete* $f''(0) = f''(1) = 0$, *kao i prirodni splajn. Greška aproksimacije funkcije je reda veličine* $O(h^4)$, *prve derivacije* $O(h^3)$, *a druge derivacije* $O(h^2)$.

7.5. Interpolacija polinomnim splajnovima — za matematičare

Iskustvo s polinomnom interpolacijom ukazuje da polinomi imaju dobra lokalna svojstva aproksimacije, ali da globalna uniformna pogreška može biti vrlo velika. Niti posebnim izborom čvorova interpolacije ne možemo ukloniti taj fenomen. Nameće se prirodna ideja da izbjegavamo visoke stupnjeve polinoma, ali da konstruiramo polinome niskog stupnja na nekoj subdiviziji segmenta od interesa, tj. da razmotrimo **po dijelovima polinomnu interpolaciju**.

Ako je funkcija koju želimo interpolirati glatka, želimo sačuvati što je moguće veću glatkoću takvog interpolanta. To nas vodi na zahtijev da za po dijelovima linearne aproksimacije zahtijevamo globalnu neprekidnost, za po dijelovima parabolične

aproksimacije globalnu diferencijabilnost, itd. Po dijelovima polinomne funkcije koje zadovoljavaju zadane uvjete glatkoće zovemo **polinomne splajn funkcije**. Koeficijente u nekoj reprezentaciji polinomnog splajna odredit ćemo iz uvjeta interpolacije, kao i u slučaju polinomne interpolacije. Takav specijalni izbor splajna zove se **interpolacijski polinomni splajn**.

U sljedeća dva odjeljka istražit ćemo konstrukciju i svojstva aproksimacije linearnog i kubičnog splajna. Dok je za linearni splajn očito moguće zahtijevati najviše neprekidnost na cijelom segmentu od interesa (zahtijev za "lijepljenjem" prve derivacije vodi na funkciju koja je globalno linearna), za kubične je splajnove moguće zahtijevati pripadnost prostorima C^1 ili C^2, tj. moguće je naći dva kubična splajna.

Splajnovi parnog stupnja mogu biti problematični, kao što pokazuje sljedeća intuitivna diskusija. Zamislimo da je segment od interesa za interpolaciju $[a, b]$, i neka je neka njegova subdivizija (podjela na podintervale) zadana mrežom čvorova

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b. \tag{7.5.1}$$

Paraboličk splajn S_2 mora biti polinom stupnja najviše 2 (parabola) na svakom intervalu subdivizije, tj. imamo po 3 nepoznata parametra (koeficijenti polinoma stupnja 2) na svakom intervalu. Ukupno dakle treba naći $3N$ slobodnih parametara.

Zahtijev da vrijedi $S_2 \in C^1[a, b]$ vezuje $2(N-1)$ od tih parametara (neprekidnost S_2 i S_2' u $N-1$ unutrašnjih čvorova $x_1, \ldots, x_{N-1}$). Ostaju dakle $N+2$ slobodna parametra. Zahtijevamo li da S_2 bude interpolacijski, tj. da vrijedi

$$S_2(x_i) = f_i, \quad i = 0, \ldots, N,$$

ostaje slobodan samo jedan parametar. Taj bismo parametar mogli odrediti dodavanjem još jednog čvora interpolacije, ili nekim dodatnim uvjetom na rubu cijelog intervala — recimo, zadavanjem derivacije. Međutim, jasno je da se taj parametar ne može odrediti **simetrično** iz podataka. To je problem i s ostalim splajn interpolantima parnog stupnja.

Zadatak 7.5.1. *Nađite što je u gornjoj diskusiji neformalno, i što je potrebno za precizan matematički dokaz. Ako je prostor polinomnih splajnova $\mathcal{S}(n)$ stupnja n definiran zahtjevima:*

(1) $s \in \mathcal{S}(n) \Longrightarrow s|_{[x_i, x_{i+1}]} \in \mathcal{P}_n$ *($\mathcal{P}_n$ je prostor polinoma stupnja n);*

(2) $s \in C^{n-1}[x_0, x_N]$,

pokažite da je $\mathcal{S}(n)$ vektorski prostor, i dokažite da je $\dim \mathcal{S}(n) = n + N$.

7.5.1. Linearni splajn

Najjednostavniji **linearni interpolacijski splajn** S_1 određen je uvjetom globalne neprekidnosti i uvjetom interpolacije

$$S_1(x_i) = f_i, \quad i = 0, \ldots, N,$$

na mreži čvorova — subdiviziji segmenta $[a, b]$ zadanoj s (7.5.1). Očito imamo

$$S_1(x) = f_i \frac{x_{i+1} - x}{h_i} + f_{i+1} \frac{x - x_i}{h_i} = f_i + \frac{x - x_i}{h_i}(f_{i+1} - f_i), \quad x \in [x_i, x_{i+1}],$$

gdje je $h_i = x_{i+1} - x_i$, za $i = 0, \ldots, N-1$. Algoritam za računanje je trivijalan, pa možemo odmah ispitati pogrešku, odnosno, razmotriti svojstva interpolacijskog linearnog splajna obzirom na glatkoću funkcije koja se interpolira, u raznim normama koje se koriste za aproksimaciju. U dokazima ćemo koristiti jednu sporednu lemu.

Lema 7.5.1. *Ako je $f \in C[a, b]$ i α, β imaju isti znak, tada postoji $\xi \in [a, b]$ tako da vrijedi*

$$\alpha f(a) + \beta f(b) = (\alpha + \beta) f(\xi).$$

Dokaz:

Ako je $f(a) = f(b)$ tvrdnja je očigledna, jer možemo uzeti $\xi = a$ ili $\xi = b$. Ako je $f(a) \neq f(b)$, tada funkcija $\psi(x) = \alpha f(a) + \beta f(b) - (\alpha + \beta) f(x)$ poprima suprotne predznake na krajevima intervala, pa zbog neprekidnosti postoji $\xi \in (a, b)$ tako da je $\psi(\xi) = 0$. Tvrdnja vrijedi i ako je $\alpha = 0$, uz $\xi = b$, odnosno, $\beta = 0$, uz $\xi = a$. ■

Za precizno određivanje reda konvergencije aproksimacija neprekidne funkcije f zgodno je uvesti oznake

$$\omega_i(f) = \max_{x', x'' \in [x_i, x_{i+1}]} |f(x'') - f(x')|, \quad i = 0, \ldots, N-1,$$
$$\omega(f) = \max_{0 \leq i \leq N-1} \omega_i(f).$$

Vrijednost $\omega_i(f)$ zovemo **oscilacija** funkcije f na podintervalu $[x_i, x_{i+1}]$, a $\omega(f)$ je (očito) najveća oscilacija po svim podintervalima mreže.

Uočite da glatkoća funkcije f nije potrebna u definiciji ovih veličina, pa ih koristimo za ocjenu greške u slučaju da je f samo neprekidna, ali ne i derivabilna funkcija. Isto vrijedi i za zadnju (najvišu) **neprekidnu** derivaciju funkcije.

Također, kod ocjene grešaka, zgodno je uvesti skraćenu oznaku D za operator deriviranja funkcije f jedne varijable, kad je iz konteksta očito po kojoj varijabli se derivira. Onda n-tu derivaciju funkcije f u točki x možemo pisati u bilo kojem od sljedeća tri oblika

$$D^n f(x) = \frac{d^n}{dx^n} f(x) = f^{(n)}(x).$$

Pokazuje se da je prvi oblik najpregledniji u zapisu nekih dugačkih formula.

Da bismo olakšali razumijevanje teorema o ocjenama greške splajn interpolacije koji slijede, objasnimo odmah osnovnu ideju za uvođenje oznake $\omega(f)$ i njezinu ulogu u ocjeni greške. Jednostavno rečeno, $\omega(f)$ služi tome da napravimo finu razliku između ograničenosti i neprekidnosti funkcije f na nekom intervalu. Za dobivanje korisnih ocjena, obično, uz ograničenost, pretpostavljamo još i integrabilnost funkcije. Neprekidnost je, očito, jače svojstvo.

Za ilustraciju, uzmimo da je f derivabilna funkcija na $[a, b]$. Onda je prva derivacija Df i ograničena funkcija na $[a, b]$, čim postoji derivacija u svakoj točki segmenta, s tim da uzimamo jednostrane derivacije (limese) u rubovima. Drugim riječima, postoji njezina ∞–norma

$$\|Df\|_\infty = \sup_{x\in[a,b]} |Df(x)|.$$

Ako je prva derivacija i integrabilna, to označavamo s $f \in L^1_\infty[a, b]$. Gornji indeks 1 označava da je riječ o prvoj derivaciji funkcije f, a donji indeks ∞ označava ograničenost (preciznija definicija prostora $L^1_\infty[a, b]$ zahtijeva teoriju mjere i integrala). Naravno, prva derivacija **ne mora** biti neprekidna na $[a, b]$, da bi bila integrabilna. Ako je Df i neprekidna, onda je $f \in C^1[a, b]$ (oznaka koju smo odavno koristili).

