SVEUČILIŠTE U ZAGREBU
PMF – MATEMATIČKI ODJEL

Mladen Rogina, Sanja Singer i Saša Singer

Numerička analiza

Predavanja i vježbe

Zagreb, 2003.

Sadržaj

1. Mogućnosti današnjih računala 1

1.1. Efikasni i neefikasni algoritmi 2

2. Greške 6

2.1. Računalo je izbacilo... Neće mi prihvatiti! 6

2.2. Mjere za grešku 7

2.3. Greške modela 8

2.4. Greške u ulaznim podacima 8

2.5. Greške metoda za rješavanje problema 9

2.6. Greške aritmetike računala 12

2.7. Propagiranje grešaka u aritmetičkim operacijama 14

2.8. Primjeri iz života 18

3. Vektorske i matrične norme 23

3.1. Vektorske norme 23

3.2. Matrične norme 25

4. Stabilnost problema i algoritama 30

4.1. Jednostavni model problema 30

4.2. Uvjetovanost problema 34

5. Rješavanje linearnih sustava 43

5.1. Kako se sustavi rješavaju u praksi 43

5.2. Gaussove eliminacije 44

5.3. LR faktorizacija 50

5.4. Teorija perturbacije linearnih sustava 54
5.5. Greške zaokruživanja kod rješavanja trokutastog linearnog sustava 60
5.6. Greška unaprijed za trokutasti sustav 69
5.7. Greške zaokruživanja za LR faktorizaciju 74
5.8. Pivotni rast 79
5.9. Posebni tipovi matrica 80

6. Faktorizacija Choleskog i QR faktorizacija 87
6.1. Faktorizacija Choleskog 87
6.2. Analiza greške za faktorizaciju Choleskog 92
6.3. QR faktorizacija 97
6.3.1. Givensove rotacije 100
6.3.2. Householderovi reflektori 102

7. Iterativne metode za rješavanje linearnih sustava 104
7.1. Općenito o iterativnim metodama 104
7.2. Jacobijeva metoda 109
7.3. Gauss–Seidelova metoda 110
7.4. JOR metoda (Jacobi overrelaxation) 111
7.5. SOR metoda (Successive overrelaxation) 113
7.6. Konvergencija Jacobijeve i Gauss–Seidelove metode 114
7.7. Konvergencija JOR i SOR metode 117
7.8. Optimalni izbor relaksacijskog parametra 123
7.9. Primjeri — akademski i praktični 136

8. Izvrednjavanje funkcija 149
8.1. Hornerova shema 151
8.1.1. Računanje vrijednosti polinoma u točki 152
8.1.2. Hornerova shema je optimalan algoritam 153
8.1.3. Stabilnost Hornerove sheme 155
8.1.4. Dijeljenje polinoma linearnim faktorom oblika $x - x_0$ . . . 155
8.1.5. Potpuna Hornerova shema 157

8.1.6. "Hornerova shema" za interpolacijske polinome 159
8.1.7. Hornerova shema za kompleksne argumente realnog polinoma 159
8.1.8. Računanje parcijalnih derivacija kompleksnog polinoma . . 163
8.2. Generalizirana Hornerova shema 165
8.2.1. Izvrednjavanje rekurzivno zadanih funkcija 167
8.2.2. Izvrednjavanje Fourierovih redova 171
8.2.3. Klasični ortogonalni polinomi 175
8.3. Stabilnost rekurzija i generalizirane Hornerove sheme 181
8.4. Besselove funkcije i Millerov algoritam 182
8.4.1. Opća forma Millerovog algoritma 183
8.4.2. Izvrednjavanje Besselovih funkcija 184
8.5. Asimptotski razvoj 191
8.6. Verižni razlomci i racionalne aproksimacije 200
8.6.1. Brojevni verižni razlomci 201
8.6.2. Uzlazni algoritam za izvrednjavanje brojevnih verižnih razlomaka 202
8.6.3. Eulerova forma verižnih razlomaka i neki teoremi konvergencije 206
8.6.4. Silazni algoritam za izvrednjavanje brojevnih verižnih razlomaka 210
8.6.5. Funkcijski verižni razlomci 210

9. Rješavanje nelinearnih jednadžbi 214
9.1. Općenito o iterativnim metodama 214
9.2. Metoda raspolavljanja (bisekcije) 215
9.3. Regula falsi (metoda pogrešnog položaja) 219
9.4. Metoda sekante 221
9.5. Metoda tangente (Newtonova metoda) 225
9.6. Metoda jednostavne iteracije 231
9.7. Newtonova metoda za višestruke nultočke 235
9.8. Hibridna Brent–Dekkerova metoda 237
9.9. Primjeri 238

10. Aproksimacija i interpolacija 240
10.1. Opći problem aproksimacije 240
10.1.1. Linearne aproksimacione funkcije 241
10.1.2. Nelinearne aproksimacione funkcije 242
10.1.3. Kriteriji aproksimacije 242
10.2. Interpolacija polinomima 245
10.2.1. Egzistencija i jedinstvenost interpolacionog polinoma . . . 245
10.2.2. Potrebni algoritmi 247
10.2.3. Lagrangeov oblik interpolacionog polinoma 252
10.2.4. Ocjena greške interpolacionog polinoma 254
10.2.5. Newtonov oblik interpolacionog polinoma 255
10.2.6. Koliko je dobar interpolacioni polinom? 259
10.2.7. Konvergencija interpolacionih polinoma 297
10.2.8. Hermiteova i druge interpolacije polinomima 298
10.3. Interpolacija po dijelovima polinomima 303
10.3.1. Po dijelovima linearna interpolacija 304
10.3.2. Po dijelovima kubna interpolacija 306
10.3.3. Po dijelovima kubna Hermiteova interpolacija 309
10.3.4. Numeričko deriviranje 311
10.3.5. Po dijelovima kubna kvazihermiteova interpolacija 315
10.3.6. Kubična splajn interpolacija 319
10.4. Interpolacija polinomnim splajnovima — za matematičare 327
10.4.1. Linearni splajn 329
10.4.2. Hermiteov kubični splajn 334
10.4.3. Potpuni kubični splajn 339
10.5. Diskretna metoda najmanjih kvadrata 350
10.5.1. Linearni problemi i linearizacija 350
10.5.2. Matrična formulacija linearnog problema najmanjih kvadrata 356
10.5.3. Karakterizacija rješenja 356
10.5.4. Numeričko rješavanje problema najmanjih kvadrata 359

10.6. Opći oblik metode najmanjih kvadrata 371
10.6.1. Težinski skalarni produkti 371
10.7. Familije ortogonalnih funkcija 371
10.8. Neka svojstva ortogonalnih polinoma 371
10.9. Trigonometrijske funkcije 376
10.9.1. Diskretna ortogonalnost trigonometrijskih funkcija 378
10.10. Minimaks aproksimacija 385
10.10.1. Remesov algoritam 391
10.11. Skoro minimaks aproksimacije 392
10.12. Interpolacija u Čebiševljevim točkama 395
10.13. Čebiševljeva ekonomizacija 396
10.14. Diskretne ortogonalnosti polinoma T_n 399
10.15. Thieleova racionalna interpolacija 403

11. Numerička integracija 408
11.1. Općenito o integracionim formulama 408
11.2. Newton–Cotesove formule 410
11.2.1. Trapezna formula 410
11.2.2. Simpsonova formula 416
11.2.3. Produljene formule 421
11.2.4. Primjeri 424
11.2.5. Midpoint formula 427
11.3. Rombergov algoritam 428
11.4. Težinske integracione formule 437
11.5. Gaussove integracione formule 440
11.5.1. Gauss–Legendreove integracione formule 446
11.5.2. Druge Gaussove integracione formule 457

12. Metode za rješavanje običnih diferencijalnih jednadžbi 465
12.1. Uvod 465
12.2. Runge–Kutta metode 465

12.2.1. Varijabilni korak za Runge–Kutta metode 468
12.2.2. Runge–Kutta metode za sustave jednadžbi 468
12.3. Višekoračne metode 469
12.4. Krute (stiff) diferencijalne jednadžbe 470

13. Rubni problem za obične diferencijalne jednadžbe 471
13.1. Egizstencija i jedinstvenost rješenja 471
13.2. Metoda gađanja za linearne diferencijalne jednadžbe 2. reda . . . 472
13.3. Nelinearna metoda gađanja 473
13.4. Metoda konačnih razlika 473

14. Rješavanje parcijalnih diferencijalnih jednadžbi 475
14.1. Paraboličke PDJ — Provođenje topline 475
14.1.1. Eksplicitna metoda 475
14.1.2. Crank–Nicolsonova metoda 476
14.2. Hiperboličke PDJ — Valna jednadžba 477
14.2.1. Eksplicitna metoda 477

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.

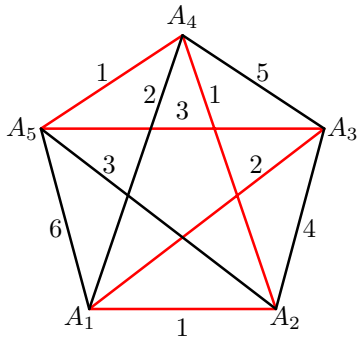
Možda je potrebno prokomentirati samo posljednja tri zahtjeva. Uobičajeno je da algoritmi imaju ulazne podatke, ali to nije nužno. Na primjer, ako želimo izračunati konstantu π nekim algoritmom, takav program vjerojatno neće zahtijevati ulazne podatke. Sasvim je obrnuta situacija s izlaznim podacima. 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*

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

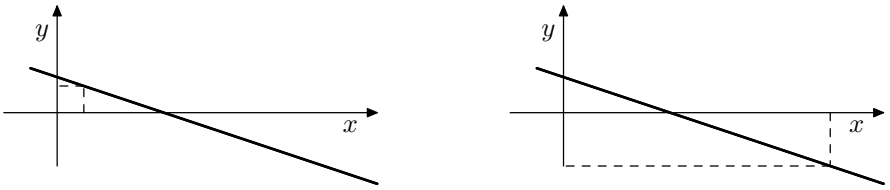
i

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

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)*

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

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

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.

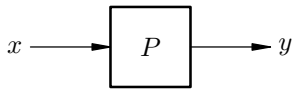
4.1. Jednostavni model problema

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.

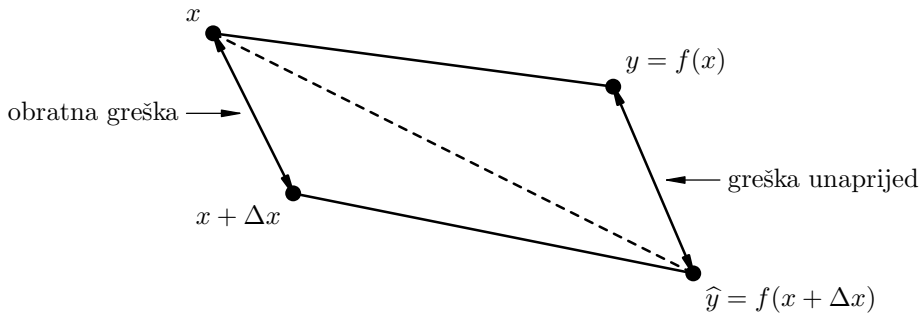
Ilustrirajmo to na najjednostavnijem primjeru funkcije $f : \mathbb{R} \to \mathbb{R}$. Pretpostavimo da je $\widehat{y}$ aproksimacija rješenja nekog problema $y = f(x)$, izračunata nekim algoritmom u realnoj aritmetici računala (tzv. aritmetika pomičnog zareza, ili engl. floating point arithmetic) 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.

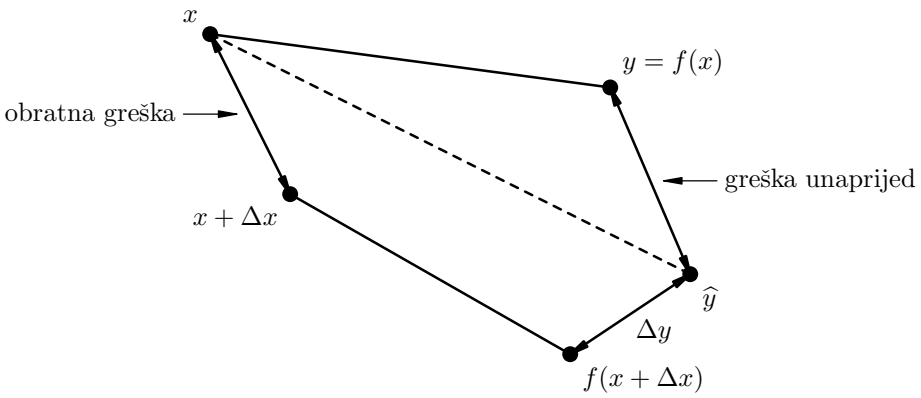
Nažalost, u nekim slučajevima nije moguće postići pravu obratnu stabilnost u relativnom smislu. Odmah je očito da probleme možemo očekivati u blizini nule.

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, ali i modernije perturbacije po komponentama. Pitanje na koje odgovaraju takve teorije perturbacije je koliko se (po normi) rješenje linearnog sustava (5.1.1) promijeni ako se po normi/po komponentama 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?

Sve prethodne ocjene su, zapravo, bile ocjene greške unaprijed. Kako izgleda ocjena greške unazad?

Teorem 5.4.2. (Rigal i Gaches) *Greška unatrag po normi definira se kao*

$$\eta_{E,f}(x+\Delta x) := \min\{\varepsilon \;\Big|\; (A+\Delta A)\,(x+\Delta x) = b+\Delta b,\ \|\Delta A\| \leq \varepsilon\|E\|,\ \|\Delta b\| \leq \varepsilon\|f\|\}.$$

Greška $\eta_{E,f}(x + \Delta x)$ može se dostići i jednaka je

$$\eta_{E,f}(x + \Delta x) = \frac{\|r\|}{\|E\|\,\|x + \Delta x\| + \|f\|}, \tag{5.4.6}$$

pri čemu je $r = b - A\,(x + \Delta x)$.

Dokaz:

Dokaz relacije (5.4.6) se provodi u dva koraka. Prvi je pokazati da vrijedi

$$\eta_{E,f}(x + \Delta x) \geq \frac{\|r\|}{\|E\|\,\|x + \Delta x\| + \|f\|},$$

a drugi da postoji takva perturbacija da se ocjena dostigne.

Premjestimo li članove lijeve i desne strane jednakosti $(A + \Delta A)\,(x + \Delta x) = b + \Delta b$, dobivamo

$$r = \Delta A\,(x + \Delta x) - \Delta b.$$

Primjenom norme s obje strane te uvrštavanjem ocjena, dobivamo

$$\begin{aligned}\|r\| &= \|\Delta A\,(x + \Delta x) - \Delta b\| \leq \|\Delta A\|\,\|x + \Delta x\| + \|\Delta b\| \\ &\leq \varepsilon(\|E\|\,\|x + \Delta x\| + \|f\|).\end{aligned}$$

Dijeljenjem lijeve i desne strane s $\|E\|\,\|x + \Delta x\| + \|f\|$, dobivamo traženu relaciju. Ostaje još pokazati da se donja ograda može dostići.

Donja se ograda dostiže za

$$\Delta A_{\min} = \frac{\|E\|\,\|x+\Delta x\|}{\|E\|\,\|x+\Delta x\| + \|f\|}\, r z^T, \quad \Delta b_{\min} = -\frac{\|f\|}{\|E\|\,\|x+\Delta x\| + \|f\|}\, r,$$

pri čemu je vektor z dualan vektoru $x+\Delta x$, tj. $z^T(x+\Delta x) = 1$. ■

Vrijednost $r := b - A\,(x+\Delta x)$ zovemo (egzaktni) rezidual približnog rješenja $x+\Delta x$. Naravno, rezidual pravog rješenja x je nula. Intuitivno očekujemo da je vektor koji daje mali rezidual (recimo, po normi) ujedno i "dobro" približno rješenje sustava. Prethodni teorem precizno opravdava to očekivanje.

Vrlo često je ovakva ocjena osjetljivosti linearnog sustava po normi pregruba i daje izuzetno pesimistične ocjene točnosti izračunatog rješenja u aritmetici računala. Za dobivanje boljih ocjena trebamo analizu po komponentama. Pokažimo dva teorema koja su u komponentnom smislu ekvivalentna prethodnim. U komponentnim ocjenama standardno pretpostavljamo da E i f imaju nenegativne elemente, tj. da vrijedi $|E| = E$ i $|f| = f$.

Teorem 5.4.3. *Neka je $Ax = b$ i $(A+\Delta A)\,(x+\Delta x) = b+\Delta b$, gdje je $|\Delta A| \le \varepsilon E$ i $|b| \le \varepsilon f$. Također, pretpostavimo da je $\varepsilon\,\||A^{-1}|\,E\| < 1$, gdje je $\|\cdot\|$ neka apsolutna norma. Za $x \neq 0$ vrijedi*

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\varepsilon}{1-\varepsilon\,\||A^{-1}|\,E\|}\,\frac{\||A^{-1}|\,E\,|x| + |A^{-1}|\,f\|}{\|x\|}, \tag{5.4.7}$$

a za ∞-normu se ocjena može dostići barem približno, do prvog reda veličine u ε.

Dokaz:

Prvi dio dokaza sličan je dokazu teorema 5.4.1. i slijedi iz iste jednakosti

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

U ∞-normi se ocjena približno dostiže stavljanjem $\Delta A = \varepsilon\, D_1 E D_2$, $\Delta b = -\varepsilon D_1 f$, gdje su $D_2 = \mathrm{diag}(\mathrm{sign}(x_i))$, $D_1 = \mathrm{diag}(\xi_j)$, $\xi_j = \mathrm{sign}(A^{-1})_{kj}$ i

$$\||A^{-1}|\,E\,|x| + |A^{-1}|\,f\|_\infty = (|A^{-1}|\,E\,|x| + |A^{-1}|\,f)_k,$$

tj. k je indeks komponente na kojoj se dostiže ∞-norma lijeve strane. ■

Ovakvim komponentnim perturbacijama od A i b, koje su ograničene s E, f i faktorom ε, odgovara broj uvjetovanosti, mjeren u ∞-normi za x, definiran sa

$$\mathrm{cond}_{E,f}(A,x) := \lim_{\varepsilon\to 0}\sup\left\{\frac{\|\Delta x\|_\infty}{\varepsilon\,\|x\|_\infty}\;\middle|\;(A+\Delta A)\,(x+\Delta x) = b+\Delta b,\right.$$
$$\left.|\Delta A| \le \varepsilon E,\ |\Delta b| \le \varepsilon f\right\}.$$

Prethodni teorem pokazuje da je ta uvjetovanost u ∞-normi jednaka

$$\mathrm{cond}_{E,f}(A,x) = \frac{\| \, |A^{-1}| \, E \, |x| + |A^{-1}| \, f \|_\infty}{\|x\|_\infty},$$

jer se ocjena (5.4.7) dostiže do prvog reda veličine u ε, tj. na limesu $\varepsilon \to 0$ ostaje upravo drugi faktor na desnoj strani (5.4.7).

Ako za E i f uzmemo prirodne vrijednosti $E = |A|$ i $f = |b|$, obično se koristi tzv. Skeelov broj uvjetovanosti

$$\mathrm{cond}(A,x) := \frac{\| \, |A^{-1}| \, |A| \, |x| \, \|_\infty}{\|x\|_\infty},$$

koji se od $\mathrm{cond}_{|A|,|b|}(A,x)$ razlikuje najviše za faktor 2. Dozvolimo li da se x mijenja, pa uzmemo maksimum među svim takvim uvjetovanostima, dobivamo uvjetovanost $\mathrm{cond}(A)$ koja ovisi samo o A

$$\mathrm{cond}(A,x) \leq \| \, |A^{-1}| \, |A| \, \|_\infty := \mathrm{cond}(A) = \kappa_{BS,\infty}(A),$$

s tim da se jednakost u ∞-normi dostže za $x = e := (1,\ldots,1)^T$. Ovo je poznata Bauer–Skeelova uvjetovanost matrice A, u ovom slučaju, generirana ∞-normom. Ta komponentna uvjetovanost može biti generirana i bilo kojom drugom matričnom normom $\kappa_{BS}(A) := \| \, |A^{-1}| \, |A| \, \|$.

I teorem 5.4.2. može se napisati u komponentnom obliku ako definiramo komponentnu grešku unatrag s

$$\omega_{E,f}(x+\Delta x) := \min\{\varepsilon \;\Big|\; (A+\Delta A)\,(x+\Delta x) = b + \Delta b,\ |\Delta A| \leq \varepsilon E,\ |\Delta b| \leq \varepsilon f\}.$$

Teorem 5.4.4. (Oettli i Prager) *Greška unatrag po komponentama jednaka je*

$$\omega_{E,f}(x+\Delta x) := \max_i \frac{|r_i|}{(E\,|x+\Delta x| + |f|)_i}, \tag{5.4.8}$$

gdje je $r = b - A\,(x+\Delta x)$. *Moguće dijeljenje s* 0 *interpretira se na sljedeći način:* $\xi/0$ *jednako je* 0, *ako je* $\xi = 0$, *a inače je* ∞.

Dokaz:

Ponovno, lako je dokazati da je desna strana relacije (5.4.8) donja ograda za $\omega_{E,f}(x+\Delta x)$. Ta donja ograda se dostiže za $\Delta A = D_1 E D_2$, $\Delta b = -D_1 f$, gdje su

$$D_1 = \mathrm{diag}\left(\frac{r_i}{(E\,|x+\Delta x| + |f|)_i}\right), \quad D_2 = \mathrm{diag}(\mathrm{sign}(x+\Delta x)_i).$$

■

U prethodnim teoremima i ocjenama nismo precizno navodili kojim prostorima pripadaju pojedini objekti, posebno A i b. Pažljivijim pogledom lako je ustanoviti da sve vrijedi i u realnom i u kompleksnom slučaju.

5.5. Greške zaokruživanja kod rješavanja trokutastog linearnog sustava

Kao što smo vidjeli, nalaženje rješenja općeg linearnog sustava na kraju se svede na rješavanje trokutastog linearnog sustava. Osim LR faktorizacije, koju smo već upoznali, i druge faktorizacije (koje ćemo tek upoznati), kao što su QR faktorizacija ili faktorizacija Choleskog, vode na trokutaste linearne sustave, pa ćemo njihovo rješavanje analizirati neovisno o nastanku.

Neka je T trokutasta matrica (ako eksplicitno ne kažemo, može biti gornjetrokutasta ili donjetrokutasta). Naš cilj je analizirali točnost rješenja linearnog sustava

$$Tx = b$$

supstitucijom unaprijed/unazad, kad računanje provodimo u aritmetici pomičnog zareza, tj. računalom. Zbog toga pretpostavljamo da računanje radimo u realnoj aritmetici, tj. da su svi objekti realni. Svi daljnji rezultati mogu se proširiti i na kompleksni slučaj, ali dobivene konstante u ocjenama neće biti iste i ovise o tome kako se kompleksne aritmetičke operacije realiziraju putem realnih.

Za početak, bez smanjenja općenitosti, možemo pretpostaviti da je linearni sustav gornjetrokutasti, tj. da vršimo supstituciju unazad. Varijablu x_i nalazimo kao

$$x_i = \frac{1}{t_{ii}} \left(b_i - \sum_{j=i+1}^{n} t_{ij} x_j \right), \tag{5.5.1}$$

što daje komponente vektora x u poretku od zadnje prema prvoj. Da bismo precizno mogli analizirati greške zaokruživanja u prethodnoj relaciji, potrebno je znati točan algoritam kako se ona izvrednjava. Recimo, sljedeća dva odsječka kôda, koja oba računaju (5.5.1), ne moraju imati iste greške zaokruživanja.

1. kôd

```
for i := n downto 1 do
  begin
  sum := 0.0;
  for j := i + 1 to n do
    sum := sum + T[i, j] * x[j];
  x[i] := (b[i] − sum)/T[i, i];
  end;
```

2. kôd

```
x[n] := b[n]/T[n, n];
for i := n − 1 downto 1 do
  begin
  sum := b[i];
  for j := i + 1 to n do
    sum := sum − T[i, j] * x[j];
  x[i] := sum/T[i, i];
  end;
```

Zašto? Osnovne aritmetičke operacije su binarne. Da bismo izračunali x_i iz relacije (5.5.1), desnu stranu treba svesti na niz binarnih operacija u nekom poretku. Matematički gledano, u egzaktnoj aritmetici, to možemo napraviti na mnogo ekvivalentnih načina — u smislu da svi daju isti rezultat, koristeći asocijativnost, pa čak

i komutativnost zbrajanja. Međutim, zbrajanje u aritmetici računala više nije asocijativno, pa egzaktno ekvivalentni algoritmi ne moraju dati iste rezultate. Poredak operacija postaje bitan.

Ako bolje pogledamo, u 1. kôdu prvo računamo cijelu sumu iz (5.5.1) u prirodnom poretku indeksa, a zatim ju oduzmemo od b_i i na kraju dijelimo s t_{ii}. U 2. kôdu, od b_i redom oduzimamo član po član te sume, u istom poretku indeksa. U praksi se, obično, koristi 2. varijanta, bar za sekvencijalno računanje. Motivacija je bazirana na promatranju kraćenja. Pretpostavimo da je $|t_{ii}x_i| \ll |b_i|$. To znači da negdje u algoritmu mora doći do kraćenja u računanju izraza u zagradi, prije završnog dijeljenja. U 1. kôdu, cijelo to kraćenje se događa na samom kraju, u operaciji $b[i] - sum$. Tj. odjednom se skrati puno znamenki, što može rezultirati velikim gubitkom točnosti. Za razliku od toga, u 2. kôdu oduzimamo član po član. Ako su članovi sume podjednakih veličina, kraćenje ide "malo–pomalo", a ne odjednom. U većini slučajeva to daje bolji rezultat.

Za algoritamsku realizaciju, pedantnije je relaciju (5.5.1) napisati u obliku

$$x_i = \Big(b_i - \sum_{j=i+1}^{n} t_{ij}x_j\Big)\Big/t_{ii}. \tag{5.5.2}$$

Naime, u algoritmu **ne** računamo prvo inverz $1/t_{ii}$, pa onda zagradu množimo s njim, nego zagradu dijelimo s t_{ii}. Ovom uštedom jedne aritmetičke operacije (množenja), osim ubrzanja, dobivamo i manju grešku zaokruživanja.

Potpuno analogno možemo napisati i algoritam za supstituciju unaprijed koja rješava linearni sustav s donjetrokutastom matricom T. Želimo da naša analiza bude primjenjiva na oba trokutasta sustava $Ly = b$ i $Rx = y$ koja dolaze iz LR faktorizacije, pa treba voditi računa o tome da za $T = L$ nema dijeljenja u pripadnoj rekurziji ($\ell_{ii} = 1$).

Kako ćemo napraviti analizu grešaka zaokruživanja? Da bismo si olakšali indeksiranje, napišimo izraz ekvivalentan jednom koraku rekurzije (5.5.2), samo unaprijed, i analizirajmo ga. Dakle, računamo

$$y = \Big(c - \sum_{i=1}^{k-1} a_i b_i\Big)\Big/b_k. \tag{5.5.3}$$

Na prvi pogled, čini se da pokušavamo varati u zaključku. Naime, u (5.5.2) imamo **rekurziju**, tj. komponente x_i koje računamo ovise jedna o drugoj (jer se i mijenja). Naprotiv, u relaciji (5.5.3) se takva međusobna ovisnost ne vidi! Svi podaci, a to su c, a_i, b_i i b_k, mogu biti i točni. Ako želimo pravu vezu između tih dviju relacija, trebalo bi dozvoliti da, recimo, a_i "glume" x_i i ulaze u račun s nekom greškom obzirom na prave, kao ranije izračunati objekti.

Ali, ... prijevare nema. Sve ovisi o tome koju vrstu analize grešaka zaokruživanja radimo. Kad bismo radili analizu unaprijed, prethodna primjedba bi bila

potpuno na mjestu. Međutim, u obratnoj analizi, a nju ćemo napraviti, možemo (barem donekle) birati kojim objektima dozvoljavamo ili pripisujemo perturbacije ekvivalentne greškama zaokruživanja.

Stvarni izračunati rezultat rekurzije (5.5.2) su neki brojevi $\widehat{x}_i$, komponente **izračunatog** rješenja $\widehat{x}$ polaznog linearnog sustava $Tx = b$. Naravno, općenito ne vrijedi $T\widehat{x} = b$. Što onda vrijedi za $\widehat{x}$? Možemo birati oblik sustava čije **egzaktno** rješenje je $\widehat{x}$. Uzmimo da perturbacije dozvoljavamo samo u matrici T, a vektor b je fiksan. Dakle, izračunati $\widehat{x}$ interpretiramo kao egzaktno rješenje perturbiranog linearnog sustava

$$(T + \Delta T)\,\widehat{x} = b$$

i tražimo odgovor na pitanje: koliko veliku perturbaciju ΔT treba napraviti u matrici T da, u egzaktnoj aritmetici kao rješenje sustava (s fiksnim b), dobijemo **izračunati** vektor $\widehat{x}$. Obratnom analizom tražimo ocjenu veličine te perturbacije ΔT obzirom na T. Naravno, želimo što manji ΔT, tj. što bolju ocjenu.

Trenutno nas uopće **ne zanima** koliko je točna aproksimacija $\widehat{x}$, odnosno ocjena za $\Delta x = x - \widehat{x}$. To prepuštamo analizi unaprijed za cijeli problem, koji može uključivati i prethodno računanje matrice T i vektora b.

Zašto smo onda fiksirali b? Upravo zato da možemo iskoristiti relaciju (5.5.3) za pojednostavljenje analize. Sjetimo se, nakon LR faktorizacije, rješavamo dva sustava $Ly = b$ i $Rx = y$. U prvom je b ulazni podatak i možemo ga smatrati fiksnim, do na polaznu grešku zaokruživanja spremanjem u računalo. Tu grešku možemo, također, prepustiti finalnoj analizi unaprijed. Međutim, u drugom sustavu je desna strana prethodno izračunato rješenje y, ili preciznije, $\widehat{y}$. Ocjena perturbacije te desne strane, kad bismo to dozvolili, izlazi analizom unaprijed za prethodni problem. A to još nemamo, i želimo izbjeći, ako je to moguće. Osim toga, spremljeni b i izračunati $\widehat{y}$ su stvarne desne strane ovih sustava kad računamo rješenja (a pravi y nemamo).

Ako sad usporedimo (5.5.2) i (5.5.3)

$$x_i = \Big(b_i - \sum_{j=i+1}^{n} t_{ij}x_j\Big)\Big/t_{ii}, \qquad y = \Big(c - \sum_{i=1}^{k-1} a_i b_i\Big)\Big/b_k,$$

možemo uzeti da komponenta b_i vektora b iz prve relacije odgovara broju c u drugoj relaciji i obje veličine se ne perturbiraju. Analogno, x_i odgovara y i ne perturbiraju se. Onda možemo uzeti da x_j odgovara broju a_i (uz odgovarajuću vezu indeksa j, i), opet bez perturbacija (jer ne diramo ranije izračunate komponente od x). Na kraju, t_{ij}, t_{ii}, odgovaraju brojevima b_i, b_k, respekitvno, i samo oni se perturbiraju. Dakle, imamo korektnu vezu ovih dviju relacija, bez "prevare".

Ostaje još pokazati da se obratna analiza relacije (5.5.3) može provesti uz prethodno opisano ograničenje na perturbacije pojedinih veličina u njoj.

Lema 5.5.1. *Izraz* (5.5.3) *za* y *računamo u aritmetici pomičnog zareza sljedećim odsječkom kôda*

$s := c;$
for $i := 1$ **to** $k-1$ **do**
  $s := s - a[i] * b[i];$
$y := s/b[k];$

uz pretpostavku sekvencijalnog izvršavanja petlje. Izračunati $\widehat{y}$ *onda zadovoljava*

$$\widehat{y}\, b_k\,(1+\theta_k) = c - \sum_{i=1}^{k-1} a_i\, b_i\,(1+\theta_i),$$

pri čemu je

$$|\theta_i| \le \gamma_i := \frac{iu}{1-iu}.$$

Dokaz:

Označimo s kapicom vrijednosti izračunate u aritmetici pomičnog zareza. Stanja varijable s parametriziramo indeksom i petlje. Neka je $\widehat{s}_0$ stanje varijable s prije početka petlje, a $\widehat{s}_i$ izračunata vrijednost od s **nakon** i-tog prolaza kroz petlju. Na samom početku algoritma je

$$\widehat{s}_0 = c,$$

jer je c već spremljen, pa uzimamo da je egzaktno prikaziv, tj. nema greške zaokruživanja pri dodjeljivanju ili kopiranju vrijednosti. Analogno, vrijednosti a_i, b_i i b_k, također, smatramo egzaktnim, jer su već spremljene u memoriji računala.

Kako dalje? Za svaku pojedinu aritmetičku operaciju $\circ$ koristimo standardni model (2.6.2) aritmetike računala. Podsjetimo, za izračunati rezultat operacije $x \circ y$ nad prikazivim, već spremljenim operandima x i y vrijedi

$$f\ell(x \circ y) = (1+\varepsilon)\,(x \circ y), \quad |\varepsilon| \le u, \tag{5.5.4}$$

uz uvjet da je $x \circ y$ u dozvoljenom rasponu, gdje je u jedinična greška zaokruživanja u odabranoj točnosti računanja. Može se pokazati da za osnovnu grešku zaokruživanja, osim (2.6.1), vrijedi i

$$f\ell(x) = \frac{x}{1+\varepsilon}, \quad |\varepsilon| \le u,$$

ako je x u dozvoljenom (prikazivom) rasponu brojeva. Posljedica toga je da vrijedi i sljedeća modifikacija (5.5.4)

$$f\ell(x \circ y) = \frac{x \circ y}{1+\varepsilon}, \quad |\varepsilon| \le u, \tag{5.5.5}$$

ako je $x \circ y$ u dozvoljenom rasponu.

U svim ovakvim analizama grešaka zaokruživanja, standardno pretpostavljamo da su svi međurezultati u dozvoljenom rasponu prikazivih brojeva, tako da za svaku pojedinu operaciju možemo koristiti jednu od prethodne dvije relacije, kako nam odgovara.

U svakom koraku petlje imamo dvije operacije — množenje i oduzimanje, i to tim redom, zbog prioriteta aritmetičkih operacija. Prema (5.5.4), za izračunate vrijednosti vrijedi

$$\widehat{s}_i = f\ell(\widehat{s}_{i-1} - a_i \cdot b_i) = (1+\delta_i)\,(\widehat{s}_{i-1} - (1+\varepsilon_i)\,a_i \cdot b_i), \quad i = 1, \ldots, k-1,$$

gdje ε_i označava grešku prilikom množenja, a δ_i grešku prilikom oduzimanja. Po pretpostavci modela, za te greške vrijedi $|\delta_i| \leq u$, $|\varepsilon_i| \leq u$.

Na kraju algoritma, umjesto y, posljednje dijeljenje izračuna $\widehat{y}$ i prema (5.5.5) vrijedi

$$\widehat{y} = f\ell\left(\frac{\widehat{s}_{k-1}}{b_k}\right) = \frac{\widehat{s}_{k-1}}{b_k\,(1+\delta_k)},$$

uz $|\delta_k| \leq u$.

Još moramo "pokupiti" sve prethodne greške. Za izračunate vrijednosti $\widehat{s}_i$ varijable s nakon svakog od $k-1$ koraka petlje, supstitucijom unaprijed dobivamo

$$\begin{aligned}
\widehat{s}_1 &= (1+\delta_1)\,(c - (1+\varepsilon_1)\,a_1 \cdot b_1) \\
&= (1+\delta_1)\,c - (1+\delta_1)\,(1+\varepsilon_1)\,a_1 \cdot b_1, \\
\widehat{s}_2 &= (1+\delta_2)\,(\widehat{s}_1 - (1+\varepsilon_2)\,a_2 \cdot b_2) \\
&= (1+\delta_1)\,(1+\delta_2)\,c - (1+\delta_1)\,(1+\delta_2)\,(1+\varepsilon_1)\,a_1 \cdot b_1 \\
&\qquad - (1+\delta_2)\,(1+\varepsilon_2)\,a_2 \cdot b_2, \\
\cdots &= \cdots \\
\widehat{s}_{k-1} &= c\prod_{i=1}^{k-1}(1+\delta_i) - \sum_{i=1}^{k-1} a_i \cdot b_i\,(1+\varepsilon_i)\prod_{j=i}^{k-1}(1+\delta_j).
\end{aligned}$$

Relaciju za $\widehat{y}$ možemo napisati u obliku

$$\widehat{y}\,b_k\,(1+\delta_k) = \widehat{s}_{k-1}.$$

Kad uvrstimo $\widehat{s}_{k-1}$ izlazi

$$\widehat{y}\,b_k\,(1+\delta_k) = c\prod_{i=1}^{k-1}(1+\delta_i) - \sum_{i=1}^{k-1} a_i \cdot b_i\,(1+\varepsilon_i)\prod_{j=i}^{k-1}(1+\delta_j).$$

Ova relacija, međutim, još uvijek nema željeni oblik. Faktor uz c odgovara nekoj relativnoj perturbaciji od c, a to ne želimo, jer se c ne perturbira. Kad podijelimo

cijelu relaciju s faktorom uz c, dobivamo

$$\widehat{y}\, b_k \frac{1+\delta_k}{\prod_{i=1}^{k-1}(1+\delta_i)} = c - \sum_{i=1}^{k-1} a_i \cdot b_i \frac{1+\varepsilon_i}{\prod_{j=1}^{i-1}(1+\delta_j)},$$

što ima traženi oblik, jer sve perturbacione faktore možemo interpretirati kao perturbacije b_i-ova, uključujući b_k, dok $\widehat{y}$, c i a_i ostaju neperturbirani.

Faktore uz b_i napišemo u obliku $(1+\theta_i)$, za $i = 1, \ldots, k$. Vidimo da se svaki takav faktor $(1+\theta_i)$ sastoji od točno i faktora oblika $(1+\delta)$ ili $1/(1+\delta)$, uz $|\delta| \leq u$. Ostaje još samo pokazati da je tada

$$|\theta_i| \leq \frac{iu}{1-iu}, \quad i = 1, \ldots, k.$$

Taj rezultat je direktna posljedica sljedeće leme. Usput, vidimo da smo dokaz mogli provesti i bez (5.5.5), koristeći samo (5.5.4). ■

U obratnoj analizi grešaka zaokruživanja stalno se pojavljuju produkti faktora istog oblika kao u prethodnom dokazu. U literaturi postoji nekoliko standardnih načina za njihovo pojednostavljenje i ocjenu. Relativno jednostavan i elegantan je sljedeći način.

Lema 5.5.2. *Neka je $u > 0$ realni broj i $n \in \mathbb{N}$ takav da vrijedi $nu < 1$. Ako je $|\delta_i| \leq u$ i $p_i \in \{-1, 1\}$, za $i = 1, \ldots, n$, onda vrijedi*

$$\prod_{i=1}^{n}(1+\delta_i)^{p_i} = 1 + \theta_n, \tag{5.5.6}$$

uz ocjenu

$$|\theta_n| \leq \gamma_n := \frac{nu}{1-nu}.$$

Dokaz:

Dokaz se provodi indukcijom po n. Za $n = 1$, pretpostavimo da je $u < 1$. Ako je $p_1 = 1$, onda je $\theta_1 = \delta_1$, pa je

$$|\theta_1| \leq u \leq \frac{u}{1-u}.$$

Ako je $p_1 = -1$, onda je $1 + \delta_1 \geq 1 - u > 0$. Iz $1 + \theta_1 = 1/(1+\delta_1)$ je

$$\theta_1 = \frac{1}{1+\delta_1} - 1 = -\frac{\delta_1}{1+\delta_1},$$

odakle slijedi ocjena

$$|\theta_1| = \frac{|\delta_1|}{1+\delta_1} \leq \frac{u}{1-u}.$$

Općenito, iz $nu < 1$ slijedi $u < 1$, pa je lijeva strana relacije (5.5.6) produkt pozitivnih faktora, a θ_n je dobro definiran i vrijedi $\theta_n > -1$.

Pretpostavimo da tvrdnja vrijedi za neki $n \in \mathbb{N}$. Ako produkt $n+1$ faktora napišemo u obliku

$$1+\theta_{n+1} = \prod_{i=1}^{n+1}(1+\delta_i)^{p_i} = \prod_{i=1}^{n}(1+\delta_i)^{p_i}\cdot(1+\delta_{n+1})^{p_{n+1}} = (1+\theta_n)(1+\delta_{n+1})^{p_{n+1}},$$

onda možemo koristiti pretpostavku indukcije za θ_n. Za $p_{n+1} = 1$ dobivamo

$$\theta_{n+1} = \theta_n + \delta_{n+1} + \theta_n\delta_{n+1}$$

pa je (relacija trokuta)

$$|\theta_{n+1}| \le \frac{nu}{1-nu} + u + \frac{nu^2}{1-nu} = \frac{(n+1)u}{1-nu} < \frac{(n+1)u}{1-(n+1)u},$$

s tim da koristimo $(n+1)u < 1$.

Za $p_{n+1} = -1$ dobivamo

$$\theta_{n+1} = \frac{1+\theta_n}{1+\delta_{n+1}} - 1 = \frac{\theta_n - \delta_{n+1}}{1+\delta_{n+1}},$$

pa je

$$|\theta_{n+1}| \le \frac{|\theta_n| + |\delta_{n+1}|}{1+\delta_{n+1}}.$$

Iz pretpostavke indukcije i $1+\delta_{n+1} \ge 1-u > 0$ slijedi

$$|\theta_{n+1}| \le \frac{\dfrac{nu}{1-nu} + u}{1-u} = \frac{(n+1)u - nu^2}{1-(n+1)u+nu^2} < \frac{(n+1)u}{1-(n+1)u},$$

uz $(n+1)u < 1$. Dakle, tvrdnja vrijedi i za $n+1$. ■

Sljedeći teorem daje obratnu grešku zaokruživanja kod povratne supstitucije, kad se koristi 2. kôd za računanje (5.5.2).

Teorem 5.5.1. *Rješenje $\widehat{x}$ gornjetrokutastog sustava $Tx = b$, izračunato u aritmetici pomičnog zareza 2. kôdom, možemo interpretirati kao egzaktno rješenje gornjetrokutastog sustava*

$$(T + \Delta T)\,\widehat{x} = b,$$

gdje je

$$|\Delta t_{ij}| \le \begin{cases} \gamma_{n-i+1}\,|t_{ii}|, & za\ i = j, \\ \gamma_{|i-j|}\,|t_{ij}|, & za\ i \ne j. \end{cases}$$

Dokaz:

Tvrdnja izlazi iz prethodne dvije leme i veze između (5.5.2) i (5.5.3). ■

Analogno se može dobiti rezultat za donjetrokutasti sustav, uz odgovarajuću promjenu indeksa, zbog supstitucije unaprijed. Osim toga, perturbacije dijagonalnih elementa imaju jedan faktor manje, jer nema zadnjeg dijeljenja. Međutim, taj rezultat nećemo posebno navesti, jer, kao i prethodni, strogo ovisi o poretku aritmetičkih operacija, tj. redosljedu zbrajanja ili oduzimanja članova odgovarajuće sume.

Ako ne želimo striktno fiksirati poredak operacija u relaciji (5.5.3), onda vrijedi sljedeća lema.

Lema 5.5.3. *Izraz* (5.5.3) *za* y

$$y = \left(c - \sum_{i=1}^{k-1} a_i b_i\right) \Big/ b_k$$

računamo u aritmetici pomičnog zareza. Bez obzira na poredak operacija, tj. redosljed zbrajanja ili oduzimanja u sumi u zagradi, izračunati $\widehat{y}$ *zadovoljava*

$$\widehat{y}\, b_k\, (1 + \theta_k^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i\, (1 + \theta_k^{(i)}),$$

pri čemu je $|\theta_k^{(i)}| \leq \gamma_k$ *za sve* i. *Ako je* $b_k = 1$, *tako da nema dijeljenja, onda je* $|\theta_k^{(i)}| \leq \gamma_{k-1}$ *za sve* i.

Dokaz:

Dokaz bi išao istim putem kao i dokaz leme 5.5.1., ali ga nije lako formalno korektno zapisati jer moramo uzeti u obzir bilo koji poredak operacija. S druge strane, sasvim je jednostavno zaključiti da bilo koji b_i, za $i = 1, \ldots, k-1$, sudjeluje u točno jednom množenju i najviše $k-1$ zbrajanja ili oduzimanja, jer ukupno imamo $k-1$ aditivnih operacija u zagradi. Pripadna perturbacija ima najviše k osnovnih faktora, od kojih se barem jedan skrati kad podijelimo s faktorom uz c. Iz leme 5.5.2. onda slijedi $|\theta_k^{(i)}| \leq \gamma_{k-1}$, za $i = 1, \ldots, k-1$. Sličan argument vrijedi i za perturbaciju uz b_k, s tim da treba dodati jedan faktor ako ima dijeljenja. ■

Ako se koristi bilo koji poredak računanja u teoremu 5.5.1., dolazi samo do promjene konstanti u ogradama. Obično se koristi sljedeća **uniformna** ocjena.

Teorem 5.5.2. *Neka je* $T \in \mathbb{R}^{n\times n}$ *regularna trokutasta matrica reda* n. *Ako rješenje sustava* $Tx = b$ *računamo u aritmetici pomičnog zareza supstitucijom (unaprijed ili unatrag), za izračunato rješenje* $\widehat{x}$ *vrijedi*

$$(T + \Delta T)\, \widehat{x} = b, \quad |\Delta T| \leq \gamma_n |T|.$$

Dokaz:

Iskoristimo vezu između (5.5.3) i algoritma supstitucije unatrag ili unaprijed, zatim primijenimo lemu 5.5.3. i uočimo da je $\gamma_k \leq \gamma_n$ za sve komponente rješenja sustava. Detalje možete naći u dokazu sljedećeg teorema. ■

Za trokutaste sustave koje dobivamo iz LR faktorizacije, katkad je zgodno imati nešto preciznije ocjene po recima.

Teorem 5.5.3. *Neka su L i R trokutasti faktori iz LR faktorizacije neke regularne matrice $A \in \mathbb{R}^{n\times n}$. Neka su $\widehat{y}$ i $\widehat{x}$ izračunata rješenja linearnih sustava $Ly = b$ i $Rx = y$, dobivena algoritmom supstitucije u aritmetici pomičnog zareza. Onda vrijedi*

$$(L + \Delta L)\,\widehat{y} = b,$$
$$(R + \Delta R)\,\widehat{x} = \widehat{y},$$

uz ocjene

$$|\Delta L| \leq \operatorname{diag}(\gamma_0, \gamma_1, \ldots, \gamma_{n-1})\,|L|,$$
$$|\Delta R| \leq \operatorname{diag}(\gamma_n, \gamma_{n-1}, \ldots, \gamma_1)\,|R|,$$

za bilo koji poredak operacija u računanju svake pojedine komponente od $\widehat{y}$ i $\widehat{x}$.

Dokaz:

Rješenje sustava $Ly = b$ računamo supstitucijom unaprijed

$$y_i = b_i - \sum_{j=1}^{i-1} \ell_{ij}\, y_j, \quad i = 1, \ldots, n,$$

ne vodeći računa o poretku operacija na desnoj strani. Za primjenu leme 5.5.3. na y_i, usporedbom s (5.5.3), zaključujemo da treba uzeti $k = i$. Osim toga, L ima jedinice na dijagonali pa nema dijeljenja. Za izračunatu komponentu $\widehat{y}_i$ dobivamo ocjenu greške s faktorom γ_{i-1} uz elemente i-tog retka matrice L. Na kraju, faktore uz retke najlakše je zapisati u obliku produkta s dijagonalnom matricom tih faktora kao prvim (lijevim) članom.

Analogno, rješenje sustava $Rx = \widehat{y}$ računamo supstitucijom unatrag

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

koristeći izračunate vrijednosti $\widehat{y}_i$ umjesto (nepoznatih) pravih, ne vodeći računa o poretku operacija na desnoj strani. Usporedbom s (5.5.3), uz translaciju indeksa tako da suma počinje od 1, zaključujemo da za x_i treba uzeti $k = n - i + 1$. U ovom slučaju imamo dijeljenja, pa za izračunatu komponentu $\widehat{x}_i$ dobivamo ocjenu greške s faktorom γ_{n-i+1} uz elemente i-tog retka matrice R. ■

U dokazu koristimo samo sekvencijalni poredak računanja pojedinih komponenti rješenja (unaprijed ili unatrag), a poredak operacija za računanje svake pojedine komponente nije bitan. Ovim pristupom dobivamo općenitost rezultata i ocjena

grešaka, što nam omogućava njihovu primjenu na razne algoritme, ne vodeći računa o redoslijedu izvršavanja operacija u algoritmu. Stvarna greška, naravno, ovisi o poretku operacija u konkretnom algoritmu. Za određene podatke, ta ovisnost ne mora biti mala, što se lijepo može vidjeti iz primjera za zbrajanje računalom.

Napomenimo da u prethodnom teoremu **nije** bitno da li su matrice L i R egzaktne ili izračunate iz A, sve dok ih možemo interpretirati kao **egzaktne** faktore u LR faktorizaciji **neke** regularne matrice (A ili perturbirane $A + \Delta A$).

Ovi rezultati kažu da izračunato rješenje $\widehat{x}$ trokutastog linearnog sustava ima malu obratnu relativnu grešku po komponentama. Drugim riječima, obratna greška je skoro toliko mala koliko smijemo očekivati — puno bolje ne treba niti tražiti!

5.6. Greška unaprijed za trokutasti sustav

Sljedeći korak je procjena greške unaprijed. U toj analizi, cilj nam je pokazati sljedeće:

- isplati se raditi analizu po komponentama;
- parcijalno i potpuno pivotiranje u LR faktorizaciji se isplati, jer daje bolje ocjene grešaka;
- pivotiranje treba koristiti i u drugim faktorizacijama, da se dobiju što bolji trokutasti faktori.

Analizu provodimo za opće trokutaste sustave, bez obzira na "porijeklo" i detaljni algoritam rješavanja, pa koristimo teorem 5.5.2. za ocjenu greške unatrag. Teorem 5.5.3. daje nešto bolje ocjene za algoritam supstitucije, ali ih je teže zapisati.

Obratnu analizu grešaka zaokruživanja proveli smo tako da se samo matrica sustava perturbira, a ne i vektor desne strane. To koristimo u teoremima perturbacije koji daju grešku unaprijed tako da uzmemo $f = 0$. Teorem 5.5.2. daje ocjenu perturbacije ΔT po komponentama. Zbog toga koristimo teorem 5.4.3., s tim da za E možemo uzeti da je $E = |T|$ i $\varepsilon = \gamma_n$. Tako iz (5.4.7) dobivamo ocjenu relativne greške unaprijed za izračunati $\widehat{x} = x - \Delta x$. Primijetimo još, kad uzmemo $E = |T|$ i $f = 0$, za Skeelov broj uvjetovanosti vrijedi $\mathrm{cond}(T, x) = \mathrm{cond}_{|T|,0}(T, x)$, pa u ∞-normi ocjena glasi

$$\frac{\|\Delta x\|_\infty}{\|x\|_\infty} \leq \frac{\gamma_n \,\mathrm{cond}(T, x)}{1 - \gamma_n \,\mathrm{cond}(T)},$$

gdje je

$$\mathrm{cond}(T, x) = \frac{\| \,|T^{-1}|\,|T|\,|x|\,\|_\infty}{\|x\|_\infty}, \quad \mathrm{cond}(T) = \kappa_{BS,\infty}(T) = \| \,|T^{-1}|\,|T|\,\|_\infty,$$

uz pretpostavku da je $\gamma_n \,\mathrm{cond}(T) < 1$.

Ova ocjena može biti proizvoljno bolja od odgovarajuće ocjene koja sadrži standardni broj uvjetovanosti po normi

$$\kappa_\infty(T) = \|T^{-1}\|_\infty \|T\|_\infty.$$

Osnovni razlog za to smo već opisali — norma "vidi" samo velike elemente.

Za trokutaste sustave može se naći još detaljnije opravdanje. Klasični broj uvjetovanosti $\kappa(T)$ može biti velik iz dva razloga:

- veličina dijagonalnih elemenata,
- veliki vandijagonalni elementi u relativnom smislu, obzirom na pripadni dijagonalni element u odgovarajućem retku.

Lako se vidi da je komponentni broj uvjetovanosti $\mathrm{cond}(T, x)$ invarijantan na tzv. skaliranje redaka, tj. na množenje sprijeda bilo kojom regularnom dijagonalnom matricom D, jer je $\mathrm{cond}(DT, x) = \mathrm{cond}(T, x)$. Zbog toga, veličina dijagonalnih elemenata ne igra ulogu i $\mathrm{cond}(T, x)$ može biti velik samo iz drugog razloga.

Unatoč tomu, $\mathrm{cond}(T, x)$ može biti po volji velik. Primjer za to je familija gornjih trokutastih matrica $R(\alpha)$ s elementima

$$R(\alpha) = (r_{ij}), \qquad r_{ij} = \begin{cases} 1, & \text{za } i = j, \\ -\alpha, & \text{za } i < j. \end{cases}$$

Za inverze vrijedi

$$(R(\alpha)^{-1})_{ij} = \begin{cases} 1, & \text{za } i = j, \\ \alpha(1+\alpha)^{j-i-1}, & \text{za } i < j. \end{cases}$$

Ako uzmemo $x = e := (1, \ldots, 1)^T$, onda je

$$\mathrm{cond}(R(\alpha), e) = \mathrm{cond}(R(\alpha)) \approx 2\alpha^{n-1}, \quad \text{kad } \alpha \to \infty.$$

Dakle, **ne** vrijedi zaključak da **sve** trokutaste linearne sustave možemo riješiti s visokom točnošću!

S druge strane, za svaku trokutastu matricu T postoji bar jedan sustav za koji dobivamo visoku točnost. Ako je T gornja trokutasta, to je sustav $Tx = e_1$, a ako je T donja trokutasta, to je sustav $Tx = e_n$. U oba slučaja je $\mathrm{cond}(T, x) = 1$, a rješenje se svodi na jedno računanje recipročne vrijednosti, u zadnjem koraku supstitucije. Naravno, slično vrijedi i za skalarni višekratnik takve desne strane.

Prirodno je pitanje da li za neke trokutaste sustave možemo dobiti i bolji rezultat. Posebno, za one koje dobivamo određenim faktorizacijama s pivotiranjem. Odgovor je potvrdan, a iskazujemo ga za regularne gornje trokutaste matrice.

Lema 5.6.1. *Neka je $R \in \mathbb{R}^{n\times n}$ regularna gornje trokutasta matrica za koju vrijedi*

$$|r_{ii}| \geq |r_{ij}|, \quad \text{za svaki } j > i, \tag{5.6.1}$$

tj. u svakom retku se najveći element po apsolutnoj vrijednosti nalazi na dijagonali. Onda za jediničnu gornju trokutastu matricu $S = |R^{-1}|\,|R|$ vrijedi $s_{ij} \leq 2^{j-i}$, za svaki $j > i$.

Dokaz:

Neka je $D = \operatorname{diag}(r_{ii})$ i $V = D^{-1}R$. Onda je V jedinična gornja trokutasta matrica za koju vrijedi $|v_{ij}| \leq 1$. Osim toga je $S = |V^{-1}|\,|V|$ (invarijantnost na skaliranje redaka). Iz $V^{-1}V = I$, lako se pokaže da za elemente inverza vrijedi ocjena

$$|(V^{-1})_{ij}| \leq 2^{j-i-1}, \quad j > i.$$

Za elemente s_{ij} matrice S, uz $j > i$, dobivamo

$$s_{ij} = \sum_{k=i}^{j} |(V^{-1})_{ik}|\,|v_{kj}| \leq 1 + \sum_{k=i+1}^{j} 2^{k-i-1} \cdot 1 = 2^{j-i}.$$

Tvrdnja vrijedi i za $i = j$, s tim da vrijedi jednakost, jer je $s_{ii} = 1$. ■

Tvrdnju Leme 5.6.1. možemo interpretirati i na sljedeći način. Ako matrica R zadovoljava (5.6.1), onda je komponentni broj uvjetovanosti $\operatorname{cond}(R)$ ograđen odozgo za fiksni n, neovisno o tome koliko je velik klasični broj uvjetovanosti $\kappa(R)$. Točna vrijednost ove ocjene ovisi o normi kojom mjerimo $\operatorname{cond}(R)$. Na primjer, u ∞-normi dobivamo $\operatorname{cond}(R) \leq 2^n - 1$.

Primijenimo ovaj rezultat na ocjenu greške unaprijed za izračunato rješenje takvog sustava.

Teorem 5.6.1. *Uz pretpostavke prethodne leme, za algoritmom supstitucije izračunato rješenje $\widehat{x}$ sustava $Rx = b$ vrijedi*

$$|x_i - \widehat{x}_i| \leq 2^{n-i+1}\gamma_n \max_{j\geq i} |\widehat{x}_j|, \quad i = 1, \ldots, n.$$

Dokaz:

Prema teoremu 5.5.2., za izračunato rješenje vrijedi $(R + \Delta R)\widehat{x} = b$, uz ocjenu $|\Delta R| \leq \gamma_n |R|$. Onda je $R(x - \widehat{x}) = \Delta R\,\widehat{x}$, odakle izlazi ocjena

$$|x - \widehat{x}| = |R^{-1}\,\Delta R\,\widehat{x}| \leq \gamma_n |R^{-1}|\,|R|\,|\widehat{x}|.$$

Primjenom leme 5.6.1., uz $S = |R^{-1}|\,|R|$, po komponentama dobivamo

$$|x_i - \widehat{x}_i| \leq \gamma_n \sum_{j=i}^{n} s_{ij}|\widehat{x}_j| \leq \gamma_n \max_{j\geq i} |\widehat{x}_j| \cdot \sum_{j=i}^{n} 2^{j-i} = (2^{n-i+1} - 1)\gamma_n \max_{j\geq i} |\widehat{x}_j|,$$

pa vodeći član desne strane daje ocjenu iz tvrdnje. Preciznija ocjena nema praktičnu vrijednost, jer nas zanima samo red veličine pogreške. ■

Ocjena greške $|x_i - \widehat{x}_i|$ za komponente izračunatog rješenja u teoremu 5.6.1. može biti velika, ako je n velik, a i mali. Međutim, ona eksponencijalno pada s rastom i, što znači da su donje komponente od x uvijek izračunate s visokom relativnom točnošću obzirom na već izračunate elemente.

Analogoni leme 5.6.1. i teorema 5.6.1. vrijede za donje trokutaste matrice L koje zadovoljavaju

$$|l_{ii}| \geq |l_{ij}|, \quad \text{za svaki } j < i. \tag{5.6.2}$$

tj. opet se u svakom retku najveći element po apsolutnoj vrijednosti nalazi na dijagonali.

Treba, međutim, uočiti da ako gornja trokutasta matrica R zadovoljava (5.6.1), onda donja trokutasta matrica R^T **ne** mora zadovoljavati (5.6.2). Primjer za to je matrica

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

za koju je

$$\mathrm{cond}(R) = 5, \quad \mathrm{cond}(R^T) = 1 + \frac{2}{\varepsilon},$$

pa $\mathrm{cond}(R^T)$ može biti proizvoljno velik. "Krivac" je element $r_{12} = 1$ koji dominira svojim stupcem, tj. svojim retkom u R^T, a nije na dijagonali.

Teorem 5.6.1. i analogni rezultat za donje trokutaste matrice možemo primijeniti na trokutaste sustave koje dobivamo raznim faktorizacijama ili postupcima eliminacije s pivotiranjem:

- donje trokutaste matrice iz Gaussovih eliminacija s parcijalnim ili potpunim pivotiranjem,
- gornje trokutaste matrice iz Gaussovih eliminacija s potpunim pivotiranjem,
- gornje trokutaste matrice iz faktorizacije Choleskog s potpunim pivotiranjem ili QR faktorizacije s pivotiranjem po stupcima (vidjeti sljedeće poglavlje).

Dakle, pivotiranje se **isplati**.

Kako se to vidi kod LR faktorizacije, odnosno Gaussovih eliminacija? Ranije smo pokazali da su elementi matrice L u LR faktorizaciji upravo multiplikatori u Gaussovim eliminacijama, tj. vrijedi $l_{ij} = m_{ij}$, za $i > j$. Osim toga, uz isto pivotiranje (matrica P), na kraju dobivamo iste gornje trokutaste matrice R.

Kod parcijalnog pivotiranja, pivotni element je po apsolutnoj vrijednosti najveći u ostatku svog stupca (na dijagonali ili ispod nje). Zbog toga, kad ga dovedemo

na dijagonalu, u j-tom koraku eliminacije vrijedi

$$|l_{ij}| \leq 1 = l_{jj}, \quad \text{za svaki } i > j, \quad j = 1, \ldots, n-1.$$

Na prvi pogled imamo zaključak da su dijagonalni elementi najveći u svom stupcu, a ne retku. No, svi elementi u strogo donjem trokutu od L su ispod 1 po apsolutnoj vrijednosti, a to znači da su dijagonalni elementi najveći i u svom retku! Dakle, za L vrijedi (5.6.2).

Potpuno isto vrijedi i za potpuno pivotiranje. Tada je pivotni element apsolutno najveći u cijelom ostatku matrice, pa to vrijedi i za ostatak njegovog stupca. S druge strane, on je apsolutno najveći i u ostatku svog retka. Kad ga dovedemo na dijagonalu, pripadni redak je upravo novi redak matrice R, jer ga ostatak eliminacije ili faktorizacije više ne mijenja. Dakle, nakon i-tog koraka, za i-ti redak matrice R vrijedi (5.6.1).

Napomena 5.6.1. Pažljivijom analizom izgleda kao da nismo u potpunosti iskoristili sva svojstva parcijalnog i potpunog pivotiranja — u smislu da bismo iste rezultate mogli dobili i nešto "slabijim" varijantama pivotiranja.