Jedan od rezultata koje želimo dobiti ocjenom greške je uniformna konvergencija splajn interpolacije kad povećavamo broj čvorova, tj. "profinjujemo" mrežu (barem uz neke blage uvjete). Za uniformnu konvergenciju, očito, treba promatrati maksimalnu grešku na cijelom intervalu, tj. zanimaju nas tzv. uniformne ocjene — u ∞–normi. Iz iskustva polinomne interpolacije, jasno je da moramo iskoristiti **lokalno** ponašanje funkcije i splajn interpolacije na podintervalima mreže.

Kako ćemo lokalnost dobro ugraditi u ocjenu greške? S jedne strane, kvaliteta ocjene mora ovisiti o svojstvima (glatkoći) funkcije koju aproksimiramo (interpoliramo). Dakle, trebamo dobru globalnu mjeru lokalnog ponašanja funkcije. Za ograničene (integrabilne) funkcije koristimo ∞–normu na $[a, b]$, koja, očito, postoji. Nažalost, lokalnost tu ne pomaže, jer maksimum normi po podintervalima daje upravo normu na cijelom intervalu. Neprekidna funkcija je, naravno, i ograničena i integrabilna. Međutim, za neprekidne funkcije, $\omega(f)$ daje bitno precizniju uniformnu ocjenu greške od globalne norme, jer uključuje lokalno ponašanje po podintervalima — najveća lokalna oscilacija može biti mnogo manja od globalne oscilacije na cijelom intervalu!

S druge strane, ocjena greške mora uključivati ovisnost o nekoj veličini koja mjeri "gustoću" mreže, odnosno razmak čvorova. Ako profinjavanjem mreže želimo dobiti konvergenciju, odmah je jasno da to profinjavanje mora biti "ravnomjerno"

u cijelom $[a,b]$, tj. maksimalni razmak susjednih čvorova

$$\bar{h} := \max_{0\le i\le N-1} h_i$$

mora težiti prema nuli. Da bismo izbjegli ovisnost o svim h_i, standardno se ocjene greške izražavaju upravo u terminima veličine $\bar{h}$, koja se još zove i **dijametar mreže**.

Teorem 7.5.1. (Uniformna ocjena pogreške linearnog splajna)

Neka je $S_1(x)$ linearni interpolacijski splajn za funkciju f. Obzirom na svojstva glatkoće funkcije f vrijedi:

(1) *ako je $f \in C[a,b]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \omega(f);$$

(2) *ako je $f \in L^1_\infty[a,b]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}}{2}\,\|Df\|_\infty;$$

(3) *ako je $f \in C[a,b] \bigcap_{i=0}^{N-1} C^1[x_i, x_{i+1}]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}}{4}\,\omega(Df);$$

(4) *ako je $f \in C[a,b] \bigcap_{i=0}^{N-1} L^2_\infty[x_i, x_{i+1}]$ tada je*

$$\|S_1(x) - f(x)\|_\infty \le \frac{\bar{h}^2}{8}\,\|D^2 f\|_\infty.$$

Dokaz:

Neka je $t := (x - x_i)/h_i$. Prema (7.5.1.) greška je

$$E(x) := S_1(x) - f(x) = (1-t)f_i + t f_{i+1} - f(x), \quad x \in [x_i, x_{i+1}]. \tag{7.5.2}$$

Uočimo da je $x \in [x_i, x_{i+1}]$ ekvivalentno s $t \in [0,1]$, pa $(1-t)$ i t imaju isti (pozitivni) predznak, ili je jedan od njih jednak nula.

Ako je $f \in C[a,b]$, onda prema Lemi 7.5.1. postoji $\xi \in [x_i, x_{i+1}]$ takav da vrijedi $E(x) = f(\xi) - f(x)$, pa je $|E(x)| \le \omega_i(f) \le \omega(f)$.

Ako je prva derivacija ograničena i integrabilna, vrijedi

$$f_i = f(x) + \int_x^{x_i} Df(v)\,dv, \quad f_{i+1} = f(x) + \int_x^{x_{i+1}} Df(v)\,dv.$$

Supstitucijom u (7.5.2) dobijemo

$$E(x) = -(1-t)\int_{x_i}^{x} Df(v)\,dv + t\int_{x}^{x_{i+1}} Df(v)\,dv$$

i

$$|E(x)| \le (1-t)\int_{x_i}^{x} |Df(v)|\,dv + t\int_{x}^{x_{i+1}} |Df(v)|\,dv.$$

Slijedi

$$|E(x)| \le \left[(1-t)\int_{x_i}^{x} dv + t\int_{x}^{x_{i+1}} dv\right] \|Df\|_\infty = 2t(1-t)\,h_i\,\|Df\|_\infty.$$

Kako parabola $2t(1-t)$ ima maksimum $1/2$ u $t = 1/2$, dokazali smo da vrijedi

$$|E(x)| \le \frac{1}{2}\,\bar{h}\,\|Df\|_\infty.$$

Neka je sada $f \in C[a,b]$ klase C^1 na svakom podintervalu mreže (eventualni prekidi prve derivacije mogu biti samo u čvorovima mreže). Prema Taylorovoj formuli s Lagrangeovim oblikom ostatka

$$f_i = f(x) - t\,h_i\,Df(\xi), \quad f_{i+1} = f(x) + (1-t)\,h_i\,Df(\eta), \quad \xi,\eta \in [x_i, x_{i+1}].$$

Supstitucijom u (7.5.2) dobijemo

$$E(x) = t(1-t)\,h_i\,(Df(\eta) - Df(\xi)),$$

odakle slijedi

$$|E(x)| \le t(1-t)\,h_i\,\omega_i(Df),$$

pa opet ocjenom parabole na desnoj strani dobijemo da vrijedi

$$|E(x)| \le \frac{1}{4}\,\bar{h}\,\omega(Df).$$

Na kraju, ako f ima na svakom podintervalu ograničenu i integrabilnu drugu derivaciju, tada vrijedi Taylorova formula s integralnim oblikom ostatka

$$f_i = f(x) - t\,h_i\,Df(x) + \int_{x}^{x_i} (x_i - v)\,D^2 f(v)\,dv$$

$$f_{i+1} = f(x) + (1-t)\,h_i\,Df(x) + \int_{x}^{x_{i+1}} (x_{i+1} - v)\,D^2 f(v)\,dv,$$

pa iz formule (7.5.2) slijedi

$$E(x) = (1-t)\int_x^{x_i}(x_i - v)\,D^2 f(v)\,dv + t\int_x^{x_{i+1}}(x_{i+1} - v)\,D^2 f(v)\,dv.$$

Odavde lako slijedi

$$|E(x)| \le \frac{1}{2}\,h_i^2\,t(1-t)\,\|D^2 f\|_\infty \le \frac{1}{8}\,\bar{h}^2\,\|D^2 f\|_\infty.$$

■

Zadatak 7.5.2. *Dokažite da se u slučajevima* (3) *i* (4) *teorema* 7.5.1. *može ocijeniti i greška u derivaciji, točnije, da vrijedi:*

(3) $\|DS_1(x) - Df(x)\|_\infty \le \omega(Df)$;

(4) $\|DS_1(x) - Df(x)\|_\infty \le \dfrac{\bar{h}}{2}\,\|D^2 f\|_\infty$.

Teoremi poput teorema 7.5.1. pripadaju grupi teorema koji se nazivaju **direktni teoremi teorije aproksimacija**. Iako u daljnjem nećemo slijediti ovaj pristup do krajnjih detalja, primijetimo da se prirodno pojavljuju dva važna pitanja.

(1) Da li su navedene ocjene najbolje moguće, tj. da li smo zbog tehnike dokazivanja napravili na nekom mjestu pregrubu ocjenu, iskoristili nedovoljno "finu" nejednakost, pa zapravo možemo dobiti bolji red konvergencije? Da li su i konstante u ocjeni greške najbolje moguće?

(2) Ako dalje povećavamo glatkoću funkcije koja se interpolira, možemo li dobiti sve bolje i bolje ocjene za grešku, na primjer, u slučaju linearnog splajna, ocjene s h^2, h^3, i tako redom?

Teoremi koji se bave problematikom kao u (2) zovu se **inverzni teoremi teorije aproksimacija**. U većini slučajeva to su iskazi tipa "red aproksimacije naveden u direktnom teoremu je najbolji mogući". Doista, da nije tako, trebalo bi dopuniti ili popraviti direktni teorem! Ocjena optimalnosti konstanti je neugodan problem, koji za opći stupanj splajna nije riješen — treba konstruirati primjer funkcije na kojoj se dostiže konstanta iz direktnog teorema.

Slični su i tzv. **teoremi zasićenja teorije aproksimacija**, koji pokušavaju odgovoriti na drugo pitanje: može li se bolje aproksimirati funkcija ako su pretpostavke na glatkoću jače? I ovi teoremi su u principu negativnog karaktera — na primjer, za linearni splajn možemo staviti da je $f \in C^\infty[a,b]$, ali red aproksimacije će ostati h^2. Sam prostor u kojem se aproksimira jednostavno ne može točnije reproducirati funkciju koja se aproksimira, nedostaje mu "snage aproksimacije". Iako se u daljnjem nećemo baviti općim teoremima aproksimacije, svi direktni teoremi koji slijede optimalni su u smislu postojanja odgovarajućih inverznih teorema i teorema zasićenja.

Zadatak 7.5.3. *Pokažite da u slučaju* (4) *teorema* 7.5.1. *funkcija* $f(x) = x^2$ *igra ulogu* **ekstremale**, *tj. da vrijedi "=" umjesto "≤", pa je ocjena ujedno i najbolja moguća.*

Zadatak 7.5.4. *Ako je* $f \in C[a,b] \bigcap_{i=0}^{N-1} C^3[x_i, x_{i+1}]$, *tada vrijedi*

$$DS\Big(x_i + \frac{h_i}{2}\Big) = Df\Big(x_i + \frac{h_i}{2}\Big) + O(h_i^2), \quad i = 0, \ldots, N-1.$$

Iz toga možemo zaključiti da red aproksimacije derivacije u specijalno izabranim točkama može biti i **viši** *od optimalnog; to je efekt* **superkonvergencije**.

7.5.2. Hermiteov kubični splajn

Kao i u slučaju Hermiteove interpolacije polinomima, možemo razmatrati i Hermiteovu interpolaciju splajn funkcijama. Ako preskočimo paraboličke splajnove (v. raniju diskusiju), prvi je netrivijalni slučaj po dijelovima kubičnih splajnova s globalno neprekidnom derivacijom.

Definicija 7.5.1. *Neka su u čvorovima* $a = x_0 < x_1 < \cdots < x_N = b$ *zadane vrijednosti* f_i, f_i', *za* $i = 0, \ldots, N$. *Hermiteov interpolacijski kubični splajn je funkcija* $H \in C^1[a,b]$ *koja zadovoljava*

(1) $H(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3$, *za svaki* $x \in [x_i, x_{i+1}]$;

(2) $H(x_i) = f_i$, $DH(x_i) = f_i'$, *za* $i = 0, \ldots, N$.

Koristeći Hermiteovu bazu iz teorema 7.2.4. na svakom podintervalu mreže $[x_i, x_{i+1}]$, lagano vidimo da vrijedi

$$H(x) = \varphi_1(t) f_i + \varphi_2(t) f_{i+1} + \varphi_3(t) h_i f_i' + \varphi_4(t) h_i f_{i+1}', \quad t = \frac{x - x_i}{h_i}, \tag{7.5.3}$$

gdje je

$$\varphi_1(t) = (1-t)^2(1+2t), \qquad \varphi_2(t) = t^2(3-2t),$$
$$\varphi_3(t) = t(1-t^2), \qquad \varphi_4(t) = -t^2(1-t).$$

Napomenimo još samo da kod računanja treba prvo izračunati koeficijente A_i i B_i formulama

$$\begin{aligned} A_i &= -2\,\frac{f_{i+1} - f_i}{h_i} + (f_i' + f_{i+1}'), \\ B_i &= -A_i + \frac{f_{i+1} - f_i}{h_i} - f_i', \end{aligned} \quad \text{za} \quad i = 0, \ldots, N-1, \tag{7.5.4}$$

i zapamtiti ih. Za zadanu točku $x \in [x_i, x_{i+1}]$, Hermiteov splajn računamo formulom

$$H(x) = f_i + (th_i)\,[f_i' + t(B_i + tA_i)]. \tag{7.5.5}$$

Obzirom na činjenicu da su nam derivacije f_i' najčešće nepoznate, preostaje nam samo da ih aproksimiramo iz zadanih vrijednosti funkcije. To je problem **približne Hermiteove interpolacije**, i tada ne možemo više očekivati isti red konvergencije. Vrijednost Hermiteove interpolacije je, međutim, više teorijska nego praktična, kao što ćemo vidjeti kasnije. U tom smislu koristit ćemo sljedeći direktni teorem.

Teorem 7.5.2. *Za Hermiteov kubični splajn, ovisno o glatkoći funkcije f, vrijede sljedeće uniformne ocjene pogreške:*

(1) *ako je $f \in C^1[a,b]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{3}{8}\,\bar{h}\,\omega(Df);$$

(2) *ako je $f \in L^2_\infty[a,b]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{16}\,\bar{h}^2\,\|D^2 f\|_\infty;$$

(3) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} C^2[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{32}\,\bar{h}^2\,\omega(D^2 f);$$

(4) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L^3_\infty[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{96}\,\bar{h}^3\,\|D^3 f\|_\infty;$$

(5) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} C^3[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{192}\,\bar{h}^3\,\omega(D^3 f);$$

(6) *ako je $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L^4_\infty[x_i, x_{i+1}]$ tada je*

$$\|H(x) - f(x)\|_\infty \leq \frac{1}{384}\,\bar{h}^4\,\|D^4 f\|_\infty.$$

Dokaz:

U svim slučajevima treba analizirati grešku

$$E(x) := H(x) - f(x) = f_i\varphi_1(t) + f_{i+1}\varphi_2(t) + h_i f_i'\varphi_3(t) + h_i f_{i+1}'\varphi_4(t) - f(x).$$

Ako f_i, f_{i+1} zamijenimo njihovim Taylorovim razvojem oko točke $x = x_i + th_i$ s ostatkom u Lagrangeovom obliku, dobijemo

$$E(x) = h_i\,[(1-t)\,\varphi_2(t)\,Df(\xi) - t\,\varphi_1(t)\,Df(\eta) + \varphi_3(t) f_i' + \varphi_4(t) f_{i+1}'].$$

U daljnjem oznake ξ, η, ... označavaju točke u $[x_i, x_{i+1}]$. Prema Lemi 7.5.1. možemo grupirati članove istog znaka (prvi i treći, drugi i četvrti), pa dobijemo

$$E(x) = h_i\, t(1-t)(1+2t-2t^2)\,[Df(\bar{\xi}) - Df(\bar{\eta})],$$

odakle slijedi ocjena greške po točkama

$$|E(x)| \le h_i\, t(1-t)(1+2t-2t^2)\,\omega_i(Df).$$

Odavde odmah slijedi tvrdnja (1), uzimanjem maksimuma polinoma u varijabli t.

Ako f ima drugu derivaciju ograničenu i integrabilnu, razvijemo opet f_i, f_i', f_{i+1}, f_{i+1}' oko točke x, ali koristeći Taylorovu formulu s integralnim oblikom ostatka. Nakon kraćeg računa dobijemo integralnu reprezentaciju greške

$$\begin{aligned} E(x) = &\int_{x_i}^{x} (1-t)^2\,[-th_i + (1+2t)(v-x_i)]\,D^2f(v)\,dv \\ &+ \int_{x}^{x_{i+1}} t^2\,[-(1-t)h_i + (3-2t)(x_{i+1}-v)]\,D^2f(v)\,dv. \end{aligned}$$

Zamjenom varijable $v - x_i = \tau h_i$ dobivamo

$$E(x) = h_i^2 \left\{ \int_0^t \psi_1(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau \right\}, \quad (7.5.6)$$

gdje je

$$\begin{aligned} \psi_1(t,\tau) &= (1-t)^2\,[(1+2t)\tau - t], \\ \psi_2(t,\tau) &= t^2\,[(3-2t)(1-\tau) - (1-t)]. \end{aligned}$$

Ne možemo upotrijebiti teorem o srednjoj vrijednosti za integrale, jer $\psi_1(t,\tau)$ mijenja znak; točnije $\psi_1(t,\tau^*) = 0$ za $\tau^* = t/(1+2t)$. Međutim, $[0,t] = [0,\tau^*) \cup [\tau^*, t]$, a na svakom od podintervala ψ_1 je konstantnog znaka, pa teorem srednje vrijednosti za integrale možemo upotrijebiti po dijelovima.