U oba slučaja, dobivena matrica L, osim (5.6.2), ima još neka svojstva, koja nismo iskoristili. Dijagonala dominira i stupcima i cijelim donjim trokutom, a mi koristimo samo dominaciju po recima. Međutim, L ima jediničnu (konstantnu) dijagonalu, pa su sva ova svojstva dominacije ekvivalentna. Pivotiranje osigurava dominaciju po stupcima, pa i sve ostalo. Teško bi bilo konstruirati algoritam pivotiranja koji direktno daje dominaciju dijagonale po recima, jer se L računa stupac po stupac. Osim toga, to nema smisla, jer L ima jediničnu dijagonalu, pa su dominacije ekvivalentne. Izlazi da smo "maksimum u ostatku stupca" u potpunosti iskoristili, barem za trokutasti sustav s matricom L.

Za R **ne** vrijedi isti argument, jer R nema jediničnu dijagonalu. Od potpunog pivotiranja iskoristili smo samo to da je pivotni element najveći po apsolutnoj vrijednosti u ostatku svog stupca i retka. Takvih elemenata koji dominiraju ostatkom svog retka i stupca može biti i više, pa ih je, općenito, lakše naći. Naravno, može se dogoditi da je apsolutno najveći element u cijelom ostatku matrice jedini takav — on sigurno dominira svojim retkom i stupcem. Dakle, za isti rezultat mogli smo i "slabije" pivotirati!

Možemo zaključiti da je polazna kritika, uglavnom, korektna. Zaista, u analizi trokutastih sustava nismo u potpunosti iskoristili sva svojstva pivotiranja. Međutim, odgovor na to je sasvim jednostavan. Točnost rješenja dobivenih trokutastih sustava **nije** jedina svrha pivotiranja. Ono se ozbiljno koristi i u analizi LR faktorizacije, za kontrolu i ocjenu kvalitete izračunatih faktora L i R. Naime, parcijalno pivotiranje ima utjecaja i na matricu R, a ne samo na L. To pogotovo vrijedi za potpuno pivotiranje.

5.7. Greške zaokruživanja za LR faktorizaciju

Analiza grešaka zaokruživanja kod rješavanja lineranog sustava korištenjem LR faktorizacije koristi dvije stvari: analizu grešaka zaokruživanja izraza oblika (5.5.3)

$$y = \Big(c - \sum_{i=1}^{k-1} a_i b_i\Big) \Big/ b_k.$$

i analizu rješenja trokutastog sustava.

Rekurzije koje koristimo za računanje elemenata r_{kj} i ℓ_{jk} matrica R i L su

$$\begin{aligned} r_{kj} &= a_{kj} - \sum_{i=1}^{k-1} \ell_{ki}\, r_{ij}, \quad j = k, \ldots, n, \\ \ell_{ik} &= \Big(a_{ik} - \sum_{j=1}^{k-1} \ell_{ij}\, r_{jk}\Big) \Big/ r_{kk}, \quad i = k+1, \ldots, n. \end{aligned} \tag{5.7.1}$$

Primijetite da su to izrazi istog oblika koje smo analizirali u lemi 5.5.1., a potom u lemi 5.5.3. za bilo koji poredak sumacije.

Sada je jednostavno primjeniti lemu 5.5.3. na svaki od izraza u (5.7.1), uz korektno "nazivanje" objekata. Ulogu y "igra" lijeva strana svakog od izraza, ulogu c odgovarajući element a_{kj}, odnosno a_{ik} matrice A, ulogu a-ova dotad izračunati elementi matrice L i ulogu b-ova dotad izračunati elementi matrice R (izračunate elemente označimo kapicom).

Mi želimo ocijeniti koliko se razlikuju elementi matrice A obzirom na izračunate elemente matrica L i R. Primjenimo li lemu 5.5.3. za r_{kj} dobivamo

$$\widehat{r}_{kj}\,(1+\theta_k^{(0)}) = a_{kj} - \sum_{i=1}^{k-1} (1+\theta_k^{(i)})\,\widehat{\ell}_{ki}\,\widehat{r}_{ij},$$

pri čemu je $|\theta_k^{(i)}| \le \gamma_k$. Prebacivanjem sume s jedne strane na drugu, te uzimanjem apsolutnih vrijednosti dobivamo

$$\Big|a_{kj} - \widehat{r}_{kj} - \sum_{i=1}^{k-1} \widehat{\ell}_{ki}\,\widehat{r}_{ij}\Big| \le \gamma_k \sum_{i=1}^{k-1} |\widehat{\ell}_{ki}|\,|\widehat{r}_{ij}|. \tag{5.7.2}$$

Na sličan način, korištenjem leme 5.5.3., analiziramo i drugu relaciju u (5.7.1)

$$\widehat{\ell}_{ik}\,\widehat{r}_{kk}\,(1+\theta_k^{(0)}) = a_{ik} - \sum_{j=1}^{k-1} (1+\theta_k^{(i)})\,\widehat{\ell}_{ij}\,\widehat{r}_{jk}.$$

Ponovno premještanjem pribrojnika, pa uzimanjem apsolutnih vrijednosti, dobivamo

$$\Big|a_{ik} - \sum_{j=1}^{k} \widehat{\ell}_{ij}\,\widehat{r}_{jk}\Big| \le \gamma_k \sum_{j=1}^{k} |\widehat{\ell}_{ij}|\,|\widehat{r}_{jk}|. \tag{5.7.3}$$

Koje je značenje relacija (5.7.2) i (5.7.3)? Prvo, primijetite da za sve elemente gornjeg trokuta matrice A vrijedi relacija (5.7.2), a za elemente strogo donjeg trokuta relacija (5.7.3), pa zajedno pokrivaju čitav A. Napišemo li to matrično, za čitav A, onda (5.7.2) i (5.7.3) daju

$$|A - \widehat{L}\widehat{R}| \leq \gamma_n |\widehat{L}|\,|\widehat{R}|.$$

Time smo dokazali sljedeći teorem.

Teorem 5.7.1. *U aritmetici pomičnog zareza računamo LR faktorizaciju zadane matrice A reda n. Pretpostavimo da je algoritam uspješno završio (bez pojave prevelikih ili premalih brojeva koji nisu prikazivi, i bez pokušaja dijeljenja s nulom). Izračunati trokutasti faktori $\widehat{L}$ i $\widehat{R}$ onda zadovoljavaju*

$$\widehat{L}\,\widehat{R} = A + \Delta A, \quad |\Delta A| \leq \gamma_n |\widehat{L}|\,|\widehat{R}|.$$

Uz malo dodatnog napora, prethodni teorem, zajedno s teoremom o analizi grešaka zaokruživanja za trokutaste sustave, daje obratnu grešku zaokruživanja za rješenje linearnog sustava $Ax = b$ korištenjem LR faktorizacije.

Teorem 5.7.2. *U aritmetici pomičnog zareza rješavamo linearni sustav $Ax = b$ s matricom A reda n. Neka su $\widehat{L}$ i $\widehat{R}$ izračunati trokutasti faktori u LR faktorizaciji matrice A, i neka je $\widehat{x}$ izračunato rješenje sustava $Ax = b$. Onda postoji perturbacija ΔA matrice A za koju vrijedi*

$$(A + \Delta A)\,\widehat{x} = b, \quad |\Delta A| \leq \gamma_{3n} |\widehat{L}|\,|\widehat{R}|.$$

Dokaz:

Greške u rješenju lineranog sustava interpretiramo kao egzaktno rješenje linearnog sustava s malo perturbiranom matricom. Raščlanimo li perturbacije u matrici A, možemo vidjeti da su one posljedica:

- LR faktorizacije matrice (tu perturbaciju označimo s ΔA_1),
- rješavanja trokutastih linearnih sustava $Ly = b$ i $Rx = y$.

Iz prethodnog teorema, za LR faktorizaciju dobivamo ocjenu

$$\widehat{L}\,\widehat{R} = A + \Delta A_1, \quad |\Delta A_1| \leq \gamma_n |\widehat{L}|\,|\widehat{R}|. \tag{5.7.4}$$

Prema teoremu 5.5.2., rješavanje trokutastog sustava u aritmetici pomičnog zareza možemo interpretirati kao točno rješavanje malo perturbiranog sustava. Tj., za zadane T i b, izračunamo rješenje $\widehat{x}$, i onda postoji perturbacija ΔT za koju vrijedi

$$(T + \Delta T)\,\widehat{x} = b, \quad |\Delta T| \leq \gamma_n |T|.$$

Primijenimo to na dva trokutasta sustava koja moramo riješiti. Njihove matrice su izračunati faktori $\widehat{L}$, odnosno $\widehat{R}$, a izračunata rješenja supstitucijama unaprijed, odnosno unatrag, neka su $\widehat{y}$ i $\widehat{x}$. Dobivamo da postoje perturbacije $\Delta\widehat{L}$ i $\Delta\widehat{R}$ za koje vrijedi

$$\begin{aligned} (\widehat{L}+\Delta\widehat{L})\,\widehat{y} &= b, \quad |\Delta\widehat{L}| \le \gamma_n|\widehat{L}|, \\ (\widehat{R}+\Delta\widehat{R})\,\widehat{x} &= \widehat{y}, \quad |\Delta\widehat{R}| \le \gamma_n|\widehat{R}|. \end{aligned} \tag{5.7.5}$$

Tada b možemo napisati kao

$$\begin{aligned} b &= (\widehat{L}+\Delta\widehat{L})\,(\widehat{R}+\Delta\widehat{R})\,\widehat{x} = (\widehat{L}\widehat{R}+\Delta\widehat{L}\,\widehat{R}+\widehat{L}\,\Delta\widehat{R}+\Delta\widehat{L}\,\Delta\widehat{R})\,\widehat{x} \\ &= (A+\Delta A_1+\Delta\widehat{L}\,\widehat{R}+\widehat{L}\,\Delta\widehat{R}+\Delta\widehat{L}\,\Delta\widehat{R})\,\widehat{x} \\ &:= (A+\Delta A)\,\widehat{x}. \end{aligned}$$

Za ovako definiranu perturbaciju ΔA, iz (5.7.4) i (5.7.5) dobivamo ocjenu

$$\begin{aligned} |\Delta A| &= |\Delta A_1+\Delta\widehat{L}\,\widehat{R}+\widehat{L}\,\Delta\widehat{R}+\Delta\widehat{L}\,\Delta\widehat{R}| \\ &\le |\Delta A_1|+|\Delta\widehat{L}|\,|\widehat{R}|+|\widehat{L}|\,|\Delta\widehat{R}|+|\Delta\widehat{L}|\,|\Delta\widehat{R}| \\ &\le (3\gamma_n+\gamma_n^2)\,|\widehat{L}|\,|\widehat{R}|. \end{aligned}$$

Na kraju, za konstantu na desnoj strani vrijedi

$$\begin{aligned} 3\gamma_n+\gamma_n^2 &= 3\,\frac{nu}{1-nu}+\Big(\frac{nu}{1-nu}\Big)^2 = \frac{3nu(1-nu)+(nu)^2}{(1-nu)^2} \\ &= \frac{3nu-2(nu)^2}{1-2nu+(nu)^2} \le \frac{3nu}{1-2nu} \le \frac{3nu}{1-3nu} = \gamma_{3n}, \end{aligned}$$

uz uvjet da je $3nu<1$, pa dobivamo ocjenu iz tvrdnje teorema. ■

U prethodnom dokazu iskoristili smo uniformne ocjene konstantom γ_n za perturbacije kod rješavanja oba trokutasta sustava. Znamo da teorem 5.5.3. daje i finije ocjene po recima

$$\begin{aligned} |\Delta\widehat{L}| &\le \operatorname{diag}(\gamma_0,\gamma_1,\ldots,\gamma_{n-1})\,|\widehat{L}| \\ |\Delta\widehat{R}| &\le \operatorname{diag}(\gamma_n,\gamma_{n-1},\ldots,\gamma_1)\,|\widehat{R}|. \end{aligned}$$

Nažalost, njih ne možemo zgodno iskoristiti u dokazu prethodnog teorema. Naime, faktor $\operatorname{diag}(\gamma_n,\gamma_{n-1},\ldots,\gamma_1)$, koji djeluje na retke od $|\widehat{R}|$, možemo interpretirati i tako da djeluje na stupce od $|\widehat{L}|$. Međutim, on ne mora komutirati s $|\widehat{L}|$, pa ga ne možemo "prebaciti" ispred $|\widehat{L}|$, tako da i on djeluje na retke od $|\widehat{L}|$.

Kako možemo interpretirati prethodni teorem? U idealnom slučaju željeli bismo da je $|\Delta A|\le u|A|$. To bi odgovaralo grešci koju napravimo zaokruživanjem elemenata matrice A pri inicijalnom spremanju podataka. Ali, nad svakim elementom matrice A vrši se još najviše n operacija, i zato ne možemo očekivati nešto bolje od ocjene oblika

$$|\Delta A| \le c_n u|A|,$$

gdje je c_n konstanta reda veličine n. Takva će ocjena vrijediti pod uvjetom da $\widehat{L}$ i $\widehat{R}$ zadovoljavaju da je

$$|\widehat{L}|\,|\widehat{R}| = |\widehat{L}\widehat{R}|, \tag{5.7.6}$$

što, naravno, ne mora uvijek biti slučaj. Ako vrijedi (5.7.6), onda iz teorema 5.7.1 izlazi

$$|\widehat{L}|\,|\widehat{R}| = |\widehat{L}\widehat{R}| = |A + \Delta A| \leq |A| + \gamma_n |\widehat{L}|\,|\widehat{R}|,$$

pa, prebacivanjem članova dobivamo

$$|\widehat{L}|\,|\widehat{R}| \leq \frac{1}{1-\gamma_n}\,|A|.$$

Ako tu relaciju uvrstimo u teorem 5.7.2., onda imamo

$$(A + \Delta A)\,\widehat{x} = b, \quad |\Delta A| \leq \frac{\gamma_{3n}}{1-\gamma_n}\,|A|, \tag{5.7.7}$$

tj. izračunato rješenje $\widehat{x}$ ima malu obratnu relativnu grešku po komponentama.

Pitanje je koje matrice zadovoljavaju (5.7.6). Na primjer, ako LR faktorizacija daje nenegativne elemente u L i R, takve će matrice sigurno zadovoljavati traženo svojstvo. Takva je, na primjer, klasa **totalno pozitivnih/totalno nenegativnih** matrica.

Totalno pozitivne/nenegativne matrice su kvadratne matrice kojima su determinante bilo koje kvadratne podmatrice pozitivne/nenegativne. Inverzi totalno nenegativnih matrica, također, imaju svojstvo da su im matrice L i R nenegativne (po komponentama). Nažalost, kod totalno nenegativnih matrica parcijalno pivotiranje (ili zamjene redaka) će općenito uništiti svojstvo totalne nenegativnosti. Na primjer, uzmite samo zamjenu 2 retka i bilo koju 2×2 matricu nastalu zamjenom redaka. Njezina će determinanta promijeniti znak. I kod inverza totalno nenegativnih matrica, pivotiranje uništava svojstvo nenegativnosti L i R. Zbog toga je za totalno nenegativne matrice i njihove inverze najbolje koristiti Gaussove eliminacije, odnosno LR faktorizaciju **bez** pivotiranja.

No vratimo se na stabilnost LR faktorizacije. Važno svojstvo koje proizlazi iz teorema 5.7.1. i teorema 5.7.2. je da stabilnost rješenja linearnog sustava ne ovisi o veličini multiplikatora, već o veličini elemenata koji se javljaju u matrici $|\widehat{L}|\,|\widehat{R}|$. Ta matrica može imati male elemente iako su joj multiplikatori $m_{ij} = \ell_{ij}$ veliki, ali može imati i velike elemente, a da su joj multiplikatori reda veličine 1.

Da bismo lakše proučavali stabilnost Gaussovih eliminacija (ili ekvivalentno LR faktorizacije + rješenja sustava), koristit ćemo norme. Po ugledu na izvod (5.7.7), razumno je proučavati omjer

$$\frac{\|\,|\widehat{L}|\,|\widehat{R}|\,\|}{\|A\|}. \tag{5.7.8}$$

Bez pivotiranja, omjer normi (5.7.8) može biti proizvoljno velik. Na primjer, pokažite da je za matricu

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}$$

taj je omjer ε^{-1}.

Kod parcijalnog pivotiranja vrijedi da je

$$|\ell_{ij}| \leq 1 \quad \text{za sve } i \geq j,$$

pa nije teško (indukcijom po koracima) pokazati da zbog $m_{ik} = \ell_{ik}$ i

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

vrijedi

$$|r_{ij}| \leq 2^{i-1} \max_{k \leq i} |a_{kj}|.$$

Dakle, kod parcijalnog je pivotiranja L malen, a R ograđen relativno obzirom na A.

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 5.7.3. (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*

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

Dokaz:

Uz parcijalno pivotiranje za egzaktne faktore L i R vrijede ocjene $|L| \leq 1$ i $|R| \leq \rho_n$, a faktor n^2 je posljedica prijelaza na $\| \ \|_\infty$. Striktno govoreći, za izračunate faktore vrijede malo slabije ocjene $|\widehat{L}| \leq 1+u$ i $|\widehat{R}| \leq \widehat{\rho}_n$, gdje je $\widehat{\rho}_n$ izračunati faktor rasta, pa desnu stranu treba tako i shvatiti. Međutim, ideja cijele tvrdnje je da se analizira "pravi" faktor rasta ρ_n. ■

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

5.8. Pivotni rast

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	$< 5.005\,.$

5.9. 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 5.9.1. (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 dijagonalno dominantna po stupcima. Za $j = 2, \ldots, n$ vrijedi

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

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

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

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

$$L = \begin{bmatrix} 1 & & & \\ \ell_2 & 1 & & \\ & \ddots & \ddots & \\ & & \ell_{n-1} & 1 \\ & & & \ell_n & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} r_1 & e_1 & & \\ & r_2 & e_2 & \\ & & \ddots & \ddots \\ & & & r_{n-1} & e_{n-1} \\ & & & & r_n \end{bmatrix}. \tag{5.9.1}$$

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

$$\begin{aligned} &r_1 = d_1, \\ &\text{za } i = 2, \dots, n: \\ &\quad \ell_i = c_i / r_{i-1}, \\ &\quad r_i = d_i - \ell_i\, e_{i-1}. \end{aligned} \tag{5.9.2}$$

Računamo li te vrijednosti u aritmetici pomičnog zareza, onda za izračunate vrijednosti vrijedi

$$\begin{aligned} &(1+\varepsilon_i)\widehat{\ell}_i = \frac{c_i}{\widehat{r}_{i-1}}, \qquad |\varepsilon_i| \le u, \\ &(1+\theta_i)\widehat{r}_i = d_i - \widehat{\ell}_i\, e_{i-1}(1+\delta_i), \qquad |\theta_i|, |\delta_i| \le u. \end{aligned}$$

Premještanjem pribrojnika, te korištenjem apsolutne vrijednosti dobivamo

$$|c_i - \widehat{\ell}_i\,\widehat{r}_{i-1}| \le u\,|\widehat{\ell}_i\,\widehat{r}_{i-1}|,$$
$$|d_i - \widehat{\ell}_i\,e_{i-1} - \widehat{r}_i| \le u\,(|\widehat{\ell}_i\,e_{i-1}| + |\widehat{r}_i|).$$

Ako ove relacije napišemo matrično, onda je

$$A = \widehat{L}\widehat{R} + \Delta A, \quad |\Delta A| \le u\,|\widehat{L}|\,|\widehat{R}|. \tag{5.9.3}$$

Rješavanje linearnog sustava $Ax = b$, nakon LR faktorizacije napravi još dodatnu grešku prilikom supstitucije unaprijed i unatrag. Na sličan način kao kod rješavanja trokutastog sustava, nije teško pokazati da za tako izračunato rješenje $\widehat{x}$ vrijedi

$$(\widehat{L} + \Delta\widehat{L})\,(\widehat{R} + \Delta\widehat{R})\,\widehat{x} = b, \quad |\Delta\widehat{L}| \le u|\widehat{L}|, \quad |\Delta\widehat{R}| \le (2u + u^2)|\widehat{R}|. \tag{5.9.4}$$

Kombiniranjem (5.9.3) i (5.9.4), dobivamo da je ukupna greška

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

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

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

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 5.9.2. *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, $L \ge 0$ i $R \ge 0$,

(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 $A = D_1BD_2$, gdje su $|D_1| = |D_2| = I$.

Dokaz:

Dokažimo samo tvrdnju (a). Za simetričnu pozitivno definitnu matricu, možemo LR faktorizaciju napisati u obliku

$$A = LDL^T.$$

Kako se to pokazuje? U običnoj LR faktorizaciji, ako postoji, (a postoji, što se lako dokazuje, jer je dijagonala uvijek pozitivna), faktor R se rastavi na produkt $R = DM^T$ dijagonalne matrice D i gornjetrokutaste matrice M^T s jedinicama na dijagonali. Dobivamo

$$A = LDM^T, \quad M \text{ donjetrokutasta, regularna.}$$

Zbog simetrije vrijedi

$$A = A^T = MDL^T,$$

pa je

$$LDM^T = MDL^T.$$

Množenjem slijeva s L^{-1} i zdesna s L^{-T} dobivamo

$$DM^T L^{-T} = L^{-1}MD.$$

Primijetimo da na lijevoj strani imamo produkt gornjetrokutastih matrica, a na desnoj strani donjetrokutastih, pa zaključujemo da su ti produkti dijagonalne matrice. Uočimo još da su te dijagonalne matrice baš jednake D (jer i M i L imaju na dijagonali jedinice), pa imamo

$$L^{-1}MD = D \implies MD = LD \implies M = L.$$

I ne samo to, D mora imati pozitivne elemente, jer bi inače postojao vektor x takav da je $(Ax, x) \leq 0$, tj. A ne bi bila pozitivno definitna.

Sada možemo, uz malo razmišljanja i raspisivanja, zaključiti da je

$$|L|\,|R| = |L|\,D\,|L^T| = |LDL^T| = |LR|.$$

Pažljivo pokažite da je srednja jednakost korektna! ■

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 5.9.3. *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*

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

Dokaz:

Uspoređujemo elemente matrica $|L|\,|R|$ i A. Za elemente na sporednim dijagonalama, direktnim množenjem iz (5.9.1) dobivamo da vrijedi

$$(|L|\,|R|)_{ij} = |a_{ij}|, \quad \text{za } |i-j| = 1.$$

Ostalo je još pokazati što se zbiva s dijagonalnim elementima. Dovoljno je pokazati da vrijedi

$$|\ell_i\, e_{i-1}| + |r_i| \le 3\,|d_i|.$$

U nastavku dokaza pretpostavljamo da je A dijagonalno dominantna po recima (za dijagonalnu dominantnost po stupcima dokaz ide slično). Prvo tvrdimo da je

$$|e_i| \le |r_i|,$$

za sve indekse i. Dokaz se provodi indukcijom. Za $i = 1$ to je očito iz dijagonalne dominantnosti. Pretpostavimo da to vrijedi za $i-1$, a zatim iz (5.9.2) imamo redom

$$\begin{aligned} |r_i| &\ge |d_i| - |\ell_i|\,|e_{i-1}| = |d_i| - \frac{|c_i|}{|r_{i-1}|}\,|e_{i-1}| \\ &\ge |d_i| - |c_i| \ge |e_i|. \end{aligned}$$

Na sličan način pokazuje se da je $|r_i| \le |d_i| + |c_i|$. Konačno, dobivamo

$$\begin{aligned} |\ell_i\, e_{i-1}| + |r_i| &= \left|\frac{c_i}{r_{i-1}}\, e_{i-1}\right| + |r_i| \le |c_i| + |r_i| \\ &\le |c_i| + (|d_i| + |c_i|) \le 3|d_i|. \end{aligned}$$

■

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

Teorem 5.9.4. *Ako je zadana nesingularna trodijagonalna matrica A tipa* (a)–(d) *iz teorema* 5.9.2. *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*

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

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.

Dokaz:

Za matricu A tipa (a)–(c), pretpostavka da je u dovoljno mali osigurava pozitivnost izračunatih dijagonalnih elemenata $\widehat{r}_i$ matrice $\widehat{R}$, jer $\widehat{r}_i \to r_i > 0$, kad $u \to 0$. Lako se vidi da $\widehat{r}_i > 0$ povlači da je $|\widehat{L}|\,|\widehat{R}| = |\widehat{L}\widehat{R}|$ i za izračunate faktore. Sličan argument vrijedi i ako je A tipa (d).

Zadnji dio tvrdnje za dijagonalno dominantne matrice A je trivijalan. ■

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.

S druge strane, u postupku eliminacija mogu se pojaviti i veliki multiplikatori, ali oni nemaju negativnih posljedica na stabilnost. Na primjer, uzmimo M-matricu

$$A = \begin{bmatrix} 2 & -2 & 0 \\ \varepsilon - 2 & 2 & 0 \\ 0 & -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ (\varepsilon - 2)/2 & 1 & 0 \\ 0 & -1/\varepsilon & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & 3 \end{bmatrix} = LR,$$

gdje je $0 < \varepsilon < 2$. Multiplikator ℓ_{32} nije ograničen kad $\varepsilon \to 0$, i na limesu dobivamo singularnu matricu A koja nema LR faktorizaciju (bez pivotiranja). Međutim, za $\varepsilon > 0$ vrijedi $|L|\,|R| = |A|$ i Gaussove eliminacije (bez pivotiranja) su vrlo stabilne, upravo onako kao što teorem 5.9.4. kaže da moraju biti.

6. Faktorizacija Choleskog i QR faktorizacija

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 $D = \mathrm{diag}(r_{ii})$ 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}

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

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 "skalarana" (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. Analiza greške za faktorizaciju Choleskog

Ocjenu greške zaokruživanja za faktorizaciju Choleskog dobivamo na sličan način kao i kod LR faktorizacije. Dovoljno je primijetiti da se algoritam svodi na isti oblik rekurzije kao i ranije, osim za dijagonalne elemente matrice R, gdje imamo druge korijene umjesto dijeljenja.

Standardnom modelu aritmetike (2.6.2) dodajemo odgovarajuću pretpostavku za korijen. Neka je x egzaktno prikaziv broj, tj. već spremljen u memoriji računala. Za izračunatu vrijednost od $\sqrt{x}$ onda vrijedi

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

gdje je u jedinična greška zaokruživanja u odabranoj točnosti računanja.

Analogon leme 5.5.3. za izraze koji se javljaju u računanju dijagonalnih elemenata matrice R ima sljedeći oblik.

Lema 6.2.1. *Izraz*

$$y = \Big(c - \sum_{i=1}^{k-1} a_i b_i\Big)^{1/2}$$

računamo u aritmetici pomičnog zareza. Bez obzira na poredak operacija, tj. redosljed zbrajanja ili oduzimanja u sumi u zagradi, izračunati $\widehat{y}$ zadovoljava

$$\widehat{y}^2\,(1+\theta_{k+1}) = c - \sum_{i=1}^{k-1} a_i b_i\,(1+\theta_{k-1}^{(i)}),$$

pri čemu je $|\theta_{k-1}^{(i)}| \le \gamma_{k-1}$, za $i = 1, \ldots, k-1$, i $|\theta_{k+1}| \le \gamma_{k+1}$.

Dokaz:

Dokaz ide istim putem kao i dokaz leme 5.5.3., samo je dijeljenje zamijenjeno korijenom. Kvadriranje završne relacije kvadrira i grešku vađenja drugog korijena. Zato uz $\widehat{y}^2$ dobivamo jedan osnovni faktor više, što daje θ_{k+1} s navedenom ocjenom, a ne θ_k s ocjenom $|\theta_k| \le \gamma_k$, kao ranije. ■

Neka je $\widehat{R}$ izračunati faktor Choleskog za simetričnu pozitivno defintinu matricu A. Za elemente strogo gornjeg trokuta matrice R koristimo lemu 5.5.3., jer pripadne relacije imaju dijeljenje bez korijena. Dobivamo ocjenu

$$\Big|a_{ij} - \sum_{k=1}^{i} \widehat{r}_{ki}\,\widehat{r}_{kj}\Big| \le \gamma_i \sum_{k=1}^{i} |\widehat{r}_{ki}|\,|\widehat{r}_{kj}|,$$

s tim da je $i < j$, tako da sume idu samo do i. Ista relacija, zbog simetrije matrice A, vrijedi i za a_{ji}, s istim γ_i, a ne γ_j. Za dijagonalne elemente od R iz leme 6.2.1. slijedi ocjena

$$\Big|a_{jj} - \sum_{k=1}^{j} \widehat{r}_{kj}^2\Big| \le \gamma_{j+1} \sum_{k=1}^{j} \widehat{r}_{kj}^2.$$

Spajanjem ovih ocjena za sve elemente matrice A odmah dobivamo ocjenu greške unatrag.

Teorem 6.2.1. *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica. Pretpostavimo da algoritam za nalaženje faktorizacije Choleskog završava bez greške u aritmetici računala. Tako izračunati faktor $\widehat{R}$ zadovoljava*

$$\widehat{R}^T\,\widehat{R} = A + \Delta A, \quad |\Delta A| \le \gamma_{n+1} |\widehat{R}^T|\,|\widehat{R}|.$$

Kad ovom teoremu dodamo rezultate za pripadne trokutaste sustave dobivamo sljedeći rezultat za rješenje linearnog sustava $Ax = b$.

Teorem 6.2.2. *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica i neka je $\widehat{x}$ izračunato rješenje linearnog sustava $Ax = b$ u aritmetici pomičnog zareza, na bazi faktorizacije Choleskog matrice A i pripadnih supstitucija unaprijed i unatrag. Onda je*

$$(A + \Delta A)\,\widehat{x} = b, \quad |\Delta A| \leq \gamma_{3n+1}|\widehat{R}^T|\,|\widehat{R}|.$$

Dokaz:

Dokaz ide analogno dokazu teorema 5.7.2. za LR faktorizaciju. ■

Napomenimo da perturbacija ΔA iz ovog teorema ne mora biti simetrična. Naime, kad rješavamo trokutaste sustave s matricama R i R^T, općenito, dobivamo prave matrice grešaka unatrag koje ne moraju biti jedna drugoj transponirane. Može se pokazati postoji i "mala" simetrična perturbacija ΔA za koju vrijedi teorem 6.2.2., ali uz nešto lošiju ocjenu.

Uz ponešto truda, iz ovih rezultata možemo dobiti dobru ocjenu za omjer normi

$$\frac{\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|}{\|A\|}.$$

Kako to izlazi? Za egzaktnu faktorizaciju Choleskog $A = R^T R$ u 2-normi vrijedi sljedeća nejednakost

$$\|\,|R^T|\,|R|\,\|_2 = \|\,|R|\,\|_2^2 \leq n\|R\|_2^2 = n\|R^T R\|_2 = n\|A\|_2.$$

Prva i pretposljednja jednakost izlaze direktno iz definicije 2-norme, a nejednakost iz ponašanja 2-norme prema apsolutnoj vrijednosti (v. poglavlje o normama).

Prema teoremu 6.2.1., izračunati faktor $\widehat{R}$ je egzaktni faktor Choleskog za matricu $A + \Delta A$. Kad prethodnu nejednakost napišemo za $\widehat{R}$, dobivamo

$$\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|_2 \leq n\|A + \Delta A\|_2 \leq n(\|A\|_2 + \|\Delta A\|_2),$$

a zatim iskoristimo ocjenu iz teorema 6.2.1. i "monotonost" 2-norme

$$\|\Delta A\|_2 \leq \|\,|\Delta A|\,\|_2 \leq \gamma_{n+1}\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|_2.$$

Spajanjem ovih nejednakosti izlazi

$$\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|_2 \leq n\|A\|_2 + n\gamma_{n+1}\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|_2,$$

odakle, uz pretpostavku da je $n\gamma_{n+1} < 1$, dobivamo

$$\|\,|\widehat{R}^T|\,|\widehat{R}|\,\|_2 \leq n(1 - n\gamma_{n+1})^{-1}\|A\|_2.$$

Odavde odmah možemo zaključiti da faktorizacija Choleskog ima savršenu stabilnost unatrag po normi.

Ovu ocjenu možemo iskoristiti u teoremu 6.2.2. za ocjenu greške unatrag kod rješavanja linearnog sustava. Dobivamo

$$\|\Delta A\|_2 \leq \| \, |\Delta A| \, \|_2 \leq \gamma_{3n+1} n (1 - n\gamma_{n+1})^{-1} \|A\|_2.$$

Ako još pretpostavimo da je $(3n+1)u < 1/2$ i $n\gamma_{n+1} \leq 1/2$, onda je

$$\|\Delta A\|_2 \leq 4n(3n+1)u\|A\|_2,$$

tj. imamo ocjenu oblika $\|\Delta A\|_2 \leq c_n u \|A\|_2$, u kojoj c_n ovisi samo o n i to kvadratno.

Da smo radili u M-normi, $\|A\|_M = \max_{i,j} |a_{ij}|$, mogli smo dobiti još bolje izgledajući rezultat

$$\|\Delta A\|_M \leq \gamma_{3n+1}(1 - \gamma_{n+1})^{-1} \|A\|_M,$$

iz koje slijedi $\|\Delta A\|_M \leq c_n u \|A\|_M$, s tim da c_n ovisi samo linearno o n.

Standardnom teorijom perturbacije po normi dobivamo ocjenu za grešku unaprijed izračunatog rješenja $\widehat{x}$ u obliku

$$\frac{\|x - \widehat{x}\|}{\|x\|} \leq c_n u \, \kappa(A) + O(u^2),$$

gdje je c_n neka blago rastuća funkcija od n, očito ovisna i o normi u kojoj mjerimo greške i broj uvjetovanosti.

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

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

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 $r_{11} \ge \cdots \ge r_{nn} > 0$.

Na kraju, spomenimo još neke modernije rezultate koji daju dublji uvid u faktorizaciju Choleskog. Prvi daje potencijalno poboljšanje teorema 6.2.1. promatranjem dijagonale matrice A.

Teorem 6.2.3. (Demmel) *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica. Pretpostavimo da algoritam za nalaženje faktorizacije Choleskog završava bez greške u aritmetici računala. Tako izračunati faktor $\widehat{R}$ zadovoljava*

$$\widehat{R}^T \widehat{R} = A + \Delta A, \quad |\Delta A| \le (1 - \gamma_{n+1})^{-1} \gamma_{n+1}\, d d^T,$$

gdje je d vektor korijena dijagonalnih elemenata matrice A, tj. $d_i = a_{ii}^{1/2}$.

Ideja je simetrično izlučiti dijagonalnu skalu od A, tj. napisati A u obliku $A = DHD$, gdje je $D = \mathrm{diag}(A)^{1/2}$ korijen iz dijagonale od A. Matrica H sad ima jediničnu dijagonalu. Poznati teorem van der Sluisa kaže da vrijedi

$$\kappa_2(H) \leq n \min_{\Delta} \kappa_2(\Delta A \Delta),$$

gdje je Δ bilo koja dijagonalna matrica. Drugim riječima, matrica D skoro minimizira uvjetovanost u 2-normi po svim dijagonalnim skaliranjima. Zbog toga je sigurno

$$\kappa_2(H) \leq n\kappa_2(A),$$

tj. uvjetovanost od H nije pretjerano narasla obzirom na A, ali se može dogoditi da je

$$\kappa_2(H) \ll \kappa_2(A),$$

ako je A loše skalirana. Uočimo još da je $1 \leq \|H\|_2 \leq n$, jer je H pozitivno definitna s jediničnom dijagonalom.

Iz teorema 6.2.3., umjesto klasične ocjene s $\kappa(A)$, možemo dobiti potencijalno mnogo bolju ocjenu preko $\kappa(H)$.

Teorem 6.2.4. (Demmel, Wilkinson) *Neka je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica i neka je $\widehat{x}$ izračunato rješenje linearnog sustava $Ax = b$ u aritmetici pomičnog zareza, na bazi faktorizacije Choleskog matrice A i pripadnih supstitucija unaprijed i unatrag. Ako A napišemo u obliku $A = DHD$, gdje je $D = \mathrm{diag}(A)^{1/2}$, onda za skaliranu grešku $D(x - \widehat{x})$ vrijedi*

$$\frac{\|D(x-\widehat{x})\|_2}{\|Dx\|_2} \leq \frac{\varepsilon\,\kappa_2(H)}{1-\varepsilon\,\kappa_2(H)},$$

gdje je $\varepsilon = n(1-\gamma_{n+1})^{-1}\gamma_{3n+1}$.

6.3. QR faktorizacija

U mnogim je primjenama simetrična matrica A reda n zadana svojim, generalno, pravokutnim faktorom G, tako da je

$$A = G^T G.$$

Na prvi je pogled jasno da je tako definirana A simetrična. Lako dokazujemo da je tako definirana matrica pozitivno semidefinitna, jer je

$$x^T Ax = (x^T G^T)(Gx) = (Gx)^T(Gx) \geq 0. \tag{6.3.1}$$

Da bi A bila pozitivno definitna, potrebni su još neki uvjeti na oblik faktora G.

- Ako je G tipa $m \times n$, onda mora biti $m \geq n$. U protivnom, kad bi bilo $m < n$, onda bi bio $\text{rang}(G) \leq m$, što povlači da je i $\text{rang}(A) \leq m$, tj. A je singularna.
- G mora imati puni stupčani rang, tj. mora biti $\text{rang}(G) = n$. U tom je slučaju nul-potprostor od G (tj. svi oni vektori za koje je $Gx = 0$) trivijalan (samo $x = 0$), pa za sve $x \neq 0$ vrijedi $Gx \neq 0$ i u (6.3.1) izlazi $x^T Ax > 0$.

Dakle, pretpostavimo da je pozitivno definitna matrica A zadana svojim faktorom G tipa $m \times n$, $m \geq n$, $\text{rang}(G) = n$. Znamo da se svaka takva matrica A može rastaviti faktorizacijom Choleskog u $A = R^T R$. Pitamo se može li se to napraviti i ako je A implicitno zadana faktorom G. Očito je da može i to eksplicitnim formiranjem matrice G. S numeričke strane takvo eksplicitno formiranje matrice A nije jako poželjno, prvo zato jer formiranjem elemenata od A radimo neke greške zaokruživanja, a drugo, takav proces predugo traje.

Kad bismo mogli matricu G odmah faktorizirati tako da je

$$G = QR = Q \begin{bmatrix} R_0 \\ 0 \end{bmatrix}, \tag{6.3.2}$$

gdje je Q ortogonalna matrica reda m a R_0 gornjetrokutasta matrica reda n s pozitivnim dijagonalnim elementima, onda bismo lako pročitali faktorizaciju Choleskog matrice A, jer je

$$A = G^T G = R^T Q^T QR = R^T R = R_0^T R_0,$$

pa bi R_0 bio baš traženi faktor.

Faktorizacija (6.3.2) za G punog stupčanog ranga uvijek postoji i zove se **QR faktorizacija**. Primijetite da smo (6.3.2) mogli pisati i u jednostavnijoj formi, ako prvih n stupaca matrice Q označimo s Q_0 (pazite Q_0 je tipa $m \times n$), onda je

$$G = QR = Q_0 R_0, \quad Q_0^T Q_0 = I_n.$$

Ostaje samo pokazati da QR faktorizacija matrice G postoji.

Teorem 6.3.1. *Neka je $G \in \mathbb{R}^{m \times n}$, $m \geq n$ i neka je* $\text{rang}(G) = n$. *Tad postoji jedinstvena faktorizacija oblika*

$$G = Q_0 R_0,$$

pri čemu je Q_0 tipa $m \times n$, $Q_0^T Q_0 = I_n$, a R_0 gornjetrokutasta s pozitivnim dijagonalnim elementima.

Dokaz:

Najjednostavniji je dokaz ovog teorema je korištenjem Gram-Schmidtove ortogonalizacije. Ako stupce matrice $G = [g_1, g_2, \ldots, g_n]$ ortogonaliziramo slijeva udesno, dobit ćemo ortonormalni niz vektora q_1 do q_n koji razapinje isti potprostor kao i stupci od G. Stavimo li $Q_0 = [q_1, q_2, \ldots, q_n]$, dobili smo $m \times n$ ortogonalnu

matricu. Također Gram–Schmidtov postupak ortogonalizacije računa i koeficijente $r_{ji} = q_j^T g_i$ koji polazni stupac g_i izražavaju kao linearnu kombinaciju prvih i vektora q_j ortonormirane baze, tako da je

$$g_i = \sum_{j=1}^{i} r_{ji} q_j.$$

Elementi r_{ji} su elementi matrice R_0. Iz Gram–Schmidtovog algoritma bit će jasno da se može uzati $r_{ii} > 0$. ■

Iako je ovaj dokaz ortogonalizacijom elegantan, u praksi se **nikad** ne koristi Gram–Schmidtov (CGS) postupak ortogonalizacije, jer je nestabilan kad su stupci od G skoro linearno zavisni. Umjesto toga, može se koristiti tzv. modificirani Gram–Schmidtov postupak (MGS) koji je mnogo stabilniji, ali i kod njega se može dogoditi da je izračunati Q_0 vrlo daleko od ortogonalnog, tj. $\|Q_0^T Q_0 - I\| \gg u$ kad je G loše uvjetovana.

Potpunosti radi, dajemo i CGS i MGS algoritam.

Algoritam 6.3.1. (Klasični i modificirani Gram–Schmidt)

for $i := 1$ **to** n **do**
  {`Nađi` i`-ti stupac od` Q `i` R}
  begin
  $q_i = g_i$;
  for $j := 1$ **to** $i - 1$ **do**
    {`Oduzmi komponentu u` q_j `u smjeru` g_i}
    begin
    $r_{ji} := q_j^T g_i$; {`kod CGS-a`} `ili` $r_{ji} := q_j^T q_i$; {`kod MGS-a`}
    $q_i := q_i - r_{ji} q_j$;
    end;
  $r_{ii} := \|q_i\|_2$;
  if $r_{ii} = 0$ **do**
    begin
    error := *true*;
    exit;
    end
  else
    begin
    error := *false*;
    $q_i := q_i / r_{ii}$;
    end;
  end;

Pokažite da su dvije formule za r_{ji} koje koriste CGS i MGS matematički ekvivalentne.

Kad želimo ortogonalan Q, koristimo ili **Givensove rotacije** ili **Householderove reflektore** kojima poništavamo odgovarajuće elemente u matrici G, što će nam, ponovno, dati konstruktivni dokaz teorema 6.3.1.

6.3.1. Givensove rotacije

Matrica

$$R(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

je Givensova rotacija koja svaki vektor $x \in \mathbb{R}^2$ rotira obrnuto od kazaljke na satu za kut φ.

Možemo definirati i Givensovu rotaciju u (i, j) ravnini s

$$R(i,j,\varphi) = \begin{bmatrix} & & \overset{i}{\downarrow} & & & & \overset{j}{\downarrow} & & & \\ 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & \cos\varphi & & & -\sin\varphi & & & \leftarrow i \\ & & & & 1 & & & & & \\ & & & & & \ddots & & & & \\ & & & & & & 1 & & & \\ & & & \sin\varphi & & & \cos\varphi & & & \leftarrow j \\ & & & & & & & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{bmatrix}$$

Za zadani vektor $x \in \mathbb{R}^m$, lako poništavamo njegovu j-tu komponentu x_j korištenjem rotacije $R(i, j, \varphi)$. Primijetite da množenjem matrice $R(i, j, \varphi)$ na x mijenjamo samo i-tu i j-tu komponentu u x, pa dobivamo

$$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} x_i' \\ 0 \end{bmatrix}.$$

Drugi redak u matričnoj jednadžbi je

$$\sin\varphi\, x_i + \cos\varphi\, x_j = 0.$$

Ako je $x_j = 0$ nemamo što poništavati, a ako nije, dobivamo

$$\operatorname{ctg}\varphi = -\frac{x_i}{x_j},$$

odakle, korištenjem trigonometrijskog identiteta

$$1 + \operatorname{ctg}^2 \varphi = \frac{1}{\sin^2 \varphi}$$

slijedi

$$\sin^2 \varphi = \frac{x_j^2}{x_i^2 + x_j^2}, \quad \cos^2 \varphi = 1 - \sin^2 \varphi = \frac{x_i^2}{x_i^2 + x_j^2}.$$

Sada možemo izabrati predznake za $\sin \varphi$ i $\cos \varphi$, tako da x_i' bude pozitivan. Ako stavimo

$$\sin \varphi = -\frac{x_j}{\sqrt{x_i^2 + x_j^2}}, \quad \cos \varphi = \frac{x_i}{\sqrt{x_i^2 + x_j^2}},$$

dobivamo

$$x_i' = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}\, x_i + \frac{x_j}{\sqrt{x_i^2 + x_j^2}}\, x_j = \frac{x_i^2 + x_j^2}{\sqrt{x_i^2 + x_j^2}} = \sqrt{x_i^2 + x_j^2} > 0.$$

Primijetite da je element x_i' dobiven nakon transformacije upravo norma i-te i j-te komponente polaznog vektora.

Sistematskim poništavanjem elemenata, konstruirat ćemo QR faktorizaciju matrice G. Počnimo s prvim stupcem. Redom, možemo poništavati elemente g_{j1}, $j = 2, \ldots m$ korištenjem rotacija $R(1, j, \varphi)$, tj. rotacija koje "nabacuju" normu prvog stupca na prvi element u stupcu. Zatim to možemo ponoviti za drugi, treći i svaki daljnji stupac. Primijetite, da time nećemo "pokvariti" već sređene nule u prethodnim stupcima. Grafički, za jednu matricu tipa 5×3 to izgleda ovako

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \rightarrow \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \rightarrow \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix} \rightarrow \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Za ocjenu greške zaokruživanja postoji i bolji raspored poništavanja elemenata, jer se ovakvom transformacijom, pri "sređivanju" prvog stupca prvi redak mijenja $m - 1$ puta. Kad bismo ujednačili da se svaki redak podjednak broj puta transformira, onda bi naša ocjena greške bila bolja. To možemo postići korištenjem niza nezavisnih rotacija koje ne zahvaćaju iste retke. Osim toga, takav raspored odvijanja rotacija dozvoljava paralelizaciju algoritma.

Naravno, na kraju algoritma, na mjestu matrice G piše matrica R. Kako ćemo pronaći Q? Ako promatramo transformacije koje obavljamo, dobivamo

$$R(n, m, \varphi_{nm}) \cdot R(n, m-1, \varphi_{n,m-1}) \cdots R(1, 2, \varphi_{12}) G := Q^{-1} G = R.$$

Primijetimo da smo matricu G slijeva množili produktom ortogonalnih matrica, koji je i sam ortogonalna matrica, a možemo je označiti s Q^{-1}, pa je $G = QR$.

6.3.2. Householderovi reflektori

Umjesto da elemente u stupcu poništavamo jedan po jedan, korištenjem Householderovih reflektora, možemo odjednom poništiti sve elemente, osim jednog, u dijelu odgovarajućeg stupca.

Matrica H definirana s

$$H = I - 2uu^T, \quad \|u\|_2 = 1$$

zove se Householderov reflektor. Matrica H je simetrična, što je očito, i ortogonalna. Vrijedi

$$\begin{aligned} HH^T = H^2 &= (I - 2uu^T)(I - 2uu^T) = I - 4uu^T + 4uu^T uu^T \\ &= I - 4uu^T + 4u(u^T u)u^T = I - 4uu^T + 4\|u\|_2^2 uu^T = I. \end{aligned}$$

Zašto baš ime reflektor? Promatrajmo hiperravninu koja je okomita na u i prolazi ishodištem. Reflektor H sve vektore x preslikava u simetrični obzirom na tu ravninu.

Ako imamo zadan vektor x, jednostavno je pronaći u za Householderov reflektor tako da poništimo sve osim prve komponente vektora x, tj. tražimo da je

$$Hx = \begin{bmatrix} c \\ 0 \\ \vdots \\ 0 \end{bmatrix} = c \cdot e_1.$$

Raspišimo tu jednadžbu

$$Hx = (I - 2uu^T)x = x - 2u(u^T x) = c \cdot e_1.$$

($u^T x$ je broj!) Odatle, premještanjem pribrojnika, slijedi

$$u = \frac{1}{2(u^T x)}(x - ce_1),$$

tj. u mora biti linearna kombinacija od x i ce_1, tj. mora biti

$$u = \alpha(x - ce_1).$$

Također, zbog unitarne invarijantnosti mora biti

$$\|x\|_2 = \|Hx\|_2 = |c|,$$

pa u mora biti paralelan s vektorom

$$\tilde{u} = x \pm \|x\|_2 e_1, \tag{6.3.3}$$

a jedinične norme, pa je

$$u = \frac{\tilde{u}}{\|\tilde{u}\|_2}.$$

Oba izbora znakova u (6.3.3) zadovoljavaju $Hx = ce_1$, sve dok je $\tilde{u} \neq 0$, ali se u praksi, zbog stabilnosti koristi

$$\tilde{u} = x + \text{sign}(x_1)\|x\|_2 e_1,$$

jer to znači da nema kraćenja pri računanju prve komponente od $\tilde{u}$, koja je jednaka

$$\tilde{u}_1 = x_1 + \text{sign}(x_1)\|x\|_2,$$

tj. oba su pribrojnika istog znaka.

Primijetite da se računanje u može izbjeći, ako definiramo

$$H = I - 2\frac{\tilde{u}\tilde{u}^T}{\tilde{u}^T\tilde{u}}.$$

Ponovno, sustavna primjena Householderovih reflektora na poništavanje elemenata prvog stupca, zatim elemenata drugog stupca od dijagonalnog mjesta nadolje daje konstrukciju QR faktorizacije.

7. Iterativne metode za rješavanje linearnih sustava

7.1. Općenito o iterativnim metodama

Umjesto direktnih metoda za rješavanje linearnih sustava, u praksi se često koriste iterativne metode, posebno za šuplje sustave vrlo velikih redova.

Pretpostavimo da je A regularna matrica reda n. Iterativna metoda koja pronalazi približno rješenje sustava $Ax = b$ zadana je početnim vektorom $x^{(0)}$ i generirana je nizom iteracija $x^{(m)}$, $m \in \mathbb{N}$, koji (nadamo se) konvergira prema rješenju linearnog sustava x.

U praksi se gotovo isključivo koriste iterativne metode prvog reda, koje iz jednog prethodnog vektora $x^{(m)}$ nalaze sljedeću aproksimaciju $x^{(m+1)}$.

Ideja iterativnih metoda je brzo računanje $x^{(m+1)}$ iz $x^{(m)}$. Kriterij zaustavljanja sličan je kao kod svih metoda "limes" tipa – tj. kad je $x^{(m)}$ dovoljno dobra aproksimacija za pravo rješenje x. Naravno, i tu postoji problem, jer pravi x ne znamo. Zbog toga, koristimo svojstvo da je konvergentan niz Cauchyjev, tj. da susjedni članovi niza moraju postati po volji bliski. Standardno, uzima se da su $x^{(m+1)}$ i $x^{(m)}$ dovoljno bliski ako je

$$\|x^{(m+1)} - x^{(m)}\| \leq \varepsilon,$$

gdje je ε neka unaprijed zadana točnost (reda veličine u, odnosno $n \cdot u$), a $\| \ \|$ neka vektorska norma.

Da bismo definirali iterativnu metodu, potrebno je pažljivo rastaviti matricu A.

Definicija 7.1.1. Rastav *matrice A je par matrica (M, K) (obje reda n) za koje vrijedi*

(a) $A = M - K$,

(b) M *je regularna.*

Bilo koji rastav matrice A generira iterativnu metodu na sljedeći način:

$$Ax = Mx - Kx = b \implies Mx = Kx + b,$$

pa zbog regularnosti od M izlazi

$$x = M^{-1}Kx + M^{-1}b.$$

Ako označimo $R = M^{-1}K$, $c = M^{-1}b$ onda je prethodna relacija ekvivalentna s

$$x = Rx + c.$$

Time smo definirali iterativnu metodu

$$x^{(m+1)} = Rx^{(m)} + c, \quad m \in \mathbb{N}_0. \tag{7.1.1}$$

Pravo rješenje x je fiksna točka iteracione funkcije (7.1.1), tj. fiksna točka preslikavanja

$$f(x) = Rx + c.$$

To znači da u analizi konvergencije možemo koristiti poznate teoreme o fiksnoj točki (poput Banachovog u potpunim prostorima).

Kriterij konvergencije ovakvih metoda je jednostavan.

Lema 7.1.1. *Niz iteracija* $(x^{(m)})$, $m \in \mathbb{N}_0$, *generiran relacijom* (7.1.1) *konvergira prema rješenju linearnog sustava* $Ax = b$ *za* **sve početne vektore** $x^{(0)}$ *i* **sve desne strane** b, *ako je*

$$\|R\| < 1,$$

pri čemu je $\|\ \|$ *proizvoljna* **operatorska norma**.

Dokaz:

Oduzmimo $x = Rx + c$ od $x^{(m+1)} = Rx^{(m)} + c$. Dobivamo

$$x^{(m+1)} - x = R(x^{(m)} - x),$$

pa uzimanjem norme dobivamo

$$\|x^{(m+1)} - x\| \leq \|R\|\,\|x^{(m)} - x\| \leq \|R\|^{m+1}\,\|x^{(0)} - x\|.$$

No, zbog $\|R\| < 1$ slijedi $\|R\|^{m+1} \to 0$ za $m \to \infty$, odakle izlazi $\|x - x^{(m+1)}\| \to 0$, pa iz neprekidnosti norme slijedi $x^{(m+1)} \to x$ za svaki $x^{(0)}$. ■

No, može se dobiti i nešto bolji rezultat, korištenjem veze spektralnog radijusa i operatorske norme matrice.

Lema 7.1.2. *Za sve operatorske norme* $\|\ \|$ *vrijedi*

$$\rho(R) \leq \|R\|.$$

Za sve R *i sve* $\varepsilon > 0$ *postoji operatorska norma* $\|\ \|_*$ *takva da je*

$$\|R\|_* \leq \rho(R) + \varepsilon.$$

Norma $\|\ \|_*$ *ovisi i o* R*, i o* ε*.*

Dokaz:

Da bismo dokazali da je $\rho(R) \leq \|R\|$ za bilo koju operatorsku normu, neka je $\rho(R) = |\lambda|$, a x pripadni svojstveni vektor od λ. Tada vrijedi

$$\|R\| = \max_{y \neq 0} \frac{\|Ry\|}{\|y\|} \geq \frac{\|Rx\|}{\|x\|} = \frac{\|\lambda x\|}{\|x\|} = \frac{|\lambda|\,\|x\|}{\|x\|} = |\lambda| = \rho(R).$$

S druge strane, da bismo konstruirali operatorsku normu $\|\ \|_*$ takvu da je $\|R\|_* \leq \rho(R) + \varepsilon$, svedimo matricu R na njenu Jordanovu formu. Znamo da postoji regularna matrica S za koju je $J = S^{-1}RS$ Jordanova forma matrice R. Jedinice iznad dijagonale u J "kvare" spektralni radijus, pa je ideja da umjesto njih dobijemo zadani $\varepsilon > 0$. To se postiže dijagonalnom matricom sličnosti. Neka je

$$D_\varepsilon = \operatorname{diag}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1}).$$

Tada je

$$(SD_\varepsilon)^{-1} R (SD_\varepsilon) = D_\varepsilon^{-1} J D_\varepsilon = \left[\begin{array}{ccccc:ccc:c} \lambda_1 & \varepsilon & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & \ddots & \varepsilon & & & & & \\ & & & \lambda_1 & & & & & \\ \hdashline & & & & & & & & \\ & & & & & \lambda_2 & \varepsilon & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & \ddots & \varepsilon \\ & & & & & & & & \lambda_2 \\ \hdashline & & & & & & & & \ddots \end{array}\right].$$

Uzmimo kao vektorsku normu

$$\|x\|_* = \|(SD_\varepsilon)^{-1} x\|_\infty$$

(pokažite da je to vektorska norma!), koja generira operatorsku normu. U toj operatorskoj normi vrijedi

$$\|R\|_* = \max_{x\neq 0} \frac{\|Rx\|_*}{\|x\|_*} = \max_{x\neq 0} \frac{\|(SD_\varepsilon)^{-1}Rx\|_\infty}{\|(SD_\varepsilon)^{-1}x\|_\infty} = \max_{y\neq 0} \frac{\|(SD_\varepsilon)^{-1}R(SD_\varepsilon)y\|_\infty}{\|y\|_\infty}$$
$$= \|(SD_\varepsilon)^{-1}R(SD_\varepsilon)\|_\infty \leq \max_i |\lambda_i| + \varepsilon = \rho(R) + \varepsilon.$$

■

Konačno, prethodne dvije leme daju potpunu karakterizaciju konvergencije iterativnih metoda.

Teorem 7.1.1. *Niz iteracija* $(x^{(m)})$, $m \in \mathbb{N}_0$, *generiran relacijom* (7.1.1) *konvergira prema rješenju linearnog sustava* $Ax = b$ *za* **sve početne vektore** $x^{(0)}$ *i* **sve desne strane** b, *ako i samo ako je*

$$\rho(R) < 1,$$

pri čemu je $\rho(R)$ *spektralni radijus matrice* $R = M^{-1}K$. *Uočite da* $\rho(R)$ *ovisi samo o* A *i njenom rastavu, a ne o* b.

Dokaz:

Ako je $\rho(R) \geq 1$, izaberimo startnu aproksimaciju $x^{(0)}$ tako da je $x^{(0)} - x$ svojstveni vektor koji pripada svojstvenoj vrijednosti λ, $\rho(R) = |\lambda|$. Tada vrijedi

$$(x^{(m+1)} - x) = R(x^{(m)} - x) = \cdots = R^{m+1}(x^{(0)} - x) = \lambda^{m+1}(x^{(0)} - x),$$

pa, očito $(m+1)$-a potencija broja koji je po apsolutnoj vrijednosti veći ili jednak 1, ne može težiti u 0.

S druge strane, ako je $\rho(R) < 1$, onda možemo izabrati $\varepsilon > 0$ takav da je

$$\rho(R) + \varepsilon < 1,$$

a zatim po lemi 7.1.2. i operatorsku normu $\| \ \|_*$ takvu da vrijedi

$$\|R\|_* \leq \rho(R) + \varepsilon < 1. \tag{7.1.2}$$

Budući da je $\| \ \|_*$ operatorska norma, ponovno, po prvom dijelu leme 7.1.2. slijedi da je

$$\rho(R) \leq \|R\|_*. \tag{7.1.3}$$

Relacije (7.1.2) i (7.1.3) zajedno daju da je

$$\|R\|_* < 1,$$

pa primjenom leme 7.1.1. dobivamo traženi rezultat. ■

Lema 7.1.1. i teorem 7.1.1., zapravo nam daju i brzinu konvergencije. Prisjetimo se što je brzina konvergencije.

Definicija 7.1.2. *Za niz aproksimacija $x^{(m)}$, $m \in \mathbb{N}_0$, reći ćemo da konvergira prema x s redom p ako je*

$$\|x^{(m+1)} - x\| \leq c\|x^{(m)} - x\|^p, \quad c \in \mathbb{R}_0^+.$$

Ako je $p = 1$ (tzv. linearna konvergencija), mora biti $c < 1$ (tzv. geometrijska konvergencija s faktorom c).

U slučaju iterativnih metoda, konvergencija je linearna, a faktor je $\rho(R) < 1$, tj. vrijedi

$$\|x^{(m+1)} - x\|_* \leq \rho(R)\|x^{(m)} - x\|_*.$$

Podijelimo li prethodnu relaciju s $\rho(R) \cdot \|x^{(m+1)} - x\|_*$, a zatim logaritmiramo, dobivamo

$$-\log \rho(R) \leq \log \|x^{(m)} - x\|_* - \log \|x^{(m+1)} - x\|_*, \tag{7.1.4}$$

pa zbog toga broj

$$r(R) = -\log \rho(R)$$

možemo definirati kao brzinu konvergencije iteracija. Što nam kaže relacija (7.1.4)? Broj $r(R)$ je porast broja korektnih decimalnih znamenki u rješenju po iteraciji. Dakle, što je manji $\rho(R)$, to je veća brzina konvergencije iteracija.

Naravno, sljedeći cilj nam je odgovoriti na pitanje kako odrediti rastav matrice $A = M - K$ koji zadovoljava

(1) $Rx = M^{-1}Kx$ i $c = M^{-1}b$ se lako računaju,

(2) $\rho(R)$ je malen?

Odmah nam se nameću neka jednostavna rješenja za ova dva suprotna cilja. Na primjer, izaberemo li $M = I$, M je regularna, i Rx i c se lako računaju, ali nije jasno da smo zadovoljili da je $\rho(R) < 1$. S druge strane, izbor $K = 0$ je izvrsan za drugi cilj ($\rho(R) = 0$), ali nije dobar za prvi cilj, jer je $c = A^{-1}b$, tj. dobivamo polazni problem kojeg treba riješiti ($x = c$).

Dakle, rastav koji bi uvijek dobro radio nije lako konstruirati. Međutim tu će nam pomoći praksa. Matrice koje se javljaju u praksi su ili pozitivno definitne ili (strogo) dijagonalno dominantne, pa će za takve tipove matrica biti mnogo lakše konstruirati iterativne metode i pokazati da one konvergiraju.

Uvedimo sljedeću notaciju. Pretpostavimo da A nema nula na dijagonali. Tada A možemo zapisati kao

$$A = D - \widetilde{L} - \widetilde{U} = D(I - L - U),$$

pri čemu je D dijagonala od A, $-\widetilde{L}$ striktno donji trokut od A, a $-\widetilde{U}$ striktno gornji trokut od A. Jednako tako, vrijedi $DL = \widetilde{L}$, $DU = \widetilde{U}$.

7.2. Jacobijeva metoda

Općenito govoreći Jacobijeva metoda u petlji prolazi kroz jednadžbe linearnog sustava, mijenjajući j-tu varijablu, tako da j-ta jednadžba bude ispunjena. Dakle, u $(m+1)$-om koraku vrijednost varijable x_j, u oznaci $x_j^{(m+1)}$, računamo iz j-te jednadžbe korištenjem aproksimacija iz m-tog koraka za preostale varijable, tj. vrijedi

$$a_{jj}x_j^{(m+1)} + \sum_{\substack{k=1\\k\neq j}}^{n} a_{jk}x_k^{(m)} = b_j, \quad j = 1, \ldots, n. \tag{7.2.1}$$

Naravno, to ide ako i samo ako je $a_{jj} \neq 0$ za sve j.

Vidimo da na nove komponente djeluju samo dijagonalni elementi matrice A, dok svi ostali djeluju na stare (prethodne ili prošle) komponente. Skupimo li sve jednadžbe iz (7.2.1) za jedan korak, onda ih zajedno možemo zapisati kao

$$Dx^{(m+1)} = (\widetilde{L} + \widetilde{U})x^{(m)} + b,$$

ili

$$x^{(m+1)} = D^{-1}(\widetilde{L} + \widetilde{U})x^{(m)} + D^{-1}b := R_{Jac}x^{(m)} + c_{Jac},$$

uz

$$R_{Jac} = D^{-1}(\widetilde{L} + \widetilde{U}) = L + U, \quad c_{Jac} = D^{-1}b.$$

Pripadni rastav matrice A je

$$A = D - (\widetilde{L} + \widetilde{U}),$$

tj. $M = D$, $K = \widetilde{L} + \widetilde{U}$.

Algoritam 7.2.1. (Jedan korak Jacobijeve metode)

for $j := 1$ **to** n **do**
$$x_j^{(m+1)} := \left(b_j - \sum_{\substack{k=1\\k\neq j}}^{n} a_{jk}x_k^{(m)}\right) \Big/ a_{jj};$$

Uočimo da petlju po j u ovom algoritmu, odnosno prolaz kroz jednadžbe sustava u (7.2.1), možemo napraviti **bilo kojim** redom po j — ne nužno sekvencijalnim. Komponente novog vektora $x^{(m+1)}$ ovise samo o komponentama starog vektora $x^{(m)}$, a ne i o nekim novim komponentama. Zbog toga je Jacobijeva metoda idealna za paralelno računanje, jer pojedine komponente novog vektora možemo računati potpuno nezavisno.

7.3. Gauss–Seidelova metoda

Ako komponente novog vektora u Jacobijevoj metodi zaista računamo sekvencijalno, od prve prema zadnjoj, odmah se nameće ideja za poboljšanje. Naime, u aproksimaciji j-te varijable u $(m+1)$-om koraku koristimo aproksimaciju svih ostalih varijabli iz prethodnog m-tog koraka, iako već imamo poboljšane varijable $x_i^{(m+1)}$, za $i < j$, iz novog koraka. Iskoristimo li sve poznate nove komponente umjesto starih, onda (7.2.1) glasi

$$a_{jj}x_j^{(m+1)} + \sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} + \sum_{k=j+1}^{n} a_{jk}x_k^{(m)} = b_j, \quad j = 1, \ldots, n. \tag{7.3.1}$$

Ovu metodu zovemo Gauss–Seidelova metoda. Poredak prolaska kroz jednadžbe sustava postaje potpuno određen i strogo sekvencijalan, od prve prema zadnjoj. Opet, to ide ako i samo ako je $a_{jj} \neq 0$ za sve j.

Na nove komponente djeluju, osim dijagonalnih, i svi elementi donjeg trokuta matrice A, dok samo strogo gornji trokut djeluje na stare komponente. Sve jednadžbe iz (7.3.1) za jedan korak iteracija možemo zajedno zapisati u obliku

$$(D - \widetilde{L})x^{(m+1)} = \widetilde{U}x^{(m)} + b,$$

ili

$$x^{(m+1)} = (D - \widetilde{L})^{-1}\widetilde{U}x^{(m)} + (D - \widetilde{L})^{-1}b := R_{GS}x^{(m)} + c_{GS},$$

uz

$$R_{GS} = (D - \widetilde{L})^{-1}\widetilde{U} = (I - L)^{-1}U, \quad c_{GS} = (D - \widetilde{L})^{-1}b = (I - L)^{-1}D^{-1}b.$$

Matrica R_{GS} je singularna, jer je U strogo gornja trokutasta ($\det U = 0$). Pripadni rastav matrice A je

$$A = (D - \widetilde{L}) - \widetilde{U},$$

tj. $M = D - \widetilde{L}$, $K = \widetilde{U}$.

Algoritam 7.3.1. (Jedan korak Gauss–Seidelove metode)

for $j := 1$ **to** n **do**

$$x_j^{(m+1)} := \left(b_j - \sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} - \sum_{k=j+1}^{n} a_{jk}x_k^{(m)}\right) \Big/ a_{jj};$$

Zgodna je stvar da u implementaciji Gauss–Seidelove metode ne moramo pamtiti dva vektora, već samo jedan, tako da nove izračunate komponente prepisujemo preko starih u istom polju x.

S druge strane, redosljed "popravaka" varijabli u Jacobijevom algoritma ne igra nikakvu ulogu, u smislu da bilo koji poredak izvršavanja petlje po j daje isti

rezultat i, naravno, istu iterativnu metodu. Nasuprot tome, redosljed "popravaka" kod Gauss–Seidelovog algoritma je bitan. U (7.3.1) i u algoritmu 7.3.1. uzeli smo prirodni redosljed — od prve prema zadnjoj jednadžbi, odnosno, varijabli (petlja po j od 1 do n).

To **ne** znači da je to i jedini mogući redosljed. U principu, možemo uzeti i bilo koji drugi redosljed, tj. bilo koju drugu od mogućih $n!$ permutacija jednadžbi. No, zbog sekvencijalnosti popravaka, rezultat nije isti, tj. dobivamo drugačiju iterativnu metodu. U praksi se katkad i koriste drugačiji redosljedi, ali za sasvim posebne linearne sustave koji nastaju diskretizacijom parcijalnih diferencijalnih jednadžbi. Na primjer, za Laplaceovu jednadžbu u dvije dimenzije koristi se tzv. "crveno–crni" poredak (engl. "red–black" ordering), koji naliči šahovskoj ploči s crvenim i crnim poljima, s tim da se prvo računaju sva polja crvene boje, a zatim sva polja crne boje. Zbog posebne strukture sustava, ovaj poredak dozvoljava i efikasnu paralelizaciju.

7.4. JOR metoda (Jacobi overrelaxation)

Kad jednom znamo konstruirati iterativni proces, nameće se vrlo jednostavna ideja za njegovo poboljšanje, uvođenjem jednog realnog parametra. Nove aproksimacije možemo računati u dva koraka. Prvo iz $x^{(m)}$ nađemo (jednostavnu) pomoćnu sljedeću aproksimaciju $x_*^{(m+1)}$, a zatim za "pravu" novu aproksimaciju $x^{(m+1)}$ uzmemo težinsku sredinu prethodne aproksimacije $x^{(m)}$ i pomoćne nove aproksimacije $x_*^{(m+1)}$

$$x^{(m+1)} = (1-\omega)x^{(m)} + \omega x_*^{(m+1)} = x^{(m)} + \omega(x_*^{(m+1)} - x^{(m)}), \tag{7.4.1}$$

gdje je ω težinski parametar kojeg možemo birati. Očito, za $\omega = 1$ dobivamo $x^{(m+1)} = x_*^{(m+1)}$, pa je ova metoda proširenje metode za nalaženje pomoćnih aproksimacija. Obično se uzima $\omega \in \mathbb{R}$ i $\omega \neq 0$, da ne dobijemo stacionaran niz.

Ideja za ovakav postupak dolazi iz općih metoda za rješavanje jednadžbi i minimizaciju funkcionala. Pomoćna aproksimacija $x_*^{(m+1)}$ daje **smjer** korekcije prethodne aproksimacije $x^{(m)}$ u kojem treba ići da bismo se približili pravom rješenju sustava ili smanjili rezidual $r(x) = Ax - b$ u nekoj normi. No, ako je $x_*^{(m+1)} - x^{(m)}$ dobar smjer korekcije, onda možemo dodati i izbor duljine koraka ω u smjeru te korekcije, tako da dobijemo što bolji $x^{(m+1)}$. Općenito očekujemo da je $\omega > 0$, tj. da idemo u smjeru vektora korekcije, a ne suprotno od njega, a stvarno želimo dobiti $\omega > 1$, tako da se još više maknemo u dobrom smjeru i približimo pravom rješenju ili točki minimuma.

Naziv "relaksacija" dolazi upravo iz minimizacijskih metoda, a odnosi se na sve iterativne metode koje koriste neki oblik minimizacije ili pokušaja minimizacije reziduala. U tom smislu, često se koristi i tradicionalni naziv "relaksacijski parametar" za ω.

Obzirom na vrijednost parametra ω, imamo tri različita slučaja. Ako je $\omega = 1$, onda se metoda svodi na pomoćnu metodu i to je tzv. obična ili standardna relaksacija. Ako je $\omega < 1$, onda takvu metodu zovemo podrelaksacija (engl. "underrelaxation"), a ako je $\omega > 1$ onda metodu zovemo nad- ili pre-relaksacija (engl. "overrelaxation").

U općem slučaju, ω se posebno računa u svakoj pojedinoj iteraciji, tako da dobijemo što bolji $x^{(m+1)}$. Postupak se svodi na jednodimenzionalnu optimizaciju (kao i određivanje koraka u višedimenzionalnoj optimizaciji), a ovisi o kriteriju optimalnosti za mjerenje "kvalitete" aproksimacija.

Srećom, za neke klase linearnih sustava, koje su izrazito bitne u praksi, unaprijed se može dobro odabrati optimalni ili skoro optimalni parametar ω za maksimalno ubrzanje konvergencije iterativnih metoda i to tako da isti ω vrijedi za sve iteracije. U većini slučajeva dobivamo $\omega > 1$ za optimalni ω, pa se takve metode standardno zovu "OverRelaxation" i skraćeno označavaju s OR. Obzirom na to da se ω zadaje ili bira unaprijed, a zatim koristi za sve iteracije, metoda je ovisna o jednom parametru i standardno koristimo oznaku OR(ω).

Ako se pomoćna nova aproksimacija $x_*^{(m+1)}$ iz (7.4.1) računa po Jacobijevoj metodi, dobivamo Jacobijevu nadrelaksaciju ili JOR metodu. Iz (7.2.1) za komponente pomoćne aproksimacije $x_{Jac}^{(m+1)}$ vrijedi

$$x_{j,Jac}^{(m+1)} = \Big(b_j - \sum_{\substack{k=1\\k\neq j}}^{n} a_{jk}x_k^{(m)}\Big)\Big/a_{jj}, \quad j = 1, \ldots, n.$$

Kad to uvrstimo u (7.4.1) dobivamo sljedeći algoritam.

Algoritam 7.4.1. (Jedan korak JOR(ω) metode)

for $j := 1$ **to** n **do**
$$x_j^{(m+1)} := (1-\omega)x_j^{(m)} + \frac{\omega}{a_{jj}}\Big(b_j - \sum_{\substack{k=1\\k\neq j}}^{n} a_{jk}x_k^{(m)}\Big);$$

Kao i kod obične Jacobijeve metode, petlju po j možemo izvršiti bilo kojim redom po j, uz isti rezultat, pa se i ovdje komponente novog vektora $x^{(m+1)}$ mogu paralelno računati.

Vektorski oblik iteracija u JOR(ω) metodi je

$$x^{(m+1)} = (1-\omega)x^{(m)} + \omega(R_{Jac}x^{(m)} + c_{Jac}) := R_{JOR(\omega)}x^{(m)} + c_{JOR(\omega)}$$

pa je

$$R_{JOR(\omega)} = (1-\omega)I + \omega R_{Jac} = (1-\omega)I + \omega(L+U),$$
$$c_{JOR(\omega)} = \omega c_{Jac} = \omega D^{-1}b.$$

Pripadni rastav matrice A je

$$A = \frac{1}{\omega}D - \Big(\frac{1-\omega}{\omega}D + \widetilde{L} + \widetilde{U}\Big),$$

tj. $M = \frac{1}{\omega}D$, $K = \frac{1-\omega}{\omega}D + \widetilde{L} + \widetilde{U}$. Naravno, za $\omega = 1$ dobivamo Jacobijevu metodu.

Iz oblika matrice $R_{JOR(\omega)}$ vidimo da se optimalni parametar ω koji minimizira njen spektralni radijus može odrediti unaprijed, prije početka iteracija. Drugim riječima, treba koristiti isti ω u svim iteracijama, naravno, pod uvjetom da imamo konvergenciju.

7.5. SOR metoda (Successive overrelaxation)

Relacija (7.4.1) koristi ideju težinske sredine ili duljine koraka na nivou vektorskih aproksimacija $x^{(m)}$. To prirodno odgovara Jacobijevoj metodi i paralelnom računanju. Međutim, potpuno istu ideju možemo koristiti i za poboljšanje svake pojedine varijable $x_j^{(m)}$, tj. pojedinačnih komponenti vektora $x^{(m)}$, što odgovara "Gauss–Seidelovskom" pristupu. Dakle, nova aproksimacija j-te varijable ima oblik

$$x_j^{(m+1)} = (1-\omega)x_j^{(m)} + \omega x_{j,*}^{(m+1)} = x_j^{(m)} + \omega(x_{j,*}^{(m+1)} - x_j^{(m)}), \quad j = 1, \ldots, n, \quad (7.5.1)$$

gdje je $x_{j,*}^{(m+1)}$ neka pomoćna nova aproksimacija j-te varijable, koju računamo tog trenutka kad nam treba, za svaki pojedini j.

SOR metoda (engl. "Successive OverRelaxation" ili ponovljena nad- ili prerelaksacija) je proširenje ili poboljšanje Gauss–Seidelove metode u smislu da se pomoćna nova aproksimacija $x_{j,*}^{(m+1)}$ iz (7.5.1) računa po Gauss–Seidelovoj metodi, pa ju označavamo s $x_{j,GS}^{(m+1)}$. Relacija (7.5.1) za j-tu komponentu nove aproksimacije u SOR(ω) metodi ima oblik

$$x_j^{(m+1)} = (1-\omega)x_j^{(m)} + \omega x_{j,GS}^{(m+1)}, \quad j = 1, \ldots, n. \quad (7.5.2)$$

Iz (7.3.1), trenutna pomoćna Gauss–Seidelova aproksimacija $x_{j,GS}^{(m+1)}$ koju možemo izračunati iz poznatih prvih $j-1$ komponenti novog vektora $x^{(m+1)}$ i preostalih komponenti iz prethodne aproksimacije $x^{(m)}$ je

$$x_{j,GS}^{(m+1)} = \Big(b_j - \sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} - \sum_{k=j+1}^{n} a_{jk}x_k^{(m)}\Big)\Big/ a_{jj}.$$

Kad to uvrstimo u (7.5.2), možemo izračunati j-tu komponentu $x_j^{(m+1)}$ nove aproksimacije po SOR(ω) metodi.

Algoritam 7.5.1. (Jedan korak SOR(ω) metode)

for $j := 1$ **to** n **do**

$$x_j^{(m+1)} := (1-\omega)x_j^{(m)} + \frac{\omega}{a_{jj}}\Big(b_j - \sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} - \sum_{k=j+1}^{n} a_{jk}x_k^{(m)}\Big);$$

Iz algoritma odmah vidimo da je

$$a_{jj}x_j^{(m+1)} + \omega\sum_{k=1}^{j-1} a_{jk}x_k^{(m+1)} = (1-\omega)a_{jj}x_j^{(m)} - \omega\sum_{k=j+1}^{n} a_{jk}x_k^{(m)} + \omega b_j, \quad j = 1,\dots,n.$$

Ove jednadžbe možemo zapisati u vektorskom obliku

$$(D - \omega\widetilde{L})x^{(m+1)} = ((1-\omega)D + \omega\widetilde{U})x^{(m)} + \omega b,$$

odakle slijedi

$$R_{SOR(\omega)} = (I - \omega L)^{-1}((1-\omega)I + \omega U),$$
$$c_{SOR(\omega)} = \omega(I - \omega L)^{-1}D^{-1}b.$$

I ovdje vidimo da se dobra vrijednost za ω, ako postoji, može odrediti unaprijed za sve iteracije.

Nakon analize konvergencije ovih iterativnih metoda, pokazat ćemo da za neke klase matrica možemo naći optimalni izbor parametra ω koji ubrzava konvergenciju, i da vrijedi $\omega > 1$, što opravdava naziv OR u ovim metodama.

7.6. Konvergencija Jacobijeve i Gauss–Seidelove metode

U ovom poglavlju nabrojit ćemo, a u većini slučajeva i dokazati, koji su dovoljni uvjeti za konvergenciju pojedine metode.

Teorem 7.6.1. *Ako je A strogo dijagonalno dominantna matrica po recima, onda i Jacobijeva i Gauss–Seidelova metoda konvergiraju i vrijedi*

$$\|R_{GS}\|_\infty \le \|R_{Jac}\|_\infty < 1.$$

Dokaz:

Prije dokaza, uočimo da relacija $\|R_{GS}\|_\infty \le \|R_{Jac}\|_\infty$ znači da jedan korak u najgorem slučaju (problemu) za Gauss–Seidelovu metodu konvergira barem jednako brzo kao i jedan korak u najgorem slučaju za Jacobijevu metodu. To ne znači da će Gauss–Seidelova metoda konvergirati brže nego Jacobijeva za bilo koji problem $Ax = b$. Naprosto, može se dogoditi da Jacobijeva metoda u nekom koraku "slučajno" ima manju grešku.

Prvo pokažimo da je $\|R_{Jac}\|_\infty < 1$. Zbog stroge dijagonalne dominantnosti po recima

$$|a_{jj}| > \sum_{\substack{k=1 \\ k\neq j}}^{n} |a_{kj}|, \quad j = 1, \ldots, n$$

vrijedi da je

$$1 > \max_j \frac{1}{|a_{jj}|} \sum_{\substack{k=1 \\ k\neq j}}^{n} |a_{kj}| = \|R_{Jac}\|_\infty.$$

Sada nam preostaje dokazati da je $\|R_{GS}\|_\infty \leq \|R_{Jac}\|_\infty$.

Matrice iteracija možemo napisati u obliku $R_{Jac} = L+U$, i $R_{GS} = (I-L)^{-1}U$. Želimo dokazati da vrijedi

$$\|R_{GS}\|_\infty = \| \, |R_{GS}| \, e\|_\infty \leq \| \, |R_{Jac}| \, e\|_\infty = \|R_{Jac}\|_\infty, \tag{7.6.1}$$

pri čemu je e vektor sa svim komponentama jednakim 1, tj. $e = (1, \ldots, 1)^T$. Ovu relaciju možemo lako pokazati ako dokažemo jaču komponentnu nejednakost

$$|(I-L)^{-1}U| \cdot e = |R_{GS}| \cdot e \leq |R_{Jac}| \cdot e = (|L| + |U|) \cdot e. \tag{7.6.2}$$

Krenimo slijeva i iskoristimo relaciju trokuta. Vrijedi

$$|(I-L)^{-1}U| \cdot e \leq |(I-L)^{-1}| \, |U| \cdot e. \tag{7.6.3}$$

Ako je $\rho(L) < 1$, (a je, jer je $\rho(L) = 0$!), onda je $I - L$ regularna i možemo $(I-L)^{-1}$ razviti u red

$$(I-L)^{-1} = I + L + L^2 + \cdots + L^n + \cdots$$

Primijetimo da je L strogo donjetrokutasta (nilpotentna) matrica, pa je $L^k = 0$ za $k \geq n$, pa prethodni red postaje

$$(I-L)^{-1} = I + L + L^2 + \cdots + L^{n-1}.$$

Uvrstimo to u (7.6.3), pa ponovnim korištenjem relacije trokuta i formule za inverz matrice oblika $(I - A)$, dobivamo

$$|(I-L)^{-1}U| \cdot e \leq \Big| \sum_{i=0}^{n-1} L^i \Big| \, |U| \cdot e \leq \sum_{i=0}^{n-1} |L|^i \, |U| \cdot e \leq (I - |L|)^{-1} \, |U| \cdot e.$$

Relacija (7.6.2) će vrijediti ako dokažimo još jaču komponentnu nejednakost

$$(I - |L|)^{-1} \, |U| \cdot e \leq (|L| + |U|) \cdot e. \tag{7.6.4}$$

Budući da su svi članovi u redu za $(I - |L|)^{-1}$ nenegativni, dovoljno je pokazati da je

$$|U| \cdot e \leq (I - |L|) \, (|L| + |U|) \cdot e = (|L| + |U| - |L|^2 - |L| \, |U|) \cdot e,$$

odnosno

$$0 \leq (|L| - |L|^2 - |L|\,|U|) \cdot e = |L|\,(I - |L| - |U|) \cdot e. \tag{7.6.5}$$

Ponovno, budući da su svi elementi $|L|$ nenegativni, prethodna nejednakost bit će ispunjena ako je

$$0 \leq (I - |L| - |U|) \cdot e,$$

odnosno

$$(|L| + |U|) \cdot e \leq e.$$

Budući da je $|R_{Jac}| = |L + U| = |L| + |U|$, jer se elementi L i U nigdje ne zbrajaju, onda je posljednja nejednakost ekvivalentna s

$$|R_{Jac}| \cdot e \leq e.$$

Ovu posljednju jednakost možemo dokazati jer je

$$\| \, |R_{Jac}| \cdot e\|_\infty = \|R_{Jac}\|_\infty < 1.$$

Čitanjem dokaza u obratnom redosljedu, dobivamo tražena svojstva (7.6.4), (7.6.2), a onda i (7.6.1). Iako je zadnja nejednakost stroga, kad se vraćamo unatrag, nejednakost u (7.6.5) više ne mora biti stroga, na primjer, za $L = 0$. Dakle, može biti $\|R_{GS}\|_\infty = \|R_{Jac}\|_\infty$. ■

Analogni se rezultat, tj. da Jacobijeva i Gauss–Seidelova metoda konvergiraju, može se dobiti i za strogo dijagonalno dominantne matrice po stupcima, samo onda ulogu norme ∞ igra norma 1.

Uvjeti prethodnog teorema mogu se još malo oslabiti, do ireducibilnosti i slabe dijagonalne dominantnosti. Da bismo definirali ireducibilnu matricu, moramo definirati vezu između matrica i grafova.

Definicija 7.6.1. *Matrici A odgovara usmjereni graf $G(A)$ s čvorovima $1, \ldots, n$. Usmjereni brid tog grafa od čvora i do čvora j postoji ako i samo ako je $a_{ij} \neq 0$.*

Definicija 7.6.2. *Usmjereni graf je jako povezan ako postoji put iz svakog čvora u svaki čvor. Komponenta jake povezanosti usmjerenog grafa je podgraf koji je jako povezan i ne može se povećati, a da ostane jako povezan.*

Definicija 7.6.3. *Matrica A je reducibilna ako i samo ako postoji matrica permutacije P takva da je*

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

pri čemu su A_{11} i A_{22} kvadratni blokovi reda manjeg od n, tj. PAP^T je blok gornjetrokutasta matrica. Matrica A je ireducibilna, ako nije reducibilna, tj. ako ne postoji matrica permutacije P za koju je PAP^T blok gornjetrokutasta matrica.

Najjednostavnija karakterizacija ireducibilnih matrica je preko jako povezanih pripadnih grafova.

Lema 7.6.1. *Matrica A je ireducibilna ako i samo ako je $G(A)$ jako povezan.*

Dokaz:

Ako je A reducibilna

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

onda ne postoji "povratak" iz čvorova koji odgovaraju A_{22} u čvorove koji odgovaraju A_{11}, pa $G(A)$ nije jako povezan. Obratno, ako $G(A)$ nije jako povezan, postoji komponenta jake povezanosti koja ne sadrži sve čvorove grafa. Ako matricu prepermutiramo tako da čvorovi iz te komponente dođu na početak (u blok A_{11}), dobit ćemo traženi blok gornjetrokutasti oblik. ■

Definicija 7.6.4. *Matrica A je slabo dijagonalno dominantna po recima ako za svaki j vrijedi*

$$|a_{jj}| \geq \sum_{\substack{k=1 \\ k \neq j}}^{n} |a_{kj}|,$$

a stroga nejednakost se javlja barem jednom.

Sada možemo oslabiti uvjete teorema 7.6.1., s tim da ovaj rezultat navodimo bez dokaza.

Teorem 7.6.2. *Ako je A ireducibilna i slabo dijagonalno dominantna matrica po recima, onda i Jacobijeva i Gauss–Seidelova metoda konvergiraju i vrijedi*

$$\|R_{GS}\|_\infty \leq \|R_{Jac}\|_\infty < 1.$$

Unatoč navedenim rezultatima da je pod nekim uvjetima Gauss–Seidelova metoda brža nego Jacobijeva, ne postoji nikakav generalni rezultat te vrste. Dapače, postoje nesimetrične matrice za koje Jacobijeva metoda konvergira, a Gauss–Seidelova ne, kao i matrice za koje Gauss–Seidelova metoda konvergira, a Jacobijeva divergira.

7.7. Konvergencija JOR i SOR metode

Promotrimo sad konvergenciju relaksacijskih metoda JOR(ω) i SOR(ω) u ovisnosti o parametru ω. Obzirom na to da se ove metode za $\omega = 1$ svode na Jacobijevu, odnosno, Gauss–Seidelovu metodu, usput ćemo dobiti i neke rezultate o konvergenciji ovih osnovnih metoda.

Za početak, promatramo JOR(ω) metode, jer su bitno jednostavnije za analizu.

Teorem 7.7.1. *Ako Jacobijeva metoda za rješenje linearnog sustava $Ax = b$ konvergira za svaku početnu iteraciju $x^{(0)}$, onda za bilo koji $\omega \in (0, 1]$ konvergira i JOR(ω) metoda za svaku početnu iteraciju.*

Dokaz:

Prema teoremu 7.1.1., pretpostavka o konvergenciji Jacobijeve metode za svaku početnu iteraciju ekvivalentna je s činjenicom da vrijedi $\rho(R_{Jac}) < 1$. Neka su μ_j, $j = 1, \ldots, n$, svojstvene vrijednosti matrice R_{Jac}. Onda je $|\mu_j| < 1$ za sve j. Ako μ_j napišemo kao kompleksne brojeve u obliku $\mu_j = \alpha_j + i\beta_j$, uz α_j, $\beta_j \in \mathbb{R}$, onda iz $|\mu_j| < 1$ slijedi $\alpha_j^2 + \beta_j^2 < 1$ i $|\alpha_j| < 1$.

Matrica iteracije u JOR(ω) metodi je

$$R_{JOR(\omega)} = (1 - \omega)I + \omega R_{Jac} = (1 - \omega)I + \omega(L + U),$$

što je polinom u funkciji od R_{Jac}, pa za svojstvene vrijednosti λ_j matrice $R_{JOR(\omega)}$ vrijedi

$$\lambda_j = 1 - \omega + \omega\mu_j, \quad j = 1, \ldots, n,$$

odakle izlazi

$$|\lambda_j|^2 = |(1 - \omega + \omega\alpha_j) + i\omega\beta_j|^2 = (1 - \omega + \omega\alpha_j)^2 + \omega^2\beta_j^2.$$

Ako je $0 < \omega \leq 1$, onda vrijedi ocjena

$$\begin{aligned} |\lambda_j|^2 &\leq (1 - \omega)^2 + 2\omega(1 - \omega)|\alpha_j| + \omega^2(\alpha_j^2 + \beta_j^2) \\ &< (1 - \omega)^2 + 2\omega(1 - \omega) + \omega^2 = (1 - \omega + \omega)^2 = 1, \end{aligned}$$

odakle slijedi $\rho(R_{JOR(\omega)}) < 1$, a to znači da JOR($\omega$) metoda konvergira za svaku početnu iteraciju. ■

Prethodni rezultat daje samo uvjetnu konvergenciju, u smislu da ako metoda konvergira za $\omega = 1$, onda konvergira i za sve ω iz skupa $(0, 1]$. Precizniju, ali negativnu informaciju daje sljedeći rezultat.

Teorem 7.7.2. *Vrijedi $\rho(R_{JOR(\omega)}) \geq |\omega - 1|$, pa je $0 < \omega < 2$ nužan uvjet za konvergenciju JOR metode.*

Dokaz:

Znamo da je trag matrice jednak zbroju svih njezinih svojstvenih vrijednosti. Promotrimo trag matrice

$$R_{JOR(\omega)} = (1 - \omega)I + \omega R_{Jac} = (1 - \omega)I + \omega(L + U).$$

Drugi član $\omega(L + U)$ ima nul-dijagonalu, pa samo prvi član daje trag

$$\operatorname{tr}(R_{JOR(\omega)}) = n(1 - \omega) = \sum_{j=1}^{n} \lambda_j.$$

Dobivamo da je

$$n|1-\omega| \leq \sum_{j=1}^{n} |\lambda_j| \leq n\rho(R_{JOR(\omega)}),$$

odakle slijedi $\rho(R_{JOR(\omega)}) \geq |\omega - 1|$. Dakle, za konvergenciju JOR(ω) metode mora vrijediti $|\omega - 1| < 1$, ili $0 < \omega < 2$. U protivnom, za neke $x^{(0)}$ metoda divergira. ■

Za simetrične (hermitske) i pozitivno definitne matrice A, možemo dobiti i pozitivnu informaciju o garantiranoj konvergenciji JOR metode.

Teorem 7.7.3. *Neka je A simetrična (hermitska) i pozitivno definitna matrica i neka za svojstvene vrijednosti μ_j matrice R_{Jac} Jacobijeve metode vrijedi*

$$\mu_j < 1, \quad j = 1, \ldots, n.$$

Onda JOR(ω) metoda konvergira za sve parametre ω za koje vrijedi

$$0 < \omega < \frac{2}{1-\mu} \leq 2, \tag{7.7.1}$$

gdje je $\mu := \min_j \mu_j$ najmanja svojstvena vrijednost matrice R_{Jac}.

Dokaz:

Prvo uočimo da je $R_{Jac} = D^{-1}(\widetilde{L} + \widetilde{L}^*) = D^{-1/2}(D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2})D^{1/2}$ slična simetričnoj (hermitskoj) matrici $D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2}$, pa su joj svojstvene vrijednosti μ_j realne. Već smo vidjeli da za svojstvene vrijednosti λ_j matrice $R_{JOR(\omega)}$ vrijedi

$$\lambda_j = 1 - \omega + \omega\mu_j, \quad j = 1, \ldots, n,$$

pa je $1 - \lambda_j = \omega(1 - \mu_j)$, odakle slijedi da je $|\lambda_j| < 1$ za sve j, ako i samo ako vrijedi

$$0 < \omega(1 - \mu_j) < 2, \quad j = 1, \ldots, n.$$

Iz $\mu_j < 1$ slijedi $1 - \mu_j > 0$, pa mora biti $\omega > 0$. S druge strane, iz $\omega(1-\mu_j) \leq \omega(1-\mu)$ izlazi uvjet $\omega(1 - \mu) < 2$, pa je $\omega < 2/(1 - \mu)$. Na kraju, iz $\mathrm{tr}(R_{Jac}) = 0$ slijedi da najmanja svojstvena vrijednost μ zadovoljava $\mu \leq 0$, što dokazuje (7.7.1). ■

Ovaj rezultat **ne** znači da Jacobijeva metoda konvergira jer se može dogoditi da je $\mu \leq -1$, pa u (7.7.1) dobivamo $\omega < 1$. Međutim, ako Jacobijeva metoda konvergira, onda konvergira i JOR, s tim da u (7.7.1) možemo uzeti i $\omega > 1$.

Korolar 7.7.1. *Neka je A simetrična (hermitska) pozitivno definitna matrica i pretpostavimo da Jacobijeva metoda konvergira. Onda konvergira i JOR(ω) metoda za sve parametre ω za koje vrijedi* (7.7.1) *i u toj relaciji je $2/(1 - \mu) > 1$.*

Dokaz:

Iz konvergencije Jacobijeve metode slijedi $-1 < \mu_j < 1$, za sve j, pa vrijedi

zaključak prethodnog teorema. Osim toga, vrijedi i $\mu > -1$, pa je $1 - \mu < 2$, što pokazuje da je gornja granica za ω u (7.7.1) veća od 1. ■

Ovo je pojačanje teorema 7.7.1. za simetrične (hermitske) pozitivno definitne matrice. Nažalost, gornju granicu za dozvoljeni $\omega > 1$ nije lako naći. Ako znamo $\rho(R_{Jac})$, možemo koristiti ocjenu $-1 < -\rho(R_{Jac}) \leq \mu$ i birati ω tako da je

$$1 < \omega < \frac{2}{1 + \rho(R_{Jac})} \leq \frac{2}{1 - \mu} \leq 2.$$

Međutim, nije jasno da ćemo takvim izborom parametra ω ubrzati konvergenciju iterativne metode. Obzirom na to da za SOR metodu možemo dobiti jače rezultate, JOR metoda se relativno rijetko koristi u praksi.

Prvi rezultat za SOR metodu je isti nužni uvjet konvergencije kao i kod JOR metode.

Teorem 7.7.4. *Vrijedi $\rho(R_{SOR(\omega)}) \geq |\omega - 1|$, pa je $0 < \omega < 2$ nužan uvjet za konvergenciju SOR metode.*

Dokaz:

Znamo da je determinanta matrice jednaka produktu svih njezinih svojstvenih vrijednosti. Izračunajmo determinantu matrice

$$R_{SOR(\omega)} = (I - \omega L)^{-1}((1 - \omega)I + \omega U)$$

koja je produkt trokutastih matrica. Iskoristimo Binet–Cauchyjev teorem i činjenicu da su L i U strogo trokutaste matrice. Zbog toga, samo dijagonale, tj. članovi s I ulaze u determinante, pa je

$$\begin{aligned} \det R_{SOR(\omega)} &= \det(I - \omega L)^{-1} \cdot \det((1 - \omega)I + \omega U) \\ &= \det I \cdot \det((1 - \omega)I) = (1 - \omega)^n. \end{aligned}$$

S druge strane je

$$\det R_{SOR(\omega)} = \prod_{j=1}^{n} \lambda_j(R_{SOR(\omega)}),$$

pa iz $|\lambda_j(R_{SOR(\omega)})| \leq \rho(R_{SOR(\omega)})$ dobivamo

$$|1 - \omega|^n = \prod_{j=1}^{n} |\lambda_j(R_{SOR(\omega)})| \leq (\rho(R_{SOR(\omega)}))^n,$$

odakle slijedi $\rho(R_{SOR(\omega)}) \geq |\omega - 1|$. Dakle, za konvergenciju SOR(ω) metode mora vrijediti $|\omega - 1| < 1$, ili $0 < \omega < 2$. U protivnom, metoda sigurno divergira, bar za neke početne vektore $x^{(0)}$. ■

Ako je A simetrična (hermitska) i pozitivno definitna, uvjet $0 < \omega < 2$ je i dovoljan za konvergenciju.

Teorem 7.7.5. *Ako je A simetrična (hermitska) i pozitivno definitna matrica tada je*

$$\rho(R_{SOR(\omega)}) < 1 \quad za \quad 0 < \omega < 2,$$

pa SOR(ω) konvergira. Posebno, uzimajući $\omega = 1$, slijedi da i Gauss–Seidelova metoda konvergira.

Dokaz:

Da bismo skratili pisanje, označimo s $R := R_{SOR(\omega)}$. Trebamo dokazati da za sve svojstvene vrijednosti matrice R vrijedi $|\lambda_j(R)| < 1$, tj. da one leže unutar jediničnog kruga u kompleksnoj ravnini. Da bismo to dokazali, trebamo iskoristiti da su svojstvene vrijednosti od A na pozitivnoj realnoj osi. Dokaz se sastoji od dva koraka. U prvom, prebacujemo problem iz jediničnog kruga u desnu otvorenu poluravninu $\operatorname{Re} z > 0$, gdje je lakše iskoristiti pozitivnu definitnost matrice A.

Za prvi korak koristimo razlomljene linearne transformacije. Lako se provjerava da takva (tzv. Möbiusova) transformacija oblika

$$\zeta(z) := \frac{1+z}{1-z}$$

bijektivno preslikava unutrašnjost jediničnog kruga $|z| < 1$ na desnu otvorenu poluravninu $\operatorname{Re}\zeta > 0$. Na isti način želimo transformirati svojstvene vrijednosti matrice R. Dakle, trebamo gledati matricu $\zeta(R)$. Obzirom na to da množenje matrica ne mora biti komutativno, pokazat će se da je zgodnije inverz pisati kao lijevi faktor (sami pogledajte put kroz dokaz ako inverz pišemo s desne strane). Definiramo matricu

$$S := (I-R)^{-1}(I+R). \tag{7.7.2}$$

Tada za svojstvene vrijednosti vrijedi $\lambda_j(S) = \zeta(\lambda_j(R))$, pa S ima svojstvene vrijednosti u desnoj otvorenoj poluravnini ako i samo ako R ima svojstvene vrijednosti unutar jediničnog kruga. Nažalost, to vrijedi samo ako je S korektno definirana, tj. ako je $I - R$ regularna. To još ne znamo, pa pokušajmo doći do relacije za S koja je uvijek korektna.

Polazna matrica A je po pretpostavci regularna. Da bismo dobili iterativnu metodu s matricom iteracije R (sasvim općenito), koristimo rastav ili cijepanje matrice $A = M - K$, pa ako je M regularna, onda je

$$R = M^{-1}K = M^{-1}(M-A) = I - M^{-1}A.$$

Tada je $I - R = M^{-1}A$ očito regularna (produkt regularnih), a $I + R = 2I - M^{-1}A$. Za S dobivamo

$$S = (I-R)^{-1}(I+R) = A^{-1}M(2I - M^{-1}A) = 2A^{-1}M - I.$$

Dakle, ako definiramo

$$S := 2A^{-1}M - I = A^{-1}(2M - A), \tag{7.7.3}$$

onda je S korektno definirana za bilo koju regularnu matricu A, čak i kad M nije regularna. Za nastavak dokaza treba iskoristiti ostale pretpostavke na A i pogledati kad svojstvene vrijednosti matrice S leže u desnoj otvorenoj poluravnini, u ovisnosti o ω u SOR metodi.

Neka je (λ, x) bilo koji svojstveni par od S, tj. $Sx = \lambda x$. Iz (7.7.3), množenjem s A, onda vrijedi i

$$(2M - A)x = ASx = \lambda Ax.$$

Množenjem s x^* slijeva dobivamo

$$x^*(2M - A)x = \lambda x^* Ax.$$

Napišimo adjungiranu (ozvjezdičenu) jednadžbu, iskoristivši simetričnost (hermitičnost) matrice $A = A^*$

$$x^*(2M^* - A)x = \bar{\lambda} x^* Ax,$$

i zbrojimo ih. Dijeljenjem s 2 dobivamo

$$x^*(M + M^* - A)x = \frac{\lambda + \bar{\lambda}}{2}\, x^* Ax = (\operatorname{Re}\lambda) x^* Ax.$$

Po pretpostavci je $x \neq 0$ (svojstveni vektor od S), pa iz pozitivne definitnosti od A slijedi $x^*Ax > 0$. Dijeljenjem dobivamo

$$\operatorname{Re}\lambda = \frac{x^*(M + M^* - A)x}{x^* Ax}, \tag{7.7.4}$$

pa je $\operatorname{Re}\lambda > 0$ ako i samo ako je brojnik pozitivan.

Sad iskoristimo da matrica M kod rastava matrice A u SOR(ω) metodi ima oblik

$$M = \omega^{-1}(D - \omega\widetilde{L}) = \omega^{-1}D - \widetilde{L},$$

pa je M korektno definirana za $\omega \neq 0$. Osim toga, iz $A = A^*$ slijedi $\widetilde{U} = \widetilde{L}^*$, ili $A = D - \widetilde{L} - \widetilde{L}^*$. Koristeći $D = D^*$, dobivamo da je

$$M + M^* - A = (\omega^{-1}D - \widetilde{L}) + (\omega^{-1}D - \widetilde{L}^*) - (D - \widetilde{L} - \widetilde{L}^*) = (2\omega^{-1} - 1)D.$$

Dijagonala D simetrične (hermitske) pozitivno definitne matrice A je i sama simetrična (hermitska) pozitivno definitna matrica. Na kraju, iz

$$x^*(M + M^* - A)x = (2\omega^{-1} - 1)x^* Dx$$

dobivamo da je brojnik u (7.7.4) pozitivan, ako i samo ako je $2\omega^{-1} - 1 > 0$ ili $0 < \omega < 2$.

Dakle, za matricu S u SOR metodi vrijedi $\operatorname{Re}\lambda_j(S) > 0$ za sve j, ako (i samo ako) je $0 < \omega < 2$. Nažalost, još uvijek ne možemo iskoristiti (7.7.2). No, inverz funkcije ζ

$$z(\zeta) = \frac{\zeta - 1}{\zeta + 1}$$

preslikava desnu otvorenu poluravninu na unutrašnjost jediničnog kruga. Rješavanjem jednadžbe (7.7.2) po S, očekujemo da je

$$R = (S - I)(S + I)^{-1}.$$

Zaista, ako je $\operatorname{Re}\lambda_j(S) > 0$ za sve j, onda je $S + I$ regularna, pa iz (7.7.3) slijedi

$$(S - I)(S + I)^{-1} = (2A^{-1}M - 2I)\,(2A^{-1}M)^{-1} = I - M^{-1}A = R,$$

a onda iz veze spektara vrijedi $|\lambda_j(R)| < 1$ za sve j

$$\begin{aligned} |\lambda_j(R)| &= \left|\frac{\lambda_j(S) - 1}{\lambda_j(S) + 1}\right| = \left|\frac{(\operatorname{Re}\lambda_j(S) - 1)^2 + (\operatorname{Im}\lambda_j(S))^2}{(\operatorname{Re}\lambda_j(S) + 1)^2 + (\operatorname{Im}\lambda_j(S))^2}\right|^{1/2} \\ &= \left|\frac{(\operatorname{Re}\lambda_j(S))^2 - 2\operatorname{Re}\lambda_j(S) + 1 + (\operatorname{Im}\lambda_j(S))^2}{(\operatorname{Re}\lambda_j(S))^2 + 2\operatorname{Re}\lambda_j(S) + 1 + (\operatorname{Im}\lambda_j(S))^2}\right|^{1/2} < 1. \end{aligned}$$

Lako se vidi da vrijedi i obrat, tj. iz $|\lambda_j(R)| < 1$ za sve j, koristeći (7.7.2), dobivamo $\operatorname{Re}\lambda_j(S) > 0$ za sve j. ■

Očito je da smo prethodni dokaz mogli provesti i mnogo brže ili kraće. Njegov deduktivni dio ima samo dva bitna koraka:

(1) definiramo $S = A^{-1}(2M - A)$ i pokažemo da je $\operatorname{Re}\lambda_j(S) > 0$ za sve j,

(2) pokažemo da je $R = (S - I)(S + I)^{-1}$, a zatim da je $|\lambda_j(R)| < 1$ za sve j.

Međutim, oblik matrice S i njezina veza s R nisu pali "s neba", niti su rezultat pogađanja. U pozadini cijele konstrukcije je lijepo matematičko opravdanje koje smo posebno željeli naglasiti.

Iz posljednja dva teorema odmah izlazi sljedeći rezultat.

Korolar 7.7.2. *Ako je A simetrična (hermitska) pozitivno definitna matrica, onda SOR(ω) konvergira ako i samo ako je $0 < \omega < 2$.*

U usporedbi s korolarom 7.7.1. za JOR metodu, ovo je bitno proširenje i pojačanje. Ovdje dobivamo bezuvjetnu konvergenciju, veći raspon za ω i maksimalnu moguću gornju granicu 2, koja ne ovisi o λ.

7.8. Optimalni izbor relaksacijskog parametra

Pokažimo još da se dobrim izborom relaksacijskog parametra ω može postići bitno ubrzanje konvergencije iteracija u SOR(ω) metodi, bar za neke klase matrica.

Dokazat ćemo klasični rezultat koji daje optimalni izbor parametra ω za klasu tzv. konzistentno poredanih matrica. Dokazao ga je D. M. Young, još 1950. godine,

a ima veliku praktičnu vrijednost jer pokriva matrice koje se javljaju u diskretizaciji nekih parcijalnih diferencijalnih jednadžbi, poput Poissonove.

Za početak, moramo definirati potrebne pojmove, koji se opet oslanjaju na vezu između matrica i pripadnih grafova.

Definicija 7.8.1. *Matrica A ima svojstvo (A) ako postoji matrica permutacije P takva da vrijedi*

$$PAP^T = \begin{bmatrix} D_1 & A_{12} \\ A_{21} & D_2 \end{bmatrix},$$

gdje su D_1 i D_2 dijagonalne matrice. Drugim riječima, u pripadnom grafu $G(A)$ čvorovi su podijeljeni u dva disjunktna skupa S_1 i S_2, s tim da ne postoji brid koji veže dva različita čvora iz istog skupa S_i. Ako ignoriramo bridove koji vežu čvor sa samim sobom, onda bridovi vežu samo čvorove iz različitih skupova. Takav se graf zove bipartitni graf.

Ako matrica A ima svojstvo (A), onda rastav ili cijepanje matrice PAP^T za iterativnu metodu ima posebnu strukturu. Tada je

$$PAP^T = \begin{bmatrix} D_1 & A_{12} \\ A_{21} & D_2 \end{bmatrix} = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ -A_{21} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -A_{12} \\ 0 & 0 \end{bmatrix} := D - \widetilde{L} - \widetilde{U}.$$

Ako je D regularna, onda možemo korektno definirati sve opisane iterativne metode za matricu PAP^T, a iz prethodne strukture dokazat ćemo posebna svojstva tih iterativnih metoda. U praksi se obično pretpostavlja da je A sa svojstvom (A) već dovedena u ovaj oblik odgovarajućom permutacijom P, tj. da polazna matrica A ima ovu strukturu.

U nastavku pretpostavljamo da je D regularna i koristimo standardne oznake $L = D^{-1}\widetilde{L}$ i $U = D^{-1}\widetilde{U}$.

Definicija 7.8.2. *Neka je $\alpha \neq 0$. Definirajmo familiju matrica*

$$R_{Jac}(\alpha) = \alpha D^{-1}\widetilde{L} + \frac{1}{\alpha} D^{-1}\widetilde{U} = \alpha L + \frac{1}{\alpha} U. \tag{7.8.1}$$

Vidimo da je $R_{Jac}(1) = R_{Jac}$ matrica iteracije u Jacobijevoj metodi.

Matrice $R_{Jac}(\alpha)$ za $\alpha \neq 1$ nemaju direktnu interpretaciju kao matrice iteracije u nekoj od standardnih iterativnih metoda koje smo opisali.

Propozicija 7.8.1. *Za matrice A sa svojstvom (A), svojstvene vrijednosti matrica $R_{Jac}(\alpha)$ ne ovise o α, s tim da D, L i U dobivamo iz rastava matrice PAP^T.*

Dokaz:

Po definiciji je

$$\begin{aligned} R_{Jac}(\alpha) &= \alpha L + \frac{1}{\alpha}U = D^{-1}\Big(\alpha\widetilde{L} + \frac{1}{\alpha}\widetilde{U}\Big) \\ &= \begin{bmatrix} D_1^{-1} & \\ & D_2^{-1} \end{bmatrix} \left(\alpha \begin{bmatrix} 0 & 0 \\ -A_{21} & 0 \end{bmatrix} + \frac{1}{\alpha}\begin{bmatrix} 0 & -A_{12} \\ 0 & 0 \end{bmatrix}\right) \\ &= -\begin{bmatrix} 0 & \frac{1}{\alpha}D_1^{-1}A_{12} \\ \alpha D_2^{-1}A_{21} & 0 \end{bmatrix}. \end{aligned}$$

Ovu relaciju možemo napisati i u obliku

$$R_{Jac}(\alpha) = \begin{bmatrix} I & \\ & \alpha I \end{bmatrix} \left(-\begin{bmatrix} 0 & D_1^{-1}A_{12} \\ D_2^{-1}A_{21} & 0 \end{bmatrix}\right) \begin{bmatrix} I & \\ & \alpha I \end{bmatrix}^{-1} = \Delta_\alpha R_{Jac}\Delta_\alpha^{-1},$$

gdje je $\Delta_\alpha = \mathrm{diag}(I, \alpha I)$ dijagonalna matrica u kojoj dimenzije blokova odgovaraju dimenzijama blokova D_1 i D_2. Zbog $\alpha \neq 0$, očito je Δ_α regularna. Zaključujemo da su $R_{Jac}(\alpha)$ i R_{Jac} slične matrice, a to povlači da imaju iste svojstvene vrijednosti, tj. svojstvene vrijednosti $R_{Jac}(\alpha)$ ne ovise o α. ■

Svojstvo (A) iz definicije 7.8.1. je geometrijsko ili grafovsko svojstvo matrice. Posljedica toga je ovo algebarsko svojstvo invarijantnosti, kojeg ćemo bitno koristiti u nastvaku, pa mu dajemo posebno ime.

Definicija 7.8.3. *Neka je A proizvoljna matrica takva da je*

$$A = D - \widetilde{L} - \widetilde{U}$$

s tim da je D regularna i

$$R_{Jac}(\alpha) = \alpha D^{-1}\widetilde{L} + \frac{1}{\alpha}D^{-1}\widetilde{U} = \alpha L + \frac{1}{\alpha}U.$$

Ako svojstvene vrijednosti matrice $R_{Jac}(\alpha)$ ne ovise o α, onda kažemo da A ima konzistentan poredak (engl. consistent ordering).

Ako A ima svojstvo (A), onda PAP^T ima konzistentan poredak. Obratno ne vrijedi, tj. ako matrica ima konzistentan poredak, ne mora imati svojstvo (A).

Primjer 7.8.1. *Blok trodijagonalne matrice oblika*

$$\begin{bmatrix} D_1 & U_1 & & \\ L_1 & \ddots & \ddots & \\ & \ddots & \ddots & U_{m-1} \\ & & L_{m-1} & D_m \end{bmatrix}$$

imaju konzistentan poredak kad su D_i regularne dijagonalne matrice bilo kojih redova, za bilo koji blok red m. Pokažite to!

Međutim, za $m > 2$, ako su vandijagonalni blokovi netrivijalni, ove matrice nemaju svojstvo (A). Takvim matricama odgovaraju tzv. slojeviti grafovi, u kojima čvorove možemo podijeliti u m disjunktnih skupova — slojeva, tako da bridovi idu samo između čvorova koji su u različitim, ali susjednim slojevima. Kao i prije, ignoriramo bridove iz čvora u samog sebe. U ovom kontekstu, bipartitni graf ima samo 2 sloja.

Za matrice koje imaju konzistentan poredak postoje jednostavne formule koje vežu svojstvene vrijednosti matrica R_{Jac}, R_{GS} i $R_{SOR(\omega)}$.

Teorem 7.8.1. *Ako matrica A ima konzistentan poredak i ako je $\omega \neq 0$, onda vrijedi:*

(a) *Svojstvene vrijednosti matrice $R_{Jac}(\alpha)$ dolaze u $\pm$ parovima. Preciznije, ako je $\mu \neq 0$ svojstvena vrijednost od R_{Jac} multipliciteta ν, onda je i $-\mu$ svojstvena vrijednost od R_{Jac} multipliciteta ν.*

(b) *Ako je μ svojstvena vrijednost od R_{Jac} i ako λ zadovoljava jednadžbu*

$$(\lambda + \omega - 1)^2 = \lambda\omega^2\mu^2, \tag{7.8.2}$$

onda je λ svojstvena vrijednost od $R_{SOR(\omega)}$.

(c) *Obratno, ako je $\lambda \neq 0$ svojstvena vrijednost od $R_{SOR(\omega)}$, onda je μ iz* (7.8.2) *svojstvena vrijednost od R_{Jac}.*

Dokaz:

Prije početka dokaza uočimo da predznak od μ ne igra ulogu u (7.8.2), što je u suglasnosti s prvom tvrdnjom. Također, relacija (7.8.2) generira isti broj vrijednosti za λ i μ, uz uvjet da je $\lambda \neq 0$.

(a) Po definiciji 7.8.3., ako A ima konzistentan poredak, onda svojstvene vrijednosti od $R_{Jac}(\alpha)$ ne ovise o α. Posebno, to znači da matrice $R_{Jac} = R_{Jac}(1)$ i $R_{Jac}(-1)$ imaju iste svojstvene vrijednosti.

S druge strane, prema definiciji (7.8.1) za $R_{Jac}(-1)$ izlazi

$$R_{Jac}(-1) = -D^{-1}\widetilde{L} - D^{-1}\widetilde{U} = -(L + U) = -R_{Jac}(1) = -R_{Jac},$$

pa matrice R_{Jac} i $-R_{Jac}$ istovremeno moraju imati iste i suprotne svojstvene vrijednosti. To je moguće ako i samo ako svojstvene vrijednosti μ dolaze u $\pm$ parovima s istim multiplicitetom, čim je $\mu \neq 0$.

(b) Neka je μ svojstvena vrijednost od R_{Jac} i pretpostavimo da λ zadovoljava jednadžbu (7.8.2). Ako je $\lambda = 0$, onda u (7.8.2) mora biti $\omega = 1$, tj. SOR metoda se svodi na Gauss–Seidelovu metodu. Tada je $R_{SOR(1)} = R_{GS} = (I - L)^{-1}U$, a znamo da je R_{GS} singularna, jer je U strogo gornja trokutasta. Dakle, $\lambda = 0$ je tada svojstvena vrijednost za $R_{SOR(1)}$.

Pretpostavimo sad da je $\lambda \neq 0$. Onda, zbog $\omega \neq 0$, iz jednadžbe (7.8.2) možemo izračunati μ

$$\mu = \frac{\lambda + \omega - 1}{\sqrt{\lambda}\omega}. \tag{7.8.3}$$

Matrica iteracije u SOR(ω) metodi ima oblik

$$R_{SOR(\omega)} = (I - \omega L)^{-1}((1-\omega)I + \omega U).$$

Pogledajmo vrijednost karakterističnog polinoma $p(\lambda) = \det(\lambda I - R_{SOR(\omega)})$ ove matrice u točki λ. Da bismo se riješili inverza u $R_{SOR(\omega)}$, uočimo da je $\det(I - \omega L) = 1$, jer je L strogo donja trokutasta. Zbog toga, po Binet–Cauchyjevom teoremu vrijedi

$$\begin{aligned} \det(\lambda I - R_{SOR(\omega)}) &= \det((I - \omega L)\,(\lambda I - R_{SOR(\omega)})) \\ &= \det(\lambda(I - \omega L) - ((1-\omega)I + \omega U)) \\ &= \det((\lambda + \omega - 1)I - \omega\lambda L - \omega U). \end{aligned} \tag{7.8.4}$$

Zadnja dva člana želimo svesti na oblik $R_{Jac}(\alpha)$ za neki α, izlučivanjem odgovarajućeg faktora. Imamo

$$\omega\lambda L + \omega U = \sqrt{\lambda}\omega\Big(\sqrt{\lambda}L - \frac{1}{\sqrt{\lambda}}U\Big) = \sqrt{\lambda}\omega\, R_{Jac}(\sqrt{\lambda}),$$

pa je $\alpha = \sqrt{\lambda}$, s tim da smo iskoristili $\lambda \neq 0$. Kad u (7.8.4) izlučimo isti faktor $\sqrt{\lambda}\omega$, uvrstimo ovu relaciju i uvažimo da je $R_{Jac}(\sqrt{\lambda}) = R_{Jac}(1) = R_{Jac}$, dobivamo

$$\begin{aligned} \det(\lambda I - R_{SOR(\omega)}) &= \det\left(\sqrt{\lambda}\omega\left(\Big(\frac{\lambda + \omega - 1}{\sqrt{\lambda}\omega}\Big)I - R_{Jac}\right)\right) \\ &= (\sqrt{\lambda}\omega)^n \det\left(\Big(\frac{\lambda + \omega - 1}{\sqrt{\lambda}\omega}\Big)I - R_{Jac}\right). \end{aligned}$$

Vidimo da je faktor uz I upravo jednak μ iz (7.8.3), pa prethodna relacija postaje

$$\det(\lambda I - R_{SOR(\omega)}) = (\sqrt{\lambda}\omega)^n \det(\mu I - R_{Jac}). \tag{7.8.5}$$

Ako je μ svojstvena vrijednost od R_{Jac}, onda je desna strana jednaka nuli, pa to vrijedi i za lijevu stranu, tj. λ je svojstvena vrijednost od $R_{SOR(\omega)}$.

(c) Relaciju (7.8.5) smo izveli baš uz pretpostavku da je $\lambda \neq 0$, pa odmah slijedi tvrdnja. ■

Korolar 7.8.1. *Ako A ima konzistentan poredak, tada je*

$$\rho(R_{GS}) = (\rho(R_{Jac}))^2,$$

što znači da Gauss–Seidelova metoda konvergira dvostruko brže nego Jacobijeva metoda (ako barem jedna od njih kovergira).

Dokaz:

Izaberemo li u prethodnom teoremu $\omega = 1$, onda je SOR metoda baš Gauss–Seidelova metoda, pa za taj ω relacija (7.8.2) glasi

$$\lambda^2 = \lambda\mu^2,$$

ili $\lambda = \mu^2$. Budući da to vrijedi za svaku svojstvenu vrijednost, onda to vrijedi i za spektralni radijus. ■

Odavde odmah slijedi da za konzistentno poredane matrice Gauss–Seidelova metoda konvergira ako i samo ako konvergira i Jacobijeva metoda. Vidjet ćemo da slično vrijedi i za SOR metodu. Međutim, dobit ćemo i puno jači rezultat koji nam kaže kako treba izabrati parametar ω za ubrzanje konvergencije u SOR metodi.

Na početku ovog poglavlja vidjeli smo da brzina konvergencije iterativne metode ovisi o spektralnom radijusu $\rho(R)$ matrice iteracija R. Manji $\rho(R)$, općenito, osigurava i bržu konvergenciju, jer vrijedi ocjena

$$\|x^{(m+1)} - x\|_* \leq \rho(R)\|x^{(m)} - x\|_*.$$

Za neke početne vektore i u nekim iteracijama možemo dobiti i manju grešku, ali znamo da je ova ocjena dostižna. Dakle, da bismo globalno ubrzali konvergenciju metode treba dobiti što manji spektralni radijus $\rho(R)$. U tom smislu, kod SOR(ω) metode, one vrijednosti parametra ω za koje je $\rho(R_{SOR(\omega)})$ globalno najmanji, zovemo **optimalnim** relaksacijskim parametrima i označavamo s ω_{opt}.

Uz blage dodatne uvjete i malo truda, iz relacije (7.8.2) možemo dobiti i taj optimalni izbor parametra ω_{opt} koji minimizira $R_{SOR(\omega)}$, tako da on ovisi samo o spektralnom radijusu $\rho(R_{Jac})$ u Jacobijevoj metodi.

Teorem 7.8.2. *Pretpostavimo da matrica A ima konzistentan poredak i da matrica R_{Jac} u Jacobijevoj metodi ima samo realne svojstvene vrijednosti. Onda SOR(ω) metoda konvergira za bilo koji početni vektor ako i samo ako je $\mu := \rho(R_{Jac}) < 1$ (tj. Jacobijeva metoda konvergira) i vrijedi $0 < \omega < 2$. Dodatno, za $\mu < 1$ onda vrijedi i*

$$\omega_{\text{opt}} = \frac{2}{1 + \sqrt{1 - \mu^2}},$$

$$\rho(R_{SOR(\omega_{\text{opt}})}) = \omega_{\text{opt}} - 1 = \frac{\mu^2}{(1 + \sqrt{1 - \mu^2})^2} = \frac{1 - \sqrt{1 - \mu^2}}{1 + \sqrt{1 - \mu^2}},$$

a za sve $\omega \in (0, 2)$ vrijedi

$$\rho(R_{SOR(\omega)}) = \begin{cases} 1 - \omega + \frac{1}{2}\omega^2\mu^2 + \omega\mu\sqrt{1 - \omega + \frac{1}{4}\omega^2\mu^2}, & za\ 0 < \omega \leq \omega_{\text{opt}}, \\ \omega - 1, & za\ \omega_{\text{opt}} \leq \omega \leq 2. \end{cases}$$

Dokaz:

Matrica A ima konzistentan poredak, pa možemo iskoristiti teorem 7.8.1.(b) da iz svojstvenih vrijednosti μ_j matrice R_{Jac} izračunamo svojstvene vrijednosti λ_j matrice $R_{SOR(\omega)}$. Relacija (7.8.2) daje vezu

$$(\lambda_j + \omega - 1)^2 = \lambda_j \omega^2 \mu_j^2,$$

što možemo napisati kao kvadratnu jednadžbu za λ_j

$$\lambda_j^2 - 2\Big(1 - \omega + \frac{1}{2}(\omega\mu_j)^2\Big)\lambda_j + (\omega - 1)^2 = 0. \tag{7.8.6}$$

Prema tvrdnji (a) teorema 7.8.1., ako $\mu_j = 0$ ima multiplicitet m, onda pripadni $\lambda_j = 1 - \omega$ ima isti multiplicitet m. Osim toga, svojstvene vrijednosti $\mu_j \neq 0$ dolaze u $\pm$ parovima, pa u rješavanju jednadžbe (7.8.6) možemo ignorirati predznak od μ_j, jer svaki $\pm$ par daje dvije svojstvene vrijednosti λ_j.

Sad iskoristimo pretpostavku da R_{Jac} ima samo realne svojstvene vrijednosti μ_j, pa jednadžba (7.8.6) ima **realne** koeficijente i kod rješavanja možemo gledati samo $\mu_j > 0$.

Znamo da SOR(ω) metoda konvergira za bilo koji početni vektor ako i samo ako je $|\lambda_j| < 1$ za sve j. Ako je pripadni $\mu_j = 0$, onda je $\lambda_j = 1 - \omega$, pa $|\lambda_j| < 1$ vrijedi ako i samo ako je $0 < \omega < 2$. Ako su to i jedine svojstvene vrijednosti od R_{Jac}, tj. $R_{Jac} = 0$, onda je prva tvrdnja dokazana.