$$\begin{aligned} \int_0^t \psi_1(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau &= D^2f(\xi)\int_0^{\tau^*} \psi_1(t,\tau)\,d\tau + D^2f(\eta)\int_{\tau^*}^t \psi_1(t,\tau)\,d\tau \\ &= \frac{t^2(1-t)^2}{2(1+2t)}\,\{4t^2\,D^2f(\eta) - D^2f(\xi)\}. \end{aligned}$$

Analogno

$$\int_0^t \psi_2(t,\tau)\,D^2f(x_i+\tau h_i)\,d\tau = \frac{(1-t)^2t^2}{2(3-2t)}\,\{4(1-t^2)\,D^2f(\bar{\xi}) - D^2f(\bar{\eta})\}.$$

Iz (7.5.6) dobivamo

$$E(x) = \frac{h_i^2\, t^2(1-t)^2}{2[3+4t(1-t)]}\,\{4t^2(3-2t)\,D^2f(\eta) - (3-2t)\,D^2f(\xi) \\ + 4(1-t^2)(1+2t)\,D^2f(\bar{\xi}) - (1+2t)\,D^2f(\bar{\eta})\}.$$

Primijenimo li lemu 7.5.1. na neprekidne funkcije istog znaka, dobivamo ocjenu

$$|E(x)| \le \frac{2h_i^2\, t^2(1-t)^2}{3+4t(1-t)}\,\omega_i(D^2f).$$

Maksimalna vrijednost desne strane postiže se za $t = 1/2$, odakle slijedi

$$|E(x)| \le \frac{1}{32}\,h_i^2\,\omega_i(D^2f),$$

što dokazuje ocjenu (3). Ocjena (2) proizlazi lagano iz iste ocjene greške po točkama.

Ako je f po dijelovima klase C^3, slično dobivamo

$$E(x) = h_i^3\left\{\int_0^t \psi_1(t,\tau)\,D^3f(x_i+\tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^3f(x_i+\tau h_i)\,d\tau\right\},$$

gdje su sada

$$\psi_1(t,\tau) = (1-t)^2\tau\left[t - \frac{(1+2t)\tau}{2}\right],$$
$$\psi_2(t,\tau) = t^2(1-\tau)\left[-(1-t) + \frac{(3-2t)(1-\tau)}{2}\right].$$

Zbog simetrije, dovoljno je razmatrati $t \in [0, 1/2]$, pa slijedi

$$|E(x)| \le \frac{2}{3}\,h_i^3\,\frac{t^2(1-t)^3}{(3-2t)^2}\,\omega_i(D^3f).$$

Odavde slijedi ocjena greške za $\|E(x)\|_\infty$. Maksimalna greška je u $x_i + h_i/2$, tj. za $t = 1/2$. Slično slijedi i ocjena (4).

Na kraju, ako f ima ograničenu i integrabilnu četvrtu derivaciju na svakom podintervalu, tada je

$$E(x) = \frac{1}{6}\,h_i^4\left\{\int_0^t \psi_1(t,\tau)\,D^4f(x_i+\tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^4f(x_i+\tau h_i)\,d\tau\right\},$$

gdje su

$$\psi_1(t,\tau) = (1-t)^2\tau^2\,[-3t + (1+2t)\tau],$$
$$\psi_2(t,\tau) = t^2(1-\tau)^2\,[-3(1-t) + (3-2t)(1-\tau)],$$

pa zaključujemo da vrijedi

$$|E(x)| \leq \frac{t^2(1-t)^2}{4!}\, h_i^4\, \|D^4 f\|_\infty, \quad t \in [0,1]. \tag{7.5.7}$$

Odavde se lagano dobije ocjena za $\|E(x)\|_\infty$. ■

Zadatak 7.5.5. *Pokažite da za* $f \in C^1[a,b] \bigcap_{i=0}^{N-1} L_\infty^4[x_i, x_{i+1}]$ *(slučaj* (6) *iz prethodnog teorema) vrijede sljedeće ocjene za derivacije:*

$$\|DH(x) - Df(x)\|_\infty \leq \frac{\sqrt{3}}{216}\, \bar{h}^3\, \|D^4 f\|_\infty,$$

$$\|D^2 H(x) - D^2 f(x)\|_\infty \leq \frac{1}{12}\, \bar{h}^2\, \|D^4 f\|_\infty,$$

$$\|D^3 H(x) - D^3 f(x)\|_\infty \leq \frac{1}{2}\, \bar{h}\, \|D^4 f\|_\infty.$$

Uputa: Treba derivirati integralnu reprezentaciju za $E(x)$*, tj. naći integralnu reprezentaciju za* $D^k E(x)$*,* $k = 1, 2, 3$*.*

Zadatak 7.5.6. *Pokušajte za prvih pet slučajeva (klasa glatkoće funkcije* f*) iz teorema* 7.5.2. *izvesti slične ocjene za one derivacije koje imaju smisla obzirom na pretpostavljenu glatkoću. Prema prošlom zadatku, ocjene treba tražiti u obliku*

$$\|D^r H(x) - D^r f(x)\|_\infty \leq C_r\, \bar{h}^{e_f - r}\, M_f, \quad r \in \{0,1,2,3\},$$

gdje su C_r *konstante ovisne o* r*, a osnovni eksponenti* e_f *i "mjere" funkcije* M_f *ovise samo o klasi funkcije (ne i o* r*), pa se mogu "pročitati" iz teorema (*$r = 0$*). Uvjerite se da ocjene imaju smisla samo za* $r \leq e_f$*, a dokazuju se sličnom tehnikom.*

Zadatak 7.5.7. (Superkonvergencija) *Uz pretpostavke dodatne glatkoće funkcije* f*, u posebno izbaranim točkama može se dobiti i viši red aproksimacije pojedinih derivacija funkcije* f*.*

(*a*) *U točkama* $x_i^* := x_i + h_i/2$ *prva derivacija može se aproksimirati s* $O(h_i^4)$*, a treća s* $O(h_i^2)$*. Točnije, vrijedi*

$$DH(x^*) = Df(x^*) - \frac{h_i^4}{1920}\, D^4 f(x^*) + O(h_i^5),$$

$$D^3 H(x^*) = D^3 f(x^*) + \frac{h_i^2}{40}\, D^4 f(x^*) + O(h_i^3).$$

(*b*) *U točkama* $\bar{x}_i := x_i + (3 \pm \sqrt{3})h_i/6$ *druga derivacija može se aproksimirati s* $O(h_i^3)$*. Točnije, vrijedi*

$$D^2 H(\bar{x}) = D^2 f(\bar{x}) \pm \frac{\sqrt{3}\, h_i^3}{540}\, D^5 f(\bar{x}) + O(h_i^4).$$

Nađite uz koje pretpostavke dodatne glatkoće funkcije f *vrijede ove tvrdnje i ocjene, i dokažte ih.*

7.5.3. Potpuni kubični splajn

Zahtijevamo li neprekidnost druge derivacije od po dijelovima kubičnih funkcija, dolazimo prirodno na definiciju **potpunog kubičnog splajna**, koji se često još zove i samo **kubični splajn**. Cilj nam je razmotriti algoritme za konstrukciju kubičnih splajnova koji interpoliraju zadane podatke — vrijednosti funkcije, ali ne i njezine derivacije, jer tražimo veću glatkoću. Takav splajn zovemo **kubični interpolacijski splajn**.

Od svih splajn funkcija, kubični interpolacijski splajn je vjerojatno najviše korišten i najbolje izučen u smislu aproksimacije i brojnih primjena, od aproksimacije u raznim normama, do rješavanja rubnih problema za obične diferencijalne jednadžbe. Ime "splajn" (eng. **"spline"**) označava elastičnu letvicu koja se mogla učvrstiti na rebra brodova kako bi se modelirao oblik oplate; točna etimologija riječi pomalo je zaboravljena. U matematičkom smislu pojavljuje se prvi put u radovima Eulera, oko 1700. godine, i slijedi mehaničku definiciju elastičnog štapa.

Središnja linija s takvog štapa (ona koja se ne deformira kod transverzalnog opterećenja) u linearnoj teoriji elastičnosti ima jednadžbu

$$-D^2(EI\, D^2 s(x)) = f(x),$$

gdje je E Youngov modul elastičnosti štapa, a I moment inercije presjeka štapa oko njegove osi. Pretpostavimo li da je štap izrađen od homogenog materijala, i da ne mijenja poprečni presjek (E i I su konstante), dolazimo na jednadžbu

$$-D^4 s = f,$$

gdje je f vanjska sila po jedinici duljine. U odsustvu vanjske sile ($f = 0$), središnja linija s elastične letvice je dakle kubični polinom.