U protivnom, postoje $\mu_j > 0$ i treba analizirati rješenja jednadžbe (7.8.6). Da bismo pojednostavnili zapis, promatramo kvadratnu jednadžbu

$$\lambda^2 + b\lambda + c = 0 \tag{7.8.7}$$

s realnim koeficijentima b i c. Tražimo kriterij (ako i samo ako uvjet) da korijeni ove jednadžbe leže unutar jediničnog kruga. Rješenja jednadžbe su

$$\lambda_{1,2} = \frac{1}{2}(-b \pm \sqrt{b^2 - 4c}).$$

Znamo da je $|c| = |\lambda_1|\,|\lambda_2|$, pa je $|c| < 1$ očiti nužni uvjet da bi oba korijena bila u jediničnom krugu.

Ako je diskriminanta negativna $b^2 - 4c < 0$, onda su rješenja konjugirano kompleksna

$$\lambda_{1,2} = \frac{1}{2}(-b \pm i\sqrt{4c - b^2}),$$

pa je

$$|\lambda_{1,2}|^2 = \frac{1}{4}(b^2 + 4c - b^2) = c. \tag{7.8.8}$$

Dakle, ako je $b^2 < 4c$, onda je $|\lambda_{1,2}| < 1$, ako i samo ako je $c < 1$. Dodatno, uočimo da je tada $c > 0$ i $|b| \leq 2\sqrt{c} < 1 + c$, zbog $(1 - \sqrt{c})^2 > 0$.

Ako je diskriminanta nenegativna $b^2 - 4c \geq 0$, onda imamo par realnih rješenja

$$\lambda_{1,2} = \frac{1}{2}(-b \pm \sqrt{b^2 - 4c}),$$

pa je

$$\max|\lambda_{1,2}| = \frac{1}{2}\max|-b \pm \sqrt{b^2 - 4c}| = \frac{1}{2}(|b| + \sqrt{b^2 - 4c}). \tag{7.8.9}$$

Iz $\max|\lambda_{1,2}| < 1$ dobivamo redom

$$\begin{aligned} |b| + \sqrt{b^2 - 4c} &< 2 \\ \sqrt{b^2 - 4c} &< 2 - |b| \\ b^2 - 4c &< (2 - |b|)^2 = 4 - 4|b| + b^2 \\ |b| &< 1 + c. \end{aligned}$$

Dakle, ako je $b^2 \geq 4c$, onda je $|\lambda_{1,2}| < 1$, ako i samo ako je $|b| < 1 + c$. Dodatno, iz $2 - |b| > 0$ slijedi $b^2 < 4$, pa mora biti i $c < 1$.

Zaključujemo da u oba slučaja iz $|\lambda_{1,2}| < 1$ slijedi $c < 1$ i $|b| < 1 + c$. Međutim, očito vrijedi i obrat, jer ovisno o odnosu b^2 i $4c$, iskoristimo pravu (jaču) od ove dvije pretpostavke. Na kraju, iz $|b| < 1 + c$ slijedi $1 + c > 0$ ili $c > -1$, što zajedno s $c < 1$, osigurava i raniji nužni uvjet $|c| < 1$.

Dokazali smo da rješenja jednadžbe (7.8.7) leže unutar jediničnog kruga, ako i samo ako vrijedi $c < 1$ i $|b| < 1 + c$.

Usporedbom (7.8.6) i (7.8.7) vidimo da je

$$b = -2\Big(1 - \omega + \frac{1}{2}(\omega\mu_j)^2\Big), \quad c = (\omega - 1)^2. \tag{7.8.10}$$

Uvjet $c < 1$ daje $(\omega - 1)^2 < 1$, ili $0 < \omega < 2$. Drugi uvjet $|b| < 1 + c$ daje

$$2\Big|1 - \omega + \frac{1}{2}(\omega\mu_j)^2\Big| < 1 + (\omega - 1)^2.$$

Ako lijevu stranu napišemo u obliku

$$|2 - 2\omega + \omega^2 + \omega^2(\mu_j^2 - 1)| = |1 + (\omega - 1)^2 + \omega^2(\mu_j^2 - 1)|,$$

uz oznaku $a := 1 + (\omega - 1)^2 > 0$ dobivamo uvjet

$$|a + \omega^2(\mu_j^2 - 1)| < a,$$

što je moguće ako i samo ako je drugi član negativan, tj. za $\mu_j^2 < 1$. Dakle, sve vrijednosti λ_j leže u jediničnom krugu, ako i samo ako je $0 < \omega < 2$ i vrijedi $\mu_j^2 < 1$ za sve j, što je ekvivalentno s $\mu := \rho(R_{Jac}) < 1$.

Time smo dokazali prvi dio tvrdnje. Drugi dio dobivamo tako da nađemo spektralni radijus $\rho(R_{SOR(\omega)})$ za svaki $\omega \in (0,2)$, a zatim ga minimiziramo po ω.

Neka je $\omega \in (0,2)$. Svojstvene vrijednosti λ_j možemo podijeliti u tri grupe (skupa). Ako je $\mu_j = 0$ za neki j, onda je pripadni $\lambda_j = 1-\omega$ i

$$|\lambda_j| = |1-\omega|. \tag{7.8.11}$$

Označimo skup svih takvih λ_j s S_0. Ako je $\mu = 0$, tj. $R_{Jac} = 0$, onda je očito $\rho(R_{SOR(\omega)}) = |1-\omega|$, pa je optimalna vrijednost parametra $\omega_{\text{opt}} = 1$. Dobivamo $\rho(R_{SOR(\omega_{\text{opt}})}) = 0$, što odgovara Gauss–Seidelovoj metodi s $R_{GS} = 0$. Dakle, i drugi dio tvrdnje vrijedi ako je $\mu = 0$.

Zbog toga možemo pretpostaviti da je $\mu > 0$, tj. da postoji barem jedna svojstvena vrijednost $\mu_j > 0$. Za $\mu_j > 0$, rješavanjem (7.8.6) dobivamo

$$\begin{aligned}(\lambda_j)_{1,2} &= 1-\omega+\frac{1}{2}(\mu_j\omega)^2 \pm \mu_j\omega\sqrt{1-\omega+\left(\frac{1}{2}\mu_j\omega\right)^2} \\ &= \left(\frac{1}{2}\mu_j\omega \pm \sqrt{1-\omega+\left(\frac{1}{2}\mu_j\omega\right)^2}\right)^2.\end{aligned} \tag{7.8.12}$$

Ponašanje ovih rješenja ovisi o vrijednostima diskriminante

$$\Delta(\mu_j) = 1-\omega+\left(\frac{1}{2}\mu_j\omega\right)^2.$$

Primijetimo da je $\Delta(\mu_j)$ rastuća funkcija od μ_j. Ovisno o predznaku $\Delta(\mu_j)$ dobivamo dvije grupe svojstvenih vrijednosti λ_j. Prva grupa S_- odgovara negativnim, a druga S_+ nenegativnim diskriminantama.

Za sve $\mu_j > 0$ za koje je $\Delta(\mu_j) < 0$, ako takvih ima, dobivamo kompleksno konjugirani par $(\lambda_j)_{1,2}$. Tada mora biti $1-\omega < 0$, što pokazuje da je S_- neprazan samo ako je $\omega > 1$, tj. ova grupa je sigurno prazna za $\omega \leq 1$. Ako uvrstimo b i c iz (7.8.10) u (7.8.8), izlazi da je

$$|(\lambda_j)_{1,2}| = \sqrt{c} = \omega - 1, \tag{7.8.13}$$

pa vidimo da ove vrijednosti **ne ovise** o μ_j.

S druge strane, za one $\mu_j > 0$ za koje je $\Delta(\mu_j) \geq 0$, ako takvih ima, dobivamo par realnih svojstvenih vrijednosti $(\lambda_j)_{1,2}$. Analogno, iz (7.8.10) i (7.8.9), izlazi da je

$$\begin{aligned}\max|(\lambda_j)_{1,2}| &= \frac{1}{2}(|b|+\sqrt{b^2-4c}) \\ &= 1-\omega+\frac{1}{2}(\mu_j\omega)^2+\mu_j\omega\sqrt{1-\omega+\left(\frac{1}{2}\mu_j\omega\right)^2} \\ &= \left(\frac{1}{2}\mu_j\omega+\sqrt{1-\omega+\left(\frac{1}{2}\mu_j\omega\right)^2}\right)^2,\end{aligned}$$

jer iz $\Delta(\mu_j) \geq 0$ slijedi $1 - \omega + (\mu_j \omega/2)^2 > 0$. Očito je $\max |(\lambda_j)_{1,2}|$ rastuća funkcija po μ_j, pa najveću vrijednost dobivamo za najveći μ_j, tj. za $\mu_j = \mu = \rho(R_{Jac})$, naravno, pod uvjetom da je $\Delta(\mu) \geq 0$.

Obzirom na to da i $\Delta(\mu_j)$ raste s μ_j, druga grupa je neprazna ako i samo ako je $\Delta(\mu) \geq 0$ ili

$$\mu^2 > 4\,\frac{\omega - 1}{\omega^2},$$

pa je ova grupa sigurno neprazna za $\omega \leq 1$. Precizni kriterij nepraznosti S_+ dobivamo rješavanjem $\Delta(\mu) \geq 0$ po ω. Iz

$$0 = \Delta(\mu) = 1 - \omega + \left(\frac{1}{2}\mu\omega\right)^2$$

dobivamo granične vrijednosti za ω

$$\omega_{1,2} = 2\frac{1 \pm \sqrt{1 - \mu^2}}{\mu^2} = \frac{2}{1 \mp \sqrt{1 - \mu^2}}.$$

Intervalu $(0, 2)$ pripada samo manja od te dvije vrijednosti koju (zasad bez opravdanja) označavamo s ω_{opt}

$$\omega_{\text{opt}} = 2\frac{1 - \sqrt{1 - \mu^2}}{\mu^2} = \frac{2}{1 + \sqrt{1 - \mu^2}}.$$

Na kraju, zaključujemo da je S_+ neprazan ako i samo ako je $\omega \leq \omega_{\text{opt}}$. Uočimo da je $\omega_{\text{opt}} > 1$, zbog $\mu > 0$. Maksimum apsolutnih vrijednosti za $\lambda_j \in S_+ \neq \emptyset$ je

$$\lambda := 1 - \omega + \frac{1}{2}(\mu\omega)^2 + \mu\omega\sqrt{1 - \omega + \left(\frac{1}{2}\mu\omega\right)^2}. \tag{7.8.14}$$

Sad konačno možemo izračunati $\rho(R_{SOR(\omega)})$ uspoređujući maksimalne apsolutne vrijednosti iz (7.8.11), (7.8.13) i (7.8.14).

Za $\omega \leq 1$ je $S_- = \emptyset$. Za $\mu_j = 0$ imamo $|\lambda_j| = 1 - \omega$, pa je očito $|\lambda_j| < \lambda$, jer su u (7.8.14) svi članovi desne strane od trećeg nadalje pozitivni, a prva dva su upravo $|\lambda_j|$. Dakle, za $\omega \leq 1$ je $\rho(R_{SOR(\omega)}) = \lambda$.

Ako je $1 < \omega \leq \omega_{\text{opt}}$, onda je S_+ sigurno neprazan. Ako S_+ sadrži sve svojstvene vrijednosti, onda je očito $\rho(R_{SOR(\omega)}) = \lambda$. U protivnom, za $S_0 \cup S_- \neq \emptyset$, u (7.8.11) i (7.8.13) dobivamo isti maksimum $\omega - 1$. Usporedimo ga s λ iz (7.8.14). Zbog $\Delta(\mu) \geq 0$ dobivamo

$$\begin{aligned} \lambda - (\omega - 1) &= 2 - 2\omega + \frac{1}{2}(\mu\omega)^2 + \mu\omega\sqrt{1 - \omega + \left(\frac{1}{2}\mu\omega\right)^2} \\ &= 2\Delta(\mu) + \mu\omega\sqrt{\Delta(\mu)} \geq 0. \end{aligned} \tag{7.8.15}$$

Jednakost se dostiže samo za $\Delta(\mu) = 0$, tj. za $\omega = \omega_{\text{opt}}$, pa je opet $\rho(R_{SOR(\omega)}) = \lambda$.

Na kraju, za $\omega_{\text{opt}} < \omega < 2$, dobivamo da je $S_+ = \emptyset$. Barem jedan od skupova S_0, S_- nije prazan. U (7.8.11) i (7.8.13) dobivamo isti maksimum $\omega - 1$, pa je $\rho(R_{SOR(\omega)}) = \omega - 1$.

Dokazali smo da je

$$\rho(R_{SOR(\omega)}) = \begin{cases} \lambda & \text{za } 0 < \omega \leq \omega_{\text{opt}}, \\ \omega - 1, & \text{za } \omega_{\text{opt}} \leq \omega \leq 2. \end{cases}$$

Time je zadnja relacija iz tvrdnje teorema u potpunosti dokazana.

Ostaje još naći najmanju moguću vrijednost od $\rho(R_{SOR(\omega)})$ za $\omega \in (0, 2)$. Kad λ iz (7.8.14) promatramo kao funkciju od ω

$$\begin{aligned} \lambda(\omega) &= 1 - \omega + \frac{1}{2}(\mu\omega)^2 + \mu\omega\sqrt{1 - \omega + \left(\frac{1}{2}\mu\omega\right)^2} \\ &= \left(\frac{1}{2}\mu\omega + \sqrt{1 - \omega + \left(\frac{1}{2}\mu\omega\right)^2}\right)^2, \end{aligned}$$

derviranjem drugog oblika po ω dobivamo

$$\begin{aligned} \frac{d\lambda}{d\omega} &= 2\sqrt{\lambda}\Big(\frac{1}{2}\mu + \frac{\frac{1}{2}\mu^2\omega - 1}{2\sqrt{1 - \omega + (\frac{1}{2}\mu\omega)^2}}\Big) \\ &= \sqrt{\lambda}\,\frac{\mu\sqrt{1 - \omega + (\frac{1}{2}\mu\omega)^2} + \frac{1}{2}\mu^2\omega - 1}{\sqrt{1 - \omega + (\frac{1}{2}\mu\omega)^2}} \end{aligned}$$

Brojnik možemo napisati u obliku $(\lambda - 1)/\omega$, pa je

$$\frac{d\lambda}{d\omega} = \frac{\sqrt{\lambda}\,(\lambda - 1)}{\omega\sqrt{\Delta(\mu)}} < 0, \quad \omega \in (0, \omega_{\text{opt}}),$$

jer je $0 < \lambda < 1$, $\omega > 0$ i $\Delta(\mu) > 0$. Dakle, $\rho(R_{SOR(\omega)}) = \lambda(\omega)$ monotono pada na $(0, \omega_{\text{opt}}]$, s tim da u ω_{opt} derivacija nije definirana (teži u $-\infty$), jer je $\Delta(\mu) = 0$.

Za $\omega \in (\omega_{\text{opt}}, 2)$ je $\rho(R_{SOR(\omega)}) = \omega - 1$, što je očito rastuća funkcija. U točki ω_{opt}, iz (7.8.15), jer je desna strana jednaka 0, slijedi

$$\rho(R_{SOR(\omega_{\text{opt}})}) = \lambda(\omega_{\text{opt}}) = \omega_{\text{opt}} - 1.$$

Dakle, $\rho(R_{SOR(\omega)})$ dostiže jedinstveni globalni minimum za $\omega = \omega_{\text{opt}}$ na $(0, 2)$, pa je i oznaka opravdana. To dokazuje i zadnji dio tvrdnje.

Vidimo da se minimum dostiže kad je $\Delta(\mu) = 0$, tj. kad je izraz pod korijenom u (7.8.12) jednak 0. Iz $\mu_j = \mu$ tada dobivamo dvostruki korijen $(\lambda_j)_{1,2} = \lambda(\omega_{\text{opt}})$. ■

Dobili smo da za optimalni parametar vrijedi $\omega_{\text{opt}} > 1$ pa je optimalni SOR zaista nadrelaksacija. Zgodno je primijetiti da što je μ bliže 1, tj. što sporije konvergira Jacobijeva metoda, to je ω_{opt} bliže 2, tj. SOR tada "produljuje" korak. I obratno, ako je μ blizu 0 (Jacobi brz), onda je ω_{opt} blizu 1, pa je optimalni SOR blizak Gauss–Seidelovoj metodi.

Ovo su slični uvjetni rezultati kao i za JOR metodu, u smislu da se pretpostavlja da Jacobijeva metoda konvergira. Srećom, za neke matrice nije teško pronaći spektralni radijus matrice u Jacobijevoj metodi i ustanoviti da Jacobijeva metoda konvergira. A tada imamo i optimalni izbor parametra za SOR, dok za JOR to nemamo. Ako usporedimo relacije koje vežu svojstvene vrijednosti matrica u JOR i SOR metodi s onima iz Jacobijeve metode

$$\begin{aligned} \lambda_j \in \sigma(R_{JOR(\omega)}): &\quad \lambda_j + \omega - 1 = \omega\mu_j, \\ \lambda_j \in \sigma(R_{SOR(\omega)}): &\quad \lambda_j + \omega - 1 = \omega\mu_j\sqrt{\lambda_j}, \end{aligned}$$

izlazi da ne bi bilo preteško dobiti neki rezultat o optimalnosti za JOR metodu. Probajte ga dobiti. Međutim, to se ne isplati. SOR metoda s optimalnim parametrom je bitno brža.

Teorem 7.8.2. o optimalnom izboru parametra u SOR metodi, osim konvergencije Jacobijeve metode, ima još i pretpostavku da su sve svojstvene vrijednosti matrice R_{Jac} realne. Kad bismo znali da je R_{Jac} simetrična (ili hermitska), onda je ta pretpostavka ispunjena. Međutim, to obično nije slučaj.

U principu znamo da je polazna matrica A simetrična ili hermitska. Tada je $R_{Jac} = D^{-1}(\widetilde{L} + \widetilde{L}^*)$, što ne mora biti simetrična matrica, osim ako D nije skalarna matrica, tj. ima konstantnu dijagonalu. Ako je A još i pozitivno definitna, onda je (vidjeti dokaz teorema 7.7.3.)

$$R_{Jac} = D^{-1}(\widetilde{L} + \widetilde{L}^*) = D^{-1/2}(D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2})D^{1/2},$$

jer je i D pozitivno definitna (dijagonalna s pozitivnim elementima), pa je R_{Jac} slična simetričnoj (hermitskoj) matrici $D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2}$, odakle slijedi da ima realne svojstvene vrijednosti μ_j.

To još uvijek ne garantira konvergenciju Jacobijeve metode, kao što ćemo pokazati na primjeru u sljedećem odjeljku. Međutim, ako je A još i konzistentno poredana, onda i Jacobijeva metoda mora konvergirati.

Teorem 7.8.3. *Neka je A simetrična (hermitska) i pozitivno definitna matrica i pretpostavimo da A ima konzistentan poredak. Onda matrica iteracije R_{Jac} u Jacobijevoj metodi ima samo realne svojstvene vrijednosti i vrijedi $\rho(R_{Jac}) < 1$, tj. Jacobijeva metoda konvergira.*

Dokaz:

Prvi dio tvrdnje da je $\mu_j \in \mathbb{R}$ smo već dokazali. Iz pozitvne definitnosti od

$A = D - \widetilde{L} - \widetilde{L}^*$ slijedi da za bilo koji vektor $x \neq 0$ vrijedi

$$0 < x^*Ax = x^*(D - \widetilde{L} - \widetilde{L}^*)x$$

pa je

$$x^*(\widetilde{L} + \widetilde{L}^*)x < x^*Dx.$$

Ako definiramo $y = D^{1/2}x$, onda je $y \neq 0$ ako i samo ako je $x \neq 0$, i za sve $y \neq 0$ vrijedi

$$y^*D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2}y < y^*y.$$

što znači da je matrica $I - D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2}$ pozitivno definitna. Zbog toga su sve svojstvene vrijednosti matrice $D^{-1/2}(\widetilde{L} + \widetilde{L}^*)D^{-1/2}$ strogo manje od 1, pa to zbog sličnosti vrijedi i za R_{Jac}.

Sad iskoristimo da A ima konzistentan poredak, pa iz teorema 7.8.1.(a) slijedi da svojstvene vrijednosti od R_{Jac} dolaze u $\pm$ parovima, tj. $\mu_j < 1$ povlači i $\mu_j > -1$. Dobivamo da je $\rho(R_{Jac}) < 1$, pa Jacobijeva metoda konvergira. ■

Naravno da tada konvergiraju i Gauss–Seidelova metoda i SOR(ω) metode za $\omega \in (0, 2)$. To vrijedi i bez pretpostavke o konzistentnom poretku, ali tada nemamo optimalni izbor parametra za SOR metodu.

Ako imamo zadanu matricu A, onda nije jednostavno provjeriti da li ona ima konzistentan poredak koristeći definiciju 7.8.3., odnosno algebarsko svojstvo (7.8.1). Mnogo je lakše provjeriti "grafovska" svojstva, poput svojstva (A), koja garantiraju konzistentan poredak za permutiranu matricu PAP^T. Zbog toga je vrlo korisno naći generalizacije svojstva (A).

Na tu temu postoje mnogi rezultati. U primjeru 7.8.1. imali smo blok trodijagonalu matricu s regularnim dijagonalnim matricama.

Definicija 7.8.4. *Matrica A ima svojstvo (A^π) ako postoji matrica permutacije P takva da se PAP^T može particionirati u blok trodijagonalnu matricu oblika*

$$PAP^T = \begin{bmatrix} D_1 & U_1 & & \\ L_1 & \ddots & \ddots & \\ & \ddots & \ddots & U_{n-1} \\ & & L_{n-1} & D_n \end{bmatrix},$$

gdje su D_i, $i = 1, \ldots, m$, regularne matrice.

Dokažite da matrice sa svojstvom (A^π) imaju konzistentan poredak. Dokaz ide slično kao u propoziciji 7.8.1., tako da se pogodnom dijagonalnom matricom sličnosti transformira $R_{Jac}(\alpha)$ i pokaže da njene svojstvene vrijednosti ne ovise o α. Ovaj rezultat je također dokazao Young, 1950. godine.

Na kraju, spomenimo da je R. S. Varga generalizirao i ovaj rezultat na tzv. p-cikličke matrice, uz $p \geq 2$, pri čemu su 2-cikličke matrice upravo one sa svojstvom (A^π).

7.9. Primjeri — akademski i praktični

Za početak, ilustrirajmo odnos između Jacobijeve i Gauss–Seidelove metode, u smislu da postoje primjeri kad jedna od njih konvergira, a druga ne.

Znamo da Gauss–Seidelova metoda konvergira za simetrične (hermitske) pozitivno definitne matrice. Jacobijeva metoda tad ne mora konvergirati. Naravno, treba uzeti matricu koja nije dijagonalno dominantna po recima ili stupcima, tj. vandijagonalni elementi trebaju biti relativno veliki.

Primjer 7.9.1. *Lako se provjerava da je matrica*

$$A = \begin{bmatrix} 1 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 1 \end{bmatrix}$$

simetrična i pozitivno definitna, jer su sve vodeće glavne minore pozitivne: $D_1 = 1$, $D_2 = 1 - (0.9)^2 = 0.19$ *i*

$$\begin{aligned} D_3 &= 1 + 2(0.9)^3 - 3(0.9)^2 = 1 + 2 \cdot 0.729 - 3 \cdot 0.81 = 1 + 1.458 - 2.43 \\ &= 2.458 - 2.43 = 0.028. \end{aligned}$$

Matrica iteracije u Jacobijevoj metodi je $(D = I)$*,*

$$R_{Jac} = - \begin{bmatrix} 0 & 0.9 & 0.9 \\ 0.9 & 0 & 0.9 \\ 0.9 & 0.9 & 0 \end{bmatrix}$$

i ima svojstvene vrijednosti $\mu_1 = \mu_2 = 0.9$ *i* $\mu_3 = -1.8$*. Dakle,* $\rho(R_{Jac}) = 1.8$*, pa Jacobijeva metoda ne konvergira za sve početne iteracije.*

Pogledajmo usput što daje teorem 7.7.3. *o konvergenciji JOR metode. Najmanja svojstvena vrijednost od* R_{Jac} *je* $\mu := \min_j \mu_j = -1.8$*. Iz* (7.7.1) *dobivamo da JOR*(ω) *metoda konvergira za sve parametre* ω *za koje vrijedi*

$$0 < \omega < \frac{2}{1 - \mu} = \frac{5}{7} < 1,$$

tj. imamo pravu podrelaksaciju.

Primjer 7.9.2. *Za matricu*

$$A = \begin{bmatrix} 1 & 1 & -1 \\ \frac{1}{2} & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

odmah vidimo da nije dijagonalno dominantna. Ipak, Jacobijeva metoda konvergira s matricom iteracije

$$R_{Jac} = \begin{bmatrix} 0 & -1 & 1 \\ -\frac{1}{2} & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix}.$$

Pokažite da je $\rho(R_{Jac}) < 1$. *Izračunajte matricu* R_{GS} *i pokažite da je* $\rho(R_{GS}) > 1$, *tj. da Gauss–Seidelova metoda ne konvergira za sve početne iteracije.*

U nastavku ovog odjeljka ilustrirat ćemo iterativne metode na jednoj klasi matrica koja se javlja u praksi i ima tipična svojstva.

Iterativne metode za rješavanje linearnih sustava koriste se kod rješavanja rubnog problema za obične diferencijalne jednadžbe i kod rješavanja parcijalnih diferencijalnih jednadžbi.

Ideja metoda koje vode na linearne sustave je diskretizacija, tj. umjesto da tražimo funkciju koja bi bila rješenje problema, aproksimiramo rješenje u odabranim čvorovima, tako da aproksimiramo derivacije koje se javljaju u problemu.

Ako je problem jednodimenzionalan, obično se interval na kojem tražimo rješenje ekvidistantno podijeli. Kod dvodimenzionalnog problema, područje se obično podijeli u pravokutnu mrežu.

Na primjer, želimo riješiti tzv. Poissonovu (eliptičku parcijalnu diferencijalnu) jednadžbu u dvije dimenzije

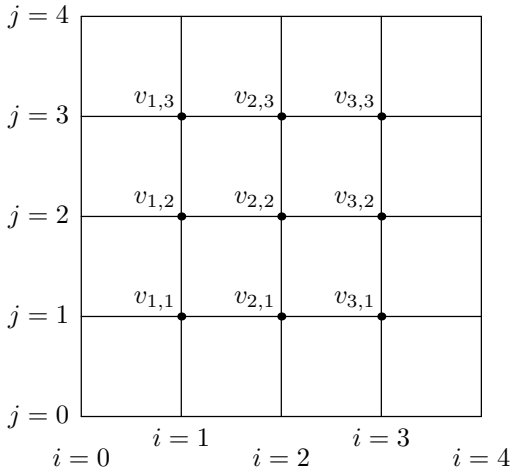
$$-\frac{\partial^2 v(x,y)}{\partial x^2} - \frac{\partial^2 v(x,y)}{\partial y^2} = f(x,y)$$

na kvadratu $\{(x,y) \mid 0 < x, y < 1\}$ uz rubni uvjet $v = 0$, tj. funkcija v je jednaka 0 na rubu kvadrata. Kvadrat podijelimo u mrežu čvorova, a da nam bude jednostavnije, pretpostavimo da je i ta mreža kvadratna, tj. korak u x i y smjeru je jednak

$$h = \frac{1}{N+1}.$$

Uz tako definirane korake, unutarnji čvorovi mreže su točke (x_i, y_j), gdje je $x_i = ih$, $y_j = jh$, za $i, j = 1, \ldots, N$. Dakle, imamo $n := N^2$ unutarnjih čvorova mreže.

Takva mreža za $N = 3$ izgleda ovako:



Vrijednost rješenja u čvoru (x_i, y_j) označavamo s $v_{i,j} := v(ih, jh)$, a funkcijsku vrijednost s $f_{i,j} := f(ih, jh)$.

Kako se aproksimiraju derivacije? Pretpostavimo da su točke x_{i-1}, x_i i x_{i+1} ekvidistantne i da je $x_{i+1} - x_i = x_i - x_{i-1} = h$. Ako za funkciju f postoji Taylorov red oko x_i, onda uvrštavanjem točaka x_{i-1} i x_{i+1} u taj red dobivamo

$$f(x_{i-1}) = f(x_i) - \frac{f'(x_i)}{1!}h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(\xi_{i,i-1})}{3!}h^3$$
$$f(x_{i+1}) = f(x_i) + \frac{f'(x_i)}{1!}h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_{i,i+1})}{3!}h^3.$$

Oduzimanjem prve jednakosti od druge, izlazi

$$f(x_{i+1}) - f(x_{i-1}) = 2\frac{f'(x_i)}{1!}h + O(h^3),$$

pa je dobra aproksimacija derivacije funkcije f u točki x_i

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h}.$$

Ta aproksimacija derivacije obično se naziva simetrična (centralna) razlika.

Prvo, izaberimo dva čvora (x_i^-, y_j) i (x_i^+, y_j) takva da je

$$x_i^- = \frac{x_i + x_{i-1}}{2}, \qquad x_i^+ = \frac{x_i + x_{i+1}}{2}.$$

Korištenjem simetrične razlike, aproksimirajmo prve parcijalne derivacije u ta dva čvora. Dobivamo

$$\frac{\partial v}{\partial x}\Big|_{x=x_i^-,\ y=y_j} \approx \frac{v_{i,j} - v_{i-1,j}}{h},$$
$$\frac{\partial v}{\partial x}\Big|_{x=x_i^+,\ y=y_j} \approx \frac{v_{i+1,j} - v_{i,j}}{h}.$$

Ponovno, primijenimo simetričnu razliku, ali ovaj puta za drugu parcijalnu derivaciju po x u (x_i, y_j), korištenjem derivacije u točkama (x_i^-, y_j) i (x_i^+, y_j). Odmah imamo

$$\frac{\partial^2 v}{\partial x^2}\Big|_{x=x_i,\ y=y_j} \approx \frac{1}{h}\Big(\frac{\partial v}{\partial x}\Big|_{x=x_i^+,\ y=y_j} - \frac{\partial v}{\partial x}\Big|_{x=x_i^-,\ y=y_j}\Big) = \frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{h^2}.$$

Na isti način dobivamo i formulu za drugu parcijalnu derivaciju po y

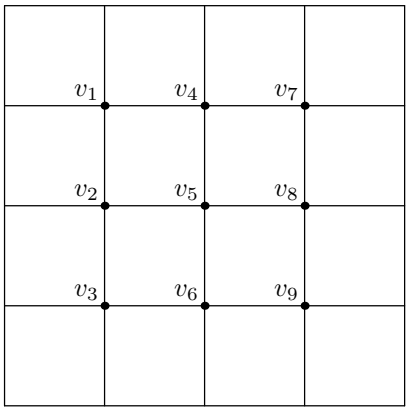
$$\frac{\partial^2 v}{\partial y^2}\Big|_{x=x_i,\ y=y_j} \approx \frac{1}{h}\Big(\frac{\partial v}{\partial y}\Big|_{x=x_i,\ y=y_j^+} - \frac{\partial v}{\partial y}\Big|_{x=x_i,\ y=y_j^-}\Big) = \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{h^2}.$$

Uvrstimo li te aproksimacije derivacija u diferencijalnu jednadžbu, dobivamo

$$4v_{i,j} - v_{i-1,j} - v_{i+1,j} - v_{i,j-1} - v_{i,j+1} = h^2 f_{i,j}, \quad 1 \le i, j \le N. \tag{7.9.1}$$

Pitanje je kako treba napisati ove jednadžbe, tako da se dobije linearni sustav s nekom strukturom. Postoje dva načina da bi se to napravilo. Jedan je sekvencijalno numeriranje $v_{i,j}$ po recima ili stupcima (slijeva nadesno, ili zdesna nalijevo, odozgo nadolje ili odozdo nagore), a drugi tzv. crveno–crni poredak čvorova.

Ako v_{ij} sekvencijalno numeriramo po stupcima odozgo nadolje, na primjer za $N = 3$, dobivamo ovakav poredak čvorova



Dakle, u (7.9.1) lako zamjenjujemo $v_{i,j}$ s v_k. Ako se na isti način transformiraju i $f_{i,j}$ u f_k, onda dobivamo linearni sustav

$$T_{N\times N} v = h^2 f,$$

gdje je $v = [v_1, v_2, \ldots, v_{N\times N}]^T$, $f = [f_1, f_2, \ldots, f_{N\times N}]^T$, a matrica $T_{N\times N}$ ima N blok-redaka i stupaca, svaki dimenzije N. Matrica

$$T_{N\times N} = \begin{bmatrix} T_N + 2I_N & -I_N & & \\ -I_N & \ddots & \ddots & \\ & \ddots & \ddots & -I_N \\ & & -I_N & T_N + 2I_N \end{bmatrix}, \tag{7.9.2}$$

pri čemu je I_N jedinična matrica reda N, a

$$T_N = \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}.$$

Može se pokazati da je T_N matrica koja nastaje diskretizacijom odgovarajuće jednodimenzionalne Poissonove jednadžbe.

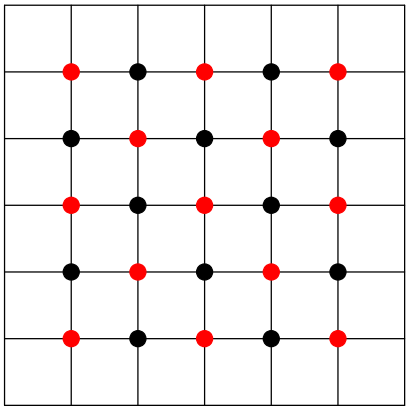
Na primjer, za $N = 3$, matrica linearnog sustava je

$$T_{3\times 3} = \left[\begin{array}{ccc:ccc:ccc} 4 & -1 & & -1 & & & & & \\ -1 & 4 & -1 & & -1 & & & & \\ & -1 & 4 & & & -1 & & & \\ \hdashline -1 & & & 4 & -1 & & -1 & & \\ & -1 & & -1 & 4 & -1 & & -1 & \\ & & -1 & & -1 & 4 & & & -1 \\ \hdashline & & & -1 & & & 4 & -1 & \\ & & & & -1 & & -1 & 4 & -1 \\ & & & & & -1 & & -1 & 4 \end{array}\right]$$

Uočite da je matrica $T_{N\times N}$ slabo dijagonalno dominantna i ireducibilna, pa će i Jacobijeva i Gauss–Seidelova metoda konvergirati. Dapače, pokazat ćemo da $T_{N\times N}$ ima svojstvo (A) i da je konzistentno poredana.

Ako čvorove $v_{i,j}$ poredamo u tzv. crveno–crni poredak, dobit ćemo konzistentno poredanu matricu. Crveno–crni poredak dobivamo tako da ih obojamo poput šahovske ploče: svaki crveni čvor (osim rubnog) je okružen s četiri crna susjeda i obratno.

Na primjer, za $N = 5$ takvo crveno–crno bojanje čvorova izgleda ovako:



Ako zatim sve čvorove koji su crveno obojani popišemo prije crnih (dodijelimo im indekse prije crnih), ili obratno, dobit ćemo blok matricu oblika

$$PT_{N\times N}P^T = \begin{bmatrix} D_1 & T_{12} \\ T_{21} & D_2 \end{bmatrix}.$$

Lako je vidjeti da su dijagonalni blokovi baš dijagonalne matrice, jer ne postoji veza između dva crvena ili dva crna čvora (osim čvora sa samim sobom).

Konkretno, crveno–crni poredak za matricu $T_{3\times 3}$ daje

$$PT_{3\times 3}P^T = \left[\begin{array}{ccccc:cccc} 4 & & & & & -1 & -1 & & \\ & 4 & & & & -1 & & -1 & \\ & & 4 & & & -1 & -1 & -1 & -1 \\ & & & 4 & & & -1 & & -1 \\ & & & & 4 & & & -1 & -1 \\ \hdashline -1 & -1 & -1 & & & 4 & & & \\ -1 & & -1 & -1 & & & 4 & & \\ & -1 & -1 & & -1 & & & 4 & \\ & & -1 & -1 & -1 & & & & 4 \end{array}\right].$$

Dakle, da bismo ispitali konvergenciju iterativnih metoda, dovoljno je naći spektralni radijus matrice R_{Jac}. Prvo, nađimo rastav (cijepanje) matrice $T_{N\times N}$

$$T_{N\times N} = 4I_{N\times N} - (4I_{N\times N} - T_{N\times N}),$$

pa je $M = 4I_{N\times N}$, $K = (4I_{N\times N} - T_{N\times N})$,

$$R_{Jac} = M^{-1}K = (4I_{N\times N})^{-1}\,(4I_{N\times N} - T_{N\times N}) = I_{N\times N} - \frac{1}{4}\,T_{N\times N}.$$

Drugim riječima, R_{Jac} je polinom od $T_{N\times N}$, pa ako je $\lambda_{i,j}$ svojstvena vrijednost od $T_{N\times N}$, onda je $1 - \lambda_{i,j}/4$ svojstvena vrijednost od R_{Jac}.

Pametnim raspisivanjem matrice $T_{N\times N}$, može se pokazati da su svojstvene vrijednosti matrice $T_{N\times N}$ jednake

$$\lambda_{i,j} = 4 - 2\Big(\cos\frac{\pi i}{N+1} + \cos\frac{\pi j}{N+1}\Big),$$

odakle slijedi da je

$$\rho(R_{Jac}) = \max_{i,j}\left|1 - \frac{\lambda_{i,j}}{4}\right| = \left|1 - \frac{\lambda_{1,1}}{4}\right| = \left|1 - \frac{\lambda_{N,N}}{4}\right| = \cos\frac{\pi}{N+1}.$$

Odmah je vidljivo da porastom N argument kosinusa ide prema nuli, pa će $\rho(R_{Jac})$ biti sve bliže 1, a iterativne će metode sve sporije konvergirati.

Čak štoviše, možemo procijeniti $\rho(R_{Jac})$ za velike N. Dovoljno dobra aproksimacija bit će prva dva člana u Taylorovom redu za funkciju kosinus

$$\rho(R_{Jac}) = \cos\frac{\pi}{N+1} \approx 1 - \frac{\pi^2}{2(N+1)^2}.$$

Spektralni radijus za Gauss–Seidelovu metodu lako je dobiti koristeći korolar 7.8.1.

$$\rho(R_{GS}) = (\rho(R_{Jac}))^2 = \cos^2\frac{\pi}{N+1}.$$

Približno, kvadriranjem prva dva člana u Taylorovom redu, vrijedi

$$\rho(R_{GS}) \approx 1 - \frac{\pi^2}{(N+1)^2}.$$

Konačno, koristeći teorem 7.8.2., za SOR(ω) dobivamo

$$\omega_{opt} = \frac{2}{1 + \sin\frac{\pi}{N+1}}, \qquad \rho(R_{SOR(\omega_{opt})}) = \frac{\cos^2\frac{\pi}{N+1}}{\Big(1 + \sin\frac{\pi}{N+1}\Big)^2}.$$

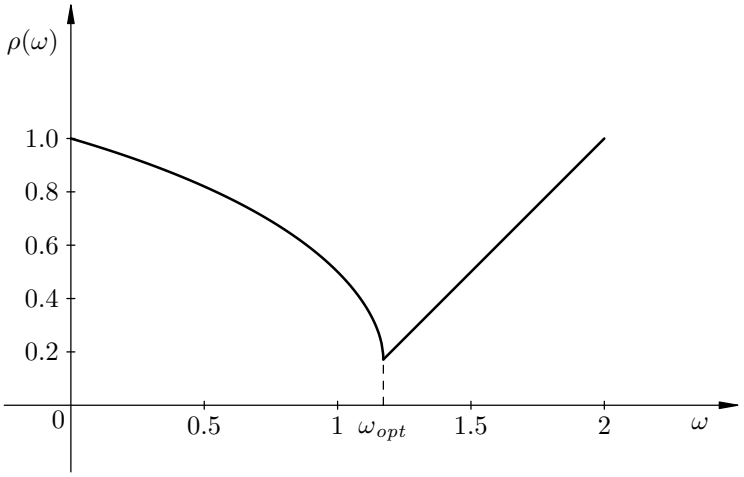
Veličinu $\rho(R_{SOR(\omega_{opt})})$ možemo približno ocijeniti za velike N. Vrijedi

$$\frac{\cos^2\frac{\pi}{N+1}}{\Big(1 + \sin\frac{\pi}{N+1}\Big)^2} = \frac{1 - \sin\frac{\pi}{N+1}}{1 + \sin\frac{\pi}{N+1}} = 1 - \frac{2\sin\frac{\pi}{N+1}}{1 + \sin\frac{\pi}{N+1}} \approx 1 - 2\sin\frac{\pi}{N+1} \approx 1 - \frac{2\pi}{N+1}.$$

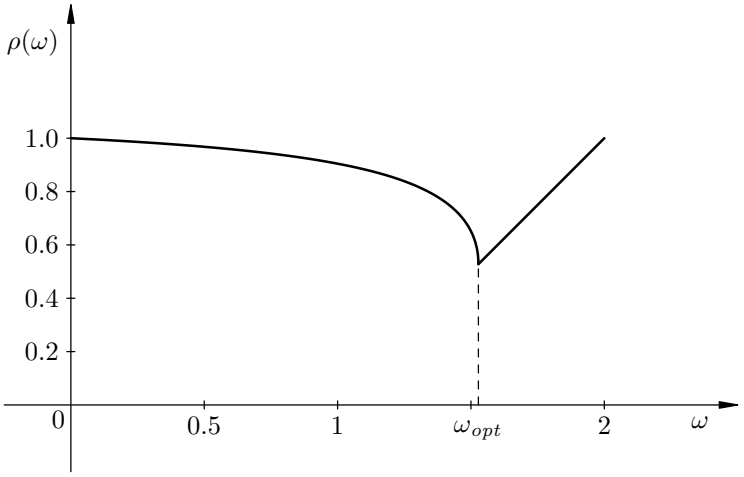
Primijetite da spektralni radijus i kod Jacobijeve metode i kod Gauss–Seidelove metode ima oblik $1 - O(1/N^2)$, dok kod SOR metode s optimalnim parametrom spektralni radijus ima oblik $1 - O(1/N)$, što pokazuje da bi SOR za optimalni izbor parametra morao biti reda veličine N puta brži i od Jacobijeve i od Gauss–Seidelove metode.

Pogledajmo kako se ponaša $\rho(R_{SOR(\omega)})$ kao funkcija od ω, te ovisno o N, kako se ponaša ω_{opt}. Redom, za $N = 3$, 9, 16, 25, dobivamo sljedeće grafove

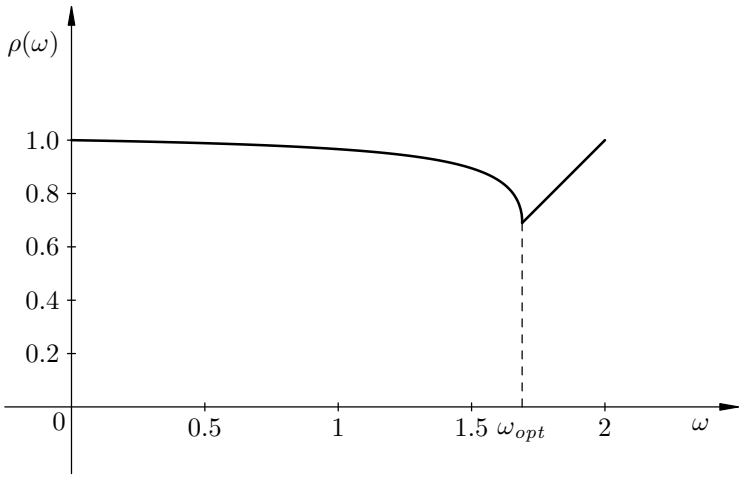
$N = 3$



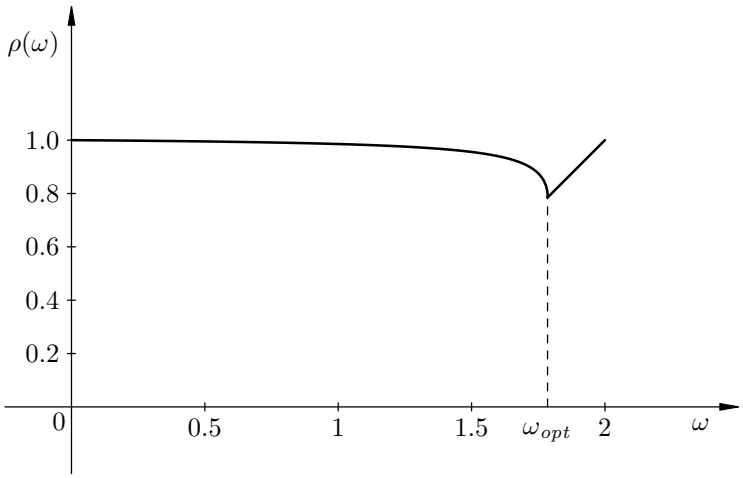
$N = 9$



$N = 16$



$N = 25$



Kao što smo očekivali, optimalni se parametar pomiče prema 2, a $\rho(R_{SOR(\omega_{opt})})$ postaje sve bliže 1.

U sljedećoj tablici dan je pregled spektralnih radijusa za različite iterativne metode za razne N.

N	$\rho(R_{Jac})$	$\rho(R_{GS})$	ω_{opt}	$\rho(R_{SOR(\omega_{opt})})$
4	0.8090169944	0.6545084972	1.2596161837	0.2596161837
9	0.9510565163	0.9045084972	1.5278640450	0.5278640450
16	0.9829730997	0.9662361147	1.6895466227	0.6895466227
25	0.9927088741	0.9854709087	1.7848590191	0.7848590191
36	0.9963974885	0.9928079552	1.8436477483	0.8436477483
49	0.9980267284	0.9960573507	1.8818383898	0.8818383898
64	0.9988322268	0.9976658174	1.9078264563	0.9078264563
81	0.9992661811	0.9985329006	1.9262204896	0.9262204896
100	0.9995162823	0.9990327986	1.9396763332	0.9396763332
121	0.9996684675	0.9993370449	1.9497968003	0.9497968003
144	0.9997652980	0.9995306511	1.9575898703	0.9575898703
169	0.9998292505	0.9996585301	1.9637127389	0.9637127389
196	0.9998728466	0.9997457093	1.9686076088	0.9686076088
225	0.9999033847	0.9998067787	1.9725803349	0.9725803349
256	0.9999252867	0.9998505789	1.9758476503	0.9758476503

Sad kad imamo sve informacije o ovim iterativnim metodama, trebalo bi još samo izabrati neki početni vektor i računati rješenje za zadani f i podjelu N.

Kako se bira početni vektor $x^{(0)}$? Ako znamo matricu iteracija R i iterativnu metodu realiziramo u obliku (7.1.1)

$$x^{(m+1)} = Rx^{(m)} + c, \quad m \in \mathbb{N}_0,$$

onda se obično bira $x^{(0)} = c$, što bi odgovaralo tome da je prethodna iteracija bio nul-vektor. Razlog za to je sasvim jednostavan, posebno u aritmetici računala. Ako je $c = 0$, onda stajemo u jednom koraku i "ne kvarimo" nule. Inače bismo nul-vektor morali dobiti kao limes vektora koji nemaju (sve) nul-komponente, tj. sigurno dolazi do kraćenja.

Međutim, niti jedan od naša 4 algoritma koje smo napisali ne provodi iteracije u tom obliku, jer se R katkad komplicirano računa, već radimo direktno s originalnim podacima A i b. Tada je zgodno zaista uzeti $x^{(0)} = 0$, za slučaj da je $b = 0$, iz istih razloga. Prethodni primjer pokazuje da sve potrebne informacije o matrici iteracija R možemo dobiti i bez da ju eksplicitno izračunamo.

Kako zaustavljamo iteracije? Najlakši način je tzv. heuristička konvergencija. Unaprijed zadamo traženu točnost ε i prekidamo iteracije čim vrijedi

$$\|x^{(m+1)} - x^{(m)}\| \leq \varepsilon,$$

u nekoj pogodno odabranoj vektorskoj normi, na primjer, ∞-normi. To znači da u svakom koraku moramo računati i ovu normu, ali to obično nije pretjerano skupo, a može se isplatiti, ako "slučajno" greška naglo padne u nekoj iteraciji.

S druge strane, ako znamo vrijednost neke operatorske norme $\|R\|$ matrice iteracija (bez pretjerano računanja), s tim da je $\|R\| < 1$, onda unaprijed možemo izračunati potreban broj iteracija. Naime, u pripadnoj vektorskoj normi vrijedi ocjena

$$\|x^{(m+1)} - x\| \leq \frac{\|R\|}{1 - \|R\|} \|x^{(m+1)} - x^{(m)}\|, \tag{7.9.3}$$

kao u Banachovom teoremu o fiksnoj točki. Dokaz ove relacije je jednostavan:

$$\begin{aligned} x^{(m+1)} - x &= R(x^{(m)} - x) = R(x^{(m)} - x^{(m+1)}) + R(x^{(m+1)} - x) \\ (I - R)(x^{(m+1)} - x) &= -R(x^{(m+1)} - x^{(m)}) \\ x^{(m+1)} - x &= -(I - R)^{-1} R(x^{(m+1)} - x^{(m)}), \end{aligned}$$

jer znamo da je $I - R$ regularna matrica. Primjenom norme i ocjenama izlazi (7.9.3). Iz te ocjene dobivamo i

$$\|x^{(m+1)} - x\| \leq \frac{\|R\|^{m+1}}{1 - \|R\|} \|x^{(1)} - x^{(0)}\|,$$

a iz ove relacije možemo, nakon prve iteracije, izračunati potreban broj iteracija $m + 1$, tako da, do na greške zaokruživanja, vrijedi

$$\|x^{(m+1)} - x\| \leq \varepsilon.$$

Još jedan praktični primjer za vježbu.

Primjer 7.9.3. *Preformulirajte linearni sustav $A'x = b'$ u $Ax = b$, gdje je*

$$A' = \begin{bmatrix} -0.25 & -0.25 & 0 & 1 \\ 0 & 1 & -0.25 & -0.25 \\ -0.25 & -0.25 & 1 & 0 \\ 1 & 0 & -0.25 & -0.25 \end{bmatrix}, \quad b' = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

tako da imamo sigurnu konvergenciju i Jacobijeve i Gauss–Seidelove metode.

(a) Ako je početna iteracije $x^{(0)} = 0$, napravite četiri iteracije Jacobijevom metodom za zadani sustav.

(b) Ako je početna iteracije $x^{(0)} = 0$, napravite četiri iteracije Gauss–Seidelovom metodom za zadani sustav.

(c) Ocijenite greške približnih rješenja dobijenih pod (a) i (b).

Rješenje

I Jacobijev i Gauss–Seidelova metoda konvergirat će za strogo dijagonalno dominante matrice, drugim riječima treba jednadžbe premjestiti tako da je matrica sustava strogo dijagonalno dominantna (pazite, premještamo i komponente vektora b, ali se to ovdje ne vidi jer su jednake)

$$A = \begin{bmatrix} 1 & 0 & -0.25 & -0.25 \\ 0 & 1 & -0.25 & -0.25 \\ -0.25 & -0.25 & 1 & 0 \\ -0.25 & -0.25 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}.$$

(a) Prvo treba naći matricu R_{Jac}. Budući da je $D = I$ u ovom slučaju, onda je

$$R_{Jac} = D^{-1}(\widetilde{L} + \widetilde{U}) = \begin{bmatrix} 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0 & 0 \\ 0.25 & 0.25 & 0 & 0 \end{bmatrix}.$$

Norma beskonačno prethodne matrice je $\|R_{Jac}\|_\infty = 0.5$, pa će Jacobijeva metoda konvergirati.

Prema formuli

$$x_j^{(m+1)} = \left(b_j - \sum_{\substack{k=1 \\ k \neq j}}^{n} a_{jk} x_k^{(m)}\right) \Big/ a_{jj},$$

uvaživši da je $x^{(0)} = 0$, dobivamo sljedeće iteracije

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
0	0.5	0.75	0.875	0.9375
0	0.5	0.75	0.875	0.9375
0	0.5	0.75	0.875	0.9375
0	0.5	0.75	0.875	0.9375

.

(b) Ponovno, treba naći matricu R_{GS}. Budući da je $D = I$ u ovom slučaju, onda

je

$$R_{GS} = (D - \widetilde{L})^{-1}\widetilde{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -0.25 & -0.25 & 1 & 0 \\ -0.25 & -0.25 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.25 & 0.25 & 1 & 0 \\ 0.25 & 0.25 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0.125 & 0.125 \\ 0 & 0 & 0.125 & 0.125 \end{bmatrix}.$$

Norma beskonačno prethodne matrice je $\|R_{GS}\|_\infty = 0.5$*, pa će Gauss–Seidelova metoda konvergirati.*

Prema formuli

$$x_j^{(m+1)} := \left(b_j - \sum_{k=1}^{j-1} a_{jk} x_k^{(m+1)} - \sum_{k=j+1}^{n} a_{jk} x_k^{(m)} \right) \Big/ a_{jj}$$

uvaživši da je $x^{(0)} = 0$*, dobivamo sljedeće iteracije*

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
0	0.5	0.875	0.96875	0.9921875
0	0.5	0.875	0.96875	0.9921875
0	0.75	0.9375	0.984375	0.99609375
0	0.75	0.9375	0.984375	0.99609375

.

(c) *Za obje iterativne metode vrijedi ocjena greške*

$$\|x^{(m+1)} - x\| \leq \frac{\|R\|}{1 - \|R\|} \|x^{(m+1)} - x^{(m)}\|,$$

Budući da je $\|R_{Jac}\|_\infty = \|R_{GS}\|_\infty = 0.5$*, imamo*

$$\|x^{(m+1)} - x\|_\infty \leq \frac{0.5}{1 - 0.5} \|x^{(m+1)} - x^{(m)}\|_\infty = \|x^{(m+1)} - x^{(m)}\|_\infty.$$

Prema tome greške $\|x^{(m+1)} - x\|_\infty$ *u odgovarajućoj iteraciji možemo ocijeniti s*

k	*Jacobi*	*Gauss–Seidel*
1	0.5	0.75
2	0.25	0.375
3	0.125	0.09375
4	0.0625	0.0234375

.

Točno rješenje datog sustava je $x^T = [1, 1, 1, 1]$.

8. Izvrednjavanje funkcija

Jedan od osnovnih zadataka koji se javlja u numeričkoj matematici je izračunavanje vrijednosti funkcije u nekoj točki ili na nekom skupu točaka (tzv. izvrednjavanje funkcije). Zašto baš to?

Efikasno računanje možemo raditi samo s onim vrstama funkcija za koje imamo dobar algoritam za izvrednjavanje. Pri tome moramo voditi računa o tome da aritmetika računala stvarno podržava samo četiri osnovne aritmetičke operacije, pa samo njih možemo koristiti u algoritmima. Osim toga, i tada računanje nije egzaktno, već u svakoj operaciji imamo greške zaokruživanja. Zbog toga, pri konstrukciji algoritama imamo dva cilja. Dobar algoritam za izvrednjavanje mora (kao, uostalom, i svaki numerički algoritam) zadovoljavati dva uvjeta:

- efikasnost ili brzina, tj. imati što manji broj aritmetičkih operacija;
- točnost, u smislu stabilnosti ili osjetljivosti na greške zaokruživanja.

Oba zahtjeva su posebno bitna baš kod izvrednjavanja, jer se ono obično puno puta koristi, pa i mala lokalna ubrzanja daju velike ukupne uštede u vremenu, a isto vrijedi i za ukupni efekt grešaka zaokruživanja. Općenito, očekujemo da brži algoritam ima i manju grešku, jer imamo manje operacija koje unose grešku u račun. Međutim, ovo **ne mora** biti istina! U mnogim slučajevima možemo drastično popraviti stabilnost algoritma tako da žrtvujemo dio efikasnosti, a katkad čak i bez toga, uz pametnu reformulaciju algoritma.

U ovom poglavlju ćemo više pažnje posvetiti efikasnosti, a manje stabilnosti, osim tamo gdje je nestabilnost opasna. Cilj nam je konstruirati efikasne algoritme i opravdati njihovu efikasnost, a ne analizirati ili dokazivati njihovu stabilnost. U skladu s tim, potencijalne nestabilnosti izlažemo opisno i ilustriramo na primjerima, bez strogih dokaza.

Pretpostavimo da je zadana funkcija $f : D \to \mathbb{R}$, gdje je $D \subseteq \mathbb{R}$ neka domena. Naš zadatak je izračunati vrijednosti te funkcije f u zadanoj točki $x_0 \in D$. Preciznije, moramo sastaviti algoritam koji računa $f(x_0)$. Naravno, točka x_0 može biti bilo koja i naš algoritam mora raditi za sve ulaze $x_0 \in D$.

Trenutno zanemarimo pitanje kako se **zadaje** funkcija f. Naime, ako je f ulaz u algoritam, onda f mora biti zadana s najviše konačno mnogo podataka o f, i ti

podaci moraju jednoznačno odrediti f. Ovo je, očito, fundamentalno ograničenje i bitno smanjuje klasu funkcija koje uopće možemo algoritamski izračunati. Odgovore na takva pitanja daje tzv. teorija izračunljivosti u okviru matematičke logike i osnova matematike.

U praksi odmah dobivamo i bitno jača ograničenja. Naime, ako imamo na raspolaganju samo 4 osnovne aritmetičke operacije, onda su **racionalne** funkcije jedine funkcije f kojima možemo izračunati vrijednost u bilo kojoj točki $x_0 \in D$. A takve funkcije možemo jednoznačno zadati konačnim brojem parametara — na primjer, ponašanjem (vrijednostima) u konačnom broju točaka (vidjeti poglavlje o interpolaciji) ili koeficijentima u nekom prikazu.

Dakle, sigurno trebamo efikasne algoritme za računanje vrijednosti racionalnih funkcija. Ako se sjetimo da racionalnu funkciju možemo napisati kao kvocijent dva polinoma, onda je zgodno imati i algoritme za izvrednjavanje polinoma. Osim toga, polinomi su još jednostavnije funkcije, jer nema dijeljenja, definirane su na cijeloj domeni (nema problema s nultočkama nazivnika), a koriste se "svagdje".

I da završimo ovo filozofiranje o izvrednjavanju funkcija. Strogo govoreći, sve ostale funkcije moramo **aproksimirati** na neki način — ako ni zbog čega drugog, onda zato jer naša aritmetika nije dovoljno jaka da bismo izračunali njihovu vrijednost u točki. To nipošto nije jedini razlog za aproksimacije funkcija, ali i on pokazuje zašto je aproksimacija centralni problem numeričke analize. Uostalom, u nastavku se gotovo isključivo bavimo raznim metodama za nalaženje različitih vrsta aproksimacija (i to, uglavnom, funkcija)!

Međutim, u praksi možemo pretpostaviti da za neke osnovne matematičke funkcije f **već imamo** dobre aproksimacije za približno računanje vrijednosti $f(x_0)$ u zadanoj točki x_0:

- procesor računala ("hardware") ima ugrađene aproksimacije i odgovarajuće instrukcije za njihov poziv (izvršavanje), ili
- koristimo neki gotovi (pot)program ("software") koji to radi.

Standardno, to su: $\sqrt{x}$ (a katkad i opće potenciranje x^α), trigonometrijske, eksponencijalne, hiperboličke i njima inverzne funkcije. Kako se nalaze takve aproksimacije za razne funkcije f za "hardware" ili "software" implementaciju — o tome kasnije, kod aproksimacija (jasno je da su polinomne ili racionalne)!

Bez obzira na realizaciju, u oba slučaja, bitno je samo to da ih možemo direktno koristiti u našim algoritmima i da znamo da izračunata vrijednost tražene funkcije f u zadanoj točki x_0 ima malu relativnu grešku u odnosu na točnost računanja. Dakle, možemo pretpostaviti da za $f\ell(f(x_0))$ vrijede iste ili slične ocjene kao i za osnovne aritmetičke operacije (vidjeti (2.6.2), ali i primjer 4.1.1.).

Reklo bi se: "gotova priča", dalje se ne trebamo brinuti oko toga. Međutim, nije baš tako. Računanje $f(x_0)$, u principu, traje **dulje**, a katkad i puno dulje, nego

što je trajanje jedne osnovne aritmetičke operacije (čak i ako sve četiri operacije nemaju isto ili podjednako trajanje). Za osnovne funkcije, taj omjer može biti 10 pa i više puta. Zato ponekad izbjegavamo puno poziva takvih funkcija, pogotovo ako se to može razumno izbjeći, tj. efikasno i bez većeg gubitka točnosti.

U mnogim slučajevima se to može napraviti i u ovom poglavlju ćemo pokazati neke opće algoritme tog tipa. Ideja je da se iskoriste neke rekurzivne relacije koje zadovoljavaju takve i slične, a za aproksimaciju važne funkcije. Na primjer, u teoriji se obično koriste ortogonalni sustavi funkcija za aproksimaciju, a funkcije takvog sustava zadovoljavaju tročlanu rekurziju, što je ključno za efikasno računanje.

8.1. Hornerova shema

Polinomi su najjednostavnije algebarske funkcije. Možemo ih definirati nad bilo kojim prstenom R u obliku

$$p(x) = \sum_{i=0}^{n} a_i x^i, \quad n \in \mathbb{N}_0,$$

gdje su $a_i \in R$ koeficijenti iz tog prstena, a x je simbolička "varijabla". Polinomi, kao simbolički objekti, također, imaju algebarsku strukturu prstena.

Međutim, polinome možemo interpretirati i kao funkcije, koje možemo izvrednjavati u svim točkama x_0 iz tog prstena R, uvrštavanjem x_0 umjesto simboličke varijable x. Dobiveni rezultat $p(x_0)$ je opet u R. Zanimaju nas efikasni algoritmi za računanje te vrijednosti.

Složenost očito ovisi o broju članova u sumi. Da broj članova ne bi bio umjetno prevelik, standardno uzimamo da je $p \neq 0$ i da je vodeći koeficijent $a_n \neq 0$, tako da je n stupanj tog polinoma p. Kada želimo naglasiti stupanj, polinom označavamo s p_n.

Algoritmi koje ćemo napraviti u principu rade nad bilo kojim prstenom R, ali neki rezultati o njihovoj složenosti vrijede samo za beskonačna neprebrojiva polja, poput $\mathbb{R}$ i $\mathbb{C}$, što su ionako najvažniji primjeri u praksi. Zbog toga, u nastavku možemo uzeti da radimo isključivo s polinomima nad $\mathbb{R}$ ili $\mathbb{C}$.

Kako zadajemo polinom? Pretpostavljamo da je polinom zadan stupnjem n i koeficijentima $a_0, \ldots, a_n$ u nekoj bazi vektorskog prostora polinoma stpunja ne većeg od n. Na početku koristimo standardnu bazu $1, x, x^2, \ldots, x^n$, a kasnije ćemo modificirati algoritme i za neke druge baze.

Složenost, naravno, mjerimo brojem osnovnih aritmetičkih operacija. Kad radimo nad $\mathbb{R}$, stvar je čista, jer aritmetika računala modelira upravo te operacije, pa je brojanje korektno. No, kad radimo nad $\mathbb{C}$, treba voditi računa o tome da

se kompleksne aritmetičke operacije realiziraju putem realnih, što znači da tek broj realnih operacija daje pravu mjeru složenosti. Baš na tu temu, u nekim kompleksnim algoritmima možemo ostvariti značajne uštede, pažljivim promatranjem realnih operacija.

8.1.1. Računanje vrijednosti polinoma u točki

Zadan je polinom stupnja n

$$p_n(x) = \sum_{i=0}^{n} a_i x^i, \quad a_n \neq 0$$

kojemu treba izračunati vrijednost u točki x_0. To se može napraviti na više načina. Prvo, napravimo to direktno po zapisu, potencirajući. Krenemo li od nulte potencije $x^0 = 1$, svaka sljedeća potencija dobiva se rekurzivno kao

$$x^k = x \cdot x^{k-1}.$$

Imamo li zapamćen x^{k-1}, lako je izračunati x^k korištenjem samo jednog množenja.

Algoritam 8.1.1. (Vrijednost polinoma s pamćenjem potencija)

$sum := a_0$;
$pot := 1$;
for $i := 1$ **to** n **do**
  begin
  $pot := pot * x_0$;
  $sum := sum + a_i * pot$;
  end;
{ Na kraju je $p_n(x_0) = sum$. }

Prebrojimo zbrajanja i množenja koja se javljaju u tom algoritmu. U unutarnjoj petlji javljaju se 2 množenja i 1 zbrajanje. Budući da se petlja izvršava n puta, ukupno imamo

$$2n \text{ množenja } + n \text{ zbrajanja.}$$

Naravno, kad smo nad $\mathbb{C}$, ove operacije su kompleksne.

Izvrednjavanje polinoma u točki može se izvesti i s manje množenja. Ako polinom p_n zapišemo u obliku

$$p_n(x) = (\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1)x + a_0.$$

Algoritam koji po prethodnoj relaciji izvrednjava polinom zove se Hornerova shema. Predložio ga je W. G. Horner, 1819. godine, ali sličan zapis je koristio i Isaac Newton, još 1669. godine.

Algoritam 8.1.2. (Hornerova shema)

$sum := a_n$;
for $i := n - 1$ **downto** 0 **do**
  $sum := sum * x_0 + a_i$;
{ Na kraju je $p_n(x_0) = sum.$ }

Odmah je očito da smo korištenjem ovog algoritma broj množenja prepolovili, tj. da je njegova složenost

$$n \text{ množenja } + n \text{ zbrajanja.}$$

8.1.2. Hornerova shema je optimalan algoritam

Bitno je napomenuti da se Hornerovom shemom izvrednjavaju opći polinomi za koje znamo da imaju većinu nenula koeficijenata. Na primjer, polinom

$$p_{100}(x) = x^{100} + 1$$

besmisleno je izvrednjavati Hornerovom shemom, jer bi to predugo trajalo (binarno potenciranje je brže). Ili, kad izvrednjavamo polinom koji ima samo parne koeficijente

$$p_{2n}(x) = \sum_{i=0}^{n} a_{2i}\, x^{2i},$$

treba modificirati Hornerovu shemu tako da koristi samo parne potencije. Isto vrijedi i za polinom koji ima samo neparne potencije. Sastavite pripadne algoritme.

Za Hornerovu shemu može se pokazati da je optimalan algoritam.

Teorem 8.1.1. (Borodin, Munro) *Za opći polinom n-tog stupnja potrebno je barem n aktivnih množenja. Pod aktivnim množenjem podrazumjevamo množenje između a_i i x.*

Dakle, Hornerova shema ima optimalan broj množenja.

Rezultat prethodnog teorema može se poboljšati samo ako jedan te isti polinom izvrednjavamo u mnogo točaka. Tada se koeficijenti polinoma prije samog izvrednjavanja **adaptiraju** ili **prekondicioniraju**, tako da bismo kasnije imali što manje operacija po svakoj pojedinoj točki.

Zanimljivo je da u slučaju polinoma stupnjeva $n = 1$, 2 i 3, Hornerova shema je optimalna čak i kad računamo vrijednost polinoma u više točaka. Pokažimo jedan primjer adaptiranja koeficijenata za polinom stupnja 4.

Primjer 8.1.1. *Uzmimo opći polinom stupnja* 4

$$p_4(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

i promatrajmo shemu računananja u formi

$$y = (x + c_0)x + c_1,$$
$$p_4(x) = ((y + x + c_2)y + c_3)c_4.$$

Primijetite da ona ima 3 *množenja i* 5 *zbrajanja, ako uspijemo odrediti* c_i *u ovisnosti o* a_i*.*

Izrazimo li ovaj oblik za p_4 *u potencijama od* x*, dobivamo*

$$p_4(x) = c_4x^4 + (2c_0c_4 + c_4)x^3 + (c_0^2 + 2c_1 + c_0c_4 + c_2c_4)x^2 + (2c_0c_1c_4 + c_1c_4 + c_0c_2c_4)x + (c_1^2c_4 + c_1c_2c_4 + c_3c_4).$$

Uočite da veza između a_i *i* c_i *nije linearna. Rješavanjem po* a_i*, dobivamo*

$$\begin{aligned} c_4 &= a_4 & c_1 &= a_1/a_4 - c_0b \\ c_0 &= (a_3/a_4 - 1)/2 & c_2 &= b - 2c_1 \\ b &= a_2/a_4 - c_0(c_0 + 1) & c_3 &= a_0/a_4 - c_1(c_1 + c_2). \end{aligned}$$

Ove relacije zahtjevaju dosta računanja, ali se to obavlja samo jednom, pa će izvrednjavanje u dovoljno točaka zahtjevati manje množenja.

Dapače, V. Pan je pokazao da vrijedi sljedeći teorem.

Teorem 8.1.2. (Pan) *Za bilo koji polinom* p_n *stupnja* $n \geq 3$ *postoje realni brojevi* c*,* d_i*,* e_i*, za* $0 \leq i \leq \lceil n/2 \rceil - 1$*, takvi da se* p_n *može izračunati korištenjem*

$$(\lfloor n/2 \rfloor + 2) \text{ množenja } + n \text{ zbrajanja}$$

po sljedećoj shemi

$$\begin{aligned} y &= x + c \\ w &= y^2 \\ z &:= \begin{cases} (a_n y + d_0)y + e_0, & n \text{ paran}, \\ a_n y + e_0, & n \text{ neparan}, \end{cases} \\ z &:= z(w - d_i) + e_i, \quad \text{za } i = 1, 2, \ldots \lceil n/2 \rceil - 1. \end{aligned}$$

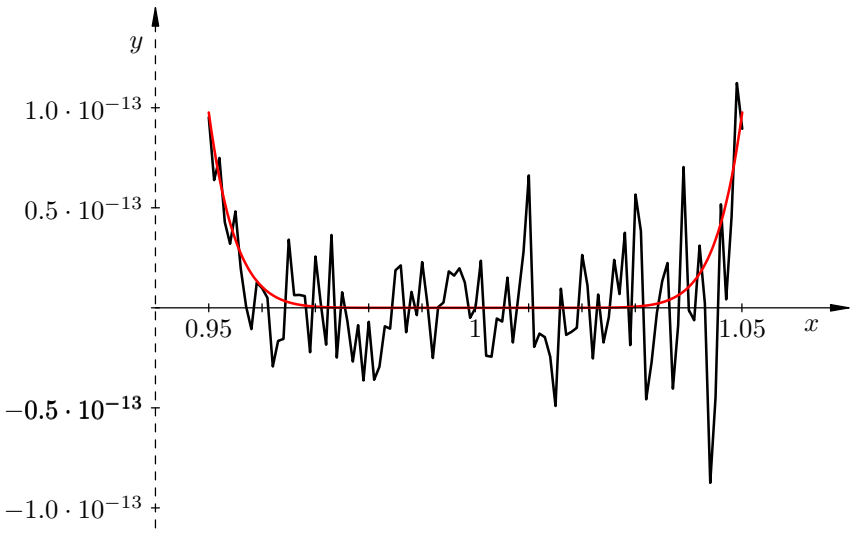
U prethodnom teoremu nismo ništa rekli o tome koliko nam je operacija potrebno za računanje c, d_i i e_i.

Teorem 8.1.3. (Motzkin, Belaga) *Slučajno odabrani polinom stupnja* n *ima vjerojatnost* 0 *da ga se može izračunati za strogo manje od* $\lceil (n + 1)/2 \rceil$ *množenja/dijeljenja ili za strogo manje od* n *zbrajanja/oduzimanja.*

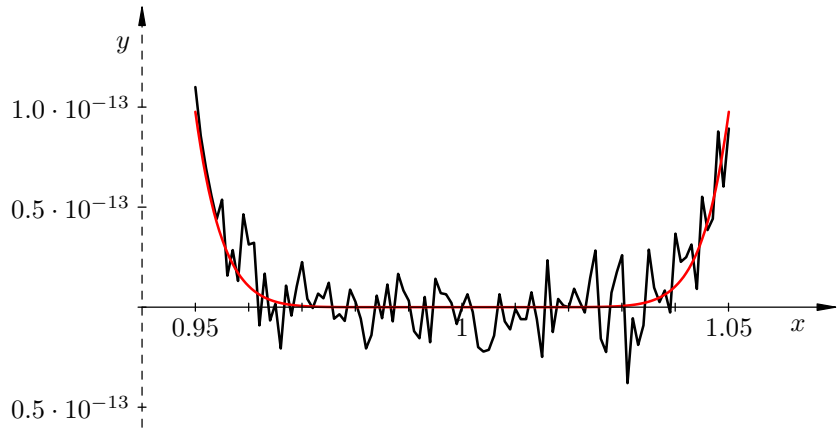
Što to znači? To znači da je red veličine broja operacija u Hornerovoj shemi optimalan za gotovo sve polinome.

8.1.3. Stabilnost Hornerove sheme

Sada znamo da je Hornerova shema optimalan algoritam u smislu efikasnosti. Pažljivom analizom grešaka zaokruživanja nije teško pokazati da je Hornerova shema i stabilan algoritam.



Izvrednjavanje polinoma $(x-1)^{10}$ razvijenog po potencijama od x: korištenjem direktne sumacije u double precision aritmetici.



Izvrednjavanje polinoma $(x-1)^{10}$ razvijenog po potencijama od x: korištenjem Hornerove sheme u double precision aritmetici.

8.1.4. Dijeljenje polinoma linearnim faktorom oblika $x-x_0$

Kako se praktično zapisuje Hornerova shema kad se radi "na ruke"? Napravi se tablica na sljedeći način. U gornjem se redu popišu svi koeficijenti polinoma p_n.

Donji se red formira na sljedeći način: vodeći se koeficijent prepiše, a svi ostali se računaju tako da se posljednji izračunati koeficijent pomnoži s x_0, a zatim mu se doda koeficijent iznad. Na kraju, ispod koeficijenta a_0 piše vrijednost polinoma u točki x_0. Pokažimo kako to funkcionira na konkretnom primjeru.

Primjer 8.1.2. *Izračunajmo vrijednost polinoma*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

u točki $x_0 = -1$.

Formirajmo tablicu:

	2	0	−1	4	0	1
−1	2	−2	1	3	−3	4

.

Dakle, $p_5(-1) = 4$.

Pogledajmo općenito što je značenje koeficijenata c_i koji se javljaju u donjem redu tablice

	a_n	a_{n-1}	$\cdots$	a_1	a_0
x_0	c_{n-1}	c_{n-2}	$\cdots$	c_0	r_0

.

Primijetite da prema algoritmu 8.1.2. (pravilu za popunjavanje tablice) vrijedi da je

$$\begin{aligned} &c_{n-1} = a_n, \\ &c_{i-1} := c_i * x_0 + a_{i-1}, \quad i = n, \ldots, 1. \end{aligned} \tag{8.1.1}$$

Očito je $r_0 = p_n(x_0)$. Promatrajmo polinom koji dobijemo dijeljenjem polinoma p_n linearnim faktorom $x - x_0$. Nazovimo taj polinom q_{n-1}. Tada vrijedi

$$p_n(x) = (x - x_0)q_{n-1}(x) + r_0. \tag{8.1.2}$$

Znamo da je q_{n-1} polinom stupnja $n - 1$ s koeficijentima

$$q_{n-1}(x) = \sum_{i=0}^{n-1} b_{i+1}x^i. \tag{8.1.3}$$

Dodatno, označimo s $b_0 = r_0$.

Uvrstimo li (8.1.3) u (8.1.2) i sredimo koeficijente uz odgovarajuće potencije, dobivamo

$$p_n(x) = b_n x^n + (b_{n-1} - x_0\, b_n)x^{n-1} + \cdots + (b_1 - x_0\, b_2)x + b_0 - x_0\, b_1.$$

Za vodeći koeficijent vrijedi $b_n = a_n$, a za a_i, uz $i < n$, je

$$a_i = b_i - x_0 \cdot b_{i+1},$$

odnosno, b_i možemo izračunati iz b_{i+1}

$$b_i = a_i + x_0 \cdot b_{i+1}.$$

Primijetite da je to relacija istog oblika kao (8.1.1), samo s pomaknutim indeksima, pa je

$$b_i = c_{i-1}, \quad i = 1, \ldots, n,$$

tj. koeficijenti koje dobijemo u Hornerovoj shemi su baš koeficijenti kvocijenta i ostatka pri dijeljenju polinoma p_n linearnim faktorom $x - x_0$.

Primjer 8.1.3. *Podijelimo*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

linearnim polinomom $x + 1$.

Primijetite da je to ista tablica kao u primjeru 8.1.2., *pa imamo*

	2	0	−1	4	0	1
−1	2	−2	1	3	−3	4

.

Odatle lako čitamo

$$2x^5 - x^3 + 4x^2 + 1 = (x + 1)\left(2x^4 - 2x^3 + x^2 + 3x - 3\right) + 4.$$

Konačno, napišimo algoritam koji nalazi koeficijente pri dijeljenju polinoma linearnim polinomom.

Algoritam 8.1.3. (Dijeljenje polinoma linearnim faktorom $(x - x_0)$**)**

$b_n := a_n$;
for $i := n - 1$ **downto** 0 **do**
  $b_i := b_{i+1} * x_0 + a_i$;

8.1.5. Potpuna Hornerova shema

Što se događa ako postupak dijeljenja polinoma linearnim faktorom nastavimo, tj. ponovimo više puta?

Vrijedi

$$\begin{aligned} p_n(x) &= (x - x_0)q_{n-1}(x) + r_0 \\ &= (x - x_0)[(x - x_0)q_{n-2}(x) + r_1] + r_0 \\ &= (x - x_0)^2 q_{n-2}(x) + r_1(x - x_0) + r_0 \\ &= \cdots \\ &= r_n(x - x_0)^n + \cdots + r_1(x - x_0) + r_0. \end{aligned}$$

Dakle, polinom p_n napisan je razvijeno po potencijama od $(x-x_0)$. Koja su značenja r_i? Usporedimo dobiveni oblik s Taylorovim polinomom oko x_0

$$p_n(x) = \sum_{i=0}^{n} \frac{p_n^{(i)}(x_0)}{i!} (x - x_0)^i.$$

Odatle odmah izlazi relacija za koeficijente

$$r_i = \frac{p_n^{(i)}(x_0)}{i!},$$

tj. potpuna Hornerova shema računa sve derivacije polinoma u zadanoj točki.

Primjer 8.1.4. *Nađite sve derivacije polinoma*

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1$$

u točki -1.

Formirajmo potpunu Hornerovu tablicu.

	2	0	-1	4	0	1
-1	2	-2	1	3	-3	4
-1	2	-4	5	-2	-1	
-1	2	-6	11	-13		
-1	2	-8	19			
-1	2	-10				
-1	2					

Odatle lako čitamo

$$\begin{aligned} p_5(-1) &= 4, & p_5^{(1)}(-1) &= -1 \cdot 1! = -1, \\ p_5^{(2)}(-1) &= -13 \cdot 2! = -26, & p_5^{(3)}(-1) &= 19 \cdot 3! = 114, \\ p_5^{(4)}(-1) &= -10 \cdot 4! = -240, & p_5^{(5)}(-1) &= 2 \cdot 5! = 240. \end{aligned}$$

Algoritam koji nalazi koeficijente r_i, odnosno derivacije zadanog polinoma u točki, može se napisati u jednom jedinom polju.

Algoritam 8.1.4. (Taylorov razvoj)

for $i := 0$ **to** n **do**
  $r_i := a_i$;
for $i := 1$ **to** n **do**
  for $j := n - 1$ **downto** $i - 1$ **do**
    $r_j := r_j + x_0 * r_{j+1}$;

8.1.6. "Hornerova shema" za interpolacijske polinome

Kao što ćemo vidjeti, kod izvrednjavanja interpolacijskog polinoma u Newtonovoj formi, treba izračunati izraz oblika

$$p_n(x) = a_n (x - x_0)(x - x_1) \cdots (x - x_{n-1}) + a_{n-1} (x - x_0)(x - x_1) \cdots (x - x_{n-2}) + \cdots + a_1 (x - x_0) + a_0,$$

pri čemu su točke x_i točke interpolacije, a x točka u kojoj želimo izračunati vrijednost polinoma.

Algoritam kojim se vrši izvrednjavanje je vrlo sličan Hornerovoj shemi. Označimo s $y_i = x - x_i$. Ponovno, postavimo zagrade kao u Hornerovoj shemi. Vrijedi

$$p_n(x) = (\cdots((a_n y_{n-1} + a_{n-1})y_{n-2} + a_{n-2})y_{n-3} + \cdots + a_1)y_0 + a_0.$$

Algoritam 8.1.5. ("Hornerova shema" za interpolacijske polinome)

$sum := a_n$;
for $i := n - 1$ **downto** 0 **do**
  $sum := sum * (x - x_i) + a_i$;

Dakle, Hornerovu shemu možemo iskoristiti i za ovakav prikaz polinoma, koji više nije u standardnoj bazi.

8.1.7. Hornerova shema za kompleksne argumente realnog polinoma

Sve što smo dosad izvodili, vrijedi općenito, tj. ako su koeficijenti polinoma iz $\mathbb{R}$ ili $\mathbb{C}$, a točka u kojoj računamo vrijednost iz istog polja.

Ako želimo izračunati vrijednost realnog polinoma u kompleksnoj točki, to se može napraviti uz određenu uštedu. Neka je $z_0 = x_0 + iy_0$. Tvrdimo da tada postoji polinom s_2 stupnja 2 s realnim koeficijentima, takav da je $s_2(z_0) = 0$. Dokaz je jednostavan, jer kao posljedica osnovnog teorema algebre slijedi, ako je $x_0 + iy_0$ nultočka polinoma s realnim koeficijentima, onda je to i $x_0 - iy_0$, tj. kod realnog polinoma kompleksne nultočke dolaze u konjugirano-kompleksnim parovima. Koeficijenti polinoma s_2 su

$$s_2(x) = x^2 + px + q = (x - x_0 - iy_0)(x - x_0 + iy_0),$$

tj.

$$p = -2x_0, \quad q = x_0^2 + y_0^2.$$

Po analogiji, podijelimo polazni realni polinom polinomom s_2. Ostatak pri dijeljenju više nije samo konstanta, nego može biti i polinom r stupnja 1

$$p_n(x) = s_2(x)q_{n-2}(x) + r(x).$$

Napišimo prošlu relaciju u zgodnijoj formi, tj. napišimo na "čudan" način polinom r

$$p_n(x) = (x^2 + px + q)\, q_{n-2}(x) + b_1(x + p) + b_0. \tag{8.1.4}$$

Izaberimo oznaku za koeficijente polinoma q_{n-2}, opet s pomaknutim indeksima

$$q_{n-2}(x) = \sum_{i=0}^{n-2} b_{i+2}x^i. \tag{8.1.5}$$

Uvrštavanjem (8.1.5) u (8.1.4) i uspoređivanjem koeficijenata uz odgovarajuće potencije, dobivamo:

$$\begin{aligned}\sum_{i=0}^{n} a_i x^i &= \sum_{i=0}^{n-2} b_{i+2}x^{i+2} + p\sum_{i=0}^{n-2} b_{i+2}x^{i+1} + q\sum_{i=0}^{n-2} b_{i+2}x^i + b_1(x+p) + b_0 \\ &= \sum_{i=2}^{n} b_i x^i + p\sum_{i=1}^{n-1} b_{i+1}x^i + q\sum_{i=0}^{n-2} b_{i+2}x^i + (b_1x + b_0) + pb_1 \\ &= \sum_{i=0}^{n} b_i x^i + p\sum_{i=0}^{n-1} b_{i+1}x^i + q\sum_{i=0}^{n-2} b_{i+2}x^i.\end{aligned}$$

Definiramo li dodatno $b_{n+1} = b_{n+2} = 0$, onda prethodnu relaciju možemo pisati kao

$$\sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n} \left(b_i + pb_{i+1} + qb_{i+2}\right) x^i.$$

Uspoređivanjem koeficijenata lijeve i desne strane, izlazi rekurzivna relacija

$$a_i = b_i + pb_{i+1} + qb_{i+2}, \quad i = n, \ldots, 0,$$

uz start $b_{n+1} = b_{n+2} = 0$. Drugim riječima, koeficijente b_i računamo iz dva koja odgovaraju dvama višim potencijama, tj.

$$b_i = a_i - pb_{i+1} - qb_{i+2}, \quad i = n, \ldots, 0, \tag{8.1.6}$$

ponovno, uz $b_{n+1} = b_{n+2} = 0$.

Što je rezultat? Ako tražimo rezultat dijeljenja kvadratnim polinomom, onda moramo izračunati sve koeficijente b_i, s tim da će indeksi $i = 2, \ldots, n$ dati koeficijente polinoma q_{n-2}, a b_1 i b_0 bit će ostaci pri dijeljenju napisani u formi (8.1.4). Ako želimo ostatak napisan u standardnoj bazi, onda ćemo r napisati kao

$$r(x) = b_1x + (b_0 + b_1p),$$

tj. na kraju računa, kad izračunamo sve koeficijente b_i, postavit ćemo

$$b_0 := b_0 + b_1 p$$

Sljedeći algoritam računa kvocijent polinoma p_n i kvadratnog polinoma s_2 u standardnoj bazi. Koeficijenti p i q su tada ulazni podaci, bez ikakvih ograničenja, tj. q ne mora biti nenegativan.

Algoritam 8.1.6. (Dijeljenje polinoma kvadratnim polinomom)

$b_n := a_n$;
$b_{n-1} := a_n - p * b_n$;
for $i := n-2$ **downto** 0 **do**
  $b_i := a_i - p * b_{i+1} - q * b_{i+2}$;
$b_0 := b_0 + p * b_1$;

Konačno, vratimo se zadatku od kojeg smo krenuli. Ako želimo izračunati vrijednost polinoma p_n u točki $x_0 + iy_0$, onda nam ne trebaju svi b_i-ovi. Promotrimo relaciju (8.1.4). Uočimo da je $s_2(x_0 + iy_0) = 0$ (tako je definiran), pa se (8.1.4) svede na

$$p_n(x_0 + iy_0) = b_1(x_0 + iy_0 + p) + b_0, \tag{8.1.7}$$

to će reći, trebaju nam samo koeficijenti b_1 i b_0 da bismo znali izračunati vrijednost u točki z_0. Relaciju (8.1.7) možemo i ljepše napisati, ako znamo da je

$$p + x_0 = -2x_0 + x_0 = -x_0,$$

pa je

$$p_n(x_0 + iy_0) = b_1(-x_0 + iy_0) + b_0 = (b_0 - x_0 b_1) + iy_0 b_1.$$

Dakle, da bismo izračunali vrijednost polinoma u kompleksnoj točki, ne moramo pamtiti čitavo polje koeficijenata b_i, nego samo "trenutna" tri koje su nam potrebna u rekurziji (8.1.6)

Algoritam 8.1.7. (Vrijednost realnog polinoma u kompleksnoj točki)

$p := -2 * x_0$;
$q := x_0^2 + y_0^2$;
$b_1 := 0$;
$b_0 := a_n$;
for $i := n-1$ **downto** 0 **do**
  begin
  $b_2 := b_1$;
  $b_1 := b_0$;
  $b_0 := a_i - p * b_1 - q * b_2$;
  end;
$\mathrm{Re}(p_n(x_0 + iy_0)) := b_0 - x_0 * b_1$;
$\mathrm{Im}(p_n(x_0 + iy_0)) := y_0 * b_1$;

Kad prebrojimo **realne** operacije u ovom algoritmu, dobivamo

$$(2n+4) \text{ množenja } + (2n+3) \text{ zbrajanja},$$

dok bi obična Hornerova shema za kompleksni polinom u kompleksnoj točki imala

$$4n \text{ množenja } + 4n \text{ zbrajanja},$$

ako smatramo da jedno kompleksno množenje realiziramo na standardan način, korištenjem 4 realna množenja i 2 realna zbrajanja, a kompleksno zbrajanje zahtijeva 2 realna zbrajanja.

U ne tako davnoj prošlosti duljina trajanja množenja u računalu bila je dosta dulja nego duljina trajanja zbrajanja, pa su se ljudi često dovijali "alkemiji" pretvaranja množenja u zbrajanja. Kod množenja kompleksnih brojeva to je lako. Vrijedi

$$(a+bi)\,(c+di) = (ac-bd)+(ad+bc)i = (ac-bd)+[(a+b)\,(c+d)-(ac+bd)]i.$$

Uočite da ova posljednja forma ima samo 3 množenja i 5 zbrajanja (produkti ac i bd se čuvaju kad se izračunaju).

Konačno, što ako želimo izračunati vrijednost polinoma p_n u točki $\bar{z}_0 = x_0 - iy_0$. Moramo li ponovno provoditi postupak? Odgovor je ne. Naime, $\bar{z}_0$ je, također nultočka od s_2, pa se (8.1.4) svede na

$$p_n(x_0 - iy_0) = b_1(-x_0 - iy_0) + b_0 = (b_0 - x_0 b_1) - iy_0 b_1,$$

tj. razlika je samo u tome da mu je imaginarni dio suprotnog predznaka, tj. treba nam još jedna dodatna operacija da istovremeno izračunamo vrijednost polinoma u z_0 i $\bar{z}_0$.

Ako "na ruke" želimo izračunati vrijednost polinoma u kompleksnoj točki, onda je tablica vrlo slična običnoj Hornerovoj shemi.

	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_2	a_1	a_0
q			$-qb_n$	$\cdots$	$-qb_4$	$-qb_3$	$-qb_2$
p		$-pb_n$	$-pb_{n-1}$	$\cdots$	$-pb_3$	$-pb_2$	$-pb_1$
$+$	b_n	b_{n-1}	b_{n-2}	$\cdots$	b_2	b_1	b_0

.

Primjer 8.1.5. *Nađite vrijednost polinoma*

$$p_3(x) = x^3 + 8x^2 + 1$$

u točkama $2 \pm i$.

Uzmimo točku $2+i$*, pa je* $x_0=2$*,* $y_0=1$*, a* $p=-4$*,* $q=5$*. Formirajmo tablicu.*

	1	8	0	1
5			-5	-60
-4		4	48	172
+	1	12	43	113

.

Posljednja dva koeficijenta su $b_1=43$*,* $b_0=113$*, pa je*

$$\mathrm{Re}(p_3(2+i))=b_0-b_1x_0=113-86=27,\quad \mathrm{Im}(p_3(2+i))=b_1y_0=43.$$

Dakle, imamo

$$p_3(2+i)=27+43i,\quad p_3(2-i)=27-43i.$$

8.1.8. Računanje parcijalnih derivacija kompleksnog polinoma

Prisjetimo se, kompleksni se polinom u varijabli z, može zapisati kao dva polinoma u i v u dvije realne varijable x i y, gdje je $z=x+iy$. Vrijedi

$$p_n(z)=u(x,y)+iv(x,y),\quad z=x+iy.$$

Drugačije gledano, funkcije u i v možemo interpretirati i kao realne funkcije kompleksnog argumenta

$$u(z)=\mathrm{Re}\,(p_n(z)),\quad v(z)=\mathrm{Im}\,(p_n(z)),$$

pa, prema prošlom odjeljku, imamo algoritam za nalaženje njihovih vrijednosti.

Da bismo oponašali dijeljenje polinoma linearnim polinomom oblika $x-x_0$, uzmimo da nam i kvadratni polinom ima oblik x^2-px-q (pa svugdje gdje u prošlom odjeljku piše p treba pisati $-p$, a gdje piše q treba pisati $-q$). Dakle, rastav u produkt kvadratnog polinoma i polinoma stupnja $n-2$ u ovakvoj notaciji glasi:

$$p_n(x)=s_2(x)q_{n-2}(x)+r(x),$$

gdje je

$$r(x)=b_1(x-p)+b_0.$$

Algoritam za nalaženje koeficijenata u ovoj formulaciji je: start $b_{n+2}=b_{n+1}=0$, a rekurzija je

$$b_i=a_i+pb_{i+1}+qb_{i+2},\quad i=n,\ldots,0. \tag{8.1.8}$$

Pokazali smo da je tad

$$p_n(z_0)=(b_0-x_0b_1)+iy_0b_1.$$

Shvatimo li $p_n(z)$ kao funkciju u odgovarajućoj točki (x, y), onda su realni i imaginarni dio te funkcije

$$u(x, y) = b_0 - xb_1, \quad v(x, y) = yb_1.$$

Parcijalne derivacije funkcija u i v možemo dobiti korištenjem deriviranja složenih funkcija. Polinomi su analitičke funkcije, pa njihove parcijalne derivacije vrijede Cauchy–Riemannovi uvjeti

$$\frac{\partial u}{\partial x}(x, y) = \frac{\partial v}{\partial y}(x, y), \quad \frac{\partial u}{\partial y}(x, y) = -\frac{\partial v}{\partial x}(x, y).$$

Zbog toga je dovoljno pronaći samo ili parcijalne derivacije jedne od funkcija ili parcijalne derivacije po jednoj varijabli. Iskoristimo li da vrijedi

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial p}\frac{\partial p}{\partial y} + \frac{\partial}{\partial q}\frac{\partial q}{\partial y},$$

dobivamo (izostavljajući pisanje točke u kojoj deriviramo)

$$\begin{aligned} \frac{\partial u}{\partial y} &= \frac{\partial b_0}{\partial y} - x\frac{\partial b_1}{\partial y} = \frac{\partial b_0}{\partial q}(-2y) - x\frac{\partial b_1}{\partial q}(-2y) = 2y\Big(x\frac{\partial b_1}{\partial q} - \frac{\partial b_0}{\partial q}\Big) \\ \frac{\partial v}{\partial y} &= b_1 + y\frac{\partial b_1}{\partial y} = b_1 + y\frac{\partial b_1}{\partial q}(-2y) = b_1 - 2y^2\frac{\partial b_1}{\partial q} \end{aligned} \tag{8.1.9}$$

Dakle, da bismo izračunali vrijednosti parcijalnih derivacija u nekoj točki, dovoljno je znati

$$\frac{\partial b_0}{\partial p}, \frac{\partial b_0}{\partial q}, \frac{\partial b_1}{\partial p}, \frac{\partial b_1}{\partial q}.$$

Uvedimo oznake

$$c_i = \frac{\partial b_i}{\partial p}, \quad d_i = \frac{\partial b_i}{\partial q}, \quad i = n+2, \ldots, 0.$$

Deriviramo li formalno relaciju (8.1.8) prvo po p, a zatim po q, dobivamo

$$\begin{aligned} c_{n+2} &= c_{n+1} = 0 \\ c_i &= b_{i+1} + pc_{i+1} + qc_{i+2}, \quad i = n, \ldots, 0 \\ d_{n+2} &= d_{n+1} = 0 \\ d_i &= b_{i+2} + pd_{i+1} + qd_{i+2}, \quad i = n, \ldots, 0 \end{aligned}$$

Odavde odmah vidimo da c_i-ovi i d_i-ovi tvore istu rekurziju, samo s indeksom translatiranim za 1, tj. vrijedi

$$c_{i+1} = d_i, \quad i = n+1, n, n-1, \ldots, 0.$$

Zbog toga, umjesto s dvije rekurzije za koeficijente možemo raditi samo s jednom, uz paralelno računanje b_i.

Algoritam dobivanja parcijalnih derivacija zove se algoritam Bairstowa.

Algoritam 8.1.8. (Algoritam Bairstowa)

$b_1 := a_n;$
$b_0 := a_{n-1} + p * b_1;$
$c_2 := 0;$
$c_1 := 0;$
$c_0 := a_n;$
for $i := n - 2$ **downto** 0 **do**
begin
$b_2 := b_1;$
$b_1 := b_0;$
$b_0 := a_i + p * b_1 + q * b_2;$
$c_2 := c_1;$
$c_1 := c_0;$
$c_0 := b_1 + p * c_1 + q * c_2;$
end;

Relacija (8.1.9) odmah daje kako se iz c-ova i b-ova dobivaju parcijalne derivacije

$$\frac{\partial u}{\partial y}(x_0, y_0) = -\frac{\partial v}{\partial x}(x_0, y_0) = 2y_0(x_0 d_1 - d_0) = 2y_0(x_0 c_2 - c_1)$$
$$\frac{\partial v}{\partial y}(x_0, y_0) = \frac{\partial u}{\partial x}(x_0, y_0) = b_1 - 2y_0^2\, d_1 = b_1 - 2y_0^2\, c_2.$$

Motivacija za ovaj algoritam potječe iz modifikacije Newtonove metode za traženje realnih nultočaka polinoma. Ako želimo naći kompleksne nultočke realnog polinoma, prvo treba izlučiti kvadratni polinom s konjugirano kompleksnim parom nultočaka. Ta generalizacija Newtonove metode zove se metoda Newton–Bairstow.

8.2. Generalizirana Hornerova shema

U prošlom odjeljku napravili smo nekoliko algoritama za izvrednjavanje polinoma i njegovih derivacija u zadanoj točki. Te algoritme, koji su egzaktni u egzaktnoj aritmetici, možemo koristiti i kao **približne** algoritme za izvrednjavanje redova potencija, tj. analitičkih funkcija.

Pretpostavimo da se funkcija f u okolini neke točke x_0 (u $\mathbb{R}$ ili $\mathbb{C}$) može razviti u red potencija oblika

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n, \tag{8.2.1}$$

s tim da znamo taj red konvergira prema f na toj okolini od x_0. Dodatno, pretpostavimo da znamo sve koeficijente a_n u ovom razvoju, u smislu da ih možemo

brzo i točno izračunati. Naravno, ovu beskonačnu sumu ne možemo efektivno algoritamski izračunati, jer zahtijeva beskonačan broj aritmetičkih operacija.

Međutim, konačne komade ovog razvoja možemo iskoristiti za aproksimaciju funkcije f na toj okolini. Iz konvergencije razvoja po točkama odmah slijedi da, za bilo koju unaprijed zadanu točnost $\varepsilon > 0$, postoji $N \in \mathbb{N}$ takav da je

$$f_N(x) = \sum_{n=0}^{N} a_n (x - x_0)^n, \tag{8.2.2}$$

aproksimacija za $f(x)$ s greškom manjom od ε. Nije bitno da li grešku mjerimo u apsolutnom ili relativnom smislu, osim ako je $f(x) = 0$. Sasvim općenito, potrebna duljina razvoja N ovisi i o ε i o x. No, ako se sjetimo da redovi potencija konvergiraju uniformno na kompaktima, možemo postići i uniformnu aproksimaciju s točnošću ε na takvim kompaktima, pa N onda ovisi samo o ε.

Kad uzmemo u obzir da ionako **približno** računamo u aritmetici računala, ovim pristupom možemo bitno povećati klasu funkcija s kojima možemo računati. Ako je greška ε dovoljno mala, recimo reda veličine osnovne greške zaokruživanja u, onda je pripadna aproksimacija $f_N(x)$ gotovo jednako dobra kao i $f(x)$.

Algoritam za računanje $f_N(x)$ u zadanoj točki x bitno ovisi u tome da li N znamo unaprijed ili ne. Ako ga **ne znamo**, onda se obično koristi sumacija unaprijed, sve dok se izračunata suma ne stabilizira na zadanu točnost. Koliko to može biti opasno, već smo vidjeli u primjeru za $\sin x$. Zbog toga se sumacija unaprijed koristi samo kao "zadnje utočište".

Vrlo često se N može unaprijed odrediti iz analitičkih svojstava funkcije f, tako da dobijemo uniformnu aproksimaciju s točnošću ε na nekom kompaktu. Obično se za taj kompakt uzima neki segment u $\mathbb{R}$, odnosno neki krug u $\mathbb{C}$. Čak nije jako bitno da N bude "savršen", tj. najmanji mogući, ako je takav N teško izračunati. Katkad je sasvim dobra i približna vrijednost za N. Tada je f_N polinom poznatog stupnja N i možemo koristiti Hornerovu shemu i njene varijacije za računanje $f_N(x)$.

Trenutno ne ulazimo u to kako se nalaze takve aproksimacije. Tome ćemo posvetiti punu pažnju u poglavlju o aproksimacijama. Zasad recimo samo to da se izbjegava direktno korištenje redova potencija (8.2.1) i pripadnih polinomnih aproksimacija u obliku (8.2.2), zbog loše uvjetovanosti sustava funkcija

$$\{1, (x - x_0), (x - x_0)^2, \ldots, (x - x_0)^n, \ldots\}$$

i nejednolikog rasporeda pogreške $e(x) = f(x) - f_N(x)$ na domeni aproksimacije.

Umjesto reda potencija (8.2.1), standardno se koriste razvoji oblika

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x), \tag{8.2.3}$$

gdje je $\{p_n \mid n \in \mathbb{N}_0\}$ neki **ortogonalni** sustav funkcija na domeni aproksimacije. U aproksimaciji elementarnih i "manje elementarnih" tzv. specijalnih funkcija vrlo često se koriste tzv. Čebiševljevi polinomi, zbog skoro jednolikog rasporeda greške na domeni. Kasnije ćemo pokazati i algoritam za nalaženje takve "kvazi–uniformne" aproksimacije iz poznatog reda potencija (tzv. Čebiševljeva ekonomizacija).

Razvoj funkcije f u red oblika (8.2.3) je očita generalizacija reda potencija. Njega, također, po istom principu, možemo iskoristiti za aproksimaciju funkcije f, ako znamo da on konvergira prema f na nekoj domeni. "Rezanjem" reda dobivamo aproksimaciju funkcije f

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x), \tag{8.2.4}$$

što je očita generalizacija polinoma iz (8.2.2). Naravno, da bismo izračunali $f_N(x)$ moramo znati sve koeficijente a_n i sve funkcije p_n. Međutim, u većini primjena **nemamo** direktnu "formulu" za računanje vrijednosti $p_n(x)$ u zadanoj točki x, za sve $n \in \mathbb{N}_0$. Umjesto toga, **znamo** da funkcije p_n zadovoljavaju neku, relativno jednostavnu rekurziju po n. Funkcije p_n ne moraju biti polinomi. Dovoljno je da ih možemo rekurzivno računati!

Pristup računanju vrijednosti $f_N(x)$ je isti kao i ranije. Ako unaprijed ne znamo N, onda se sumacija vrši unaprijed, a $p_n(x)$ računa redom iz rekurzije. S druge strane, iz teorije aproksimacija, vrlo često je moguće unaprijed naći koliko članova N treba uzeti za (uniformnu) zadanu točnost. Tada bi bilo zgodno koristiti neku generalizaciju Hornerove sheme za brzo izvrednjavanje f_N oblika (8.2.4) i to je cilj ovog odjeljka.

8.2.1. Izvrednjavanje rekurzivno zadanih funkcija

Budući da ortogonalni polinomi zadovoljavaju tročlane, homogene rekurzije, a vrlo se često koriste, posebnu pažnju posvetit ćemo baš takvim rekurzijama. Osim toga, tročlane rekurzije istog općeg oblika vrijede i za mnoge specijalne funkcije koje ne moraju biti ortogonalne. Zato pretpostavljamo da funkcije p_n, za $n \in \mathbb{N}_0$, zadovoljavaju rekurziju oblika

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots, \tag{8.2.5}$$

s tim da su poznate "početne" funkcije p_0 i p_1, i sve funkcije α_n, β_n, za $n \in \mathbb{N}$, koje su obično jednostavnog oblika.

Primijetite da potencije $p_n(x) = x^n$ zadovoljavaju dvočlanu homogenu rekurziju

$$p_n(x) - xp_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

uz $p_0(x) = 1$, pa je (8.2.5) zaista generalizacija polinomnog slučaja. Sličan algoritam za brzo izvrednjavanje f_N može se napraviti i kad p_n zadovoljavaju četveročlane ili višečlane rekurzije, ali se takve rekurzije rijetko pojavljuju u praksi.

Algoritam je vrlo sličan izvrednjavanju realnog polinoma u kompleksnoj točki. Definiramo rekurziju za koeficijente

$$\begin{aligned} B_{N+2} &= B_{N+1} = 0, \\ B_n &= a_n - \alpha_n(x)B_{n+1} - \beta_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0. \end{aligned} \tag{8.2.6}$$

Uvrštavanjem u formulu (8.2.4) za $f_N(x)$, dobivamo

$$\begin{aligned} f_N(x) &= \sum_{n=0}^{N} a_n p_n(x) = \sum_{n=0}^{N} \left(B_n + \alpha_n(x)B_{n+1} + \beta_{n+1}(x)B_{n+2}\right) p_n(x) \\ &= \sum_{n=-1}^{N-1} B_{n+1}p_{n+1}(x) + \sum_{n=0}^{N} \alpha_n(x)B_{n+1}p_n(x) + \sum_{n=1}^{N+1} \beta_n(x)B_{n+1}p_{n-1}(x) \\ &= \sum_{n=1}^{N-1} B_{n+1}(p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x)) \\ &\qquad + B_0p_0(x) + B_1p_1(x) + \alpha_0(x)B_1p_0(x) \\ &= B_0p_0(x) + B_1p_1(x) + \alpha_0(x)B_1p_0(x). \end{aligned}$$

Pripadni silazni algoritam izvrednjavanja ima sljedeći oblik.

Algoritam 8.2.1. (Generalizirana Hornerova shema za $f_N(x)$)

$B_1 := 0;$
$B_0 := a_N;$
for $k := N - 1$ **downto** 0 **do**
  begin;
  $B_2 := B_1;$
  $B_1 := B_0;$
  $B_0 := a_k - \alpha_k(x) * B_1 - \beta_{k+1}(x) * B_2;$
  end;
$f_N(x) := B_0 * p_0(x) + B_1 * (p_1(x) + \alpha_0(x) * p_0(x));$

Ako trebamo izračunati i derivaciju $f'_N(x)$, do pripadnog algoritma možemo doći deriviranjem relacije (8.2.4)

$$f'_N(x) = \sum_{n=0}^{N} a_n p'_n(x),$$

i deriviranjem rekurzije (8.2.5), tako da dobijemo i rekurziju za funkcije p'_n. Pokušajte to napraviti sami.

Međutim, postoji i jednostavniji put, deriviranjem rekurzije (8.2.6), slično kao u algoritmu Bairstowa. Ovdje je to još bitno jednostavnije, jer imamo samo jednu varijablu. Koeficijente B_n shvatimo kao funkcije od x, što oni zaista i jesu. Zatim deriviramo (8.2.6), s tim da B_n' označava derivaciju od B_n po x, u točki x. Takvim "formalnim" deriviranjem dobivamo rekurziju za koeficijente B_n'.

$$\begin{aligned}
&B_{N+2} = B_{N+1} = 0,\\
&B'_{N+2} = B'_{N+1} = 0,\\
&\quad B_n = a_n - \alpha_n(x)B_{n+1} - \beta_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0.\\
&\quad B'_n = -\alpha'_n(x)B_{n+1} - \alpha_n(x)B'_{n+1}\\
&\qquad\qquad - \beta'_{n+1}(x)B_{n+2} - \beta_{n+1}(x)B'_{n+2}, \quad n = N, \ldots, 0.
\end{aligned}$$

Odavde odmah vidimo da je i $B'_N = 0$. Uz standardnu oznaku

$$b_n = -\alpha'_n(x)B_{n+1} - \beta'_{n+1}(x)B_{n+2}, \quad n = N, \ldots, 0,$$

s tim da je očito $b_N = 0$, rekurziju za B_n' možemo napisati u obliku

$$B'_n = b_n - \alpha_n(x)B'_{n+1} - \beta_{n+1}(x)B'_{n+2}, \quad n = N, \ldots, 0,$$

što ima skoro isti oblik kao i rekurzija za B_n, osim zamjene a_n u b_n. Konačni rezultat, također, dobivamo deriviranjem ranijeg konačnog rezultata

$$f_N(x) = B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)),$$

odakle slijedi

$$\begin{aligned}
f'_N(x) = {} & B_0 p'_0(x) + B'_0 p_0(x) + B_1(p'_1(x) + \alpha'_0(x)p_0(x) + \alpha_0(x)p'_0(x)),\\
& + B'_1(p_1(x) + \alpha_0(x)p_0(x)).
\end{aligned}$$

Dakle, da bismo izračunali $f'_N(x)$, dovoljno je znati samo derivacije "početnih" funkcija p'_0 i p'_1, koje su obično jednostavne. Naravno, treba znati i derivacije α'_n, β'_n funkcija iz polazne tročlane rekurzije, ali i one su obično jednostavne. Rekurzija za derivacije p'_n nas uopće ne zanima, iako ju nije teško napisati.

Vidimo da nam za računanje $f'_N(x)$ treba i rekurzija za računanje $f_N(x)$, pa se te dvije vrijednosti obično zajedno računaju, a ne svaka posebno. Tada rekurzije za B_n i B'_n provodimo u istoj petlji. Konačni rezultati izgledaju komplicirano, ali kad u njih uvrstimo konkretne objekte, vrlo rijetko ostanu svi članovi. Obično se te formule svedu na

$$f_N(x) = B_0, \quad f'_N(x) = B'_0.$$

Algoritam 8.2.2. (Generalizirana Hornerova shema za $f_N(x)$ i $f'_N(x)$)

$B_1 := 0;$

$B_0 := a_N;$
$B'_1 := 0;$
$B'_0 := 0;$
for $k := N - 1$ **downto** 0 **do**
  begin;
  $B_2 := B_1;$
  $B_1 := B_0;$
  $B_0 := a_k - \alpha_k(x) * B_1 - \beta_{k+1}(x) * B_2;$
  $B'_2 := B'_1;$
  $B'_1 := B'_0;$
  $b := -\alpha'_k(x) * B_1 - \beta'_{k+1}(x) * B_2;$
  $B'_0 := b - \alpha_k(x) * B'_1 - \beta_{k+1}(x) * B'_2;$
  end;
$f_N(x) := B_0 * p_0(x) + B_1 * (\alpha_0(x) * p_0(x) + p_1(x));$
$f'_N(x) := B_0 * p'_0(x) + B'_0 * p_0(x) + B_1 * (p'_1(x) + \alpha'_0(x) * p_0(x) + \alpha_0(x) * p'_0(x))$
$\quad + B'_1 * (p_1(x) + \alpha_0(x) * p_0(x));$

Istim putem možemo izvesti i rekurzije za računanje viših derivacija $f_N^{(k)}(x)$, za $k \geq 2$. Zanimljivo je da u praksi to gotovo nikada nije potrebno. Razlog leži u činjenici da gotovo sve "korisne" familije funkcija p_n, $n \in \mathbb{N}$, zadovoljavaju neke diferencijalne jednadžbe **drugog** reda, s parametrom n. Jasno je da tada treba koristiti odgovarajuću diferencijalnu jednadžbu za računanje $f''_N(x)$, ali i to je vrlo rijetko potrebno.

Čak i algoritam za derivacije se rijetko koristi. Naime, ako znamo naći, tj. izračunati koeficijente a_n u prikazu

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x),$$

s dovoljnom točnošću, za $n \leq N$, tako da je pripadni $f_N(x)$ dovoljno dobra aproksimacija, onda se **ne isplati** koristiti

$$f'_N(x) = \sum_{n=0}^{N} a_n p'_n(x)$$

kao aproksimaciju za $f'(x)$, jer ona obično ima manju točnost od aproksimacije za f. Puno je bolje izračunati koeficijente a'_n (to nisu derivacije) u pravom razvoju derivacije f' po **istim** funkcijama p_n, a ne po njihovim derivacijama. Dakle, za f' koristimo aproksimaciju oblika

$$f'_{N'}(x) = \sum_{n=0}^{N'} a'_n p_n(x),$$

koja ne mora imati istu duljinu, ali zato ima željenu točnost.

Složenost ovih algoritama ključno ovisi o složenosti računanja svih potrebnih funkcija — p_0, p_1, α_n, β_n, i njihovih derivacija, pa je besmisleno brojati pojedinačne aritmetičke operacije na nivou općeg algoritma.

U praktičnim aproksimacijama se najčešće koriste tzv. ortogonalne familije funkcija p_n, koje čine ortogonalnu bazu u nekom prostoru funkcija, obzirom na neki skalarni produkt na tom prostoru. Vrlo često je p_n polinom stupnja n, za svaki $n \in \mathbb{N}_0$. Neke primjere klasičnih ortogonalnih polinoma i pripadnih rekurzija dajemo nešto kasnije.

Međutim, već smo rekli da funkcije p_n ne moraju biti polinomi i prvi primjer je baš tog tipa.

8.2.2. Izvrednjavanje Fourierovih redova

Za aproksimaciju periodičkih funkcija standardno koristimo Fourierove redove. Pretpostavimo, radi jednostavnosti, da je f periodička funkcija na segmentu $[-\pi, \pi]$. Tada, uz relativno blage pretpostavke, funkciju f možemo razviti u Fourierov red oblika

$$\sum_{n=0}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx).$$

Umjesto a_0, standardno se piše $a_0/2$, ali to nije bitna razlika. Zanemarimo trenutno pitanje konvergencije ovog reda i značenja njegove sume. Uočimo samo da ove trigonometrijske funkcije tvore ortogonalan sustav funkcija, obzirom na skalarni produkt definiran integralom.

Pretpostavimo da su nam koeficijenti a_n i b_n poznati. Naš zadatak je izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=0}^{N} a_n \cos(nx) + \sum_{n=1}^{N} b_n \sin(nx),$$

gdje je N unaprijed zadan. Ovakav izraz se često zove i trigonometrijski polinom. Vidimo da se on sastoji iz dva dijela, kosinusnog i sinusnog, pa ćemo tako i sastaviti algoritam. Usput, sjetimo se da Fourierov red parne funkcije $f(x) = f(-x)$ ima samo kosinusni dio, a Fourierov red neparne funkcije $f(x) = -f(-x)$ ima samo sinusni dio razvoja.

Pretpostavimo stoga da je f parna funkcija i trebamo izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=0}^{N} a_n \cos(nx).$$

U direktnoj sumaciji trebamo N računanja funkcije cos, za $\cos(nx)$, uz $n \geq 1$. Iako to danas više ne traje pretjerano dugo, možemo naći i bolji algoritam, koji treba samo jedno jedino računanje funkcije cos.

Da bismo dobili polazni oblik aproksimacije (8.2.4) iz generalizirane Hornerove sheme, očito treba definirati

$$p_n(x) = \cos(nx).$$

Nedostaje nam još samo tročlana homogena rekurzija za ove funkcije. Međutim, i to ide lako, ako se sjetimo formule koja sumu kosinusa pretvara u produkt

$$\cos a + \cos b = 2\cos\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right).$$

Dovoljno je uzeti $a = (n+1)x$ i $b = (n-1)x$. Dobivamo

$$\cos((n+1)x) + \cos((n-1)x) = 2\cos(nx)\cos x,$$

pa tražena rekurzija ima oblik

$$p_{n+1}(x) - 2\cos x\, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

odakle slijedi da u općoj rekurziji (8.2.5) treba uzeti

$$\alpha_n(x) = -2\cos x, \quad \beta_n(x) = 1, \quad n \in \mathbb{N}.$$

Vidimo da $\alpha_n(x)$ i $\beta_n(x)$ ne ovise o n, a $\beta_n(x)$ ne ovisi ni o x, već je konstanta.

Rekurzija (8.2.6) za B_n ima oblik

$$\begin{aligned} B_{N+2} &= B_{N+1} = 0, \\ B_n &= a_n + 2\cos x\, B_{n+1} - B_{n+2}, \quad n = N, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = 1$ i $p_1(x) = \cos x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)) \\ &= B_0 \cdot 1 + B_1(\cos x - 2\cos x \cdot 1) \\ &= B_0 - B_1 \cos x. \end{aligned}$$

Sad imamo sve elemente za generaliziranu Hornerovu shemu.

Algoritam 8.2.3. (Fourierov "red" parne funkcije)

$B_1 := 0;$
$B_0 := a_N;$
$alpha := 2 * \cos x;$
for $k := N - 1$ **downto** 0 **do**
  begin;
  $B_2 := B_1;$
  $B_1 := B_0;$
  $B_0 := a_k + alpha * B_1 - B_2;$
  end;
$f_N(x) := B_0 - 0.5 * alpha * B_1;$

Ovaj algoritam zaista "troši" jedan jedini kosinus, pod cijenu jednog množenja s 0.5. Što se stabilnosti tiče, on je podjednako stabilan kao i direktna sumacija. Male vrijednosti $\cos(nx)$ ionako ne dobivamo s malom relativnom, već malom apsolutnom greškom.

Ako trebamo izračunati i derivaciju $f'_N(x)$, za pripadni algoritam trebamo

$$\alpha'_n(x) = 2\sin x, \quad \beta'_n(x) = 0, \quad n \in \mathbb{N}.$$

Onda je

$$b_n = -2\sin x\, B_{n+1}, \quad n = N, \ldots, 0,$$

i

$$B'_n = b_n + 2\cos x\, B'_{n+1} - B'_{n+2}, \quad n = N, \ldots, 0,$$

a formalnim deriviranjem $f_N(x) = B_0 - B_1 \cos x$ dobivamo

$$f'_N(x) = B'_0 - B'_1 \cos x + B_1 \sin x.$$

Dakle, cijeli taj algoritam treba još samo jedan sinus. I tog bismo mogli izbaciti, tako da sinus izrazimo preko kosinusa,

$$\sin x = \pm\sqrt{1 - \cos^2 x},$$

ali to se već ne isplati, jer moramo paziti na znak, a oduzimanje može dovesti do nepotrebnog gubitka točnosti u sinusu.

Pretpostavimo sad da je f neparna funkcija. Trebamo izračunati aproksimaciju oblika

$$f_N(x) = \sum_{n=1}^{N} b_n \sin(nx).$$

Suma ovdje ide od 1, pa treba biti malo oprezan. Zgodniji je zapis

$$f_N(x) = \sum_{n=0}^{N-1} b_{n+1} \sin((n+1)x).$$

Nije baš lijepo ostaviti indeks N u f_N, ali sad je očito da treba definirati

$$p_n(x) = \sin((n+1)x).$$

Zatim koristimo formulu

$$\sin a + \sin b = 2\sin\left(\frac{a+b}{2}\right)\cos\left(\frac{a-b}{2}\right),$$

i uzmemo $a = (n+2)x$ i $b = nx$. Dobivamo

$$\sin((n+2)x) + \sin(nx) = 2\sin((n+1)x)\cos x,$$

pa tražena rekurzija ima oblik

$$p_{n+1}(x) - 2\cos x\, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

što je potpuno isti oblik kao i za parne funkcije, odnosno za $p_n(x) = \cos(nx)$. Dakle, rekurzija za pripadne B_n ima isti oblik, samo starta od $N-1$

$$\begin{aligned} B_{N+1} &= B_N = 0, \\ B_n &= b_{n+1} + 2\cos x\, B_{n+1} - B_{n+2}, \quad n = N-1, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = \sin x$ i $p_1(x) = \sin(2x) = 2\sin x \cos x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)) \\ &= B_0 \cdot \sin x + B_1(2\sin x \cos x - 2\cos x \cdot \sin x) \\ &= B_0 \sin x. \end{aligned}$$

Algoritam možete i sami napisati.

Za opći Fourierov red koji ima i parni i neparni dio, treba spojiti prethodne algoritme. Jedina je neugoda što je neparni za 1 kraći, jer starta s $N-1$. Ako nas to baš jako smeta, onda možemo i malo drugačije postupiti u neparnom dijelu. Umjetno definiramo da je $b_0 = 0$ i pišemo

$$f_N(x) = \sum_{n=0}^{N} b_n \sin(nx).$$

Zatim uzmemo

$$p_n(x) = \sin(nx).$$

Rekurzija za p_n, naravno, ostaje ista, a za B_n sad vrijedi "produljena" rekurzija

$$\begin{aligned} B_{N+1} &= B_N = 0, \\ B_n &= b_n + 2\cos x\, B_{n+1} - B_{n+2}, \quad n = N, \ldots, 0. \end{aligned}$$

Početne funkcije su $p_0(x) = 0$ i $p_1(x) = \sin x$, pa je konačni rezultat

$$\begin{aligned} f_N(x) &= B_0 p_0(x) + B_1(p_1(x) + \alpha_0(x)p_0(x)) \\ &= B_0 \cdot 0 + B_1(\sin x - 2\cos x \cdot 0) \\ &= B_1 \sin x. \end{aligned}$$

To pokazuje da B_0 uopće ne treba računati, ali baš to i očekujemo, kad smo rekurziju pomakli za jedan indeks naviše!

Spomenimo na kraju da obje funkcije $\cos(nx)$ i $\sin(nx)$ zadovoljavaju istu diferencijalnu jednadžbu drugog reda

$$y'' + n^2 y = 0.$$

8.2.3. Klasični ortogonalni polinomi

U aproksimacijama i rješavanju diferencijalnih jednadžbi najčešće se susrećemo s pet tipova klasičnih ortogonalnih polinoma. Za polinome

$$\{p_0, p_1, p_2, \ldots, p_n, \ldots\},$$

pri čemu indeks polinoma označava njegov stupanj, reći ćemo da su ortogonalni obzirom na težinsku funkciju w, $w(x) \geq 0$ na intervalu $[a, b]$, ako vrijedi

$$\int_a^b w(x)\, p_m(x)\, p_n(x)\, dx = 0, \quad \text{za } m \neq n.$$

Težinska funkcija određuje sistem polinoma do na konstantni faktor u svakom od polinoma. Izbor takvog faktora zove se još i standardizacija ili normalizacija.

Čebiševljevi polinomi prve vrste

Čebiševljevi polinomi prve vrste obično se označavaju s T_n. Oni su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = \frac{1}{\sqrt{1 - x^2}}.$$

Vrijedi

$$\int_{-1}^{1} \frac{T_m(x)\, T_n(x)}{\sqrt{1 - x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$T_{n+1}(x) - 2xT_n(x) + T_{n-1}(x) = 0,$$

uz start

$$T_0(x) = 1, \quad T_1(x) = x.$$

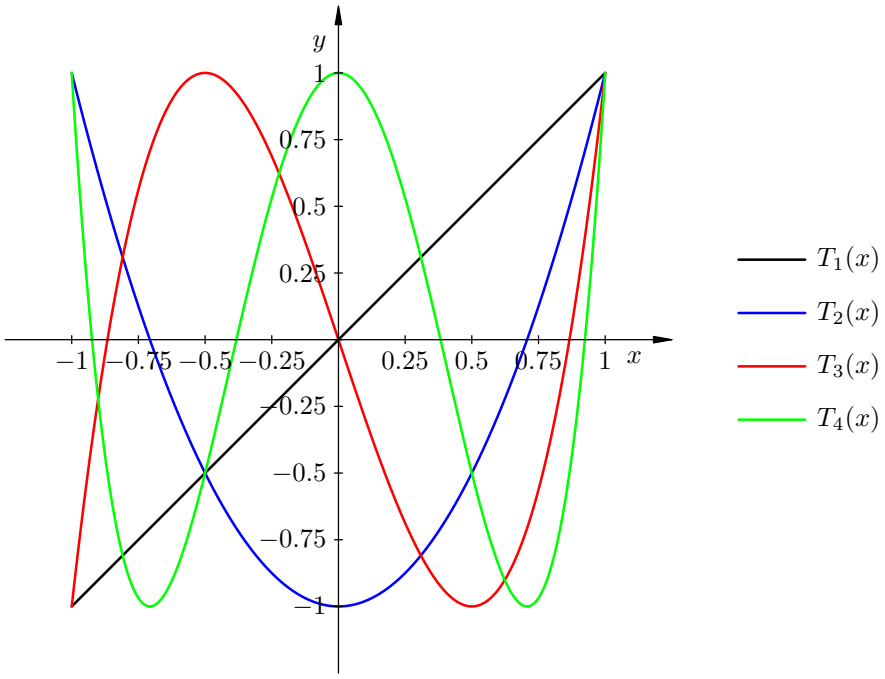
Za njih postoji i eksplicitna formula

$$T_n(x) = \cos(n \arccos x).$$

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.