Ako je letvica učvršćena u osloncima s koordinatama x_i, $i = 0, \ldots, N$, treća derivacija u tim točkama ima diskontinuitet (ova činjenica je posljedica zakona održanja momenta, i trebalo bi ju posebno izvesti). Između oslonaca, na podintervalima $[x_i, x_{i+1}]$, središnja linija je i dalje kubični polinom, ali u točkama x_i imamo prekid treće derivacije. Dakle, s je po dijelovima kubični polinom, a druga derivacija s'' je globalno neprekidna.

Definicija 7.5.2. *Neka su u čvorovima $a = x_0 < x_1 < \cdots < x_N = b$ zadane vrijednosti f_i, za $i = 0, \ldots, N$. Potpuni interpolacijski kubični splajn je funkcija $S_3 \in C^2[a, b]$ koja zadovoljava uvjete*

(1) $S_3(x) = a_{i0} + a_{i1}(x - x_i) + a_{i2}(x - x_i)^2 + a_{i3}(x - x_i)^3$, *za svaki* $x \in [x_i, x_{i+1}]$;

(2) $S_3(x_i) = f_i$, *za* $i = 0, \ldots, N$.

Kako se S_3 na svakom od N podintervala određuje s 4 koeficijenta, ukupno imamo $4N$ koeficijenata koje treba odrediti. Uvjeti glatkoće (funkcija, prva i druga derivacija u unutrašnjim čvorovima) vežu $3(N-1)$ koeficijenata, a uvjeti interpolacije $N+1$ koeficijenata. Preostaje dakle odrediti

$$4N - 3(N-1) - (N+1) = 2$$

dodatna koeficijenta. Dodatni uvjeti obično se zadaju u rubovima intervala, stoga naziv **rubni uvjeti**. U praksi se najčešće koriste sljedeći rubni uvjeti:

$$\begin{array}{lll}
(R1)\ DS_3(a) = Df(a), & DS_3(b) = Df(b), & \text{(potpuni rubni uvjeti);} \\
(R2)\ D^2S_3(a) = 0, & D^2S_3(b) = 0, & \text{(prirodni rubni uvjeti);} \\
(R3)\ D^2S_3(a) = D^2f(a), & D^2S_3(b) = D^2f(b); & \\
(R4)\ DS_3(a) = DS_3(b), & D^2S_3(a) = D^2S_3(b), & \text{(periodički rubni uvjeti).}
\end{array} \tag{7.5.8}$$

Tradicionalno se naziv **potpuni splajn** koristi za splajn oderđen rubnim uvjetima $(R1)$ interpolacije prve derivacije u rubovima. Splajn određen prirodnim rubnim uvjetima $(R2)$ zove se **prirodni splajn**. Njega možemo smatrati specijalnim slučajem rubnih uvjeta $(R3)$ interpolacije druge derivacije u rubovima, naravno, uz uvjet da sama funkcija zadovoljava prirodne rubne uvjete. Na kraju, splajn određen periodičkim rubnim uvjetima $(R4)$ zove se **periodički splajn**, a koristi se za interpolaciju periodičkih funkcija f s periodom $[a,b]$ (tada je $f_0 = f_N$ i f zadovoljava periodičke rubne uvjete).

Algoritam za konstrukciju interpolacijskog kubičnog splajna možemo izvesti na dva načina. U prvom, za nepoznate parametre koje treba odrediti uzimamo vrijednosti **prve** derivacije splajna u čvorovima. Tradicionalna oznaka za te parametre je $m_i := DS_3(x_i)$, za $i = 0, \ldots, N$. U drugom, za nepoznate parametre uzimamo vrijednosti **druge** derivacije splajna u čvorovima, koristeći globalnu neprekidnost D^2S_3, uz tradicionalnu oznaku $M_i := D^2S_3(x_i)$, za $i = 0, \ldots, N$. Napomenimo odmah da se ta dva algoritma dosta ravnopravno koriste u praksi, a za ocjenu greške trebamo i jednog i drugog, pa ćemo napraviti oba izvoda.

Prvi algoritam dobivamo primijenom Hermiteove interpolacije, ali ne zadajemo derivacije, već nepoznate derivacije m_i ostavljamo kao parametre, koje treba odrediti tako da se postigne globalna pripadnost splajna klasi $C^2[a,b]$.

Drugim riječima, tražimo da S_3 zadovoljava uvjete interpolacije $S_3(x_i) = f_i$, $DS_3(x_i) = m_i$, za $i = 0, \ldots, N$, gdje su f_i zadani, a m_i nepoznati. Uz standardne oznake iz prethodnog odjeljka, prema (7.5.3), S_3 možemo na svakom podintervalu napisati u obliku

$$\begin{aligned}
S_3(x) = {} & f_i\,(1-t)^2(1+2t) + f_{i+1}\,t^2(3-2t) \\
& + m_i\,h_i\,t(1-t)^2 - m_{i+1}\,h_i\,t^2(1-t),
\end{aligned} \tag{7.5.9}$$

gdje je $t = (x - x_i)/h_i$, za $x \in [x_i, x_{i+1}]$. Parametre m_i, m_{i+1} moramo odrediti tako da je druga derivacija D^2S_3 neprekidna u unutrašnjim čvorovima. Budući da je

$$D^2S_3(x) = \frac{f_{i+1} - f_i}{h_i^2}(6 - 12t) + \frac{m_i}{h_i}(-4 + 6t) + \frac{m_{i+1}}{h_i}(-2 + 6t),$$

slijedi

$$D^2S_3(x_i + 0) = 6\,\frac{f_{i+1} - f_i}{h_i^2} - \frac{4m_i}{h_i} - \frac{2m_{i+1}}{h_i},$$
$$D^2S_3(x_i - 0) = -6\,\frac{f_i - f_{i-1}}{h_{i-1}^2} + \frac{2m_{i-1}}{h_{i-1}} + \frac{4m_i}{h_{i-1}}.$$

Uz oznake

$$\mu_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \qquad \lambda_i = 1 - \mu_i, \qquad c_i = 3\Big(\mu_i\,\frac{f_{i+1} - f_i}{h_i} + \lambda_i\,\frac{f_i - f_{i-1}}{h_{i-1}}\Big),$$

uvjete neprekidnosti D^2S_3 u x_i, za $i = 1, \ldots, N-1$, možemo napisati u obliku

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = c_i, \quad i = 1, \ldots, N-1. \tag{7.5.10}$$

Dobili smo $N-1$ jednadžbi za $N+1$ nepoznanica m_i, pa nam fale još dvije jednadžbe. Naravno, uvjetima (7.5.10) treba dodati još neke rubne uvjete.

Za rubne uvjete $(R1)$, $(R2)$ i $(R3)$ dobivamo linearni sustav oblika

$$\begin{aligned} 2m_0 + \mu_0^*\, m_1 &= c_0^*, \\ \lambda_i\, m_{i-1} + 2m_i + \mu_i\, m_{i+1} &= c_i, \qquad i = 1, \ldots, N-1, \\ \lambda_N^*\, m_{N-1} + 2m_N &= c_N^*. \end{aligned} \tag{7.5.11}$$

Koeficijenti μ_0^*, c_0^*, λ_N^* i c_N^* određuju se ovisno o rubnim uvjetima. Za rubne uvjete $(R1)$ imamo

$$\mu_0^* = \lambda_N^* = 0, \quad c_0^* = 2Df(a), \quad c_N^* = 2Df(b),$$

a za rubne uvjete $(R3)$

$$\mu_0^* = \lambda_N^* = 1, \quad c_0^* = 3\,\frac{f_1 - f_0}{h_0} - \frac{h_0}{2}\,D^2f(a), \quad c_N^* = 3\,\frac{f_N - f_{N-1}}{h_{N-1}} + \frac{h_{N-1}}{2}\,D^2f(b).$$

Prirodni rubni uvjeti $(R2)$ su specijalni slučaj $(R3)$, uz $D^2f(a) = D^2f(b) = 0$.

Ako je f periodička funkcija, onda je $f_0 = f_N$ i $m_0 = m_N$ (periodički rubni uvjet na prvu derivaciju). Da bismo zapisali uvjet periodičnosti druge derivacije, možemo na periodički način produljiti mrežu, tako da dodamo još jedan čvor x_{N+1}, ali tako da je $x_{N+1} - x_N = x_1 = x_0$, tj. $h_N = h_0$. Zbog pretpostavke periodičnosti, moramo staviti $f_{N+1} = f_1$ i $m_{N+1} = m_1$. Na taj način, uvjet periodičnosti druge

derivacije postaje ekvivalentan uvjetu neprekidnosti druge derivacije u točki x_N, tj. jednadžbi oblika (7.5.10) za $i = N$. Kad iskoristimo sve pretpostavke

$$f_0 = f_N, \quad f_{N+1} = f_1, \quad m_0 = m_N, \quad m_{N+1} = m_1, \quad h_N = h_0,$$

dobivamo sustav od samo N jednadžbi

$$\begin{aligned} 2m_1 + \mu_1 m_2 + \lambda_1 m_N &= c_1, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} &= c_i, \qquad i = 2, \ldots, N-1, \\ \mu_N m_1 + \lambda_N m_{N-1} + 2m_N &= c_N. \end{aligned} \tag{7.5.12}$$

Uočite da smo jednadžbu $m_0 = m_N$ već iskoristili za eliminaciju nepoznanice m_0.