Katkad se koriste i Čebiševljevi polinomi prve vrste transformirani na interval $[0,1]$, u oznaci T_n^*. Korištenjem linearne (preciznije, afine) transformacije

$$[0,1] \ni x \mapsto \xi := 2x - 1 \in [-1,1]$$

dolazimo do svih svojstava tih polinoma. Na primjer, relacija ortogonalnosti tada postaje

$$\int_0^1 \frac{T_m^*(x)\, T_n^*(x)}{\sqrt{x - x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0, \end{cases}$$

a rekurzivna relacija

$$T_{n+1}^*(x) - 2(2x-1)T_n^*(x) + T_{n-1}^*(x) = 0,$$

uz start

$$T_0^*(x) = 1, \quad T_1^*(x) = 2x - 1.$$

Čebiševljevi polinomi druge vrste

Čebiševljevi polinomi druge vrste obično se označavaju s U_n. Oni su ortogonalni na intervalu $[-1,1]$ obzirom na težinsku funkciju

$$w(x) = \sqrt{1 - x^2}.$$

Vrijedi

$$\int_{-1}^{1} \sqrt{1-x^2}\, U_m(x)\, U_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi/2, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju istu rekurzivnu relaciju kao Čebiševljevi polinomi prve vrste

$$U_{n+1}(x) - 2xU_n(x) + U_{n-1}(x) = 0,$$

samo uz malo drugačiji start

$$U_0(x) = 1, \quad U_1(x) = 2x.$$

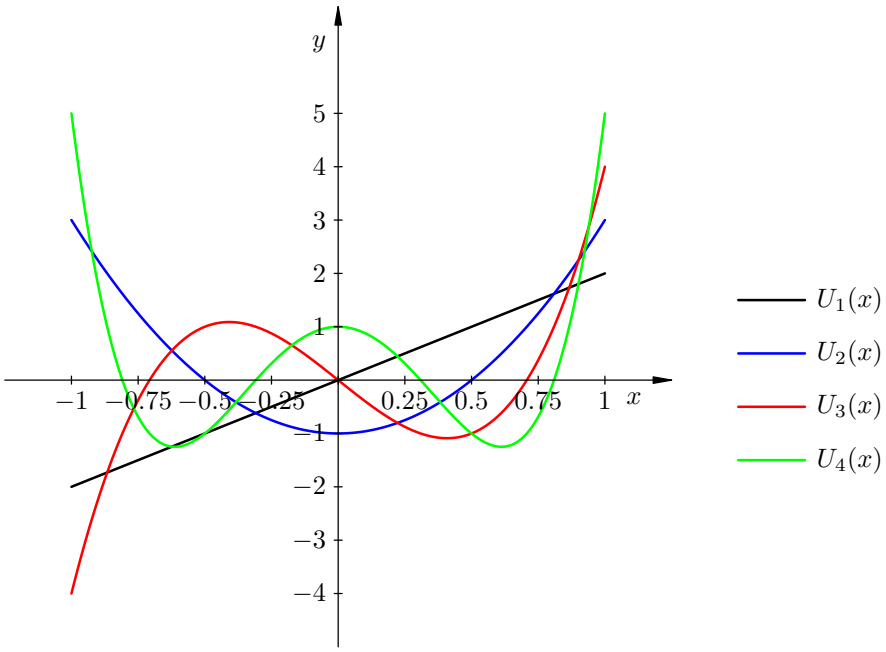
Za njih postoji i eksplicitna formula

$$U_n(x) = \frac{\sin((n+1)\arccos x)}{\sin(\arccos x)}.$$

Osim toga, n-ti Čebiševljev polinom druge vrste U_n zadovoljava diferencijalnu jednadžbu

$$(1-x^2)y'' - 3xy' + n(n+2)y = 0.$$

Graf prvih par polinoma izgleda ovako.



Legendreovi polinomi

Legendreovi polinomi obično se označavaju s P_n. Oni su ortogonalni na intervalu $[-1, 1]$ obzirom na težinsku funkciju

$$w(x) = 1.$$

Vrijedi

$$\int_{-1}^{1} P_m(x)\,P_n(x)\,dx = \begin{cases} 0, & \text{za } m \neq n, \\ 2/(2n+1), & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$(n+1)P_{n+1}(x) - (2n+1)xP_n(x) + nP_{n-1}(x) = 0,$$

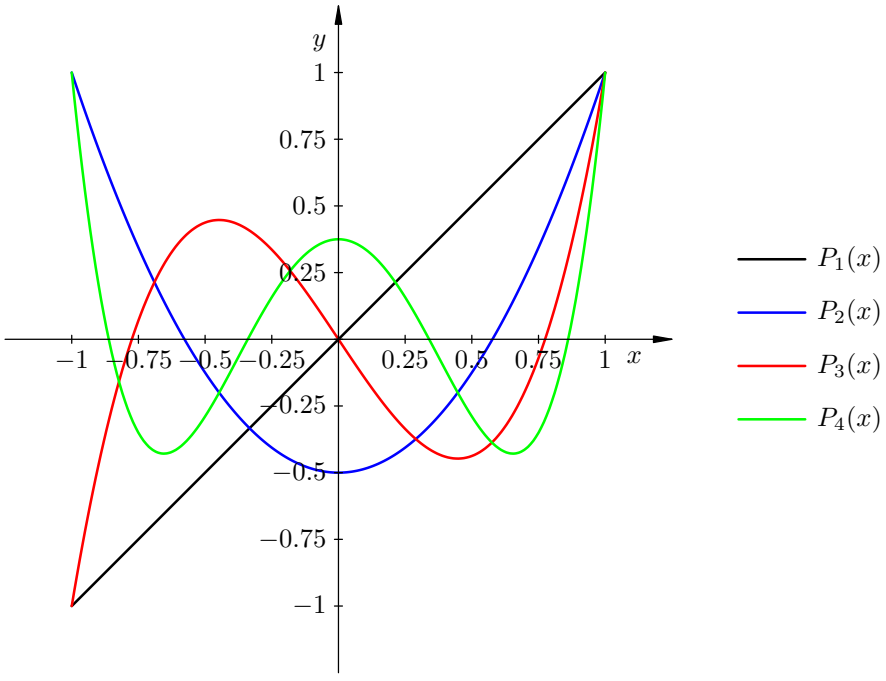
uz start

$$P_0(x) = 1, \quad P_1(x) = x.$$

Osim toga, n-ti Legendreov polinom P_n zadovoljava diferencijalnu jednadžbu

$$(1-x^2)y'' - 2xy' + n(n+1)y = 0.$$

Graf prvih par polinoma izgleda ovako.



Laguerreovi polinomi

Laguerreovi polinomi obično se označavaju s L_n. Oni su ortogonalni na intervalu $[0, \infty)$ obzirom na težinsku funkciju

$$w(x) = e^{-x}.$$

Vrijedi

$$\int_0^\infty e^{-x}\, L_m(x)\, L_n(x)\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ 1, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$(n+1)L_{n+1}(x) + (x - 2n - 1)L_n(x) + nL_{n-1}(x) = 0,$$

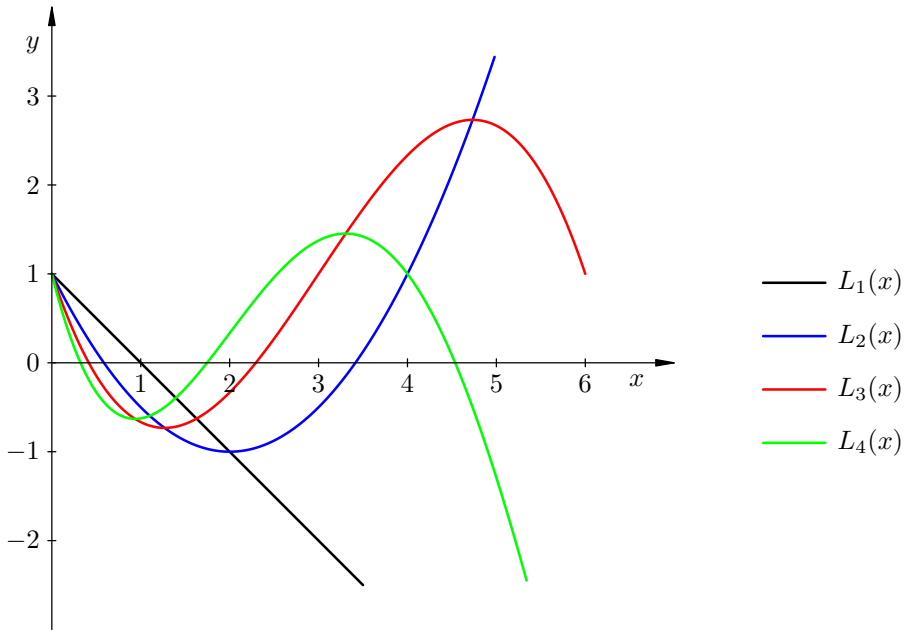
uz start

$$L_0(x) = 1, \quad L_1(x) = 1 - x.$$

Osim toga, n-ti Laguerreov polinom L_n zadovoljava diferencijalnu jednadžbu

$$xy'' + (1 - x)y' + ny = 0.$$

Graf prvih par polinoma izgleda ovako.



U literaturi se često nailazi na još jednu rekurziju za Laguerreove polinome

$$\widetilde{L}_{n+1}(x) + (x - 2n - 1)\widetilde{L}_n(x) + n^2\widetilde{L}_{n-1}(x) = 0,$$

uz jednaki start

$$\widetilde{L}_0(x) = 1, \quad \widetilde{L}_1(x) = 1 - x.$$

Uspoređivanjem ove i prethodne rekurzije dobivamo da je

$$\widetilde{L}_n(x) = n!\, L_n(x),$$

tj. radi se samo o drugačijoj normalizaciji ortogonalnih polinoma. Lako je pokazati da vrijedi

$$\int_0^\infty e^{-x}\,\widetilde{L}_m(x)\,\widetilde{L}_n(x)\,dx = \begin{cases} 0, & \text{za } m \neq n, \\ (n!)^2, & \text{za } m = n. \end{cases}$$

Hermiteovi polinomi

Hermiteovi polinomi obično se označavaju s H_n. Oni su ortogonalni na intervalu $(-\infty, \infty)$ obzirom na težinsku funkciju

$$w(x) = e^{-x^2}.$$

Vrijedi

$$\int_{-\infty}^\infty e^{-x^2}\,H_m(x)\,H_n(x)\,dx = \begin{cases} 0, & \text{za } m \neq n, \\ 2^n\,n!\sqrt{\pi}, & \text{za } m = n. \end{cases}$$

Oni zadovoljavaju rekurzivnu relaciju

$$H_{n+1}(x) - 2xH_n(x) + 2nH_{n-1}(x) = 0,$$

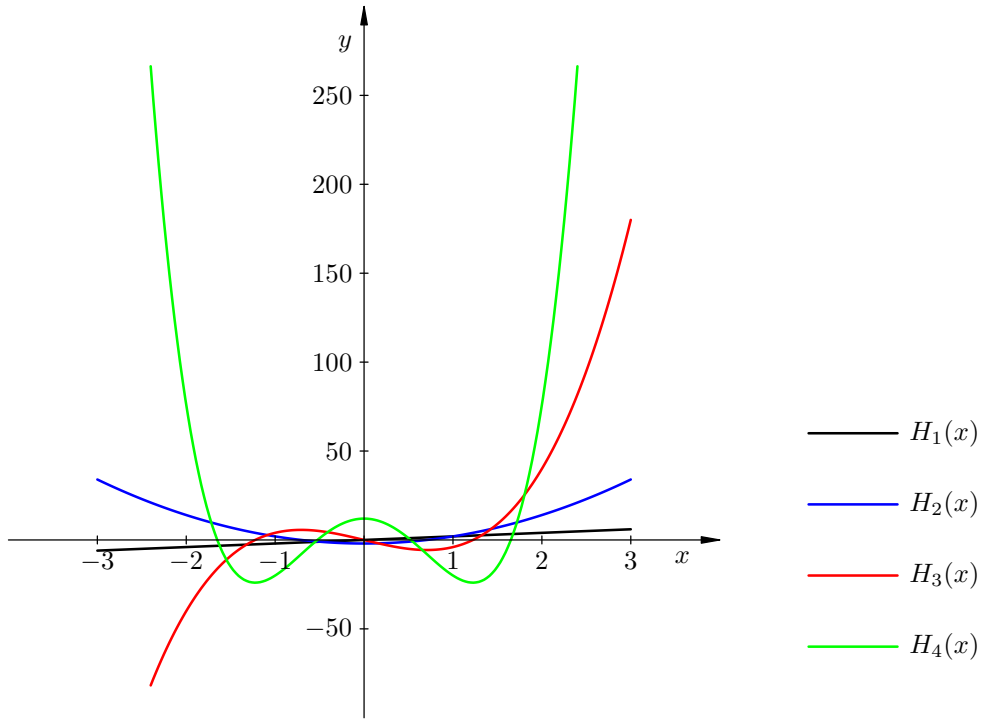
uz start

$$H_0(x) = 1, \quad H_1(x) = 2x.$$

Osim toga, n-ti Hermiteov polinom H_n zadovoljava diferencijalnu jednadžbu

$$y'' - 2xy' + 2ny = 0.$$

Graf prvih par polinoma izgleda ovako.



8.3. Stabilnost rekurzija i generalizirane Hornerove sheme

Stabilnost generalizirane Hornerove sheme u velikoj mjeri ovisi o tome koliko je stabilna rekurzija za funkcije p_n. Naime, ako u razvoju

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x),$$

uzmemo da je $a_N = 1$, i $a_n = 0$, za $n < N$, onda je $f_N = p_N$. Dakle, generaliziranu Hornerovu shemu možemo koristiti i kao silazni algoritam za izvrednjavanje funkcija p_N. Pitanje je da li je to bolje od direktnog računanja p_N unaprijed po osnovnoj rekurziji

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots, N-1.$$

Za precizan odgovor, treba analizirati stabilnost ove rekurzije za p_n.

Prije toga, pogledajmo koliki je utjecaj te stabilnosti na generaliziranu Hornerovu shemu. Stvar, naravno, bitno ovisi i o koeficijentima a_n u prikazu f_N. Za "lijepe" funkcije, ti koeficijenti obično relativno brzu "trnu", tj. teže prema nuli.

Takvi "mali" koeficijenti a_n, za veće n, bitno prigušuju greške u računanju $p_n(x)$. Isti efekt, samo manje vidljivo, vrijedi i u silaznom algoritmu.

U tom smislu, prethodni primjer je "ekstremni", jer je zadnji koeficijent jednak 1, a svi prethodni su 0. Isto kao što su polinomi analitičke funkcije, ali konačnost razvoja uništava potrebu za smanjivanjem koeficijenata kad $n \to \infty$. Dakle, na "polinomnom" slučaju će se eventualna nestabilnost najjače vidjeti. Iako to nije sasvim precizan argument, očito je ključno analizirati baš "polinomni" slučaj. Zbog toga, a i radi jednostavnosti, u nastavku gledamo samo "polinomni" slučaj $f_N = p_N$.

Generalno možemo odmah zaključiti da opasnost nastupa kad niz vrijednosti

$$p_0(x), p_1(x), \ldots, p_N(x)$$

naglo pada po apsolutnoj vrijednosti. Tada očekujemo kraćenja u osnovnoj rekurziji, što rezultira i gubitkom točnosti. Dva su pitanja na koja bi bilo zgodno naći odgovor.

- Kako se tada ponaša silazni algoritam?
- Može li se nekim "trikom", poput okretanja rekurzije, popraviti stabilnost?

Umjesto općeg odgovora, koji bi nas odveo predaleko, ilustrirajmo situaciju na jednom klasičnom primjeru.

Neka je $p_n(x) = e^{nx}$. Ove funkcije generiraju tzv. eksponencijalne polinome

$$f_N(x) = \sum_{n=0}^{N} a_n e^{nx}.$$

Očito je da možemo sastaviti razne rekurzije za p_n. Dvočlana ima oblik

$$p_{n+1}(x) - e^x p_n(x) = 0, \quad n \in \mathbb{N}_0.$$

Nije teško napraviti i tročlanu homogenu rekurziju po ugledu na trigonometrijske funkcije.

$$p_{n+1}(x) - 2 \operatorname{ch} x \, p_n(x) + p_{n-1}(x) = 0, \quad n \in \mathbb{N},$$

gdje je $\operatorname{ch} x = (e^x + e^{-x})/2$.

Očito je da $p_n(x)$ monotono raste za $x > 0$ i monotono pada za $x < 0$. Testirajmo stabilnost ove rekurzije i pripadne generalizirane Hornerove sheme za računanje $p_n(x) = e^{nx}$ u točkama $x = 1$ i $x = -1$.

8.4. Besselove funkcije i Millerov algoritam

Općenito nije potrebno znati funkcije p_0 i p_1 da bi se mogla koristiti silazna varijanta za računanje p_N. Dovoljno je znati neku vezu među funkcijama p_n koja se

lako računa, a takve su često poznate. Na primjer, to su funkcije izvodnice oblika

$$F(x) = \sum_{n=0}^{\infty} q_n \, p_n(x),$$

gdje se $F(x)$ računa nekom analitičkom formulom bez upotrebe $p_n(x)$, tj. $F(x)$ možemo naći neovisno o funkcijama p_n.

Millerov algoritam (po J. C. P. Milleru, 1954. godine) primjenjuje se kada vrijednosti funkcija p_n vrlo brzo padaju kad n raste (za sve x ili u nekom području vrijednosti za argumente), a greška zaostaje.

Pretpostavimo da funkcije p_n zadovoljavaju neku homogenu rekurziju, na primjer tročlanu, koja je najčešća u praksi

$$p_{n+1}(x) + \alpha_n(x)p_n(x) + \beta_n(x)p_{n-1}(x) = 0, \quad n = 1, 2, \ldots$$

Poznavanje bilo kojeg p_n (čak ni p_0 niti p_1) nije potrebno. Treba znati samo koeficijente α_n i β_n.

8.4.1. Opća forma Millerovog algoritma

Pokažimo kako funkcionira Millerov algoritam.

Odaberimo startnu vrijednost indeksa M od koje ćemo početi – ovisno o vrijednosti N indeksa funkcije koju tražimo.

Ako tražimo $p_N(x)$ (ili $p_N(x), \ldots, p_0(x)$, ili samo neke od njih), M se obično odabere tako da je $M > N$ i vrijedi

$$\frac{p_M(x)}{p_N(x)} \approx \text{točnost računanja.}$$

To obično garantira i da je

$$F_M(x) := \sum_{n=0}^{M} q_n \, p_n(x),$$

barem jednako točna aproksimacija za $F(x)$, što ćemo kasnije iskoristiti.

Stavimo $\tilde{p}_{M+1} = 0$, $\tilde{p}_M = 1$ i računamo brojeve $\tilde{p}_n$, za $n = M-1, \ldots, 0$, unatrag po rekurziji za $p_n(x)$:

$$\tilde{p}_n = \frac{-(\alpha_{n+1}(x)\tilde{p}_{n+1} + \tilde{p}_{n+2})}{\beta_{n+1}(x)}, \quad n = M-1, \ldots, 0.$$

Zbog homogenosti rekurzije, dobiveni niz vrijednosti

$$\tilde{p}_M, \ldots, \tilde{p}_0$$

je vrlo približno proporcionalan stvarnim vrijednostima

$$p_M(x), \ldots, p_0(x),$$

barem u području od $p_N(x)$ do $p_0(x)$, tj. vrijedi $p_n(x) \approx \tilde{p}_n \cdot c$, za $n \leq N$. Treba još naći normalizacioni faktor c.

Sada iskoristimo činjenicu da znamo koeficijente q_n u razvoju funkcije izvodnice F po funkcijama p_n

$$F(x) = \sum_{n=0}^{\infty} q_n \, p_n(x).$$

Umjesto nepoznatih vrijednosti $p_n(x)$ uvrstimo $\tilde{p}_n$ i numeričkim zbrajanjem izračunamo aproksimaciju $\tilde{F}_M$

$$\widetilde{F}_M := \sum_{n=0}^{M} q_n \, \tilde{p}_n.$$

Gornji indeks sumacije može biti i bitno manji od M, ako znamo da $p_n(x)$ vrlo brzo padaju kad n raste. U prethodnoj sumi dovoljno je uzeti toliko članova da se izračunata vrijednost $\widetilde{F}_M$ stabilizira na točnost računala ili traženu točnost.

Zatim direktno analitički izračunamo $F(x)$ po poznatoj formuli i stavimo

$$c := \frac{F(x)}{\widetilde{F}_M},$$

što je traženi normalizacioni faktor, uz pretpostavku da je $F_M(x)$ dovoljno dobra aproksimacija za $F(x)$. Na kraju izračunamo

$$p_n(x) = \tilde{p}_n \cdot c$$

za sve one n između 0 i N koji nas zanimaju, jer u tom području vrijedi vrlo dobra proporcionalnost $p_n(x) \sim \tilde{p}_n$.

Vrlo često se startna vrijednost M određuje iz nekih poznatih relacija za familiju funkcija $p_n(x)$ ili eksperimentalno, povećavanjem n sve dok se ne postigne željena točnost za $p_N(x)$. Naravno, ovim se algoritmom može računati i $p_0(x)$.

8.4.2. Izvrednjavanje Besselovih funkcija

Besselove funkcije prvi puta je uveo Bessel, 1824. godine, promatrajući jedan problem iz tzv. dinamičke astronomije, vezan uz zgodan način zapisa položaja planeta koji se kreće po elipsi oko Sunca. Da bi dobio formulu prikladnu za praktično računanje, Bessel je traženu veličinu prikazao kao red funkcija poznatih podataka. U tom redu se, kao koeficijenti, javljaju funkcije oblika

$$J_n(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin\theta - n\theta) \, d\theta, \quad n \in \mathbb{N}, \tag{8.4.1}$$

koje zovemo Besselovim funkcijama prve vrste. Očito se ova definicija može proširiti na $n \in \mathbb{Z}$ i tada je $J_{-n}(x) = (-1)^n J_n(x)$. Nažalost, ni za jednu od ovih funkcija ne postoji neka jednostavna "formula" ili oblik za računanje.

Relaciju (8.4.1) možemo iskoristiti i za numeričko računanje vrijednosti $J_n(x)$, za zadane $n \in \mathbb{N}_0$ i $x \in \mathbb{R}$, tako da upotrijebimo neku od metoda numeričke integracije. Međutim, postoje i mnogo brži algoritmi za postizanje iste tražene točnosti izračunate vrijednosti $J_n(x)$.

U klasičnom pristupu preko funkcija izvodnica, Besselove funkcije možemo definirati kao koeficijente uz t^n u razvoju

$$\exp\left(x\,\frac{t-1/t}{2}\right) = \sum_{n=-\infty}^{\infty} J_n(x)\,t^n. \tag{8.4.2}$$

Nije teško dokazati da je ova definicija ekvivalentna integralnoj reprezentaciji (8.4.1). Iz (8.4.2) mogu se izvesti mnoge važne relacije za Besselove funkcije. Na primjer, Besselove funkcije zadovoljavaju tročlanu rekurziju

$$J_{n+1}(x) - \frac{2n}{x} J_n(x) + J_{n-1}(x) = 0, \quad n \in \mathbb{N}. \tag{8.4.3}$$

Također, vrijedi $J_1(x) = -J_0'(x)$. Dakle, kad bismo znali izračunati $J_0(x)$ i $J_1(x)$ (ili $J_0'(x)$), onda bismo iz (8.4.3) mogli izračunati i $J_n(x)$. Osim toga, generaliziranom Hornerovom shemom mogli bismo onda računati i razne razvoje po Besselovim funkcijama. Nažalost, to je tako samo u teoriji. Rekurzija (8.4.3) je izrazito nestabilna unaprijed. Da bismo to pokazali, promotrimo ponašanje vrijednosti Besselovih funkcija $J_n(x)$ u ovisnosti o n i x.

Iz (8.4.2) može se pokazati da Besselove funkcije J_n zadovoljavaju diferencijalnu jednadžbu

$$x^2y'' + xy' + (x^2 - n^2)y = 0.$$

Ovu jednadžbu možemo promatrati na cijeloj kompleksnoj ravnini i u slučaju kad n nije cijeli broj. Tad se, umjesto n, obično koristi oznaka ν za parametar jednadžbe. Jedno od rješenja ove jednadžbe su Besselove funkcije prve vrste J_ν, koje imaju bitno svojstvo da su ograničene u 0 kad je $\operatorname{Re}\nu \geq 0$. Analitički im je oblik

$$J_\nu(x) = \left(\frac{1}{2}x\right)^\nu \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!\,\Gamma(\nu+k+1)}, \tag{8.4.4}$$

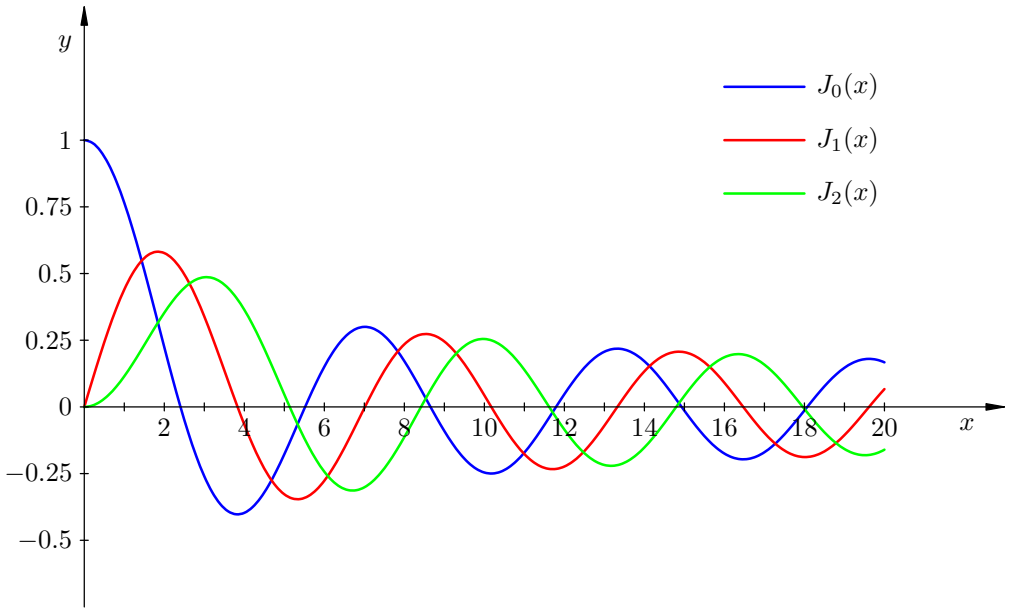
gdje su ν i x, općenito, kompleksni brojevi. U nastavku gledamo ponašanje ovih funkcija samo za nenegativne realne indekse $\nu \geq 0$ i argumente $x \geq 0$. Ako je ν cijeli broj, onda prethodna relacija glasi

$$J_n(x) = \left(\frac{1}{2}x\right)^n \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!\,(n+k)!}.$$

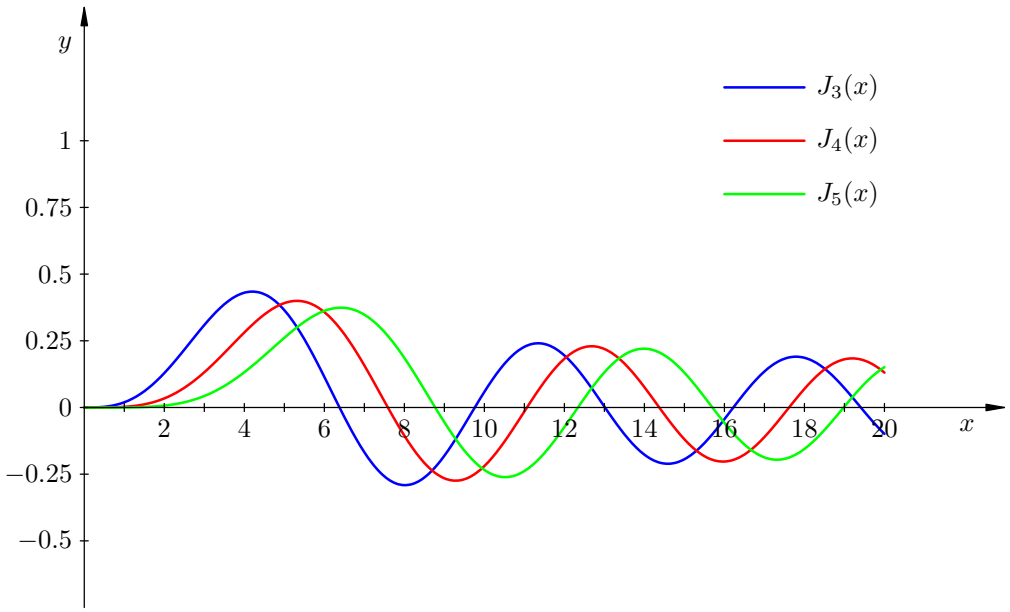
Oba reda očito konvergiraju za $x \geq 0$, a donji čak na cijelom skupu $\mathbb{C}$. Na prvi pogled izgleda kao da smo time riješili i problem računanja vrijednosti $J_0(x)$ i $J_1(x)$.

Zaista, ovaj red vrlo brzo konvergira za relativno male x, pa se može koristiti za računanje. Međutim, za malo veće x, kad je $x \approx n$ (ili ν) i dalje, dobivamo slično ponašanje kao i kod aproksimacije trigonometrijskih funkcija Taylorovim redom, tj. dolazi do sve većeg kraćenja zbrajanjem uzastopnih članova reda (8.4.4).

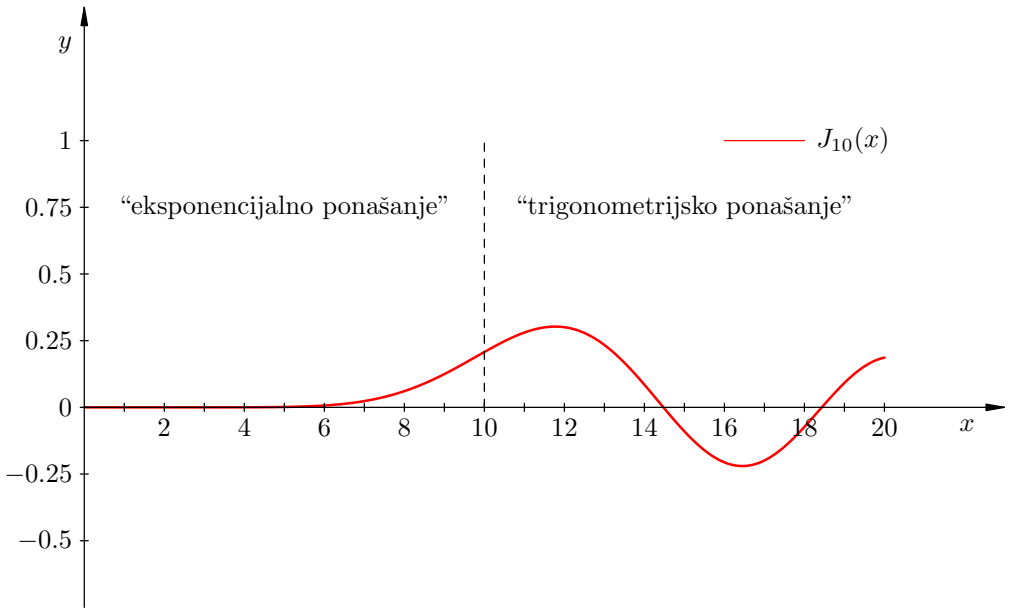
Grafički, prve tri Besselove funkcije izgledaju ovako



sljedeće tri ovako

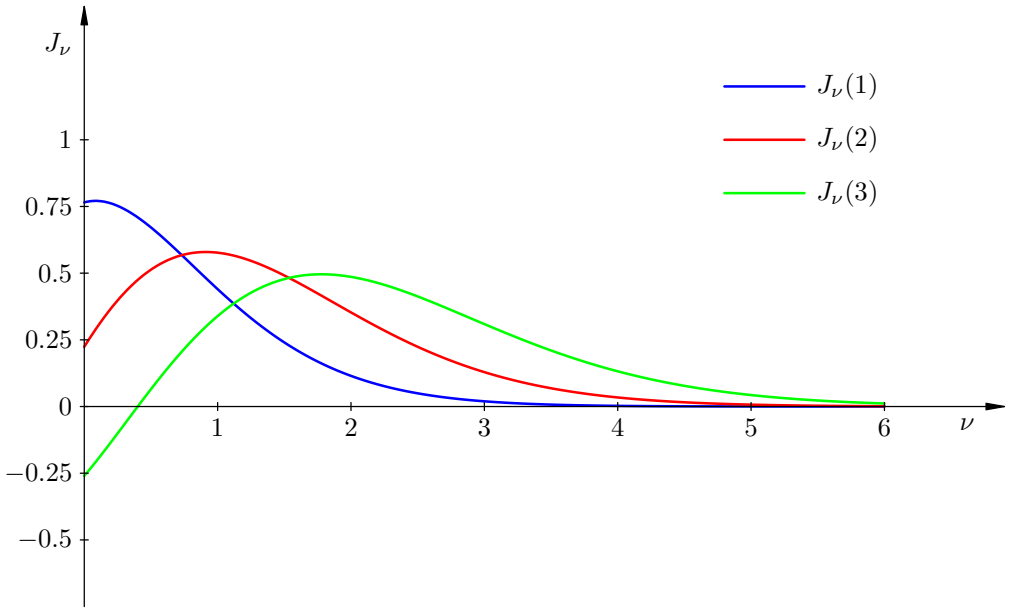


a, recimo, deseta Besselova funkcija ovako

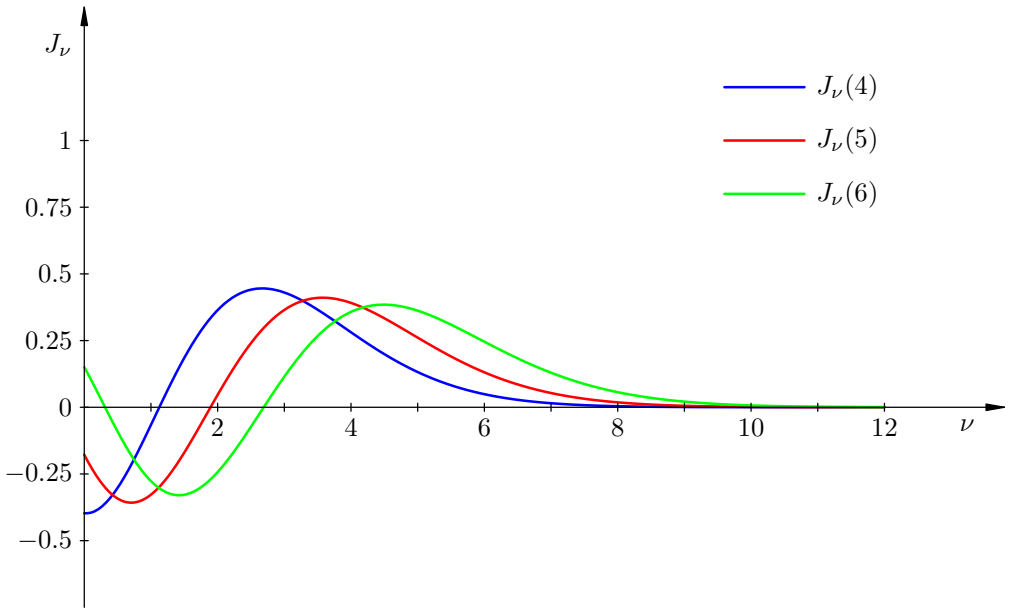


Uočite da se područje eksponencijalnog ponašanja mijenja u trigonometrijsko područje približno za $x = \nu$, kao što smo i očekivali iz Taylorovog reda.

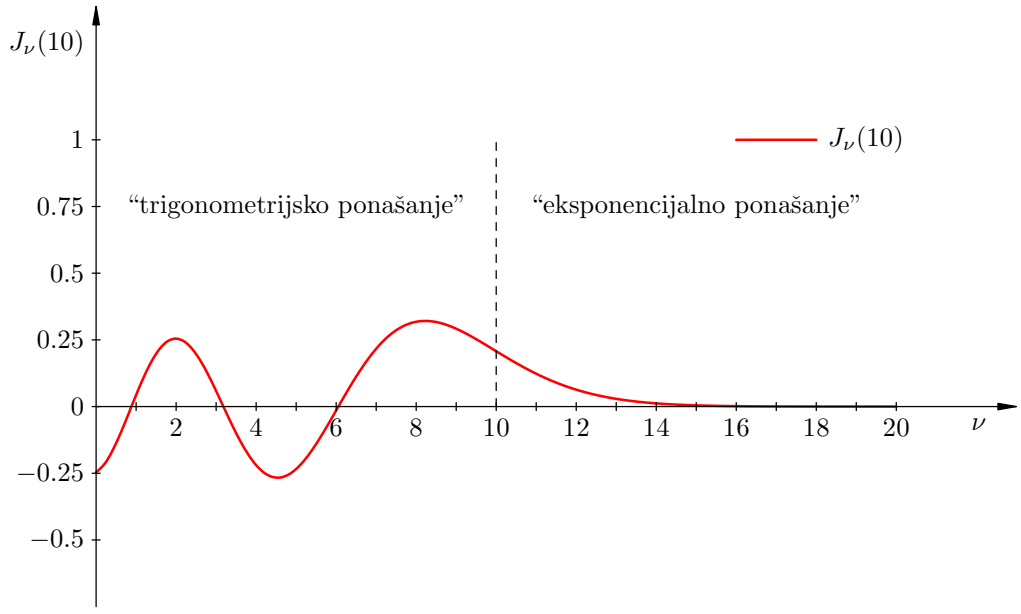
Gledamo li Besselove funkcije ne kao funkcije od x, nego za fiksni x, kao funkcije indeksa ν, onda Besselove funkcije pokazuju ovakvo ponašanje za $J_\nu(k)$, $k = 1, 2, 3,$



za $J_\nu(k)$, $k = 4, 5, 6$,



odnosno, $J_\nu(10)$ izgleda ovako:



Primijetite da i po ν postoji područje trigonometrijskog ponašanja koje za $x \approx \nu$ trne u eksponencijalno.

Vidimo da kad n raste, u rekurziji (8.4.3) dobivamo sve manje i manje brojeve, što znači da mora doći do kraćenja. To pokazuje da je rekurzija nestabilna u rastućem smjeru po n, čim uđemo u eksponencijalno područje.

Za ilustraciju nestabilnosti možemo uzeti $x = 1$ i računati vrijednosti $J_n(x)$ koristeći rekurziju (8.4.3) uzlazno po n, u `extended` preciznosti. Dobiveni rezultati na 18 decimala (apsolutno) dani su u sljedećoj tablici.

n	izračunati $J_n(1)$	točni $J_n(1)$
0	0.765197686557966552	0.765197686557966552
1	0.440050585744933516	0.440050585744933516
2	0.114903484931900481	0.114903484931900481
3	0.019563353982668406	0.019563353982668406
4	0.002476638964109955	0.002476638964109955
5	0.000249757730211237	0.000249757730211234
6	0.000020938338002418	0.000020938338002389
7	0.000001502325817779	0.000001502325817437
8	0.000000094223446486	0.000000094223441726
9	0.000000005249325991	0.000000005249250180
10	0.000000000264421352	0.000000000263061512
11	0.000000000039101058	0.000000000011980067
12	0.000000000595801917	0.000000000000499972

Kraćenje, a time i gubitak relativne točnosti počinje odmah za $n = 2$, ulaskom u eksponencijalno područje. Međutim, to se ne vidi u ovoj tablici, jer su rezultati prikazani apsolutno, a ne relativno. No, za $n = 11$ nemamo više niti jednu točnu znamenku, a za $n = 12$ gubimo i monotoni pad po n.

Vidimo da se događa nešto slično kao kod računanja e^{-nx}, što upućuje na okretanje rekurzije i primjenu Millerovog algoritma.

Korištenjem Millerovog algoritma dobivamo izuzetno dobre rezultate (drugi stupac tablice, koji je točan), a funkcija izvodnica koja se pritom koristi za normalizaciju je vrlo jednostavna

$$J_0(x) + 2(J_2(x) + J_4(x) + \cdots + J_{2k}(x) + \cdots) = 1.$$

Ova relacija izlazi direktno iz (8.4.2) za $t = 1$, kad iskoristimo parnost i neparnost Besselovih funkcija po n, tj. $J_{-n} = (-1)^n J_n$.

Za praktičnu primjenu Millerovog algoritma poželjno je znati precizno ponašanje rekurzije (8.4.3). Uočimo da je x fiksan, a zanima nas ponašanje $J_n(x)$ za velike n. Može se pokazati da u eksponencijalnom području vrijedi tzv. asimptotska relacija

$$J_\nu(x) \approx \frac{1}{\sqrt{2\pi\nu}} \left(\frac{ex}{2\nu}\right)^\nu, \tag{8.4.5}$$

za **fiksni** x i **velike** ν, tj. za $\nu \to \infty$. To pokazuje da se $J_n(x)$, gledano po n, za velike n ponaša kao

$$\frac{c_n}{n^{n+0.5}},$$

gdje je $c_n = (ex/2)^n/\sqrt{2\pi}$, a to vrlo brzo trne kad n raste. Uz malo pažnje, odavde se može izračunati početni indeks M za Millerov algoritam, tako da osiguramo potrebnu točnost.

Na sličan način može se opisati i ponašanje Besselovih funkcija J_ν kada je ν fiksan, a gledamo male ili velike argumente x. Za fiksni ν, kad $x \to 0$ iz prvog člana Taylorovog reda dobivamo i asimptotsku relaciju

$$J_\nu(x) \approx \left(\frac{1}{2}x\right)^\nu \frac{1}{\Gamma(\nu+1)},$$

koja je, očito, dobra aproksimacija za x u eksponencijalnom području blizu nule.

S druge strane, za $x \gtrsim n$, $J_n(x)$ se ponaša poput kosinusa, tj. oscilira. Prava relacija u tom trigonometrijskom području je

$$J_\nu(x) \approx \sqrt{\frac{2}{\pi x}} \cos\left(x - \frac{\pi}{4} - \frac{\nu\pi}{2}\right), \tag{8.4.6}$$

za fiksni ν, kad $x \to \infty$. Točno značenje relacija (8.4.5) i (8.4.6) bit će objašnjeno u sljedećem odjeljku o asimptotskim razvojima. Uobičajeno se koristi oznaka $\sim$, a ne $\approx$, za takve asimpotske relacije.

Napomenimo još da su sve potrebne vrijednosti $J_\nu(x)$ za crtanje prethodnih 6 slika izračunate sumacijom Taylorovog reda (8.4.4) unaprijed, sve dok zadnji dodani član ne padne ispod zadane točnosti. U prikazanom rasponu po ν i po x, kraćenje je minimalno (2–3 dekadske značajne znamenke). Jedini zanimljivi dio tog algoritma je računanje vrijednosti Γ funkcije, o čemu će uskoro biti više riječi.

Naravno, za crtanje grafova nam i ne treba neka velika točnost funkcijskih vrijednosti. U principu, savršeno dovoljno je imati 3 značajne znamenke u izračunatoj vrijednosti funkcije, tj. relativnu točnost reda veličine 10^{-3}. Za vrijednosti blizu nule, ne trebamo ni toliko. Dovoljna je relativna točnost istog reda veličine, ali obzirom na cijelu skalu, tj. raspon vrijednosti funkcije na cijelom grafu. Grešku od jedne tisućinke skale nitko neće ni primijetiti. Naime, ako je cijeli graf visok 10 cm, onda je ta greška na slici manja od desetinke milimetra!

Drugo je pitanje u **koliko** točaka treba izračunati vrijednost funkcije da bi se dobro nacrtao graf. Odgovor na to pitanje slijedi iz ocjena greške raznih vrsta aproksimacija, a posebno interpolacije, što ćemo napraviti u poglavlju o aproksimacijama. Zasad recimo samo to da je svaki od ovih grafova nacrtan korištenjem 101 točke, uz jednak razmak po x osi, a i to je bitno previše. Tridesetak točaka je sasvim dovoljno za vizuelno točan graf na ovim slikama.

Besselove funkcije imaju vrlo velike primjene u fizici, u mnogim modelima, počev od difrakcije svjetlosti do energetskih nivoa u kvantnoj mehanici. Korištenjem Millerovog algoritma, zajedno s Newtonovom metodom za nalaženje nultočaka funkcija, dobro se mogu izračunati i nultočke Besselovih funkcija, koje se, također, vrlo često koriste.

Za razliku od crtanja grafova Besselovih funkcija, za numeričko računanje njihovih nultočaka trebamo maksimalnu moguću točnost funkcijskih vrijednosti (barem u apsolutnom smislu), a to se postiže upravo Millerovim algoritmom.

8.5. Asimptotski razvoj

Sve aproksimacije koje smo do sada promatrali dobivene su "rezanjem" konvergentnih razvoja po nekom sustavu funkcija, tj. "rezanjem" konvergentnih redova na konačnu sumu.

U relaciji (8.2.3) koristili smo razvoj funkcije f oblika

$$f(x) = \sum_{n=0}^{\infty} a_n p_n(x),$$

kojeg smo aproksimirali konačnom parcijalnom sumom (8.2.4)

$$f_N(x) = \sum_{n=0}^{N} a_n p_n(x),$$

podrazumijevajući da je riječ o **konvergentnom** redu u točki x, tj. da ostatak reda teži prema nuli po N, za **fiksni** x

$$\lim_{N\to\infty} (f(x) - f_N(x)) = \lim_{N\to\infty} \sum_{n=N+1}^{\infty} a_n p_n(x) = 0.$$

Strogo formalno, ako se sjetimo definicije sume reda, i sam zapis za $f(x)$ u obliku reda je sinonim za konvergenciju u ovom smislu. Eventualna uniformna konvergencija za sve x na nekoj domeni je dobrodošla, ali nije bitna za ideju ove aproksimacije.

U ovom odjeljku ćemo pokazati da zamjenom uloge N i x u konvergenciji razvoja dobivamo novi pojam **asimptotskog** razvoja, kojeg vrlo efikasno možemo iskoristiti i za praktično računanje. Ovaj pristup se najčešće koristi za računanje vrijednosti integrala, pa ga je zgodno uvesti baš na takvim primjerima.

Neka je f funkcija definirana integralom

$$f(x) = \int_0^{\infty} e^{-xt} \cos t \, dt \tag{8.5.1}$$

za realne nenegativne vrijednosti parametra x. Pokušajmo ovaj intergal izračunati tako da $\cos t$ razvijemo u red potencija po t, a zatim dobiveni red integriramo član po član. Dobivamo redom

$$\begin{aligned} f(x) &= \int_0^\infty e^{-xt}\Big(1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \cdots\Big)\,dt \\ &= \int_0^\infty e^{-xt}\,dt - \int_0^\infty e^{-xt}\,\frac{t^2}{2!}\,dt + \int_0^\infty e^{-xt}\,\frac{t^4}{4!}\,dt - \cdots. \end{aligned}$$

Za integraciju član po član, treba izračunati integrale oblika

$$I_{2n}(x) = \int_0^\infty e^{-xt}\,\frac{t^{2n}}{(2n)!}\,dt,$$

za $n \in \mathbb{N}_0$. Za $n = 0$ odmah dobivamo

$$I_0(x) = \int_0^\infty e^{-xt}\,dt = -\frac{1}{x}\,e^{-xt}\Big|_{t=0}^\infty = \frac{1}{x},$$

jer ostaje samo vrijednost na donjoj granici, a na gornjoj granici znamo da $e^{-xt} \to 0$ kad $t \to \infty$. Za $2n > 0$ parcijalnom integracijom izlazi

$$\begin{aligned} I_{2n}(x) &= \int_0^\infty \frac{t^{2n}}{(2n)!}\,d\Big(-\frac{1}{x}\,e^{-xt}\Big) \\ &= -\frac{1}{x}\,e^{-xt}\,\frac{t^{2n}}{(2n)!}\Big|_{t=0}^\infty + \frac{1}{x}\int_0^\infty e^{-xt}\,\frac{t^{2n-1}}{(2n-1)!}\,dt \\ &= \frac{1}{x}\,I_{2n-1}(x). \end{aligned}$$

Odavde indukcijom slijedi

$$I_{2n}(x) = \frac{1}{x^{2n+1}}.$$

Kad ove integrale uvrstimo natrag u red za f, dobivamo

$$f(x) = I_0(x) - I_2(x) + I_4(x) - \cdots = \frac{1}{x} - \frac{1}{x^3} + \frac{1}{x^5} - \cdots.$$

Na kraju, ako je $x > 1$, onda red na desnoj strani konvergira i vrijedi

$$f(x) = \frac{x}{x^2 + 1}. \tag{8.5.2}$$

Ovaj rezultat možemo dobiti i direktno iz (8.5.1) dvostrukom parcijalnom integracijom.

$$\begin{aligned} f(x) &= \int_0^\infty e^{-xt}\cos t\,dt = \int_0^\infty e^{-xt}d(\sin t) \\ &= e^{-xt}\,\sin t\Big|_{t=0}^\infty + \frac{1}{x}\int_0^\infty e^{-xt}\,\sin t\,dt. \end{aligned}$$

Prvi član je nula na obje granice, pa ostaje

$$\begin{aligned} f(x) &= \frac{1}{x}\int_0^\infty e^{-xt}d(-\cos t) \\ &= -\frac{1}{x}\,e^{-xt}\,\cos t\Big|_{t=0}^\infty - \frac{1}{x^2}\int_0^\infty e^{-xt}\,\cos t\,dt \\ &= \frac{1}{x} - \frac{1}{x^2}\,f(x). \end{aligned}$$

Množenjem s x^2, za $x > 0$, dobivamo

$$x^2 f(x) = x - f(x),$$

pa zaključujemo da vrijedi (8.5.2), samo uz blažu pretpostavku $x > 0$.

Pokušajmo primijeniti istu ideju za funkciju g definiranu integralom

$$g(x) = \int_0^\infty \frac{e^{-xt}}{1+t}\,dt, \tag{8.5.3}$$

opet uz pretpostavku da je $x > 0$. Očekujemo još bolje rezultate, jer podintegralna funkcija još malo brže trne kad $t \to \infty$. Ovdje vrijedi $1/(1+t) \to 0$ kad $t \to \infty$, dok je $\cos t$ u funkciji f samo ograničen između -1 i 1. Supstitucijom razvoja

$$\frac{1}{1+t} = 1 - t + t^2 - \cdots$$

dobivamo redom

$$\begin{aligned} g(x) &= \int_0^\infty e^{-xt}(1 - t + t^2 - \cdots)\,dt \\ &= \int_0^\infty e^{-xt}\,dt - \int_0^\infty e^{-xt}t\,dt + \int_0^\infty e^{-xt}\,t^2\,dt - \cdots. \end{aligned}$$

Za integraciju član po član, treba izračunati integrale oblika

$$\widehat{I}_n(x) = \int_0^\infty e^{-xt}\,t^n\,dt, \quad n \in \mathbb{N}_0.$$

Usporedbom s integralima $I_n(x)$ za funkciju f iz prethodnog primjera odmah vidimo da je

$$\widehat{I}_n(x) = n!\, I_n(x), \quad n \in \mathbb{N}_0,$$

pa je

$$\widehat{I}_n(x) = \frac{n!}{x^{n+1}}, \quad n \in \mathbb{N}_0.$$

Kad ove integrale uvrstimo natrag u red za g, dobivamo

$$g(x) = \widehat{I}_0(x) - \widehat{I}_1(x) + \widehat{I}_2(x) - \cdots = \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots. \tag{8.5.4}$$

Međutim, ovaj red **divergira** za sve konačne vrijednosti x i relacija (8.5.4) je besmislena. Dakle, funkciju g iz (8.5.3) ne možemo izračunati na ovaj način.

Zašto je ovaj postupak bio uspješan u prvom primjeru, a ostao bez rezultata u drugom? Odgovor je jednostavan. Razvoj funkcije $\cos t$ konvergira za sve vrijednosti t, tj. na cijeloj domeni integracije. čak i jače, on konvergira uniformno na svakom konačnom intervalu za t. Zbog toga smijemo iskoristiti integraciju član po član na svakom konačnom intervalu, a zatim pustiti gornju granicu integracija na limes $t \to \infty$.

U drugom slučaju, razvoj funkcije $1/(1+t)$ konvergira samo za $t < 1$, a divergira za $t \geq 1$. Rezultat iz (8.5.4) treba shvatiti kao posljedicu integracije reda član po član, ali na intervalu na kojem taj red ne konvergira uniformno.

Sve dosad rečeno su standardne činjenice o redovima i integraciji iz matematičke analize. Međutim, ako cijelu stvar gledamo malo manje teorijski, a više praktično, onda nam nitko ne brani da pokušmo sumirati prvih nekoliko članova reda na desnoj strani u (8.5.4), za neku fiksnu vrijednost x. Idemo vidjeti što će se dogoditi, iako je to u potpunoj suprotnosti s poznatom teorijom!

Označimo s $g_n(x)$ parcijalne sume prvih n članova reda iz (8.5.4)

$$g_n(x) = \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots + (-1)^{n-1}\,\frac{(n-1)!}{x^n}. \tag{8.5.5}$$

Uzmimo neki malo veći x, recimo $x = 10$, tako da nazivnici relativno brzo padaju i izračunajmo prvih nekoliko vrijednosti $g_n(10)$. Za usporedbu, trebamo još i točnu vrijednost $g(10)$. Numeričkom integracijom može se izračunati da je

$$g(10) = 0.0915633339397880819\,.$$

Sljedeća tablica pokazuje izračunate vrijednosti $g_n(10)$ i pripadne pogreške.

n	izračunati $g_n(10)$	greška $g(10) - g_n(10)$
0	0.100000000000000000	−0.008436666060211918
1	0.090000000000000000	0.001563333939788082
2	0.092000000000000000	−0.000436666060211918
3	0.091400000000000000	0.000163333939788082
4	0.091640000000000000	−0.000076666060211918
5	0.091520000000000000	0.000043333939788082
6	0.091592000000000000	−0.000028666060211918
7	0.091541600000000000	0.000021733939788082
8	0.091581920000000000	−0.000018586060211918
9	0.091545632000000000	0.000017701939788082
10	0.091581920000000000	−0.000018586060211918
11	0.091542003200000000	0.000021330739788082
12	0.091589903360000000	−0.000026569420211918

Dobivene vrijednosti su sasvim dobre aproksimacije! Vidimo da je $g_9(10)$ najbolja aproksimacija, koja daje skoro 5 točnih decimala i skoro 4 točne vodeće znamenke. Naravno, za $n \geq 10$ pogreške sve više rastu i rezultati postaju beskorisni.

Sve u svemu, rezultat uopće nije loš, kad uzmemo da je nastao sumiranjem iz divergentnog reda. Idemo još naći i objašnjenje za ovaj prilično neočekivani uspjeh.

Jasno je da treba promatrati grešku n-te parcijalne sume iz (8.5.5) u fiksnoj točki x. Neka je

$$e_n(x) := g(x) - g_n(x).$$

Ako još i razvoj funkcije $1/(1+t)$ napišemo u istom obliku kao zbroj prvih n članova plus ostatak,

$$\frac{1}{1+t} = 1 - t + t^2 - \cdots + (-1)^{n-1}t^{n-1} + \frac{(-1)^n t^n}{1+t},$$

i to uvrstimo u definiciju (8.5.3) za g, dobivamo da je greška $e_n(x)$ upravo odgovarajući integral ostatka u prethodnoj relaciji,

$$e_n(x) = (-1)^n \int_0^\infty \frac{e^{-xt}t^n}{1+t}\,dt,$$

jer sad smijemo integrirati konačnu sumu član po član. Ovu grešku nije teško ocijeniti. Očito je

$$\frac{1}{1+t} \leq 1, \quad t \geq 0,$$

pa je

$$|e_n(x)| = \int_0^\infty \frac{e^{-xt}t^n}{1+t}\,dt \le \int_0^\infty e^{-xt}t^n\,dt = \frac{n!}{x^{n+1}}. \tag{8.5.6}$$

To pokazuje da je pogreška n-te parcijalne sume manja od apsolutne vrijednosti prvog odbačenog člana. Osim toga, zato što članovi alterniraju po predznaku, pogreška ima isti predznak kao i prvi odbačeni član. Dakle, članove reda možemo iskoristiti za ocjenu pogreške i zbrajanje članova treba zaustaviti točno **ispred** apsolutno najmanjeg člana.

Iz ocjene (8.5.6) za pogrešku n-te parcijalne sume odmah slijede dva zaključka. Ako gledamo konvergenciju u fiksnoj točki $x > 0$, onda je

$$|e_n(x)| \to \infty \quad \text{za} \quad n \to \infty,$$

tj. nema govora o konvergenciji parcijalnih suma $g_n(x)$ prema $g(x)$, što odgovara ranijem zaključku da pripadni red divergira u svakoj točki $x > 0$.

S druge strane, ako uzmemo da je n fiksan, onda vrijedi

$$|e_n(x)| \to 0 \quad \text{za} \quad x \to \infty, \tag{8.5.7}$$

što odgovara "konvergenciji", ali sa zamijenjenim ulogama n i x. Takvu vrstu "konvergencije" zovemo **asimptotska konvergencija**, u ovom slučaju, u okolini točke $+\infty$.

Parcijalne sume g_n iz (8.5.5) generiraju red na desnoj strani (8.5.4), za kojeg kažemo da je **asimptotski razvoj** funkcije g u okolini točke $+\infty$, a relaciju (8.5.4) pišemo u obliku

$$g(x) \sim \frac{1}{x} - \frac{1!}{x^2} + \frac{2!}{x^3} - \frac{3!}{x^4} + \cdots + (-1)^{n-1}\frac{(n-1)!}{x^n} + \cdots \quad (x \to +\infty), \tag{8.5.8}$$

gdje $\sim$ označava asimptotsku konvergenciju reda na desnoj strani ove relacije, u smislu (8.5.7).

Prije precizne definicije ovog pojma, objasnimo još malo vrijednost prethodnog zaključka. Relacija (8.5.7) vrijedi za svaki $n \in \mathbb{N}$, što znači da **svaku** od funkcija g_n možemo koristiti za aproksimaciju funkcije g, samo to moramo napraviti za dovoljno velike vrijednosti x, tj. na odgovarajućoj okolini točke $+\infty$. Takvom aproksimacijom **ne** možemo postići proizvoljno veliku točnost, odnosno proizvoljno malu grešku, kao kod obične konvergencije. Ovisno o x, postoji minimalna greška koju možemo dobiti (gledano po n) i bolje ne ide. Međutim, i to se uspješno može iskoristiti za praktično računanje.

Osim toga, ovaj rezultat ima i teorijsku vrijednost. Uzmimo da je $n = 1$ u (8.5.7). To znači da se, u prvoj aproksimaciji, funkcija g ponaša kao g_1 u okolini

točke $+\infty$, tj. da $g(x)$ pada kao $1/x$, kad $x \to +\infty$. Iz (8.5.6) dobivamo i ocjenu greške za takvu aproksimaciju. Preciznije, imamo

$$g(x) - g_1(x) = g(x) - 1/x = e_1(x)$$

i iz (8.5.6) slijedi

$$|e_1(x)| \leq \int_0^\infty e^{-xt} t \, dt = \frac{1}{x^2},$$

pa odmah vidimo da vrijede sljedeće dvije relacije asimptotskog ponašanja. Prva relacija je

$$g(x) - g_1(x) = O(1/x^2) \quad (x \to +\infty)$$

i ona služi kao osnova za definiciju asimptotskog razvoja. Druga relacija je

$$g(x) - g_1(x) = o(g_1(x)) \quad (x \to +\infty),$$

ili, ekvivalentno (jer $g_1(x) \neq 0$ za $x > 0$)

$$\lim_{x \to +\infty} \frac{g(x)}{g_1(x)} = 1.$$

Posljednju relaciju obično pišemo u obliku

$$g(x) \sim g_1(x) \quad (x \to +\infty) \tag{8.5.9}$$

i čitamo "$g(x)$ je asimptotski jednako $g_1(x)$ kad $x \to +\infty$", što znači da relativna greška između g i g_1 teži prema nuli kad $x \to +\infty$.

Sve to slijedi samo iz prvog člana asimptotskog razvoja. Ako znamo više članova, dobivamo sve preciznije informacije o ponašanju fukcije g u okolini $+\infty$.

Napomenimo da ista oznaka $\sim$ ima različita značenja u relacijama (8.5.8) i (8.5.9). U (8.5.9) $\sim$ je relacija asimptotskog ponašanja i ta relacija je simetrična, čak relacija ekvivalencije. Za razliku od toga, u (8.5.8) $\sim$ označava asimptotski razvoj i nije simetrična. Lijevo je funkcija, a desno je red potencija (red funkcija). Međutim, zbog toga što iz (8.5.8) slijede (8.5.9) i slični zaključci za aproksimacije g_n s više članova reda, ova oznaka $\sim$ se tradicionalno koristi u oba značenja. Naime, relaciju (8.5.9) možemo interpretirati i kao skraćeni zapis nekog asimptotskog razvoja, s tim da je naveden samo prvi član razvoja. Ali oprezno, tada zapis $g(x) \sim g_1(x)$ (po definiciji) znači samo

$$\lim_{x \to +\infty} \frac{g(x)}{g_1(x)} = 1,$$

i nema govora o nekoj asimptotskoj konvergenciji oblika (8.5.7), jer ne postoje ostali članovi razvoja (nema parametra n). Upravo tako treba interpretirati i asimptotske relacije za Besselove funkcije iz prethodnog odjeljka.

Precizna definicija asimptotskog razvoja u okolini neke točke bazirana je na definiciji asimptotskog niza u okolini te točke.

Definicija 8.5.1. (Asimptotski niz) *Neka je $D \subseteq \mathbb{R}$ neka domena i $c \in \operatorname{Cl} D$ neka točka iz zatvarača skupa D, s tim da c može biti i $+\infty$ ili $-\infty$. Nadalje, neka je $\varphi_n : D \to \mathbb{R}$, $n \in \mathbb{N}_0$, niz funkcija za kojeg vrijedi*

$$\varphi_n(x) = o(\varphi_{n-1}(x)) \quad (x \to c \text{ u } D),$$

za svaki $n \in \mathbb{N}$. Tada kažemo da je (φ_n) **asimptotski niz** *kad $x \to c$ u skupu D.*

Podsjetimo, to znači da svaka funkcija φ_n raste bitno sporije od prethodne funkcije φ_{n-1} u okolini točke c, u smislu da vrijedi

$$\lim_{\substack{x \to c \\ x \in D}} \frac{\varphi_n(x)}{\varphi_{n-1}(x)} = 0,$$

što uključuje i pretpostavku da je $\varphi_{n-1}(x) \neq 0$ na nekoj okolini točke c gledano u skupu D, osim eventualno u samoj točki c.

Ista definicija vrijedi i za kompleksne domene $D \subseteq \mathbb{C}$, a točka c može biti i ∞.

Definicija 8.5.2. (Asimptotski razvoj) *Neka je (φ_n), $n \in \mathbb{N}_0$, asimptotski niz kad $x \to c$ u skupu D. Formalni red funkcija*

$$\sum_{n=0}^{\infty} a_n \varphi_n$$

je **asimptotski razvoj** *funkcije f kad $x \to c$ u skupu D, u oznaci*

$$f(x) \sim \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D), \tag{8.5.10}$$

ako za svaki $N \in \mathbb{N}$ vrijedi relacija asimptotskog ponašanja

$$f(x) = \sum_{n=0}^{N-1} a_n \varphi_n(x) + O(\varphi_N(x)) \quad (x \to c \text{ u } D),$$

tj. apsolutna greška između f i $(N-1)$-e parcijalne sume reda raste najviše jednako brzo kao i N-ti član asimptotskog niza, u okolini točke c.

Navedeni red funkcija treba interpretirati samo kao oznaku, tj. u čisto formalnom smislu, jer on može biti divergentan u svakoj točki domene.

Uočimo da iz prethodne dvije definicije odmah slijedi i

$$f(x) = \sum_{n=0}^{N-1} a_n \varphi_n(x) + o(\varphi_{N-1}(x)) \quad (x \to c \text{ u } D),$$

za svaki $N \in \mathbb{N}$. To znači da apsolutna greška između f i bilo koje parcijalne sume reda raste bitno sporije od zadnjeg člana u parcijalnoj sumi, u okolini točke c.

Tipični primjeri asimptotskih nizova su obične i logaritamske potencije

$$\varphi_n(x) = (x-c)^n \qquad \text{ili} \qquad \varphi_n(x) = (\log(x-c))^n,$$

za $n \in \mathbb{N}_0$, u okolini točke $c \in \mathbb{R}$ (ili $\mathbb{C}$). Pripadni asimptotski razvoji su obični redovi potencija

$$f(x) \sim \sum_{n=0}^{\infty} a_n (x-c)^n \quad (x \to c \text{ u } D),$$

ili logaritamskih potencija

$$f(x) \sim \sum_{n=0}^{\infty} a_n (\log(x-c))^n \quad (x \to c \text{ u } D).$$

U praksi se najčešće se koriste asimptotski nizovi u okolini točke ∞ oblika

$$\varphi_n(x) = x^{-n} \qquad \text{ili} \qquad \varphi_n(x) = (\log x)^{-n},$$

za $n \in \mathbb{N}_0$, koji nastaju supstitucijom $x \to 1/x$ iz nizova u okolini točke $c = 0$. Za obične (inverzne) potencije, asimptotski razvoj (8.5.10) ima već poznati oblik

$$f(x) \sim \sum_{n=0}^{\infty} \frac{a_n}{x^n} \quad (x \to \infty \text{ u } D),$$

što, po definiciji, znači da vrijedi

$$f(x) = \sum_{n=0}^{N-1} \frac{a_n}{x^n} + O(x^{-N}) \quad (x \to \infty \text{ u } D).$$

Tada nema smisla govoriti o rastu, već o padu funkcija u okolini točke ∞. Apsolutna greška između f i $(N-1)$-e parcijalne sume reda pada barem jednako brzo kao i x^{-N}, odnosno bitno brže od zadnjeg člana u sumi, koji je proporcionalan s $x^{-(N-1)}$. Povijesno gledano, H. Poincaré je 1886. godine definirao asimptotski razvoj baš u ovom obliku.

Napomenimo još da asimptotsko ponašanje može bitno ovisiti o domeni D, ne samo zbog područja definicije funkcija, već zbog moguće restrikcije točaka po kojima se gledaju limesi kad x teži prema c.

Ako svi navedeni limesi kad $x \to c$ vrijede i kad umjesto D uzmemo osnovni skup $\mathbb{R}$ ili $\mathbb{C}$, tj. "bezuvjetno" (što nešto kaže i o domeni D), onda se koristi skraćena oznaka $(x \to c)$, bez navođenja domene D. Ako je još i $c = \infty$, onda se, tradicionalno, oznaka $(x \to \infty)$ može i ispustiti, tj. podrazumijeva se razvoj u okolini točke ∞.

Može se dogoditi da funkcija f nema asimptotski razvoj na zadanoj domeni D u smislu prethodne definicije. Ako je g poznata ili zadana funkcija takva da f/g ima asimptotski razvoj, onda se umjesto relacije (8.5.10) za f/g, često koristi i oznaka

$$f(x) \sim g(x) \cdot \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D).$$

Ako je $a_0 \neq 0$, onda prvi član ovog razvoja daje i asimptotsko ponašanje funkcije f, tj. vrijedi $f(x) \sim a_0 g(x)$, kad $x \to c$ u D. U protivnom, ako je prvih k koeficijenata razvoja jednako nuli, tj. $a_0 = \ldots = a_{k-1} = 0$ i $a_k \neq 0$, onda vrijedi slična relacija (koja?).

Analogno, ako $f - g$ ima asimptotski razvoj, gdje je g poznata funkcija, obično pišemo

$$f(x) \sim g(x) + \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad (x \to c \text{ u } D).$$

8.6. Verižni razlomci i racionalne aproksimacije

Na početku ovog poglavlja zaključili smo da efektivno možemo računati samo racionalne aproksimacije funkcija. Ako dobro pogledate sve što smo do sada napravili, onda možete zaključiti da se sve dosadašnje aproksimacije svode na polinome ili sume polinomnog oblika, u kojima su potencije zamijenjene nekim drugim funkcijama. U tim aproksimacijama, dijeljenje nismo bitno iskoristili u obliku aproksimacije, već samo za računanje funkcija po kojima se razvija.

Preciznije, sve dosadašnje aproksimacije imaju oblik **linearne kombinacije** nekih funkcija baze, pa tim aproksimacijama prirodno odgovara teorija linearnih ili vektorskih prostora, a tek za analizu konvergencije i ocjenu greške trebamo nešto "bogatiju" strukturu.

Ako se sjetimo nekih rezultata iz analize, poput Weierstraßovog teorema o uniformnoj aproksimaciji funkcije polinomima na kompaktu, moglo bi nam se učiniti da uopće nema potrebe za drugim vrstama aproksimacija. S druge strane, prirodno je očekivati da "dodavanjem" operacije dijeljenja u oblik aproksimacione funkcije možemo postići znatno bolje aproksimacije za razne klase funkcija, koristeći približno isti broj aritmetičkih operacija.

Pravu usporedbu polinomnih i racionalnih aproksimacija ostavljamo za poglavlje o aproksimacijama. Međutim, za praktičnu primjenu racionalnih aproksimacija trebamo dobre algoritme za njihovo izvrednjavanje.

Pretpostavimo da je zadana racionalna funkcija oblika

$$R(x) = \frac{P_n(x)}{Q_m(x)},$$

gdje su P_n i Q_m polinomi stupnjeva n i m, respektivno,

$$P_n(x) = \sum_{k=0}^{n} a_k x^k, \quad Q_m(x) = \sum_{k=0}^{m} b_k x^k.$$

Očito je da izvrednjavanje prethodne funkcije možemo izvršiti korištenjem dvije Hornerove sheme (za polinom u brojniku i nazivniku) i jednim dijeljenjem na kraju. Broj potrebnih operacija je $n + m$ množenja, $n + m$ zbrajanja i jedno dijeljenje.

Ipak, takvo izvrednjavanje racionalne funkcije nije idealno, jer se ono može izvršiti s manje operacija. Postoji još jedan problem kod takvog pristupa. Pretpostavite da je vrijednost funkcije $R(x_0)$ neki broj razumnog reda veličine. Nažalost, može se dogoditi, i to nije tako rijetko, da je taj broj dobiven dijeljenjem dva vrlo velika broja koja nisu prikaziva u aritmetici računala. Na neki način, tada bi trebalo vršiti neku normalizaciju u Hornerovoj shemi, čim nam kvocijent prijeđe neku zadanu veličinu, a tada algoritam postaje strašno kompliciran.

Na primjer, ako aproksimiramo vrijednost funkcije $\operatorname{th} x$ (ili bilo koju drugu funkciju f za koju vrijedi da $f(x)$ ne teži u ∞ kad $x \to \infty$) u točki x_0, x_0 veliko, racionalna aproksimacija bit će omjer dva polinoma.

Ako želimo racionalnu aproksimaciju koja ima stupanj polinoma bar jedan u brojniku i nazivniku, onda će polinomi u brojniku i nazivniku za veliki x_0 težiti u ∞, a sama vrijednost funkcije bit će broj nekog razumnog reda veličine.

U teoriji nepolinomnih aproksimacija moguće je pokazati da su, uz neke uvjete, najbolje racionalne aproksimacije one kojima je stupanj polinoma u brojniku jednak onom u nazivniku ili se eventualno razlikuje za jedan. U tom slučaju racionalnu aproksimaciju možemo napisati kao verižni razlomak, pa će i brzina računanja i problem preljeva (overflow) biti riješeni načinom izvrednjavanja takvog verižnog razlomka.

No prije no što upoznamo tzv. funkcionalne verižne razlomke, upoznajmo se s brojevnim verižnim razlomcima.

8.6.1. Brojevni verižni razlomci

Izraz oblika

$$R = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \cdots}}}}$$

zovemo (brojevni) verižni razlomak. Ovakav zapis je obično "prevelik" za matematičke knjige, pa su smišljeni alternativni zapisi. U različitoj literaturi nailazimo na tri oblika zapisa verižnih razlomaka

$$R = \left[b_0; \frac{a_1}{b_1}, \frac{a_2}{b_2}, \frac{a_3}{b_3}, \dots\right], \quad R = b_0 + \frac{a_1 \mid}{\mid b_1} + \frac{a_2 \mid}{\mid b_2} + \frac{a_3 \mid}{\mid b_3} + \cdots,$$

i možda najzgodniji zapis

$$R = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots. \tag{8.6.1}$$

Ako u beskonačnom verižnom razlomku uzmemo samo konačno mnogo članova,

$$R_n = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots \frac{a_n}{b_n}, \tag{8.6.2}$$

onda se takav izraz zove n-ta konvergencija verižnog razlomka R. Ako postoji vrijednost verižnog razlomka (8.6.1), onda se ona definira kao

$$R = \lim_{n\to\infty} R_n,$$

gdje je R_n n-ta konvergencija definirana izrazom (8.6.2).

Drugim riječima, važno je znati kako efikasno izračunati R_n.

8.6.2. Uzlazni algoritam za izvrednjavanje brojevnih verižnih razlomaka

Promatrajmo n-tu konvergenciju verižnog razlomka, koju možemo prikazati kao racionalni broj, kvocijent P_n i Q_n

$$R_n = \frac{P_n}{Q_n} = b_0 + \frac{a_1}{{b_1}^+} \frac{a_2}{{b_2}^+} \frac{a_3}{{b_3}^+} \cdots \frac{a_n}{b_n}.$$

Za nultu konvergenciju je

$$R_0 = \frac{P_0}{Q_0} = b_0,$$

pa možemo izabrati da je $P_0 = b_0$, $Q_0 = 1$ (mogli smo i drugačije birati, jedini je uvjet da je $P_0/Q_0 = b_0$). Za sljedeću konvergenciju vrijedi

$$R_1 = \frac{P_1}{Q_1} = b_0 + \frac{a_1}{b_1} = \frac{b_0 b_1 + a_1}{b_1} = \frac{b_1 P_0 + a_1}{b_1 Q_0}.$$

Ako još definiramo $P_{-1} = 1$, $Q_{-1} = 0$, onda prethodna relacija glasi

$$R_1 = \frac{P_1}{Q_1} = \frac{b_1 P_0 + a_1 P_{-1}}{b_1 Q_0 + a_1 Q_{-1}},$$

tj. ponovno možemo zatražiti da vrijedi da je brojnik jednak brojniku, a nazivnik nazivniku, tj. da je

$$\begin{aligned} P_1 &= b_1 P_0 + a_1 P_{-1}, \\ Q_1 &= b_1 Q_0 + a_1 Q_{-1}. \end{aligned}$$

Te dvije relacije su baza rekurzije. Primijetite, ako za P_n i Q_n vrijede relacije

$$\begin{aligned} P_n &= b_n P_{n-1} + a_n P_{n-2}, \\ Q_n &= b_n Q_{n-1} + a_n Q_{n-2}, \end{aligned}$$

onda za R_{n+1} vrijedi

$$R_{n+1} = \frac{P'_{n+1}}{Q'_{n+1}},$$

gdje je

$$\begin{aligned} P'_{n+1} &= \left(b_n + \frac{a_{n+1}}{b_{n+1}}\right) P_{n-1} + a_n P_{n-2} \\ &= (b_n P_{n-1} + a_n P_{n-2}) + \frac{a_{n+1}}{b_{n+1}} P_{n-1} = P_n + \frac{a_{n+1}}{b_{n+1}} P_{n-1}, \\ Q'_{n+1} &= \left(b_n + \frac{a_{n+1}}{b_{n+1}}\right) Q_{n-1} + a_n Q_{n-2} \\ &= (b_n Q_{n-1} + a_n Q_{n-2}) + \frac{a_{n+1}}{b_{n+1}} Q_{n-1} = Q_n + \frac{a_{n+1}}{b_{n+1}} Q_{n-1}. \end{aligned}$$

Definiramo li

$$\begin{aligned} P_{n+1} &= b_{n+1} P'_{n+1}, \\ Q_{n+1} &= b_{n+1} Q'_{n+1}, \end{aligned}$$

onda R_{n+1} ostaje nepromijenjen (brojnik i nazivnik su skalirani), a prethodna rekurzija postaje

$$\begin{aligned} P_{n+1} &= b_{n+1} P_n + a_{n+1} P_{n-1}, \\ Q_{n+1} &= b_{n+1} Q_n + a_{n+1} Q_{n-1}, \end{aligned}$$

čime smo dokazali korak indukcije.

Drugim riječima, definiramo li

$$P_{-1} = 1, \quad Q_{-1} = 0, \quad P_0 = b_0, \quad Q_0 = 1, \tag{8.6.3}$$

onda, indukcijom dobivamo tzv. uzlazni algoritam izvrednjavanja verižnog razlomka

$$\begin{aligned} P_k &= b_k P_{k-1} + a_k P_{k-2}, \\ Q_k &= b_k Q_{k-1} + a_k Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n. \tag{8.6.4}$$

Primijetite da se u ovakvom zapisu algoritma lako mogu dodavati novi a_k i b_k, tzv. karike u verižnom razlomku.

Iz (8.6.4) lako se čita da su P_k i Q_k dva rješenja diferencijske jednadžbe

$$y_k - b_k y_{k-1} - a_k y_{k-2} = 0.$$

Uočite da bi nam u algoritmu (8.6.4) odgovaralo da su ili a_k ili b_k jednaki 1, tako da ne moramo množiti tim koeficijentima. To se može postići korištenjem tzv. ekvivalentne transformacije.

Neka su w_k, za $k \geq 1$, proizvoljni brojevi različiti od 0 i $w_{-1} = w_0 = 1$. Tvrdimo da izvrednjavanjem verižnog razlomka

$$R' = b_0 + \frac{w_0 w_1 a_1}{w_1 b_1{}^+} \frac{w_1 w_2 a_2}{w_2 b_2{}^+} \frac{w_2 w_3 a_3}{w_3 b_3{}^+} \cdots. \tag{8.6.5}$$

dobijemo isti R kao u (8.6.1). Označimo sa S_n i T_n brojnik i nazivnik n-te konvergencije prethodnog verižnog razlomka

$$R'_n = \frac{S_n}{T_n} = b_0 + \frac{w_0 w_1 a_1}{w_1 b_1{}^+} \frac{w_1 w_2 a_2}{w_2 b_2{}^+} \frac{w_2 w_3 a_3}{w_3 b_3{}^+} \cdots \frac{w_{n-1} w_n a_n}{w_n b_n}.$$

Pogledajmo u kojem su odnosu S_k i T_k obzirom na P_k i Q_k. Prvo napišimo rekurzije za S_k i T_k, jednostavno supstituirajući "nove", proširene a_k i b_k

$$\begin{aligned} S_k &= w_k b_k S_{k-1} + w_{k-1} w_k a_k S_{k-2}, \\ T_k &= w_k b_k T_{k-1} + w_{k-1} w_k a_k T_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz

$$S_{-1} = 1, \quad T_{-1} = 0, \quad S_0 = b_0, \quad T_0 = 1.$$

Tvrdimo da postoji veza između P_k i S_k, te Q_k i T_k. Vrijedi

$$S_k = P_k \cdot \prod_{i=1}^{k} w_i, \quad T_k = Q_k \cdot \prod_{i=1}^{k} w_i, \quad k = 1, \ldots, n.$$

Dokaz se provodi indukcijom po k. Za bazu indukcije uzmimo $k = 1, 2$. Iz rekurzija dobivamo:

$$\begin{aligned} P_1 &= b_1 P_0 + a_1 P_{-1}, \\ P_2 &= b_2 P_1 + a_2 P_0, \\ S_1 &= w_1 b_1 S_0 + w_0 w_1 a_1 S_{-1} = w_1 b_1 P_0 + w_1 a_1 P_{-1} \\ &= w_1 (b_1 P_0 + a_1 P_{-1}) = w_1 P_1, \\ S_2 &= w_2 b_2 S_1 + w_1 w_2 a_2 S_0 = w_2 b_2 w_1 P_1 + w_1 w_2 a_2 P_0 \\ &= w_1 w_2 (b_2 P_1 + a_2 P_0) = w_1 w_2 P_2. \end{aligned}$$

Za korak indukcije, pretpostavimo da vrijedi

$$S_k = P_k \cdot \prod_{i=1}^{k} w_i, \quad S_{k-1} = P_{k-1} \cdot \prod_{i=1}^{k-1} w_i$$

za neke k, $k-1$. Tada vrijedi

$$\begin{aligned} S_{k+1} &= w_{k+1}b_{k+1}S_k + w_k w_{k+1} a_{k+1} S_{k-1} \\ &= w_{k+1}b_{k+1}P_k \cdot \prod_{i=1}^{k} w_i + w_k w_{k+1} a_{k+1} P_{k-1} \cdot \prod_{i=1}^{k-1} w_i \\ &= (b_{k+1}P_k + a_{k+1}P_{k-1}) \cdot \prod_{i=1}^{k+1} w_i = P_{k+1} \cdot \prod_{i=1}^{k+1} w_i \end{aligned}$$

što je i trebalo pokazati. Na sličan se način dokazuje i relacija za T_k i Q_k.

Dakle, vrijedi

$$R'_n = \frac{S_n}{T_n} = \frac{P_n \cdot \prod_{i=1}^{n+1} w_i}{Q_n \cdot \prod_{i=1}^{n+1} w_i} = \frac{P_n}{Q_n} = R_n.$$

Sada možemo pojednostavniti verižni razlomak R, tj. svesti ga na alternativnu formu, tako da u rekurziji (8.6.4) ili a_k ili b_k budu jednaki 1. Pretpostavimo da su $a_k \neq 0$ za sve $k \geq 1$. Budući da je izbor w_k, $k \geq 1$, proizvoljan (do na to da ne smiju biti 0), onda w_k izaberemo tako da vrijedi

$$w_1 a_1 = 1, \quad w_1 w_2 a_2 = 1, \quad \ldots, \quad w_{n-1} w_n a_n = 1.$$

Odatle odmah slijedi da je

$$w_1 = \frac{1}{a_1}, \quad w_2 = \frac{1}{w_1 a_2} = \frac{a_1}{a_2}, \quad w_3 = \frac{1}{w_2 a_3} = \frac{a_2}{a_1 a_3}, \quad \ldots,$$

odnosno općenito

$$w_{2k} = \frac{a_1 a_3 \cdots a_{2k-1}}{a_2 a_4 \cdots a_{2k}}, \quad w_{2k+1} = \frac{a_2 a_4 \cdots a_{2k}}{a_1 a_3 \cdots a_{2k+1}},$$

što se dokazuje indukcijom. Time smo dobili tzv. II tip verižnog razlomka kojem su brojnici jednaki 1, tj. dobili smo verižni razlomak oblika

$$R' = b_0 + \frac{1}{b'_1+} \frac{1}{b'_2+} \frac{1}{b'_3+} \cdots.$$

Pripadna uzlazna rekurzija za izvrednjavanje ima oblik

$$\begin{aligned} P_k &= b'_k P_{k-1} + P_{k-2}, \\ Q_k &= b'_k Q_{k-1} + Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz start (8.6.3).

S druge strane, možemo postići i da su nazivnici jednaki 1. Pretpostavimo da su $b_k \neq 0$ za sve $k \geq 1$. Budući da je izbor w_k, $k \geq 1$, proizvoljan (do na to da ne smiju biti 0), onda w_k izaberemo tako da vrijedi

$$w_1 b_1 = 1, \quad w_2 b_2 = 1, \quad \ldots, \quad w_n b_n = 1,$$

tj. stavljanjem

$$w_k = \frac{1}{b_k}$$

dobivamo tzv. I tip verižnog razlomka kojem su nazivnici jednaki 1, tj. dobili smo verižni razlomak oblika

$$R' = b_0 + \frac{a'_1}{1^+} \frac{a'_2}{1^+} \frac{a'_3}{1^+} \cdots.$$

Koeficijenti a'_k su jednaki

$$a'_1 = \frac{a_1}{b_1}, \quad a'_2 = \frac{a_2}{b_1 b_2}, \quad a'_3 = \frac{a_3}{b_2 b_3},$$

odnosno općenito

$$a'_k = \frac{a_k}{b_{k-1} b_k}.$$

Pripadna rekurzija za uzlazno izvrednjavanje je

$$\begin{aligned} P_k &= P_{k-1} + a'_k P_{k-2}, \\ Q_k &= Q_{k-1} + a'_k Q_{k-2}, \end{aligned} \qquad k = 1, 2, \ldots, n,$$

uz start (8.6.3).

8.6.3. Eulerova forma verižnih razlomaka i neki teoremi konvergencije

Ako brojeve izaberemo tako da je zbroj brojnika i nazivnika jednak jedan (osim kod prve karike), tj. ako uzmemo

$$w_1 b_1 = 1, \quad w_{k-1} w_k a_k + w_k b_k = 1, \quad k = 2, 3, \ldots,$$

onda se verižni razlomak svede na tzv. Eulerovu formu

$$R' = b_0 + \frac{\alpha_1}{1^+} \frac{\alpha_2}{(1-\alpha_2)^+} \frac{\alpha_3}{(1-\alpha_3)^+} \cdots.$$

Da bismo uspostavili vezu početnog verižnog razlomka i dobivenog verižnog razlomka u Eulerovoj formi, napišimo rekurzije za verižni razlomak u Eulerovoj formi korištenjem varijabli S_k i T_k. Promatrajmo drugu rekurziju iz (8.6.4) za Q_k (odnosno T_k, jer nas zanima ta rekurzija za verižni razlomak u Eulerovoj formi). Uz $T_{-1} = 0$, $T_0 = 1$, dobivamo

$$T_1 = T_0 + \alpha_1 T_{-1}, \quad T_k = (1 - \alpha_k)T_{k-1} + \alpha_k T_{k-2}, \quad k \geq 2.$$

Uočite da je $T_1 = 1$, pa vrijedi

$$T_k = 1$$

za sve $k \geq 0$. Time se od dvije rekurzije za računanje n-te konvergencije verižnog razlomka koristi samo ona prva, tj. vrijedi $R_k = S_k$. Iz $R_{-1} = S_{-1} = 1$, $R_0 = S_0 = b_0$ izlazi

$$R_1 = R_0 + \alpha_1 R_{-1}, \quad R_k = (1 - \alpha_k)R_{k-1} + \alpha_k R_{k-2}, \quad k \geq 2.$$

Ne mogu se svi verižni razlomci mogu svesti na Eulerov oblik. Ako i samo ako su svi $Q_k \neq 0$ (u rekurziji za početni verižni razlomak), onda se verižni razlomak može svesti na Eulerovu formu.

Nađimo vezu između originalnog i Eulerovog verižnog razlomka. Dokaz te činjenice je ponovno korištenjem indukcije. Prvo pokažimo da za sve w_k vrijedi

$$w_k = \frac{Q_{k-1}}{Q_k}, \quad k \geq 1. \tag{8.6.6}$$

Iz $w_1 b_1 = 1$, uz pretpostavku da je $b_1 \neq 0$ slijedi da je

$$w_1 = \frac{1}{b_1} = \frac{Q_0}{Q_1},$$

što je baza indukcije. Pretpostavimo da za za neki w_n vrijedi

$$w_n = \frac{Q_{n-1}}{Q_n}.$$

Iz definicione relacije za Eulerov verižni razlomak slijedi da je

$$w_n w_{n+1} a_{n+1} + w_{n+1} b_{n+1} = 1,$$

pa primjenom pretpostavke indukcije dobivamo

$$w_{n+1} = \frac{1}{w_n a_{n+1} + b_{n+1}} = \frac{1}{\frac{Q_{n-1}}{Q_n} a_{n+1} + b_{n+1}} = \frac{Q_n}{Q_{n-1} a_{n+1} + Q_n b_{n+1}} = \frac{Q_n}{Q_{n+1}},$$

čime je dokazan korak indukcije.

Iz relacije (8.6.6) i definicije

$$\alpha_k = w_{k-1} w_k a_k$$

uz dogovor $w_0 = 1$, odmah slijedi da je

$$\alpha_k = \frac{Q_{k-2}}{Q_k}\, a_k, \quad k \geq 2. \tag{8.6.7}$$

Eulerovu formu verižnog razlomka, uglavnom ćemo koristiti pri dokazivanju tvrdnji. Da bismo dokazali konvergenciju verižnog razlomka, potraban nam je jednostavan izraz za R_k, po mogućnosti neki red.

Prvo, iz rekurzija za P_k i Q_k nađimo koliko je $R_k - R_{k-1}$, a zatim i $R_k - R_{k-2}$. Za $k \geq 1$ vrijedi

$$\begin{aligned} R_k - R_{k-1} &= \frac{P_k}{Q_k} - \frac{P_{k-1}}{Q_{k-1}} = \frac{P_k Q_{k-1} - P_{k-1} Q_k}{Q_k Q_{k-1}} \\ &= \frac{(b_k P_{k-1} + a_k P_{k-2}) Q_{k-1} - P_{k-1}(b_k Q_{k-1} + a_k Q_{k-2})}{Q_k Q_{k-1}} \\ &= \frac{-a_k (P_{k-1} Q_{k-2} - P_{k-2} Q_{k-1})}{Q_k Q_{k-1}} \\ &= \frac{a_k a_{k-1} (P_{k-2} Q_{k-3} - P_{k-3} Q_{k-2})}{Q_k Q_{k-1}} = \cdots \\ &= (-1)^{k+1} \frac{a_k a_{k-1} \cdots a_1}{Q_k Q_{k-1}}. \end{aligned} \tag{8.6.8}$$

Na sličan se način dokazuje da je

$$R_k - R_{k-2} = (-1)^k \frac{b_k a_{k-1} \cdots a_1}{Q_k Q_{k-2}}, \quad k \geq 2. \tag{8.6.9}$$

Korištenjem relacije (8.6.6) možemo (8.6.8) zapisati u terminima α_k (ponovno uz pretpostavku da su svi Q_i različiti od 0). Vrijedi

$$R_k - R_{k-1} = (-1)^{k+1} \frac{a_k a_{k-1} \cdots a_1}{Q_k Q_{k-1}} = (-1)^{k+1} \alpha_k \alpha_{k-1} \cdots \alpha_1. \tag{8.6.10}$$

Rekurzivnom primjenom (8.6.10) dobivamo da je

$$R_k = (-1)^{k+1} \alpha_k \alpha_{k-1} \cdots \alpha_1 + R_{k-1} = b_0 + \sum_{i=1}^{k} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1.$$

Ako verižni razlomak konvergira, onda je

$$R = \lim_{k \to \infty} R_k = b_0 + \sum_{i=1}^{\infty} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1, \tag{8.6.11}$$

pa je

$$|R_k - R| = \Big| \sum_{i=k+1}^{\infty} (-1)^{i+1} \alpha_i \alpha_{i-1} \cdots \alpha_1 \Big|.$$

Sada možemo izreći i dokazati neke rezultate o konvergenciji verižnih razlomaka.

Teorem 8.6.1. *Ako su $a_k, b_k > 0$, tada vrijede nejednakosti*

$$R_1 > R_3 > \cdots > R_{2k-1} > \cdots,$$
$$R_0 < R_2 < \cdots < R_{2k} < \cdots$$

i

$$R_{2m-1} > R_{2k}$$

za svako m i k.

Dokaz:

Ako su $a_k, b_k > 0$, onda su to i Q_k (po rekurziji). Nakon toga, dokaz trivijalno slijedi raspisivanjem relacije (8.6.9) za parne i neparne indekse. ■

Teorem 8.6.2. *Ako su $a_k, b_k > 0$ i vrijedi $a_k \leq b_k$ i $b_k \geq \varepsilon > 0$ za $k \geq 1$, gdje je ε neka konstanta, onda je verižni razlomak (8.6.1) konvergentan.*

Dokaz:

Budući da su $a_k, b_k > 0$, onda su to i Q_k, pa iz (8.6.7) izlazi

$$\alpha_1 \alpha_2 \cdots \alpha_i = \frac{a_1 a_2 \cdots a_i}{Q_i Q_{i-1}} > 0,$$

pa je red u (8.6.11) alternirajući. Po Leibnitzovom kriteriju dovoljno je pokazati da n-ti član tog reda teži u 0 i da mu članovi opadaju po apsolutnoj vrijednosti.

Da bismo pokazali konvergenciju, dovoljno je pokazati da je $\alpha_i \leq q < 1$ (D'Alembertov kriterij konvergencije). Iz (8.6.7) dobivamo

$$\alpha_i = a_i \frac{Q_{i-2}}{Q_i} = \frac{a_i Q_{i-2}}{b_i Q_{i-1} + a_i Q_{i-2}} < \frac{a_i Q_{i-2}}{a_i Q_{i-1} + a_i Q_{i-2}} = \frac{Q_{i-2}}{Q_{i-1} + Q_{i-2}}.$$

U prethodnoj jednakosti potrebno je samo pokazati da postoji donja ograda za Q_{i-1}, što slijedi iz uvjeta $b_k \geq \varepsilon > 0$ i rekurzije za Q_i. Točnije, lako se dokazuje da je $Q_i \geq \varepsilon Q_{i-1}$, pa odatle slijedi da je

$$\alpha_i < \frac{Q_{i-2}}{(1+\varepsilon) Q_{i-2}} = \frac{1}{1+\varepsilon} < 1.$$

Također, po Leibnitzovom kriteriju, greška koju smo napravili ako R aproksimiramo s R_k manja je ili jednaka prvom odbačenom članu u redu, tj. vrijedi

$$|R_k - R| \leq \frac{a_1 a_2 \cdots a_{k+1}}{Q_k Q_{k+1}}.$$

■

8.6.4. Silazni algoritam za izvrednjavanje brojevnih verižnih razlomaka

Vratimo se još jednom na izvrednjavanje verižnih razlomaka. Prethodni je teorem dao neke ocjene o tome koliko dobro R_n aproksimira R. Zbog toga, možemo pretpostaviti da nam je unaprijed poznato koliko konvergencija trebamo da bismo dobro aproksimirali neki verižni razlomak.

Krenemo li od "silazno" od b_n, onda će sljedeća rekurzija izvrednjavati R_n. Definiramo $F_n = b_n$ (ili $F_{n+1} = \infty$) i računamo

$$F_k = b_k + \frac{a_{k+1}}{F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

na kraju ćemo dobiti

$$R_n = F_0.$$

Primijetite da silazna rekurzija u svakom koraku ima točno jedno zbrajanje i jedno dijeljenje, za razliku od uzlazne, koja u svakom koraku (u općem slučaju) ima 4 množenja i 2 zbrajanja.

Ova dva tipa rekurzija analogon su izvrednjavanja polinoma: silazna rekurzija analogon je Hornerove sheme, a uzlazna je analogon potenciranja i zbrajanja.

Dapače, može se pokazati da je silazna rekurzija za izvrednjavanje verižnih razlomaka optimalna što se broja operacija tiče, čak iako su dozvoljene transformacije koeficijenata verižnog razlomka prije početka izvrednjavanja.

8.6.5. Funkcijski verižni razlomci

Funkcijski verižni razlomci mogu se dobiti na više načina, i mogu imati više oblika. Verižne razlomke koji će varijablu imati u brojniku zvat ćemo verižni razlomci tipa I i općenito će imati oblik

$$f(x) = \beta_0 + \frac{x - x_1}{{\beta_1}^+} \frac{x - x_2}{{\beta_2}^+} \frac{x - x_3}{{\beta_3}^+} \cdots. \tag{8.6.12}$$

Funkcijski verižni razlomci mogu imati i varijablu u nazivniku, a takve ćemo verižne razlomke zvati verižni razlomci tipa II. Općenito, oni imaju oblik

$$f(x) = b_0 + \frac{a_1}{(x + b_1)^+} \frac{a_2}{(x + b_2)^+} \frac{a_3}{(x + b_3)^+} \cdots. \tag{8.6.13}$$

Izvrednjavanje verižnih razlomaka oba tipa vrlo je slično izvrednjavanju brojevnih verižnih razlomaka. Za izvrednjavanje n-te konvergencije $f_n(x)$ verižnih

razlomaka prvog tipa, možemo koristiti silazni algoritam. Stavimo $F_n = \beta_n$ (ili $F_{n+1} = \infty$), a zatim računamo

$$F_k = \beta_k + \frac{x - x_{k+1}}{F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

i na kraju je

$$f_n(x) = F_0.$$

Za izvrednjavanje n-te konvergencije verižnih razlomaka drugog tipa možemo koristiti, također, silazni algoritam. Stavimo $F_n = b_n$ (ili $F_{n+1} = \infty$), a zatim računamo

$$F_k = b_k + \frac{a_{k+1}}{x + F_{k+1}}, \quad k = (n), n-1, \ldots, 0,$$

i na kraju je

$$f_n(x) = F_0.$$

Kako dolazimo do verižnih razlomaka? Obično je nešto lakše doći do verižnih razlomaka tipa I, a zatim ih možemo pretvoriti u tip II. Ako imamo zadanu funkciju f, uobičajeno se verižni razlomak nalazi nestandardiziranim postupkom kad se funkcija zapisuje "pomoću same sebe". Da bismo to bolje objasnili, pogledajmo to na primjeru jedne funkcije.

Primjer 8.6.1. *Razvijmo u verižni razlomak prvog tipa funkciju*

$$f(x) = \sqrt{1+x}.$$

Prvo, potrebno funkciju malo drugačije zapisati. Lako se provjerava da je

$$\sqrt{1+x} = 1 + \frac{x}{1 + \sqrt{1+x}}.$$

Ako ponovimo ovaj raspis u nazivniku razlomka, dobivamo verižni razlomak

$$\sqrt{1+x} = 1 + \frac{x}{2^+} \, \frac{x}{2^+} \, \frac{x}{2^+} \, \cdots.$$

Navedimo neke od poznatih verižnih razlomaka, bez njihova "izvoda":

$$\begin{aligned} e^x &= \frac{1}{1^-}\,\frac{x}{1^+}\,\frac{x}{2^-}\,\frac{x}{3^+}\,\frac{x}{2^-}\,\frac{x}{5^+}\,\frac{x}{2^-}\,\frac{x}{7^+}\,\cdots. \\ &= 1+\frac{x}{1^-}\,\frac{x}{2^+}\,\frac{x}{3^-}\,\frac{x}{2^+}\,\frac{x}{5^-}\,\frac{x}{2^+}\,\frac{x}{7^-}\,\cdots, \end{aligned}$$

$$\ln(x+1) = \frac{x}{1^+}\,\frac{x}{2^+}\,\frac{x}{3^+}\,\frac{4x}{4^+}\,\frac{4x}{5^+}\,\frac{9x}{6^+}\,\frac{9x}{7^+}\,\frac{16x}{8^+}\,\frac{16x}{9^+}\,\cdots,$$

$$x\operatorname{tg} x = \frac{x^2}{1^-}\,\frac{x^2}{3^-}\,\frac{x^2}{5^-}\,\frac{x^2}{7^-}\,\cdots, \quad x \neq \frac{(2n+1)\pi}{2},$$

$$x\operatorname{arctg} x = \frac{x^2}{1^+}\,\frac{x^2}{3^+}\,\frac{4x^2}{5^+}\,\frac{9x^2}{7^+}\,\frac{16x^2}{9^+}\,\cdots,$$

$$x\operatorname{th} x = \frac{x^2}{1^+}\,\frac{x^2}{3^+}\,\frac{x^2}{5^+}\,\frac{x^2}{7^+}\,\cdots$$

$$x\operatorname{Arth} x = \frac{x^2}{1^-}\,\frac{x^2}{3^-}\,\frac{4x^2}{5^-}\,\frac{9x^2}{7^-}\,\frac{16x^2}{9^-}\,\cdots.$$

Svi ovi verižni razlomci su prvog tipa. Ima li koristi znati kako bi izgledao njihov drugi tip? Na primjer šesta konvergencija verižnog razlomka za $\sqrt{1+x}$ bi izgledala ovako, redom, prvi tip, racionalna funkcija, drugi tip:

$$\begin{aligned} \sqrt{1+x} &= 1+\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2^+}\,\frac{x}{2} = \frac{7x^3+56x^2+112x+64}{x^3+24x^2+80x+64} \\ &= 7+\frac{-112}{(x+20)^+}\,\frac{-24/7}{(x+8/3)^+}\,\frac{-8/63}{(x+4/3)}. \end{aligned}$$

Što vidimo? Iako drugi tip ima kompliciranije koeficijente, ima upola manje karika za izvrednjavanje, pa će to dva puta ubrzati postupak izvrednjavanja.

Kako ćemo od prethodnog verižnog razlomka prvog tipa dobiti verižni razlomak drugog tipa? Postupak se obavlja u dva koraka.

U prvom se koraku od verižnog razlomka prvog tipa dobiva racionalna funkcija. Pogledajmo silazni algoritam za izvrednjavanje verižnog razlomka prvog tipa. F_k želimo napisati kao kvocijent dva polinoma, pa možemo definirati

$$F_k = \frac{u_k}{v_k}.$$

Tada silazna rekurzija glasi

$$\frac{u_k}{v_k} = \beta_k + \frac{(x-x_{k+1})v_{k+1}}{u_{k+1}}.$$

Kao što smo to i prije radili, izjednačimo brojnike i nazivnike funkcija s obje strane. Dobivamo

$$u_k = \beta_k u_{k+1} + (x - x_{k+1})v_{k+1},$$
$$v_k = u_{k+1}.$$

Naravno v_k možemo eliminirati uvrštavanjem iz druge jednadžbe u prvu, pa dobivamo

$$u_k = \beta_k u_{k+1} + (x - x_{k+1})u_{k+2}, \quad k = n, n-1, \ldots, 0,$$

uz start $u_{n+2} = 0$, $u_{n+1} = 1$. Konačno, n-ta je konvergencija jednaka

$$f_n(x) = F_0 = \frac{u_0}{v_0} = \frac{u_0}{u_1}.$$

Da bismo iz racionalne funkcije dobili drugi tip verižnog razlomka, potrebno je koristiti silaznu rekurziju za drugi tip i uspoređivati s u_0/u_1. Iz silazne rekurzije za drugi tip izlazi

$$\frac{u_0}{u_1} = \tilde{b}_0 + \frac{a_1}{x + F_1},$$

pa možemo pisati

$$u_0 = u_1\tilde{b}_0 + a_1\widetilde{R}_1.$$

Zatim ponovimo postupak i dobivamo

$$u_1 = \widetilde{R}_1\tilde{b}_1 + a_2\widetilde{R}_2.$$

Ova rekurzija se prekida kad je stupanj polinoma 0.

Algoritam za pretvaranje racionalne funkcije u drugi tip verižnog razlomka je sljedeći. Definira se $\widetilde{R}_{-1} = u_0$ i $\widetilde{R}_0 = u_1$. Zatim se vrti petlja

$$\widetilde{R}_{k-1} = \widetilde{R}_k\tilde{b}_k + a_{k+1}\widetilde{R}_{k+1} \quad \text{za} \quad k = 0, 1, 2, \ldots$$

sve dok ne postane $\widetilde{R}_x = 1$. Pritom je

$$\tilde{b}_k = \begin{cases} b_0, & k = 0, \\ b_k + x, & k \neq 0. \end{cases}$$

9. Rješavanje nelinearnih jednadžbi

9.1. Općenito o iterativnim metodama

Računanje nultočaka nelinearnih funkcija jedan je od najčešćih zadataka primijenjene matematike. Općenito, neka je zadana funkcija

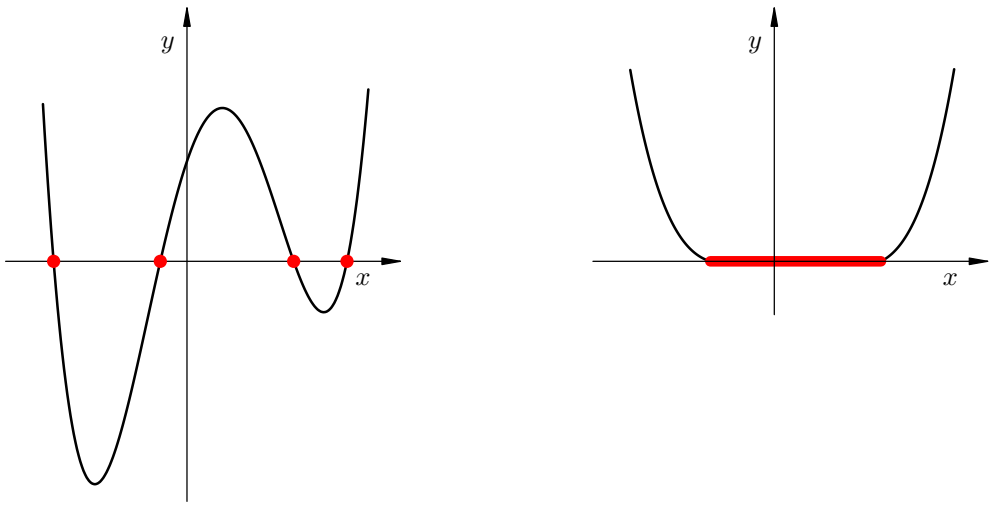
$$f : I \to \mathbb{R},$$

gdje je I neki interval. Tražimo sve one $x \in I$ za koje je

$$f(x) = 0.$$

Takvi x-evi zovu se rješenja, korijeni pripadne jednadžbe ili nultočke funkcije f.

U pravilu, pretpostavljamo da je f **neprekidna** na I i da su joj nultočke izolirane. U protivnom postojao bi problem konvergencije.



Traženje nultočki na zadanu točnost sastoji se od dvije faze.

1. Izolacije jedne ili više nultočki, tj. nalaženje intervala I unutar kojeg se nalazi bar jedna nultočka. Ovo je teži dio posla i obavlja se na temelju analize toka funkcije.
2. Iterativno nalaženje nultočke na traženu točnost.

Postoji mnogo metoda za nalaženje nultočaka nelinearnih funkcija na zadanu točnost. One se bitno razlikuju po tome hoće li uvijek konvergirati, tj. imamo li sigurnu konvergenciju ili ne i po brzini konvergencije.

Uobičajen je slučaj da brze metode nemaju sigurnu konvergenciju, dok je sporije metode imaju.

Definirajmo brzinu konvergencije metode

Definicija 9.1.1. *Niz iteracija* $(x_n, n \in \mathbb{N}_0)$ *konvergira prema točki* α *s redom konvergencije* p, $p \geq 1$ *ako vrijedi*

$$|\alpha - x_n| \leq c\,|\alpha - x_{n-1}|^p, \quad n \in \mathbb{N} \tag{9.1.1}$$

za neki $c > 0$. *Ako je* $p = 1$, *kažemo da niz konvergira linearno prema* α. *U tom je slučaju nužno da je* $c < 1$ *i obično se* c *naziva faktor linearne konvergencije.*

Relacija (9.1.1) katkad nije zgodna za linearne iterativne algoritme. Ako u (9.1.1) upotrijebimo indukciju za $p = 1$, $c < 1$, onda dobivamo da je

$$|\alpha - x_n| \leq c^n |\alpha - x_0|, \quad n \in \mathbb{N}. \tag{9.1.2}$$

Katkad će biti mnogo lakše pokazati (9.1.2) nego (9.1.1). I u slučaju (9.1.2), reći ćemo da niz iteracija konvergira linearno s faktorom c.

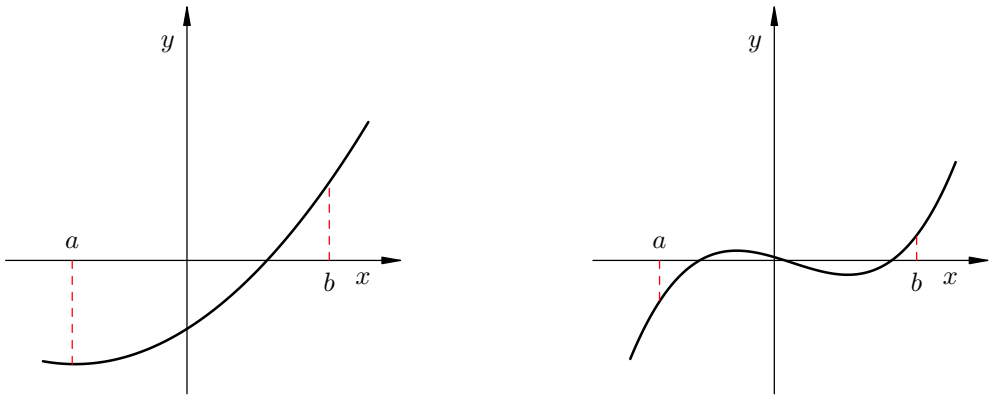
9.2. Metoda raspolavljanja (bisekcije)

Najjednostavnija metoda nalaženja nultočaka funkcije je metoda raspolavljanja. Ona funkcionira za neprekidne funkcije, ali zbog toga ima i najlošiju ocjenu pogreške.

Osnovna pretpostavka za početak algoritma raspolavljanja je **neprekidnost** funkcije f na intervalu $[a, b]$ uz pretpostavku da vrijedi

$$f(a) \cdot f(b) < 0.$$

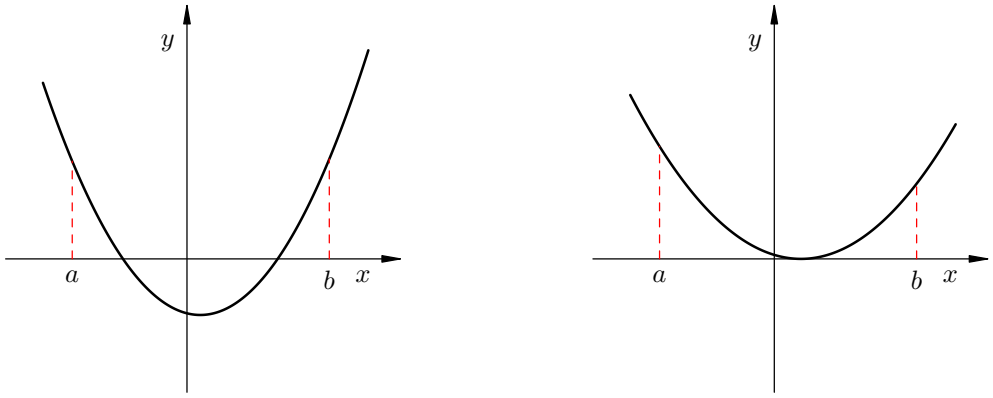
Prethodna relacija znači da funkcija f ima na intervalu $[a, b]$ **bar jednu** nultočku.



Obratno, ako je

$$f(a) \cdot f(b) > 0,$$

to **ne mora** značiti da f nema unutar $[a, b]$ nultočku. Na primjer, moglo se dogoditi da smo loše separirali nultočke i da f ima unutar $[a, b]$ paran broj nultočaka, ili nultočku parnog reda.



Dok je za prvi primjer s prethodne slike lako, boljom separacijom nultočki postići $f(a) \cdot f(b) < 0$, za drugi je primjer to nemoguće! Dakle, nultočke parnog reda nemoguće je naći metodom bisekcije.

Ako vrijede startne pretpostavke metode, metoda raspolavljanja konvergirat će prema nekoj nultočki iz intervala $[a, b]$.

Algoritam raspolavljanja je vrlo jednostavan. Označimo s α pravu nultočku funkcije, a zatim s $a_0 := a$, $b_0 := b$ i x_0 polovište $[a_0, b_0]$, tj.

$$x_0 = \frac{a_0 + b_0}{2}.$$

U n-tom koraku algoritma konstruiramo interval $[a_n, b_n]$ kojemu je duljina polovina duljine prethodnog intervala, ali tako da je nultočka ostala unutar intervala $[a_n, b_n]$.

Konstrukcija intervala $[a_n, b_n]$ sastoji se u raspolavljanju intervala $[a_{n-1}, b_{n-1}]$ točkom x_{n-1} i to tako da je

$$\begin{aligned} a_n = x_{n-1}, b_n = b_{n-1} \quad &\text{ako je} \quad f(a_{n-1}) \cdot f(x_{n-1}) > 0, \\ a_n = a_{n-1}, b_n = x_{n-1} \quad &\text{ako je} \quad f(a_{n-1}) \cdot f(x_{n-1}) < 0. \end{aligned}$$

Postupak zaustavljamo kad je

$$|\alpha - x_n| \leq \varepsilon.$$

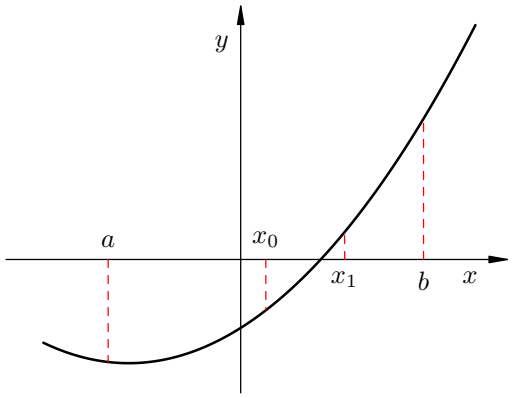
Pitanje je kako ćemo znati da je prethodna relacija ispunjena ako ne znamo α? Jednostavno! Budući da je x_n polovište intervala $[a_n, b_n]$, a $\alpha \in [a_n, b_n]$, onda je

$$|\alpha - x_n| \leq b_n - x_n,$$

pa je dovoljno staviti zahtjev

$$b_n - x_n \leq \varepsilon.$$

Grafički, metoda raspolavljanja izgleda ovako



Algoritam za metodu raspolavljanja je sljedeći.

Algoritam 9.2.1. (Metoda raspolavljanja)

$x := (a + b)/2;$
while $b - x > \varepsilon$ **do**
  begin;
  if $f(x) * f(b) < 0.0$ **then**
    $a := x$
  else
    $b := x;$
  $x := (a + b)/2;$
  end;
{ Na kraju je $x \approx \alpha$. }

Iz konstrukcije metode lako se izvodi pogreška n-te aproksimacije nultočke α. Vrijedi

$$|\alpha - x_n| \leq b_n - x_n = \frac{1}{2}(b_n - a_n) = \frac{1}{2^2}(b_{n-1} - a_{n-1}) = \cdots = \frac{1}{2^{n+1}}(b-a). \quad (9.2.1)$$

Primijetite da je

$$\frac{b-a}{2} = b - x_0,$$

pa bismo korištenjem te relacije (9.2.1) mogli pisati kao

$$|\alpha - x_n| \leq \frac{1}{2^n}(b - x_0).$$

Ova relacija podsjeća na (9.1.2), ali zdesna se nigdje ne pojavljuje $|\alpha - x_0|$. Ipak desna strana daje nam naslutiti da će konvergencija biti dosta spora.

Relacija (9.2.1) omogućava sa unaprijed odredimo koliko je koraka raspolavljanja potrebno da bismo postigli točnost ε. Da bismo postigli da je $|\alpha - x_n| \leq \varepsilon$, dovoljno je zahtijevati da je

$$\frac{1}{2^{n+1}}(b-a) \leq \varepsilon.$$

Množenjem prethodne jednadžbe s 2^{n+1} i dijeljenjem s ε dobivamo

$$\frac{b-a}{\varepsilon} \leq 2^{n+1},$$

a zatim logaritmiranjem dobivamo

$$\log(b-a) - \log\varepsilon \leq (n+1)\log 2,$$

odnosno

$$n \geq \frac{\log(b-a) - \log\varepsilon}{\log 2} - 1, \quad n \in \mathbb{N}_0.$$

Ako je funkcija f još i klase $C^1[a,b]$, tj. ako f ima i neprekidnu prvu derivaciju, može se dobiti dinamička ocjena za udaljenost aproksimacije nultočke od prave nultočke.

Po Teoremu srednje vrijednosti za funkciju f imamo

$$f(x_n) = f(\alpha) + f'(\xi)(x_n - \alpha),$$

pri čemu je ξ između x_n i α. Prvo iskoristimo da je α nultočka, tj. $f(\alpha) = 0$, a zatim uzmemo apsolutne vrijednosti obje strane. Dobivamo

$$|f(x_n)| = |f'(\xi)|\,|\alpha - x_n|. \quad (9.2.2)$$

Primijetite da je

$$|f'(\xi)| \geq m_1, \quad m_1 = \min_{x\in[a,b]} |f'(x)|.$$

Ako je $m_1 > 0$, uvrštavanjem prethodne ocjene u (9.2.2) izlazi

$$|\alpha - x_n| \leq \frac{|f(x_n)|}{m_1}.$$

Drugim riječima, ako želimo da je $|\alpha - x_n| \leq \varepsilon$, dovoljno je zahtijevati da je

$$\frac{|f(x_n)|}{m_1} \leq \varepsilon,$$

odnosno da vrijedi

$$|f(x_n)| \leq m_1\varepsilon.$$

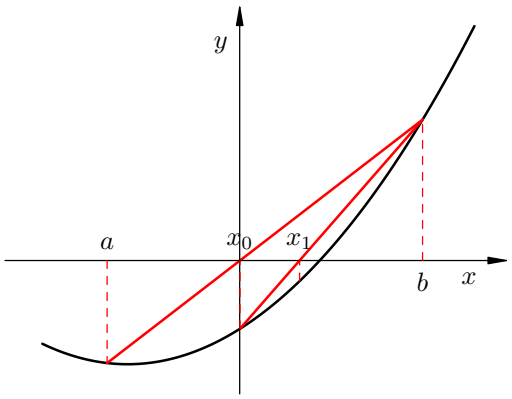
9.3. Regula falsi (metoda pogrešnog položaja)

U prethodnom poglavlju opisali smo metodu raspolavljanja, koja ima sigurnu konvergenciju, ali je vrlo spora. Prirodan je pokušaj ubrzavanja te metode je *regula falsi*. Konstruirat ćemo metodu koja će, ponovno biti konvergentna, čim se nultočka nalazi unutar $[a, b]$.

Pretpostavimo da je funkcija $f : [a, b] \to \mathbb{R}$ neprekidna na $[a, b]$ i da vrijedi

$$f(a) \cdot f(b) < 0.$$

Aproksimirajmo funkciju f pravcem koji prolazi točkama $(a, f(a))$, $(b, f(b))$. Nultočku α tada možemo aproksimirati nultočkom tog pravca, točkom x_0. Nakon toga, pomaknemo ili točku a ili točku b u x_0, ali tako da je unutar novodobivenog intervala ostala nultočka. Postupak ponavljamo sve dok nismo postigli željenu točnost.



Točka x_0 dobiva se jednostavno iz jednadžbe pravca, pa je

$$x_0 = b - f(b)\,\frac{b-a}{f(b)-f(a)}. \tag{9.3.1}$$

Postoji nekoliko ozbiljnih problema s ovom metodom, iako je aproksimacija pravcem i zatvaranje nultočke u određeni interval sasvim dobra ideja.

Izvedimo red konvergencije metode. Iskoristimo relaciju (9.3.1) za x_0, pomnožimo je s -1 i dodajmo α s obje strane. Odatle, uz oznaku ($f[a,b]$ je prva podijeljena razlika)

$$f[a,b] = \frac{f(b)-f(a)}{b-a},$$

izlazi

$$\begin{aligned}
\alpha - x_0 &= \alpha - b + \frac{f(b)}{f[a,b]} = (\alpha-b)\Big(1 + \frac{f(b)}{(\alpha-b)f[a,b]}\Big) \\
&= (\alpha-b)\left(1 + \frac{f(b)-f(\alpha)}{(\alpha-b)f[a,b]}\right) = (\alpha-b)\left(1 + (b-\alpha)\,\frac{f[b,\alpha]}{(\alpha-b)f[a,b]}\right) \\
&= (\alpha-b)\left(1 - \frac{f[b,\alpha]}{f[a,b]}\right) = (\alpha-b)\,\frac{f[a,b]-f[b,\alpha]}{f[a,b]} \\
&= -(\alpha-b)\,(\alpha-a)\frac{f[a,b,\alpha]}{f[a,b]},
\end{aligned}$$

pri čemu je po definiciji $f[a,b,\alpha]$ druga podijeljena razlika

$$f[a,b,\alpha] = \frac{f[b,\alpha]-f[a,b]}{\alpha-a}.$$

Ako je f klase $C^1[a,b]$, onda po Teoremu srednje vrijednosti imamo

$$f[a,b] = f'(\xi), \quad \xi \in [a,b].$$

Na sličan način, ako je f klase $C^2[a,b]$, lako je dokazati da je

$$f[a,b,\alpha] = \frac{1}{2}\,f''(\zeta),$$

gdje se ζ nalazi između minimima i maksimuma vrijednosti a, b, α. Iskoristimo li te dvije relacije, za funkcije klase $C^2[a,b]$ dobivamo sljedeću ocjenu

$$\alpha - x_0 = -(\alpha-b)\,(\alpha-a)\frac{f''(\zeta)}{2f'(\xi)}. \tag{9.3.2}$$

Da bismo pojednostavnili analizu, pretpostavimo da je $f''(\alpha) \neq 0$ i α je jedini korijen unutar $[a,b]$. Također, pretpostavimo da je $f''(a) \geq 0$ za sve $x \in [a,b]$. Razlikujemo dva slučaja:

Slučaj $f'(x) > 0$.

U tom je slučaju f konveksna rastuća funkcija, a spojnica točaka $(a, f(a))$ i $(b, f(b))$ se uvijek nalazi **iznad** funkcije f. Uvrštavanjem podataka o prvoj i drugoj derivaciji u (9.3.2), dobivamo da je desna strana (9.3.2) veća od 0, tj. $\alpha > x_0$, pa će se u sljedećem koraku pomaknuti a. Isto će se dogoditi u svim narednim koracima. Drugim riječima, α neprestano ostaje desno od aproksimacija x_n. Promatramo li (9.3.2), to znači da je b fiksan, pa za proizvoljnu iteraciju x_n dobivamo

$$\alpha - x_n = -(\alpha - b)\,(\alpha - a_n)\frac{f''(\zeta_n)}{2f'(\xi_n)}.$$

Uzimanjem apsolutnih vrijednosti zdesna i slijeva, slijedi da je u tom slučaju konvergencija *regule falsi* linearna.

Pogled na sličnu ocjenu za metodu bisekcije, odmah kaže da ne bi trebalo biti preteško konstruirati primjere kad je metoda bisekcije brža no *regula falsi*.

Slučaj $f'(x) < 0$.

U ovom slučaju je aproksimacija nultočke uvijek desno od α, a uvijek se pomiče b. Analiza ovog slučaja vrlo je slična prethodnom.

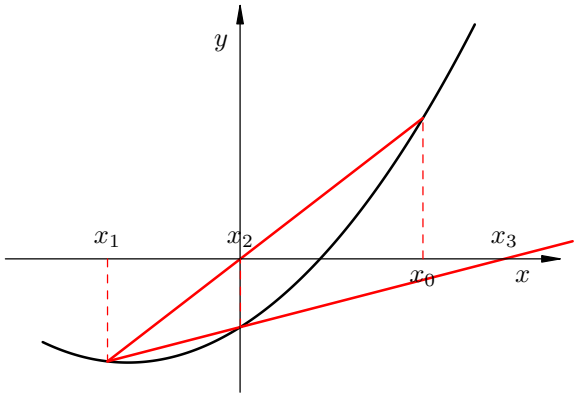
9.4. Metoda sekante

Ako graf funkcije f aproksimiramo sekantom, slično kao kod *regule falsi*, samo ne zahtijevamo da nultočka funkcije f ostane "zatvorena" unutar posljednje dvije iteracije, dobili smo metodu sekante. Time smo izgubili svojstvo sigurne konvergencije, ali se nadamo da će metoda, kad konvergira konvergirati brže nego *regula falsi*.

Počinjemo s dvije početne točke x_0 i x_1 i povlačimo sekantu kroz $(x_0, f(x_0))$, $(x_1, f(x_1))$. Ta sekanta siječe os x u točki x_2. Postupak nastavljamo povlačenjem sekante kroz posljednje dvije točke $(x_1, f(x_1))$ i $(x_2, f(x_2))$. Formule za metodu sekante dobivaju se iteriranjem početne formule za *regulu falsi*, tako da dobivamo

$$x_{n+1} = x_n - f(x_n)\,\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}. \tag{9.4.1}$$

Grafički to izgleda ovako.