Ostaje odgovoriti na očito pitanje: da li dobiveni linearni sustavi imaju jedinstveno rješenje.

Teorem 7.5.3. *Postoji jedinstveni interpolacijski kubični splajn koji zadovoljava jedan od rubnih uvjeta* $(R1)$*–*$(R4)$*.*

Dokaz:

U svim navedenim slučajevima lako se vidi da je matrica linearnog sustava strogo dijagonalno dominantna, što povlači regularnost. Naime, svi dijagonalni elementi su jednaki 2, a zbroj izvandijagonalnih elemenata je najviše $\lambda_i + \mu_i = 1$, (uz dogovor $\lambda_N^* = \lambda_N$ i $\mu_0^* = \mu_0$). ■

Algoritam 7.5.1. (Interpolacijski kubični splajn)

(1) Riješi linearni sustav (7.5.11) ili (7.5.12);
(2) Binarnim pretraživanjem nađi indeks i tako da vijedi $x \in [x_i, x_{i+1})$;
(3) Hornerovom shemom (7.5.5) izračunaj $S_3(x)$.

Primijetimo da je za rješavanje sustava potrebno samo $O(N)$ operacija, obzirom na specijalnu vrpčastu strukturu matrice. Također, matrica ne ovisi o vrijednostima funkcije koja se interpolira, pa se korak (1) u Algoritmu 7.5.1. sastoji od LR faktorizacije matrice, koju treba izračunati samo jednom.

Za računanje vrijednosti $S_3(x)$ obično se koriste formule (7.5.4)–(7.5.5). Ako je potrebno računati splajn u mnogo točaka (recimo, u svrhu brze reprodukcije grafa splajna), možemo napisati **algoritam konverzije**, tj. naći vezu između definicionog oblika splajna (v. definiciju 7.5.2.) i oblika danog formulama (7.5.4)–(7.5.5). Definiciona reprezentacija splajna kao kubične funkcije na svakom podintervalu subdivizije zove se ponekad i **po dijelovima polinomna** reprezentacija, ili skraćeno PP–reprezentacija.

Zadatak 7.5.8. *Kolika je točno ušteda u broju aritmetičkih operacija potrebnih za računanje $S_3(x)$ pri prijelazu na PP–reprezentaciju? Još "brži" oblik reprezentacije*

je standardni kubni polinom $S_3(x) = b_{i0} + b_{i1}x + b_{i2}x^2 + b_{i3}x^3$*, za svaki* $x \in [x_i, x_{i+1}]$. *Njega* **ne treba** *koristiti. Zašto?*

Kao što smo već rekli, u nekim slučajevima ugodnija je druga reprezentacija interpolacijskog kubičnog splajna, u kojoj se, umjesto m_i, kao nepoznanice javljaju $M_i := D^2S(x_i)$, za $i = 0, \ldots, N$. Zbog popularnosti i česte implementacije izvedimo ukratko i ovu reprezentaciju.

Na svakom podintervalu $[x_i, x_{i+1}]$ kubični splajn S_3 je kubični polinom kojeg određujemo iz uvjeta interpolacije funkcije i **druge** derivacije u rubovima

$$S_3(x_i) = f_i, \quad S_3(x_{i+1}) = f_{i+1}, \quad D^2S_3(x_i) = M_i, \quad D^2S_3(x_{i+1}) = M_{i+1}.$$

Ovaj sustav jednadžbi ima jedinstveno rješenje (dokažite to), odakle onda možemo izračunati koeficijente kubnog polinoma. Međutim, traženu reprezentaciju možemo jednostavno i "pogoditi", ako $S_3(x)$ na $[x_i, x_{i+1}]$ napišemo kao linearnu interpolaciju funkcijskih vrijednosti plus neka korekcija. Odmah se vidi da tražena korekcija ima oblik linearne interpolacije druge derivacije puta neki kvadratni faktor koji se poništava u rubovima. Dobivamo oblik

$$S_3(x) = f_i\,(1-t) + f_{i+1}\,t - \frac{h_i^2}{6}\,t(1-t)\,[M_i\,(2-t) + M_{i+1}\,(1+t)],$$

gdje je opet $t = (x - x_i)/h_i$, za $x \in [x_i, x_{i+1}]$ i $i = 0, \ldots, N-1$. Odavde lako izlazi

$$\begin{aligned} DS_3(x) &= \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{6}\,[M_i\,(2 - 6t + 3t^2) + M_{i+1}\,(1 - 3t^2)], \\ D^2S_3(x) &= M_i\,(1-t) + M_{i+1}\,t, \\ D^3S_3(x) &= \frac{M_{i+1} - M_i}{h_i}. \end{aligned}$$

Interpolacija druge derivacije u čvorovima ne garantira da je i prva derivacija neprekidna. To treba dodatno zahtijevati. Kako je

$$\begin{aligned} DS_3(x_i + 0) &= \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{6}\,(2M_i + M_{i+1}), \\ DS_3(x_i - 0) &= \frac{f_i - f_{i-1}}{h_{i-1}} + \frac{h_{i-1}}{6}\,(M_{i-1} + 2M_i), \end{aligned}$$

iz uvjeta neprekidnosti prve derivacije u unutrašnjim čvorovima dobivamo $N-1$ jednadžbi traženog linearnog sustava

$$\mu_i\,M_{i-1} + 2M_i + \lambda_i\,M_i = \frac{6}{h_{i-1} + h_i}\left(\frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}\right), \tag{7.5.13}$$

za $i = 1, \ldots, N-1$, gdje je, kao i prije, $\mu_i = h_{i-1}/(h_{i-1} + h_i)$ i $\lambda_i = 1 - \mu_i$.

Zadatak 7.5.9. *Napišite nedostajuće jednadžbe za rubne uvjete, i pokažite da je matrica sustava strogo dijagonalno dominantna.*

Na kraju, primijetimo da je algoritam za računanje vrijednosti $S_3(x)$ vrlo sličan ranijem, s tim što treba primijeniti malo drugačiju Hornerovu shemu (ekvivalent formula (7.5.4)–(7.5.5) za algoritam 7.5.1.):

$$S_3(x) = f_i + t\,\{(f_{i+1} - f_i) - (x_{i+1} - x)\,[(x_{i+1} - x + h_i)\widetilde{M}_i + (h_i + x - x_i)\widetilde{M}_{i+1}]\},$$

gdje je $\widetilde{M}_i := M_i/6$.

Ocjena greške za potpuni kubični splajn je teži problem nego za Hermiteov kubični splajn, budući da su koeficijenti zadani implicitno kao rješenje jednog linearnog sustava.

Teorem 7.5.4. *Neka je S_3 interpolacijski kubični splajn za funkciju f koji zadovoljava jedan od rubnih uvjeta $(R1)$–$(R4)$ u (7.5.8). Tada vrijedi*

$$\|D^r S_3(x) - D^r f(x)\|_\infty \le C_r\,\bar{h}^{e_f - r}\,M_f, \quad r = 0, 1, 2, 3,$$

gdje su C_r konstante (ovisne o r), e_f osnovni eksponenti i M_f "mjere" funkcije (e_f i M_f ovise samo o klasi funkcije, ne i o r), dani sljedećom tablicom:

Klasa funkcije	M_f	e_f	C_0	C_1	C_2	C_3
$C^1[a,b]$	$\omega(Df)$	1	$\frac{9}{8}$	4		
$L^2_\infty[a,b]$	$\|D^2 f\|_\infty$	2	$\frac{13}{48}$	0.86229		
$C^2[a,b]$	$\omega(D^2 f)$	2	$\frac{19}{96}$	$\frac{2}{3}$	4	
$L^3_\infty[a,b]$	$\|D^3 f\|_\infty$	3	$\frac{41}{864}$	$\frac{4}{27}$	$\frac{1}{2} + \frac{4\sqrt{3}}{9}$	
$C^2[a,b] \bigcap_i C^3[x_i, x_{i+1}]$	$\omega(D^3 f)$	3	$\frac{41}{1728}$	$\frac{2}{27}$	$\frac{1}{2} + \frac{2\sqrt{3}}{9}$	$1 + \frac{4\sqrt{3}}{9}\,\beta$
$C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$	$\|D^4 f\|_\infty$	4	$\frac{5}{384}$	$\frac{1}{24}$	$\frac{3}{8}$	$\frac{1}{2}\left(\frac{1}{\beta} + \beta\right)$

s tim da je

$$\beta := \frac{\max_i h_i}{\min_i h_i}$$

mjera "neuniformnosti" mreže (u zadnjem stupcu tablice).

Mjesta u tablici koja nisu popunjena znače da **ne postoje** odgovarajuće ocjene. Napomenimo, također, da rijetko korištene ocjene koje odgovaraju još nižoj glatkoći funkcije f, na primjer, $f \in C[a,b]$ ili $f \in L^1_\infty[a,b]$ nisu navedene, iako se mogu izvesti (dokaz nije trivijalan). Osim toga, nije poznato da li su sve konstante optimalne, iako se to može pokazati u nekim važnim slučajevima (na primjer, u zadnjem redu tablice, koji podrazumijeva najveću glatkoću, sve su konstante najbolje moguće).

Dokaz:

Dokažimo neke od ocjena u teoremu 7.5.4. (preostale pokašajte dokazati sami).

Neka je H Hermitski interpolacijski kubični splajn i $S := S_3$ interpolacijski kubični splajn. Tada grešku možemo napisati kao

$$E(x) := S(x) - f(x) = [H(x) - f(x)] + [S(x) - H(x)].$$

Oba interpolacijska splajna $S(x)$ i $H(x)$ možemo reprezentirati preko Hermiteove baze na svakom intervalu $[x_i, x_{i+1}]$ (v. (7.5.9), (7.5.3)), pa oduzimanjem tih reprezentacija slijedi

$$\begin{aligned} S(x) - f(x) = [H(x) - f(x)] + h_i\,[t(1-t)^2\,(m_i - Df(x_i)) \\ - (1-t)t^2\,(m_{i+1} - Df(x_{i+1}))]. \end{aligned}$$

Odavde je

$$|S(x) - f(x)| \le |H(x) - f(x)| + h_i\,t(1-t)\,\max_i |m_i - Df(x_i)|. \tag{7.5.14}$$

Za derivaciju imamo

$$\begin{aligned} DS(x) - Df(x) = [DH(x) - Df(x)] + [(1-t)(1-3t)\,(m_i - Df(x_i)) \\ - t(2-3t)\,(m_{i+1} - Df(x_{i+1}))], \end{aligned}$$

pa je stoga

$$\begin{aligned} |DS(x) - Df(x)| \le |DH(x) - Df(x)| \\ + [(1-t)|1-3t| + t|2-3t|]\,\max_i |m_i - Df(x_i)|. \end{aligned} \tag{7.5.15}$$

Ocjene za $|H(x) - f(x)|$ izveli smo u teoremu 7.5.4., a ocjene za $|DH(x) - Df(x)|$ mogu se izvesti na sličan način (v. zadatke 7.5.5. i 7.5.6.). Ostaje dakle ocijeniti drugi član na desnoj strani u (7.5.14) i (7.5.15).

Za drugu derivaciju znamo da je

$$D^2S(x) = M_i\,(1-t) + M_{i+1}\,t,$$

pa zaključujemo da je

$$\begin{aligned} D^2S(x) - D^2f(x) = (1-t)\,(M_i - D^2f(x_i)) + t\,(M_{i+1} - D^2f(x_{i+1}) \\ + (1-t)\,D^2f(x_i) + t\,D^2f(x_{i+1}) - D^2f(x). \end{aligned}$$

Ali, kako je $(1-t)D^2f(x_i)+tD^2f(x_{i+1})-D^2f(x)$ pogreška kod interpolacije funkcije D^2f linearnim splajnom S_1 (teorem 7.5.1.), možemo ju i ovako ocijeniti

$$|D^2S(x) - D^2f(x)| \le |S_1(x) - D^2f(x)| + \max_i |M_i - D^2f(x_i)|. \tag{7.5.16}$$

Slično je i za treću derivaciju

$$|D^3S(x) - D^3f(x)| \le |DS_1(x) - D^3f(x)| + \frac{2}{\min_i h_i} \max_i |M_i - D^2f(x_i)|. \tag{7.5.17}$$

Ako pogledamo nejednakosti (7.5.14), (7.5.15), (7.5.16) i (7.5.17), vidimo da preostaje ocijeniti $\max_i |m_i - Df(x_i)|$ i $\max_i |M_i - D^2f(x_i)|$. Ove ocjene, kao i sve druge, ovise o klasi funkcija.

Tvrdimo da vrijedi

$$\max_i |m_i - Df(x_i)| \le q_f,$$

gdje je q_f dan sljedećom tablicom za 6 karakterističnih klasa funkcija:

Klasa funkcije	q_f
$C^1[a,b]$	$3\,\omega(Df)$
$L^2_\infty[a,b]$	$\frac{5}{6}\,\bar{h}\,\|D^2f\|_\infty$
$C^2[a,b]$	$\frac{2}{3}\,\bar{h}\,\omega(D^2f)$
$L^3_\infty[a,b]$	$\frac{4}{27}\,\bar{h}^2\,\|D^3f\|_\infty$
$C^2[a,b]\bigcap_i C^3[x_i,x_{i+1}]$	$\frac{2}{27}\,\bar{h}^2\,\omega(D^3f)$
$C^2[a,b]\bigcap_i L^4_\infty[x_i,x_{i+1}]$	$\frac{1}{24}\,\bar{h}^3\,\|D^4f\|_\infty$

Da dokažemo ovu tablicu, pretpostavimo rubne uvjete $(R1)$ na derivaciju. Uvedemo li u linearnom sustavu (7.5.11) nove nepoznanice $q_i := m_i - Df(x_i)$, dobijemo sustav

$$\begin{aligned} q_0 &= 0, \\ \lambda_i\, q_{i-1} + 2q_i + \mu_i\, q_{i+1} &= \tilde{c}_i, \qquad i = 1,\dots,N-1, \\ q_N &= 0, \end{aligned}$$

gdje su desne strane

$$\begin{aligned} \tilde{c}_i = 3\mu_i\,\frac{f_{i+1}-f_i}{h_i} + 3\lambda_i\,\frac{f_i-f_{i-1}}{h_{i-1}} \\ - \lambda_i\,Df(x_{i-1}) - 2Df(x_i) - \mu_i\,Df(x_{i+1}). \end{aligned} \tag{7.5.18}$$

Da bismo ocijenili $|q_i|$, zapišimo ovaj sustav u matričnom obliku $Aq = \tilde{c}$, ili $q = A^{-1}\tilde{c}$. Vidimo odmah da je $A = 2I+B$, gdje je B matrica koja sadrži samo izvandijagonalne elemente λ_i i μ_i. Zbog $\lambda_i + \mu_i \leq 1$ (jednakost vrijedi u svim jednadžbama, osim prve i zadnje), slijedi $\|B\|_\infty \leq 1$. Sada nije teško ocijeniti $\|A^{-1}\|_\infty$

$$A = 2\Big(I + \frac{1}{2}\,B\Big) \implies \|A^{-1}\|_\infty \leq \frac{1}{2}\Big(1 - \frac{1}{2}\,\|B\|_\infty\Big)^{-1} \leq 1.$$

Na kraju, iz $q = A^{-1}\tilde{c}$ slijedi

$$|q_i| \leq \|q\|_\infty \leq \|A^{-1}\|_\infty \|\tilde{c}\|_\infty = \max_i |\tilde{c}_i|.$$

Drugim riječima, da bismo dokazali ocjene iz tablice za q_f, dovoljno je ocijeniti $|\tilde{c}_i|$.

Pretpostavimo da je $f \in C^1[a,b]$ i iskoristimo Lagrangeov teorem o srednjoj vrijednosti za prva dva člana u izrazu (7.5.18) za $\tilde{c}_i$. Tada je $\lambda_i + \mu_i = 1$, pa je

$$\begin{aligned}\tilde{c}_i &= 3\mu_i\,Df(\xi_{i,i+1}) + 3\lambda_i\,Df(\xi_{i-1,i}) - \lambda_i\,Df(x_{i-1}) - 2Df(x_i) - \mu_i\,Df(x_{i+1})\\ &= \lambda_i\,[Df(\xi_{i-1,i}) - Df(x_{i-1})] + 2\lambda_i\,[Df(\xi_{i-1,i}) - Df(x_i)]\\ &\quad + \mu_i\,[Df(\xi_{i,i+1}) - Df(x_{i+1})] + 2\mu_i\,[Df(\xi_{i,i+1}) - Df(x_i)],\end{aligned}$$

odakle slijedi

$$|\tilde{c}_i| \leq 3(\lambda_i + \mu_i)\,\omega(Df) = 3\,\omega(Df),$$

čime smo dokazali prvu ocjenu u tablici za q_f.

Ako je $f \in C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$, u izrazu (7.5.18) za $\tilde{c}_i$ možemo razviti f_{i-1}, $Df(x_{i-1})$, f_{i+1}, $Df(x_{i+1})$ u Taylorov red oko x_i, koristeći integralni oblik ostatka. Napomenimo da nam nije potrebna neprekidnost treće derivacije. U tom slučaju imamo dakle

$$\begin{aligned}\tilde{c}_i = 3\mu_i \Bigg\{ &Df(x_i) + \frac{h_i}{2}\,D^2f(x_i) + \frac{h_i^2}{6}\,D^3f(x_i+0)\\ &\qquad + \frac{1}{6h_i}\int_{x_i}^{x_{i+1}} (x_{i+1}-v)^3\,D^4f(v)\,dv\Bigg\}\\ + 3\lambda_i \Bigg\{ &Df(x_i) - \frac{h_{i-1}}{2}\,D^2f(x_i) + \frac{h_{i-1}^2}{6}\,D^3f(x_i-0)\\ &\qquad - \frac{1}{6h_{i-1}}\int_{x_i}^{x_{i-1}} (x_{i-1}-v)^3\,D^4f(v)\,dv\Bigg\}\\ - \mu_i \Bigg\{ &Df(x_i) + h_i\,D^2f(x_i) + \frac{h_i^2}{2}\,D^3f(x_i+0)\\ &\qquad + \frac{1}{2}\int_{x_i}^{x_{i+1}} (x_{i+1}-v)^2\,D^4f(v)\,dv\Bigg\}\end{aligned}$$

$$
\begin{aligned}
&- 2Df(x_i) \\
&- \lambda_i \Big\{ Df(x_i) - h_{i-1}\, D^2 f(x_i) + \frac{h_{i-1}^2}{2}\, D^3 f(x_i - 0) \\
&\qquad\qquad + \frac{1}{2} \int_{x_i}^{x_{i-1}} (x_{i-1} - v)^2\, D^4 f(v)\, dv \Big\}.
\end{aligned}
$$

Članovi s $Df(x_i)$, $D^2 f(x_i)$, $D^3 f(x_i + 0)$ i $D^3 f(x_i - 0)$ se skrate, pa ostaje samo

$$
\begin{aligned}
\tilde{c}_i = \frac{\mu_i}{2} &\int_{x_i}^{x_{i+1}} \left[\frac{(x_{i+1} - v)^3}{h_i} - (x_{i+1} - v)^2 \right] D^4 f(v)\, dv \\
&+ \frac{\lambda_i}{2} \int_{x_i}^{x_{i-1}} \left[-\frac{(x_{i-1} - v)^3}{h_{i-1}} - (x_{i-1} - v)^2 \right] D^4 f(v)\, dv.
\end{aligned}
$$

Zamijenimo li varijable supstitucijom $\tau h_i := v - x_i$ u prvom integralu, odnosno, $\tau h_{i-1} := v - x_{i-1}$ u drugom integralu, dobivamo

$$
\begin{aligned}
\tilde{c}_i = -\frac{\mu_i h_i^3}{2} &\int_0^1 \tau(1 - \tau)^2\, D^4 f(x_i + \tau h_i)\, d\tau \\
&+ \frac{\lambda_i h_{i-1}^3}{2} \int_0^1 \tau^2 (1 - \tau)\, D^4 f(x_{i-1} + \tau h_{i-1})\, d\tau.
\end{aligned}
$$

Odavde lagano ocijenimo

$$
\begin{aligned}
|\tilde{c}_i| &\le \frac{1}{2} \|D^4 f\|_\infty \left\{ \mu_i h_i^3 \int_0^1 \tau(1 - \tau)^2\, d\tau + \lambda_i h_{i-1}^3 \int_0^1 \tau^2 (1 - \tau)\, d\tau \right\} \\
&= \frac{1}{24} \|D^4 f\|_\infty (\mu_i h_i^3 + \lambda_i h_{i-1}^3).
\end{aligned}
$$

Uvrštavanjem μ_i, λ_i (v. 7.5.10) dobivamo

$$
|\tilde{c}_i| \le \frac{h_i h_{i-1}}{24}\, \frac{h_i^2 + h_{i-1}^2}{h_i + h_{i-1}}\, \|D^4 f\|_\infty.
$$

Na kraju, kako je

$$
\frac{h_i^2 + h_{i-1}^2}{h_i + h_{i-1}} \le \max\{h_i, h_{i-1}\},
$$

dolazimo do zadnje ocjene u tablici za q_f

$$
|\tilde{c}_i| \le \frac{1}{24}\, \bar{h}^3\, \|D^4 f\|_\infty.
$$

Upotrebom Taylorove formule, teorema o srednjoj vrijednosti i leme 7.5.1., na već poznati način, dokazuju se i ostale ocjene u tablici. Napomenimo još, da je sličnu analizu potrebno napraviti i za druge tipove rubnih uvjeta. Pokazuje se da rezultati i tehnika dokaza ne ovise mnogo o tipu rubnih uvjeta. To, naravno, vrijedi samo uz pretpostavku da funkcija f zadovoljava iste rubne uvjete kao i splajn, ako rubni uvjet ne ovisi o funkciji (na primjer, $(R2)$ ili $(R4)$). U protivnom, dobivamo slabije ocjene.

Nadalje, za ocjenu druge i treće derivacije, moramo naći ocjene oblika

$$\max_i |M_i - D^2 f(x_i)| \leq Q_f.$$

I u ovom slučaju imamo tablicu s 4 ocjene, u ovisnosti o klasi funkcije:

Klasa funkcije	Q_f
$C^2[a,b]$	$3\,\omega(D^2 f)$
$L^3_\infty[a,b]$	$\dfrac{4\sqrt{3}}{9}\,\bar{h}\,\|D^3 f\|_\infty$
$C^2[a,b] \bigcap_i C^3[x_i, x_{i+1}]$	$\dfrac{2\sqrt{3}}{9}\,\bar{h}\,\omega(D^3 f)$
$C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$	$\dfrac{1}{4}\,\bar{h}^2\,\|D^4 f\|_\infty$

Tehnika dokaza ove tablice je dosta slična onoj za prethodnu tablicu, s time da se oslanja na linearni sustav (7.5.13), pa ocjene ostavljamo kao zadatak.

Da bismo na kraju dokazali ovaj teorem, ograničimo se na "najglatkiju" klasu funkcija $C^2[a,b] \bigcap_i L^4_\infty[x_i, x_{i+1}]$; tehnika dokaza potpuno je ista i za sve druge klase. Ključna je ocjena (7.5.14):

$$|S(x) - f(x)| \leq |H(x) - f(x)| + h_i\, t(1-t)\, \max_i |m_i - Df(x_i)|.$$

Prvi dio čini greška kod interpolacije Hermiteovim splajnom, za koju, prema (7.5.7), znamo da vrijedi

$$|H(x) - f(x)| \leq \frac{t^2(1-t)^2}{4!}\, h_i^4\, \|D^4 f\|_\infty, \quad t = \frac{x - x_i}{h_i} \in [0,1],$$

a drugi dio pročitamo u tablici za $\max_i |m_i - Df(x_i)|$. Ukupno je dakle

$$|S(x) - f(x)| \leq \frac{1}{24}\, t(1-t)\,[1 + t(1-t)]\, \max_i h_i^4\, \|D^4 f\|_\infty \leq \frac{5}{384}\, \bar{h}^4\, \|D^4 f\|_\infty.$$

Zanimljivo je da ova ocjena samo 5 puta veća od ocjene za Hermiteov interpolacijski splajn, koji zahtijeva poznate derivacije funkcije f u **svim** čvorovima interpolacije, a ovdje ih koristimo samo na rubu (uz rubne uvjete $(R1)$). ■

Literatura

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, Third edition, SIAM, Philadelpia, 1999.

[2] K. E. Atkinson, *An Introduction to Numerical Analysis (2^{nd} edition)*, John Wiley & Sons, New York, 1989.

[3] W. Gautschi, *Numerical Analysis (An Introduction)*, Birkhäuser, Boston, 1997.

[4] D. Goldberg, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys, vol. 23, no. 1, March 1991.

[5] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[6] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*, SIAM, Philadelpia, 2001.

[7] A. Ralston, P. Rabinowitz, *A First Course in Numerical Analysis*, McGraw–Hill, Singapore, 1978.

[8] G. W. Stewart, J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, 1990.

[9] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963. (Also published by Prentice–Hall, Englewood Cliffs, NJ, USA. Reprinted by Dover, New-York, 1994, ISBN 0–486–67999–5.)