Primijetite da je treće iteracija izašla izvan početnog intervala, pa metoda sekante ne mora konvergirati. Jednako tako, da smo "prirodno" numerirali prve dvije točke, tako da je $x_0 < x_1$, imali bismo konvergenciju prema rješenju.

Iskoristimo li ocjenu (9.3.2) za svaki n, dobit ćemo rad konvergencije metode sekante, uz odgovarajuće pretpostavke. Imamo

$$\alpha - x_{n+1} = -(\alpha - x_n)\,(\alpha - x_{n-1})\frac{f''(\zeta_n)}{2f'(\xi_n)}. \tag{9.4.2}$$

Teorem 9.4.1. *Neka su f, f' i f'' neprekidne za sve x u nekom intervalu koji sadrži jednostruku nultočku α ($f'(\alpha) \neq 0$). Ako su početne aproksimacije x_0 i x_1 izabrane dovoljno blizu α, niz iteracija x_n konvergirat će prema α s redom konvergencije p, gdje je*

$$p = \frac{1+\sqrt{5}}{2} \approx 1.618.$$

Dokaz:

Budući da je $f'(\alpha) \neq 0$, u nekoj okolini nultočke α, $I = [\alpha - \varepsilon, \alpha + \varepsilon]$, $\varepsilon > 0$, možemo definirati

$$M = \frac{\max\limits_{x\in I} |f''(x)|}{2\min\limits_{x\in I} |f'(x)|}.$$

Za sve $x_0, x_1 \in I$, korištenjem (9.4.2), dobivamo

$$|\alpha - x_2| \leq |\alpha - x_1|\,|\alpha - x_0|\,M.$$

Da bismo skratili zapis, označimo s $e_n = \alpha - x_n$ grešku n-te iteracije (aproksimacije nultočke). Množenjem prethodne nejednakosti s M dobivamo

$$M|e_2| \leq M|e_1|\,M|e_0|.$$

Nadalje, pretpostavimo da su x_0 i x_1 izabrani tako da je

$$\delta = \max\{M|e_0|, M|e_1|\} < 1.$$

Odatle odmah slijedi da je

$$M|e_2| \leq \delta^2 < \delta.$$

Odatle zaključujemo da je

$$|e_2| < \frac{\delta}{M} = \max\{|e_0|, |e_1|\} \leq \varepsilon,$$

odnosno

$$x_2 \in [\alpha - \varepsilon, \alpha + \varepsilon] = I.$$

Primijenimo li induktivno taj argument, dobivamo

$$M|e_3| \leq M|e_2|M|e_1| \leq \delta^2 \cdot \delta = \delta^3$$
$$M|e_4| \leq M|e_3|M|e_2| \leq \delta^5.$$

Općenito, ako je

$$M|e_{n-1}| \leq \delta^{q_{n-1}}, \quad M|e_n| \leq \delta^{q_n},$$

onda je

$$M|e_{n+1}| \leq M|e_n|M|e_{n-1}| \leq \delta^{q_n+q_{n-1}} = \delta^{q_{n+1}},$$

pa je

$$q_{n+1} = q_n + q_{n-1}, \quad n \geq 1,$$

s $q_0 = q_1 = 1$. Prethodna rekurzija je rekurzija za Fibonaccijeve brojeve i lako se računa njeno eksplicitno rješenje – tj. dovoljno je riješiti diferencijsku jednadžbu

$$q_{n+1} - q_n - q_{n-1} = 0,$$

uz zadane početne $q_0 = q_1 = 1$.

Karakteristična jednadžba je

$$k^2 - k - 1 = 0,$$

pa su njena rješenja

$$k_{1,2} = \frac{1 \pm \sqrt{5}}{2}.$$

Označimo li

$$r_0 = \frac{1 + \sqrt{5}}{2}, \quad r_1 = \frac{1 - \sqrt{5}}{2},$$

onda je opće rješenje te diferencijske jednadžbe

$$q_n = c_0 r_0^n + c_1 r_1^n.$$

Konstante c_0 i c_1 određujemo iz početnih uvjeta. Dobivamo

$$1 = q_0 = c_0 + c_1$$
$$1 = q_1 = c_0 r_0 + c_1 r_1.$$

Rješavanjem ovog para jednadžbi, dobivamo

$$c_0 = \frac{1}{\sqrt{5}} r_0, \quad c_1 = -\frac{1}{\sqrt{5}} r_1,$$

pa je

$$q_n = \frac{1}{\sqrt{5}} (r_0^{n+1} - r_1^{n+1}), \quad n \geq 0.$$

Budući da je

$$r_0 \approx 1.618, \quad r_1 \approx -0.618,$$

onda za velike n $r_1^{n+1} \to 0$, pa je

$$q_n \approx \frac{1}{\sqrt{5}} (1.618)^{n+1}.$$

Vratimo se na e_n. Ovim smo pokazali da je

$$|e_n| \leq \frac{1}{M} \delta^{q_n}, \quad n \geq 0.$$

Budući da $q_n \to \infty$ za $n \to \infty$, dobivamo da $x_n \to \alpha$.

Ovaj "kvazidokaz" (jer su svugdje gornje ograde) daje nam samo ideju o redu konvergencije, koji je zaista $p = r_0$, ali je pravi dokaz mnogo teži. ■

Kod metode sekante postoji nekoliko problema. Prvi je da može divergirati ako početne aproksimacije nisu dobro odabrane.

Drugi problem koji se može javiti je kraćenje u kvocijentu

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

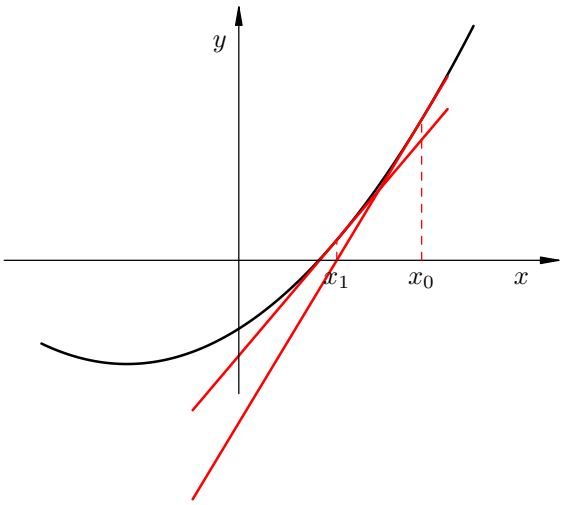
kad $x_n \to \alpha$. Osim toga, budući da iteracije ne "zatvaraju" nultočku nije lako reći kad treba zaustaviti iterativni proces.

Konačno, primijetite da je za svaku iteraciju metode sekante potrebno samo jednom izvrednjavati funkciju f i to u točki x_n, jer $f(x_{n-1})$ čuvamo od prethodne iteracije.

9.5. Metoda tangente (Newtonova metoda)

Ako graf funkcije f umjesto sekantom, aproksimiramo tangentom, dobili smo metodu tangente ili Newtonovu metodu. Slično kao i kod sekante, time smo izgubili svojstvo sigurne konvergencije, ali se nadamo da će metoda brzo konvergirati.

Pretpostavimo da je zadana početna točka x_0. Ideja metode je povući tangentu u točki $(x_0, f(x_0))$ i definirati novu aproksimaciju x_0 u točki gdje ona siječe os x.



Geometrijski izvod je jednostavan. U točki x_n napiše se jednadžba tangente i pogleda se gdje siječe os x. Jednadžba tangente je

$$y - f(x_n) = f'(x_n)(x - x_n),$$

odakle izlazi da je nova aproksimacija $x_{n+1} := x$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Primijetite da je prethodna formula usko vezana uz metodu sekante, jer je

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Do Newtonove metode može se doći i na drugačiji način. Pretpostavimo li da je funkcija f dva puta neprekidno derivabilna (na nekom području oko α), onda je možemo razviti u Taylorov red oko x_n do uključivo prvog člana. Dobivamo

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(\xi_n)}{2}(x - x_n)^2,$$

pri čemu je ξ_n između x i x_n. Uvrštavanjem $x = \alpha$, dobivamo

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{f''(\xi_n)}{2}(\alpha - x_n)^2.$$

Premještanjem, uz pretpostavku $f'(x_n) \neq 0$, izlazi

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} - (\alpha - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}.$$

Primijetite da prva dva člana zdesna daju x_{n+1}, pa dobivamo

$$\alpha - x_{n+1} = -(\alpha - x_n)^2 \frac{f''(\xi_n)}{2f'(x_n)}. \tag{9.5.1}$$

Iz (9.5.1), odmah čitamo da je Newtonova metoda, kad konvergira kvadratično konvergentna. Ipak, treba biti oprezan, jer takav zaključak vrijedi samo ako $f'(x_n)$ ne teži u 0 tijeku cijelog procesa, tj. ako je $f'(\alpha) \neq 0$ (drugim riječima, ako je nultočka jednostruka).

Slično, kao kod metode sekante, možemo dokazati sljedeći teorem o konvergenciji Newtonove metode.

Teorem 9.5.1. *Neka su f, f' i f'' neprekidne za sve x u nekom intervalu koji sadrži jednostruku nultočku α ($f'(\alpha) \neq 0$). Ako je početna aproksimacija x_0 izabrana dovoljno blizu α, niz iteracija x_n konvergirat će prema α s redom konvergencije $p = 2$. Čak štoviše, vrijedi*

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

Dokaz:

Izaberimo interval $I = [\alpha - \varepsilon, \alpha + \varepsilon]$ i neka je

$$M = \frac{\max\limits_{x\in I} |f''(x)|}{2 \min\limits_{x\in I} |f'(x)|}.$$

Za sve $x_0 \in I$, korištenjem (9.5.1), dobivamo

$$|\alpha - x_1| \leq M|\alpha - x_0|^2,$$

odnosno

$$M|\alpha - x_1| \leq (M|\alpha - x_0|)^2.$$

Izaberimo $|\alpha - x_0| \leq \varepsilon$ i $M|\alpha - x_0| < 1$. Tada je

$$M|\alpha - x_1| \leq M|\alpha - x_0|,$$

što pokazuje da je

$$|\alpha - x_1| \leq |\alpha - x_0| \leq \varepsilon.$$

Primjenom istog argumenta, induktivno dobivamo

$$|\alpha - x_n| \leq \varepsilon, \quad M|\alpha - x_n| < 1$$

za sve $n \geq 1$. Da bismo pokazali konvergenciju iskoristimo (9.5.1). Imamo

$$|\alpha - x_{n+1}| \leq M|\alpha - x_n|^2, \quad M|\alpha - x_{n+1}| \leq (M|\alpha - x_n|)^2,$$

i induktivno

$$M|\alpha - x_n| \leq (M|\alpha - x_0|)^{2^n}, \quad |\alpha - x_n| \leq \frac{1}{M}(M|\alpha - x_0|)^{2^n}.$$

Budući da je $M|\alpha - x_0| < 1$, to pokazuje da $x_n \to \alpha$ za $n \to \infty$.

Budući da u (9.5.1) ξ_n leži između x_n i α, onda mora biti $\xi_n \to \alpha$ za $n \to \infty$. Zbog toga je

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\lim_{n\to\infty} \frac{f''(\xi_n)}{2f'(x_n)} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

■

Jednostavnim riječima, ovaj teorem daje dovoljne uvjete za tzv. **lokalnu** konvergenciju Newtonove metode prema jednostrukoj nultočki. Lokalnost se odnosi na to da početna aproksimacija mora biti dovoljno blizu nultočke

$$|\alpha - x_0| \leq \varepsilon.$$

Veličina ε određena je drugim uvjetom $M|\alpha - x_0| < 1$ koji daje sigurnu konvergenciju. Naravno, tada je

$$|\alpha - x_0| < \frac{1}{M},$$

pa bi ispalo da treba uzeti $\varepsilon = 1/M$. To, nažalost, ne mora vrijediti, jer M općenito ovisi o ε. Ipak, u nekim situacijama možemo iskoristiti sličan uvjet za osiguranje konvergencije Newtonove metode.

Pretpostavimo da smo locirali nultočku funkcije f u segmentu $[a, b]$ i znamo da je f klase C^2 na tom segmentu. Neka je

$$M_2 = \max_{x\in[a,b]} |f''(x)|, \quad m_1 = \min_{x\in[a,b]} |f'(x)|.$$

Ako je f još i strogo monotona na $[a, b]$, onda je $m_1 > 0$ (a vrijedi i obrat). Tada f ima jedinstvenu jednostruku nultočku α u $[a, b]$. Umjesto "lokalnog" M, izračunamo "globalnu" veličinu

$$M' := \frac{M_2}{2m_1}.$$

Ako vrijedi

$$\frac{b-a}{2} < \frac{1}{M'},$$

onda možemo uzeti $\varepsilon = (b-a)/2$, a startna točka je polovište intervala $x_0 := (a+b)/2$. Zbog

$$|x_0 - \alpha| \leq \varepsilon < 1/M',$$

imamo sigurnu konvergenciju iteracija prema nultočki. Ako vrijedi i jači uvjet

$$b - a < \frac{1}{M'},$$

onda bilo koja startna točka $x_0 \in [a, b]$ daje sigurnu konvergenciju.

Naravno, to možemo iskoristiti samo ako imamo dovoljno informacija o funkciji f da možemo izračunati M', odnosno M_2 i m_1. Umjesto M_2, možemo uzeti i neku gornju ogradu za M_2, a umjesto m_1, neku pozitivnu donju ogradu za m_1.

Ove dvije veličine M_2 i m_1 daju i lokalne ocjene greške iteracija u Newtonovoj metodi, uz uvjet da su sve iteracije u $[a, b]$. Iz ranije relacije (9.5.1)

$$\alpha - x_n = -\frac{f''(\xi_{n-1})}{2f'(x_{n-1})}\,(\alpha - x_{n-1})^2,$$

gdje je ξ_{n-1} između α i x_{n-1}, odmah slijedi

$$|\alpha - x_n| \leq \frac{M_2}{2m_1}\,(\alpha - x_{n-1})^2.$$

Ova ocjena nije naročito korisna za praksu, jer α ne znamo. Uočite da smo sličnu ocjenu već imali u prethodnom teoremu, samo s M umjesto M'.

Za dvije susjedne iteracije u Newtonovoj metodi također vrijedi veza preko Taylorove formule

$$f(x_n) = f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) + \frac{f''(\xi_{n-1})}{2}(x_n - x_{n-1})^2,$$

pri čemu je ξ_{n-1} između x_{n-1} i x_n. Po definiciji iteracija u Newtonovoj metodi vrijedi i

$$f(x_{n-1}) + f'(x_{n-1})(x_n - x_{n-1}) = 0,$$

pa je

$$f(x_n) = \frac{f''(\xi_{n-1})}{2}(x_n - x_{n-1})^2.$$

Koristeći pretpostavku x_{n-1}, $x_n \in [a, b]$, dobivamo

$$|f(x_n)| \leq \frac{M_2}{2}(x_n - x_{n-1})^2.$$

Kao i kod metode bisekcije, ako je $m_1 > 0$, iz (9.2.2) slijedi ocjena

$$|\alpha - x_n| \leq \frac{|f(x_n)|}{m_1}.$$

Kombinacijom ovih ocjena dobivamo

$$|\alpha - x_n| \leq \frac{M_2}{2m_1}(x_n - x_{n-1})^2,$$

što se može iskoristiti u praksi. Ako je ε tražena točnost, onda test

$$\frac{M_2}{2m_1}(x_n - x_{n-1})^2 \leq \varepsilon$$

garantira da je $|\alpha - x_n| \leq \varepsilon$, do na greške zaokruživanja. Naravno, možemo koristiti i raniji test

$$\frac{|f(x_n)|}{m_1} \leq \varepsilon.$$

U ovim ocjenama greške koristili smo pretpostavku da je f strogo monotona na $[a, b]$. Ako i druga derivacija ima fiksni predznak na tom intervalu, onda možemo dobiti i **globalnu** konvergenciju Newtonove metode.

Teorem 9.5.2. *Neka je $f \in C^2[a, b]$ i $f(a) \cdot f(b) < 0$. Ako f' i f'' nemaju nultočku u $[a, b]$, tj. ako f' i f'' imaju fiksni predznak na $[a, b]$, onda Newtonova metoda konvergira prema (jedinstvenoj jednostrukoj) nultočki α funkcije f, za svaku startnu aproksimaciju $x_0 \in [a, b]$ za koju vrijedi*

$$f(x_0) \cdot f''(x_0) > 0.$$

Dokaz:

Pretpostavimo, na primjer, da je $f' > 0$ i $f'' > 0$ na cijelom $[a, b]$. Tada, jer f raste, mora biti $f(a) < 0$ i $f(b) > 0$. Zbog $f'' > 0$, startna aproksimacija mora zadovoljavati $f(x_0) > 0$. U praksi možemo uzeti $x_0 = b$, jer je to jedina točka za koju sigurno znamo da vrijedi $f(x_0) > 0$.

Neka je $(x_n, n \in \mathbb{N}_0)$, niz iteracija generiran Newtonovom metodom iz startne točke x_0 za koju je $f(x_0) > 0$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Za početak, znamo da je $x_0 > \alpha$. Tvrdimo da je $\alpha < x_n \leq x_0$ za svaki $n \in N_0$. Dokaz ide indukcijom, a bazu već imamo. Pretpostavimo da je $\alpha < x_n \leq x_0$. Onda je $f(x_n) > 0$ i $f'(x_n) > 0$, pa je

$$x_{n+1} < x_n \leq x_0,$$

što pokazuje da (x_n) monotono pada. Iz Taylorove formule je

$$0 = f(\alpha) = f(x_n) + f'(x_n)(\alpha - x_n) + \frac{f''(\xi_n)}{2}(\alpha - x_n)^2,$$

pri čemu je $\xi_n \in (\alpha, x_n) \subset [a, b]$. Zbog toga je $f''(\xi_n) > 0$, pa je

$$f(x_n) + f'(x_n)(\alpha - x_n) < 0,$$

odakle slijedi

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} > \alpha.$$

Dakle, niz (x_n) je odozdo ograničen s α i monotono pada pa postoji limes

$$\alpha' := \lim_{n\to\infty} x_n.$$

Odmah znamo i da je $\alpha \leq \alpha' \leq x_0$, tj. $\alpha' \in [a, b]$. Prijelazom na limes u formuli za Newtonove iteracije dobivamo

$$\alpha' = \alpha' - \frac{f(\alpha')}{f'(\alpha')},$$

odakle, koristeći $f'(\alpha') \neq 0$, slijedi $f(\alpha') = 0$. No, znamo da f ima jedinstvenu nultočku α u $[a, b]$ pa mora biti $\alpha = \alpha'$.

Preostala tri slučaja za predznake prve i druge derivacije se dokazuju potpuno analogno. ■

Uvjet $f(x_0) \cdot f''(x_0) > 0$ na izbor startne točke u prethodnom teoremu ima vrlo jednostavnu geometrijsku interpretaciju. Ako pogledamo graf funkcije f na $[a, b]$, startnu točku x_0 treba odabrati na "strmijoj" strani funkcije.

Primijetite da računanje u Newtonovoj metodi, iako ima veći red konvergencije nego sekanta, može trajati dulje (naravno uz istu točnost rezultata). Objašnjenje leži u činjenici da se za svaki korak Newtonove metode mora izračunati i vrijednost funkcije i vrijednost derivacije u točki. Ako se derivacija komplicirano računa, sekanta će biti brža.

Prethodni teoremi daju samo dovoljne uvjete konvergencije pojedinih iterativnih metoda. U praktičnom računanju često imamo samo interval $[a, b]$ u kojem smo locirali nultočku funkcije f, a **nemamo** dodatne informacije o funkciji f iz kojih bismo mogli izvući zaključak o konvergenciji bržih iterativnih metoda. Zbog toga se ove metode katkad kombiniraju s metodom bisekcije na sljedeći način. Prvo izračunamo novu iteraciju po bržoj metodi i ako ona ostaje u trenutnom intervalu, onda ju prihvaćamo i s njom nastavljamo iteracije i skraćujemo interval. U protivnom, radimo korak bisekcije za smanjivanje intervala.

9.6. Metoda jednostavne iteracije

Pretpostavimo da tražimo α, rješenje jednadžbe

$$x = g(x). \tag{9.6.1}$$

Definiramo jednostavnu iteracionu funkciju (iteracionu funkciju koja "pamti" samo jednu prethodnu točku) s

$$x_{n+1} = g(x_n), \quad n \geq 0,$$

uz x_0 kao početnu aproksimaciju za α. Primijetite da Newtonova metoda pripada klasi jednostavnih iteracija, jer je

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Rješenja, tj. točke za koje je $x = g(x)$, zovu se **fiksne točke** od g. Uobičajeno, mi smo zainteresirani $f(x) = 0$, pa taj problem treba reformulirati na problem (9.6.1). Postoji mnogo načina za tu reformulaciju.

Primjer 9.6.1. *Reformulirajmo problem*

$$x^2 - a = 0, \quad a > 0$$

u oblik (9.6.1). Na primjer, to možemo napraviti na jedan od sljedećih načina:

1. $x = x^2 + x - a$, *ili općenitije* $x = x + c(x^2 - a)$ *za neki* $c \neq 0$,
2. $x = a/x$,
3. $x = 0.5\,(x + a/x)$.

Prirodno je pitanje kako se različite jednostavne iteracije ponašaju. Odgovor ćemo dobiti nizom sljedećih tvrdnji.

Lema 9.6.1. *Neka je funkcija g neprekidna na intervalu $[a, b]$ i neka je*

$$a \leq g(x) \leq b, \quad \forall x \in [a, b],$$

u oznaci $g([a, b]) \subseteq [a, b]$. Tada jednostavna iteracija $x = g(x)$ ima bar jedno rješenje na $[a, b]$.

Dokaz:

Za neprekidnu funkciju $g(x) - x$ na intervalu $[a, b]$ vrijedi

$$g(a) - a \geq 0, \quad g(b) - b \leq 0.$$

Drugim riječima, funkcija $g(x) - x$ je promijenila predznak na intervalu $[a, b]$, a to može samo prolaskom kroz nultočku (neprekidna je!). ■

Lema 9.6.2. *Neka je funkcija g neprekidna na $[a,b]$ i neka je*

$$g([a,b]) \subseteq [a,b].$$

Nadalje, pretpostavimo da postoji konstanta λ, $0 < \lambda < 1$, takva da vrijedi

$$|g(x) - g(y)| \leq \lambda\,|x - y|, \quad \forall x, y \in [a,b].$$

Tada $x = g(x)$ ima jedinstveno rješenje α unutar $[a,b]$. Također, niz iteracija

$$x_n = g(x_{n-1}), \quad n \geq 1$$

konvergira prema α za proizvoljni $x_0 \in [a,b]$.

Dokaz:

Prema prethodnoj lemi, postoji bar jedno rješenje $\alpha \in [a,b]$. Pokažimo da ne postoji više od jednog rješenja. Da bismo to pokazali, pretpostavimo suprotno, tj. postoje barem dva rješenja. Uzmimo bilo koja dva od tih rješenja i nazovimo ih α i β iz $[a,b]$. Budući da su to rješenja, vrijedi

$$g(\alpha) = \alpha \quad \text{i} \quad g(\beta) = \beta.$$

Po pretpostavci, uvažavajući prethodne jednakosti, dobivamo

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| \leq \lambda|\alpha - \beta|,$$

ili drugim riječima

$$(1 - \lambda)\,|\alpha - \beta| \leq 0.$$

Budući da je $1 - \lambda > 0$, mora biti $\alpha = \beta$.

Dokažimo još konvergenciju jednostavnih iteracija za proizvoljnu startnu točku $x_0 \in [a,b]$. Prvo, uočimo da $x_{n-1} \in [a,b]$ povlači da je $x_n = g(x_{n-1}) \in [a,b]$. Nadalje, vrijedi

$$|\alpha - x_n| = |g(\alpha) - g(x_{n-1})| \leq \lambda|\alpha - x_{n-1}|,$$

odnosno indukcijom po n dobivamo

$$|\alpha - x_n| \leq \lambda^n|\alpha - x_0|, \quad n \geq 1.$$

Ako pustimo $n \to \infty$, onda $\lambda^n \to 0$, pa vrijedi $x_n \to \alpha$. ■

Ako je g derivabilna na $[a,b]$, onda je po Teoremu srednje vrijednosti

$$g(x) - g(y) = g'(\xi)(x - y), \quad \xi \text{ između } x \text{ i } y$$

za sve x, $y \in [a,b]$. Definiramo

$$\lambda = \max_{x \in [a,b]} |g'(x)|, \tag{9.6.2}$$

onda možemo pisati

$$|g(x) - g(y)| = \lambda|x - y|, \quad \forall x \in [a,b].$$

Primijetite λ može biti veći od 1!

Teorem 9.6.1. *Neka je funkcija g neprekidno diferencijabilna na $[a,b]$, neka je*

$$g([a,b]) \subseteq [a,b],$$

i neka za λ iz (9.6.2) vrijedi

$$\lambda < 1. \tag{9.6.3}$$

Tada vrijedi:

1. *$x = g(x)$ ima točno jedno rješenje na $\alpha \in [a,b]$,*
2. *za proizvoljni $x_0 \in [a,b]$, za jednostavnu iteraciju $x_{n+1} = g(x_n)$, $n \geq 0$ vrijedi*

$$\lim_{n\to\infty} x_n = \alpha,$$

$$|\alpha - x_n| \leq \lambda^n |\alpha - x_0|$$

i

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\alpha).$$

Dokaz:

Sve tvrdnje ovog teorema dokazane su u prethodne dvije leme, osim posljednje relacije o brzini konvergencije.

Vrijedi

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\xi_n)(\alpha - x_n), \quad n \geq 0,$$

gdje je ξ_n neki broj između α i x_n. Budući da $x_n \to \alpha$, onda i $\xi_n \to \alpha$, pa vrijedi

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \lim_{n\to\infty} g'(\xi_n) = g'(\alpha).$$

■

Pokažimo koliko je pretpostavka (9.6.3) značajna, tj. pretpostavimo da je $|g'(\alpha)| > 1$. Tada, ako imamo niz $x_{n+1} = g(x_n)$ i rješenje $\alpha = g(\alpha)$, vrijedi

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\xi_n)(\alpha - x_n).$$

Za x_n dovoljno blizu α, onda je i i $|g'(\xi_n)| > 1$, pa je $|\alpha - x_{n+1}| \geq |\alpha - x_n|$, pa konvergencija metode nije moguća.

Prethodni teorem se može malo i pojednostavniti, tako da se ne navodi eksplicitno interval $[a,b]$.

Teorem 9.6.2. *Neka je α rješenje jednostavne iteracije $x = g(x)$ i neka je g neprekidno diferencijabilna na nekoj okolini od α i neka je $|g'(\alpha)| < 1$. Tada vrijede svi rezultati Teorema 9.6.1., uz pretpostavku da je x_0 dovoljno blizu α.*

Dokaz:

Uzmimo $I = [\alpha - \varepsilon, \alpha + \varepsilon]$ takav da je

$$\max_{x \in I} |g'(x)| \leq \lambda < 1.$$

Tada je $g(I) \subseteq I$, jer $|\alpha - x| \leq \varepsilon$ povlači

$$|\alpha - g(x)| = |g(\alpha) - g(x)| = |g'(\xi)||\alpha - x| \leq \lambda |\alpha - x| \leq \varepsilon.$$

Sada možemo primijeniti prethodni teorem za $[a, b] = I$. ■

Primjer 9.6.2. *U primjeru 9.6.1., definirali smo tri iteracione funkcije.*

1. *Ako je* $g(x) = x^2 + x - a$, *onda je* $g'(x) = 2x + 1$ *i u nultočki* $\alpha = \sqrt{a}$ *je*

$$g'(\sqrt{a}) = 2\sqrt{a} + 1 > 1,$$

pa ta iteraciona funkcija neće konvergirati. U općenitijem je slučaju $g(x) = x + c(x^2 - a)$, *pa je* $g'(x) = 1 + 2cx$ *i*

$$g'(\sqrt{a}) = 1 + 2c\sqrt{a}.$$

Da bismo osigurali konvergenciju, mora biti

$$-1 < 1 + 2c\sqrt{a} < 1,$$

odnosno

$$-\frac{1}{\sqrt{a}} < c < 0.$$

2. *Ako je* $g(x) = a/x$, *onda je* $g'(x) = -a/x^2$, *pa je*

$$g'(\sqrt{a}) = -1.$$

3. *Ako je* $g(x) = 0.5\,(x + a/x)$, *onda je* $g'(x) = 0.5\,(1 - a/x^2)$, *pa je*

$$g'(\sqrt{a}) = 0.$$

Ovaj odjeljak završit ćemo promatranjem jednostavnih iteracionih funkcija, ali višeg reda konvergencije, kao što je, na primjer Newtonova metoda.

Teorem 9.6.3. *Neka je* α *rješenje od* $x = g(x)$ *i neka je* g p *puta neprekidno diferencijabilna za sve* x *u okolini* α, *za neki* $p \geq 2$. *Nadalje, pretpostavimo da je*

$$g'(\alpha) = \cdots = g^{(p-1)}(\alpha) = 0. \tag{9.6.4}$$

Ako je startna vrijednost x_0 *dovoljno blizu* α, *iteraciona funkcija*

$$x_{n+1} = g(x_n), \quad n \geq 0$$

imat će red konvergencije p *i*

$$\lim_{n \to \infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^p} = (-1)^{p-1} \frac{g^{(p)}(\alpha)}{p!}.$$

Dokaz:

Razvijmo $g(x)$ u okolini α do uključivo $(p-1)$-ve potencije i napišimo ostatak. Zatim, uvrstimo $x = x_n$, pa dobivamo

$$x_{n+1} = g(x_n) = g(\alpha) + g'(\alpha)(x_n - \alpha) + \cdots + \frac{g^{(p-1)}(\alpha)}{(p-1)!}(x_n - \alpha)^{p-1} + \frac{g^{(p)}(\xi_n)}{p!}(x_n - \alpha)^p,$$

za neki ξ_n između x_n i α. Iskoristimo li da je $g(\alpha) = \alpha$ i pretpostavku (9.6.4), slijedi

$$x_{n+1} = \alpha + \frac{g^{(p)}(\xi_n)}{p!}(x_n - \alpha)^p,$$

odnosno

$$\alpha - x_{n+1} = -\frac{g^{(p)}(\xi_n)}{p!}(x_n - \alpha)^p.$$

Sada možemo primijeniti prethodni Teorem, koji pokazuje da će iteraciona funkcija konvergirati. Nadalje, to znači da $x_n \to \alpha$, pa i $\xi_n \to \alpha$, što daje traženu relaciju. ■

Korištenjem prethodnog teorema možemo analizirati i Newtonovu metodu za koju je

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Deriviranjem dobivamo da je

$$g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2},$$

pa je

$$g(\alpha) = 0,$$

uz pretpostavku da je $f'(\alpha) \neq 0$. Na sličan način, dobivamo

$$g''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)},$$

pa ako je $f''(\alpha) \neq 0$, možemo pokazati da je red konvergencije Newtonove metode jednak 2. Ako je $f'(\alpha) \neq 0$, $f''(\alpha) = 0$, onda će red konvergencije biti barem 3.

9.7. Newtonova metoda za višestruke nultočke

Promotrimo što će se dogoditi s konvergencijom Newtonove metode, ako funkcija f ima neprekidnih prvih $p+1$ derivacija i p-struku, $p \geq 2$ nultočku u α. Tada vrijedi

$$f(\alpha) = f'(\alpha) = \cdots = f^{(p-1)}(\alpha) = 0, \quad f^{(p)}(\alpha) \neq 0.$$

Samu funkciju f možemo napisati i u obliku

$$f(x) = (x-\alpha)^p\, h(x), \quad h(\alpha) \neq 0. \tag{9.7.1}$$

Ograničimo se samo na cjelobrojne p i promatrajmo Newtonovu metodu kao jednostavnu iteraciju,

$$x_{n+1} = g(x_n), \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

Deriviranjem (9.7.1) dobivamo jednostavniji oblik za derivaciju

$$f'(x) = p(x-\alpha)^{p-1}h(x) + (x-a)^p h'(x),$$

pa je

$$g(x) = x - \frac{(x-\alpha)h(x)}{ph(x) + (x-\alpha)h'(x)}.$$

Deriviranjem funkcije g dobivamo

$$g'(x) = 1 - \frac{h(x)}{ph(x)+(x-\alpha)h'(x)} - (x-\alpha)\frac{d}{dx}\Big(\frac{h(x)}{ph(x)+(x-\alpha)h'(x)}\Big),$$

tako da je

$$g'(\alpha) = 1 - \frac{1}{p} \neq 0 \quad \text{za} \quad p > 1,$$

što pokazuje linearnu konvergenciju. Prema teoremu 9.6.1., faktor konvergencije bit će $g'(\alpha) = 1 - 1/p$, što je vrlo sporo. U prosjeku to je podjednako brzo kao bisekcija za $p = 2$ ili čak lošije od bisekcije za $p \geq 3$.

Kako možemo popraviti (ubrzati) Newtonovu metodu za p-struke nultočke, $p \geq 2$. Prvo pretpostavimo da znamo p. Definiramo iteracionu funkciju

$$g(x) = x - p\,\frac{f(x)}{f'(x)}.$$

Tada je

$$g'(x) = 1 - p\,\frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = 1 - p + p\,\frac{f(x)f''(x)}{(f'(x))^2}.$$

Iskoristimo li oblik funkcije f, dobivamo

$$\begin{aligned} f(x) &= (x-\alpha)^p h(x) \\ f'(x) &= (x-\alpha)^{p-1}[ph(x) + (x-\alpha)h'(x)] \\ f''(x) &= (x-\alpha)^{p-2}[p(p-1)h(x) + 2p(x-\alpha)h'(x) + (x-\alpha)^2 h''(x)], \end{aligned}$$

pa je

$$\lim_{x\to\alpha} \frac{f(x)f''(x)}{(f'(x))^2} = 1 - \frac{1}{p}.$$

Odatle odmah slijedi

$$\lim_{x\to\alpha} g'(x) = 0,$$

što pokazuje da ova modifikacija osigurava barem kvadratično konvergentnu metodu.

Što ćemo napraviti ako unaprijed ne znamo p? Primijetimo da funkcija

$$u(x) = \frac{f(x)}{f'(x)} = \frac{(x-\alpha)^p h(x)}{(x-\alpha)^{p-1}[ph(x) + (x-\alpha)h'(x)]} = \frac{(x-\alpha)h(x)}{ph(x) + (x-\alpha)h'(x)}$$

ima jednostruku nultočku u α. Drugim riječima, obična Newtonova metoda, ali primijenjena na $u(x)$ konvergirat će kvadratično,

$$x_{n+1} = x_n - \frac{u(x_n)}{u'(x_n)},$$

gdje je

$$u'(x) = \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = 1 - \frac{f''(x)}{f'(x)}\,u(x),$$

što pokazuje da ćemo dobiti kvadratičnu konvergenciju, iako ne znamo red nultočke, ali uz računanje još jedne derivacije funkcije (f'').

Slično vrijedi i za metodu sekante, koju ćemo ubrzati, kao da radimo s jednostrukim nultočkama, ako primijenimo metodu sekante za funkciju u

$$x_{n+1} = x_n - u(x_n)\,\frac{x_n - x_{n-1}}{u(x_n) - u(x_{n-1})}.$$

I u ovom slučaju postoji "cijena", a to je računanje f'.

9.8. Hibridna Brent–Dekkerova metoda

Brent–Dekkerova metoda smišljena je kao metoda koja će imati sigurnu konvergenciju, a nadamo se da će konvergirati brže nego metoda sekante, u najboljem slučaju kvadratično. Ona **ne zahtijeva** računanje derivacija, pa ako joj je red konvergencije u prosjeku bolji od sekante, možemo očekivati da će metoda po brzini biti slična Newtonovoj, ali će imati sigurnu konvergenciju.

Metoda se sastoji od tri dijela, koje grubo možemo opisati kao inverznu kvadratnu interpolaciju, metodu sekante i metodu bisekcije. Algoritam počinje metodom sekante koja generira treću točku. Ako se prema nekim kriterijima ta točka prihvaća kao dobra, možemo nastaviti raditi s kvadratnom interpolacijom kroz posljednje tri točke, ali inverznom (uloga x i y zamijenjena) i time dobivamo četvrtu točku.

Ako je treća točka odbačena kao loša, radi se jedan korak metode bisekcije. Drugim riječima, metoda se "vrti" između svoja tri sastavna dijela, a mi se nadamo da će rijetko koristiti bisekciju.

Točni parametri kad se neka aproksimacija nultočke prihvaća kao dobra, odnosno odbacuje kao loša su dosta složeni. Metoda je sastavni dio velikih numeričkih biblioteka programa, kao što je IMSL.

9.9. Primjeri

Prije konkretnih primjera, zanimljivo je napomenuti da se u praksi može sasvim dobro numerički procijeniti red konvergencije iterativne metode i taj podatak iskoristiti kao dodatna informacija o konvergenciji metode.

Kao najjednostavniji primjer za usporedbu metoda za nalaženje nultočaka uzmimo da treba izračunati $\sqrt[3]{1.5}$. Taj problem možemo interpretirati i kao traženje realne pozitivne nultočke funkcije $f(x) = x^3 - 1.5$.

Primjer 9.9.1. *Nultočka funkcije*

$$f(x) = \operatorname{arctg}(x)$$

je $x = 0$*, ali Newtonova metoda neće konvergirati iz svake startne točke* x_0*. Naći ćemo točku* β *za koju vrijedi*

$$\begin{cases} |x_0| < |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ konvergira,} \\ |x_0| > |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ divergira,} \\ |x_0| = |\beta| & \textit{Newtonova metoda sa startom } x_0 \textit{ ciklira.} \end{cases}$$

Kako ćemo naći točku "cikliranja"? Funkcija $f(x) = \operatorname{arctg} x$ *je neparna, pa da bismo dobili cikliranje, dovoljno je da tangenta na funkciju u točki* β *presiječe os* x *u točki* $-\beta$*. Jednadžba tangente na* arctg *u točki* β *je*

$$y - \operatorname{arctg}\beta = \frac{1}{1+\beta^2}(x - \beta),$$

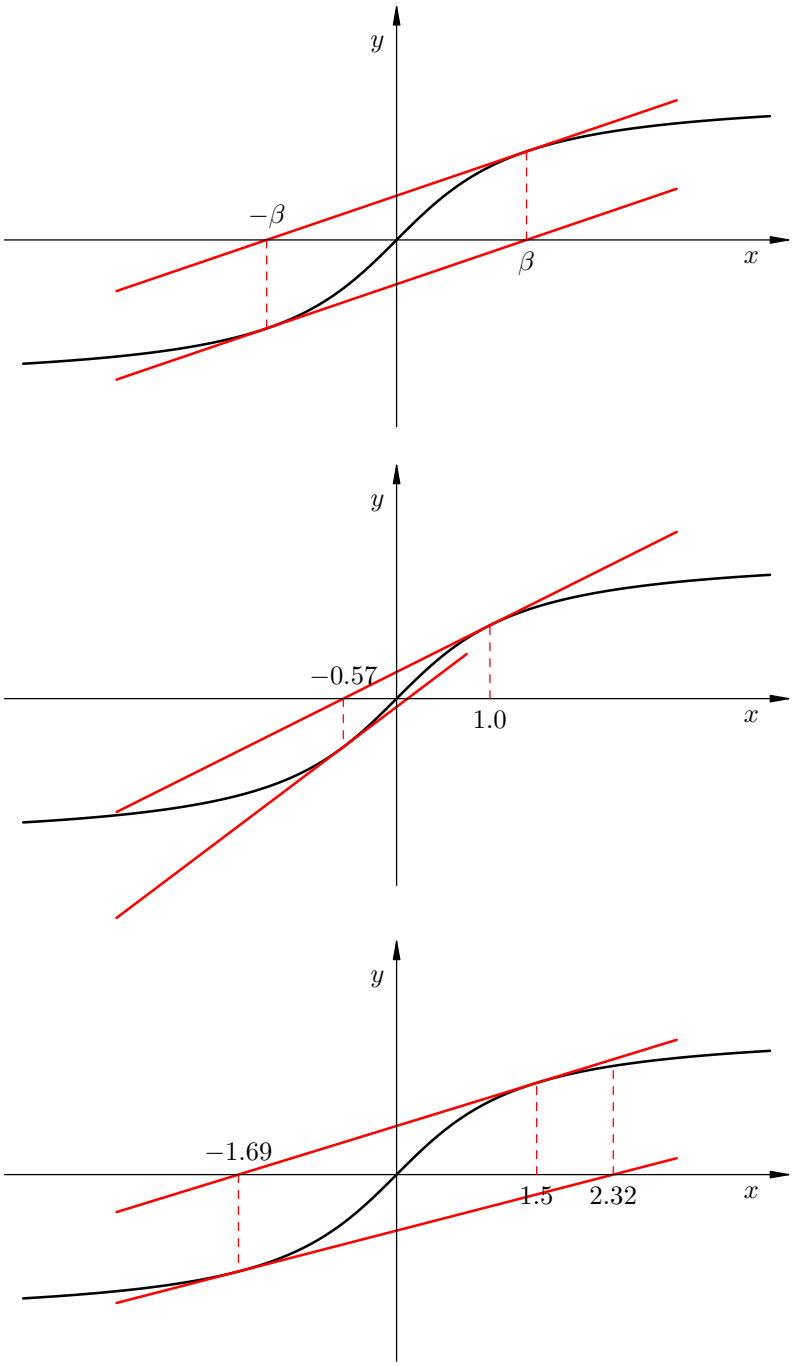
pa će tangenta sijeći os x *u* $-\beta$*, ako je*

$$\operatorname{arctg}\beta = \frac{2\beta}{1+\beta^2},$$

čime smo dobili nelinearnu jednadžbu po β*. Očito, postoje dva rješenja, suprotnih predznaka, i nije ih teško izračunati metodom bisekcije*

$$\beta = \pm 1.39174520027073489.$$

Nacrtajmo grafove Newtonove metode za sve tri mogućnosti za x_0, recimo za $x_0 = 1$, $x_0 = \beta$ i $x_0 = 1.5$.



10. Aproksimacija i interpolacija

10.1. 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.

10.1.1. 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.

10.1.2. 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.

10.1.3. 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.

10.2. 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 $a \leq x_0 < x_1 < \cdots < x_n \leq b$.

10.2.1. Egzistencija i jedinstvenost interpolacionog polinoma

Za polinomnu interpolaciju vrijedi sljedeći teorem.

Teorem 10.2.1. *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.

10.2.2. 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 10.2.1, 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 10.2.1., 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.

10.2.3. 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 $\mathcal{P}_n$

$$p_n(x) = \sum_{k=0}^{n} f_k \ell_k(x), \tag{10.2.1}$$

pri čemu je

$$\begin{aligned}\ell_k(x) &= \frac{(x-x_0)\cdots(x-x_{k-1})\,(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\ &= \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{x-x_i}{x_k-x_i} := \frac{\omega_k(x)}{\omega_k(x_k)}, \quad k=0,\ldots,n.\end{aligned} \tag{10.2.2}$$

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 (10.2.1), odmah slijedi da se suma u (10.2.1) svodi na jedan jedini član za $i = k$, tj. da vrijedi

$$p_n(x_k) = f_k.$$

Oblik (10.2.1)–(10.2.2) 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 (10.2.1) dobivamo da je

$$p_n(x) = \omega(x)\sum_{k=0}^{n}\frac{f_k}{(x-x_k)\,\omega_k(x_k)}. \tag{10.2.3}$$

Uočimo da je

$$\omega_k(x_k) = \omega'(x_k),$$

pa (10.2.3) možemo pisati kao

$$p_n(x) = \omega(x)\sum_{k=0}^{n}\frac{f_k}{(x-x_k)\,\omega'(x_k)}. \tag{10.2.4}$$

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).

10.2.4. Ocjena greške interpolacionog polinoma

Ako znamo još neke informacije o funkciji f, možemo napraviti i ocjenu greške interpolacionog polinoma.

Teorem 10.2.2. *Pretpostavimo da funkcija f ima $(n+1)$-u derivaciju na segmentu $[a,b]$ za neki $n \in \mathbb{N}_0$. Neka su $x_k \in [a,b]$, $k = 0,\dots,n$, 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*

$$x_{\min} := \min\{x_0,\dots,x_n,x\} < \xi < \max\{x_0,\dots,x_n,x\} =: x_{\max}$$

takva da za grešku interpolacionog polinoma vrijedi

$$e(x) := f(x) - p_n(x) = \frac{\omega(x)}{(n+1)!}\, f^{(n+1)}(\xi), \tag{10.2.5}$$

pri čemu je $\omega(x) := \prod_{k=0}^{n}(x - x_k)$.

Dokaz:

Ako je $x = x_k$, za neki $k \in \{0,\dots,n\}$, iz uvjeta interpolacije i definicije polinoma ω dobivamo da su obje strane u (10.2.5) 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

$$g(t) = e(t) - \omega(t)s(x) = e(t) - \omega(t)\,\frac{e(x)}{\omega(x)}, \quad t \in [a,b]. \tag{10.2.6}$$

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

$$g(x_k) = e(x_k) - \omega(x_k)\,\frac{e(x)}{\omega(x)} = 0, \quad k = 0,\dots,n.$$

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 (10.2.6) 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 (10.2.5). ■

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 (10.2.5), a korisna je ako relativno jednostavno možemo izračunati ili odozgo ocijeniti M_{n+1}.

10.2.5. 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

$$p_n(x) = p_{n-1}(x) + c(x), \tag{10.2.7}$$

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 (10.2.7). 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

$$p_n(x) = p_{n-1}(x) + (x - x_0)\cdots(x - x_{n-1})f[x_0, \ldots, x_n]. \tag{10.2.8}$$

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 (10.2.4), tj. nađimo koeficijent uz x^n u toj relaciji. Dobivamo

$$f[x_0, x_1, \ldots, x_n] = \sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)}. \tag{10.2.9}$$

Iz formule (10.2.9) 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

$$\begin{aligned}
&f[x_1,\dots,x_n]-f[x_0,\dots,x_{n-1}] \\
&\quad= \sum_{k=1}^{n-1} \frac{f(x_k)(x_n-x_0)}{(x_k-x_0)\cdots(x_k-x_{k-1})\,(x_k-x_{k+1})\cdots(x_k-x_n)} \\
&\qquad + \frac{f(x_n)(x_n-x_0)}{(x_n-x_0)\cdots(x_n-x_{n-1})} + \frac{f(x_0)(x_n-x_0)}{(x_0-x_1)\cdots(x_0-x_n)} \\
&\quad= (x_n-x_0)\sum_{k=0}^{n} \frac{f(x_k)}{\omega'(x_k)} = (x_n-x_0)\,f[x_0,\dots,x_n],
\end{aligned}$$

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 (10.2.8), 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 10.2.1. (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

$$\begin{aligned} p_{n+1}(x) &= f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)\,(x - x_1)f[x_0, x_1, x_2] \\ &\quad + \cdots + (x - x_0)\cdots(x - x_{n-1})f[x_0, x_1, \ldots, x_n] \\ &\quad + (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}] \\ &= p_n(x) + (x - x_0)\cdots(x - x_n)f[x_0, x_1, \ldots, x_n, x_{n+1}]. \end{aligned} \tag{10.2.10}$$

Budući da je

$$p_{n+1}(x_{n+1}) = f(x_{n+1}),$$

onda iz relacije (10.2.10) 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 10.2.2. (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)

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{10.2.11}$$

Zajedno s (10.2.10), 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 10.2.2. (Algoritam izvrednjavanja interpolacionog polinoma)

$sum := f[n]$;
for $i := n - 1$ **downto** 0 **do**
  $sum := sum * (x - x_i) + f[i]$;
{ Na kraju je $p_n(x) = sum$. }

10.2.6. 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 10.2.1. (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* 10.2.2*, uvažavanjem relacije* (10.2.11)*, dobivamo*

$$e_n(x) = f(x) - p_n(x) = \omega(x)\, f[x_0, x_1, \ldots, x_n, x].$$

Tvrdimo da vrijedi

$$f[x_0, x_1, \ldots, x_n, x] = f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r}(1+x_k^2)} \cdot \begin{cases} 1, & \text{ako je } n = 2r+1, \\ x, & \text{ako je } n = 2r. \end{cases} \tag{10.2.12}$$

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* (10.2.12) *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

$$e_n(x) = f(x) - p_n(x) = \omega(x)\, f(x) \cdot \frac{(-1)^{r+1}}{\prod_{k=0}^{r} (1 + x_k^2)} \cdot \begin{cases} 1, & \text{ako je } n = 2r + 1, \\ x, & \text{ako je } n = 2r. \end{cases}$$

$$= (-1)^{r+1}\, f(x)\, g_n(x),$$

gdje je

$$g_n(x) = \prod_{k=0}^{r} \frac{x^2 - x_k^2}{1 + x_k^2}. \tag{10.2.13}$$

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 (10.2.13)*, 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 $[a, b]$, Čebiševljeve točke su

$$x_k = \frac{1}{2}\left(a + b + (a - b) \cos \frac{2k + 1}{2n + 2} \right), \quad k = 0, \ldots, n.$$

Zadatak 10.2.1. *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).*

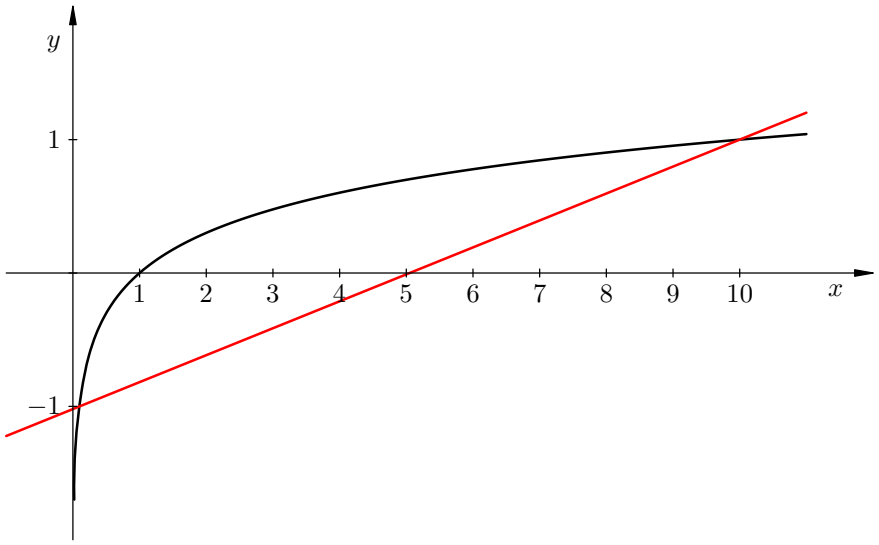
Primjer 10.2.2. *Promotrimo grafove interpolacionih polinoma stupnjeva 1–6 koji interpoliraju funkciju*

$$f(x)=\log(x) \quad za\ x\in[0.1,10]$$

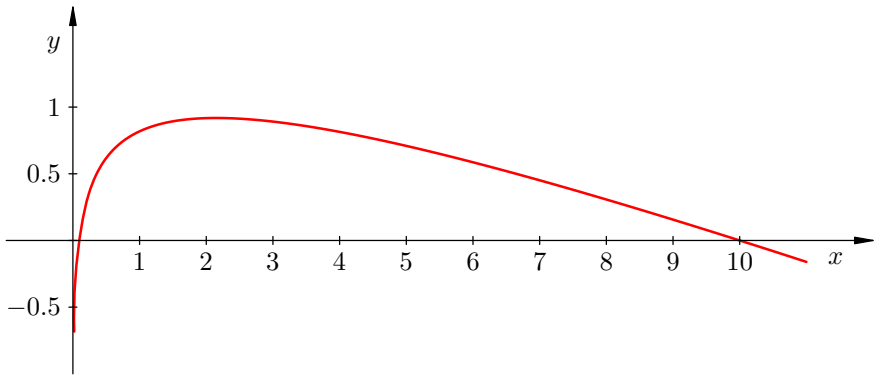
na ekvidistantnoj i Čebiševljevoj mreži.

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.*

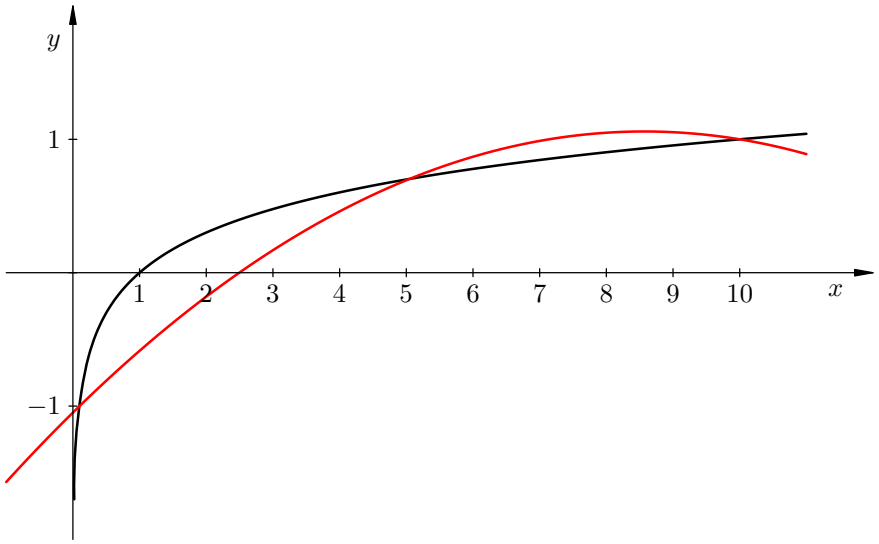
Prva grupa slika su redom funkcija (crno) i interpolacioni polinom (crveno) za ekvidistantnu mrežu, te pripadna greška, a zatim to isto za Čebiševljevu mrežu.



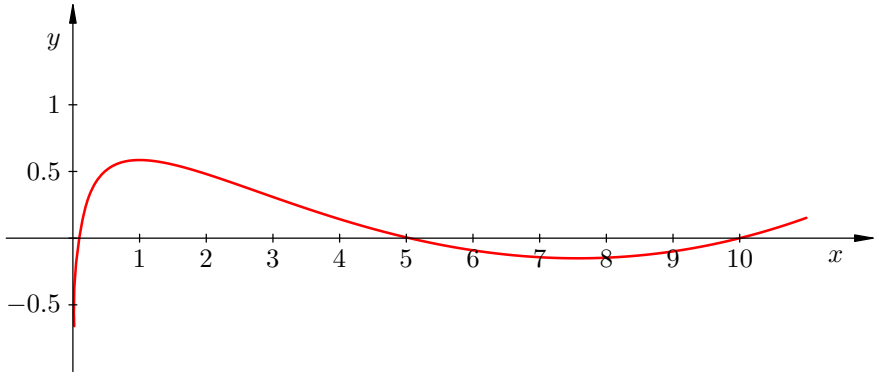
Ekvidistantna mreža, interpolacioni polinom stupnja 1.



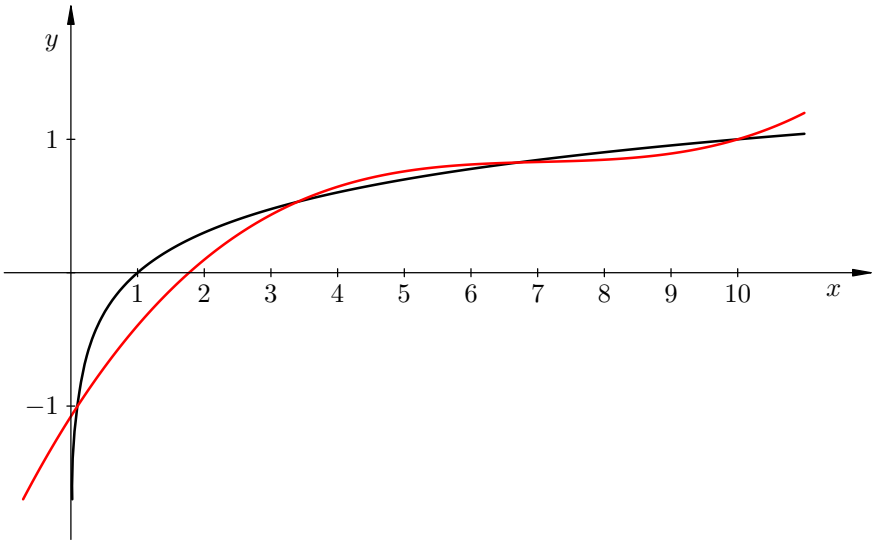
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 1.



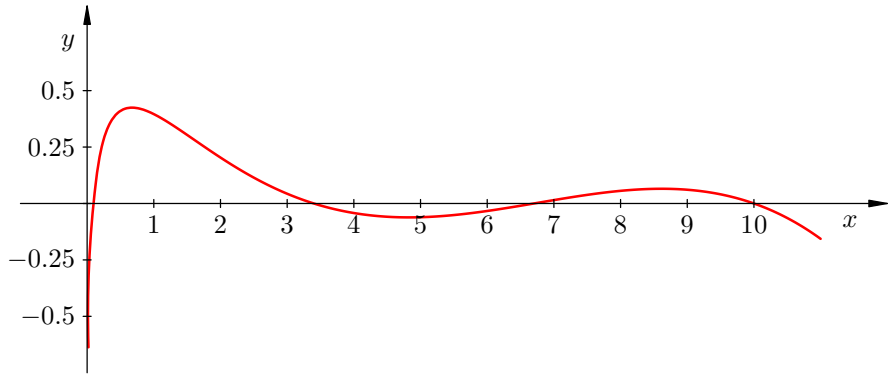
Ekvidistantna mreža, interpolacioni polinom stupnja 2.



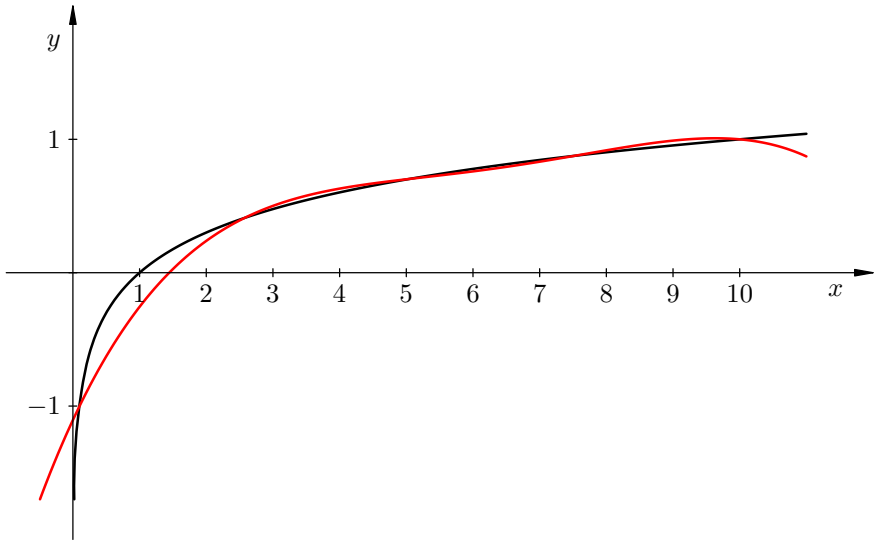
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 2.



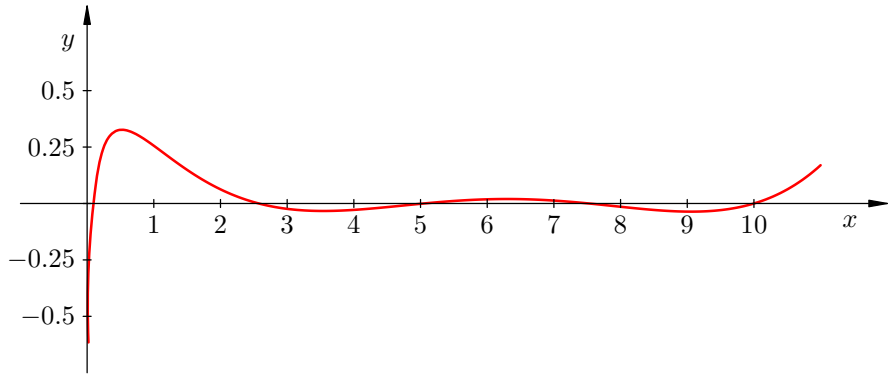
Ekvidistantna mreža, interpolacioni polinom stupnja 3.



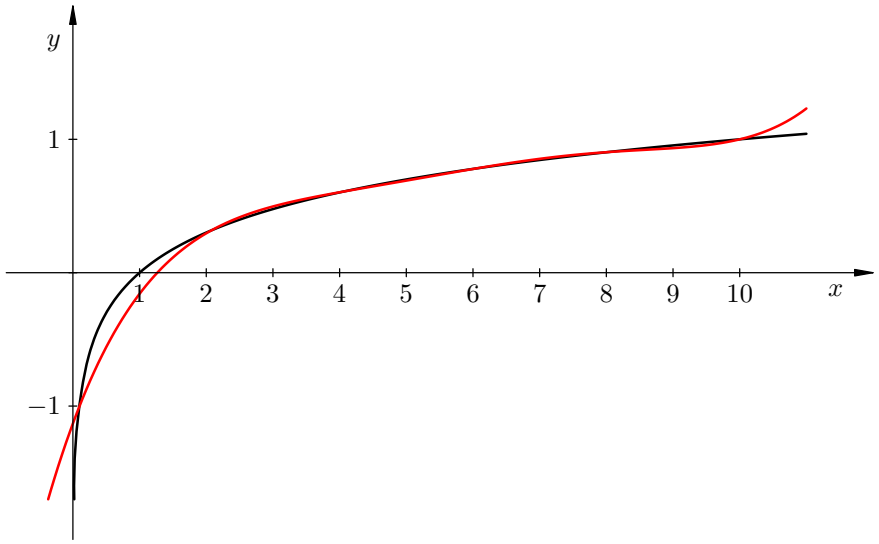
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 3.



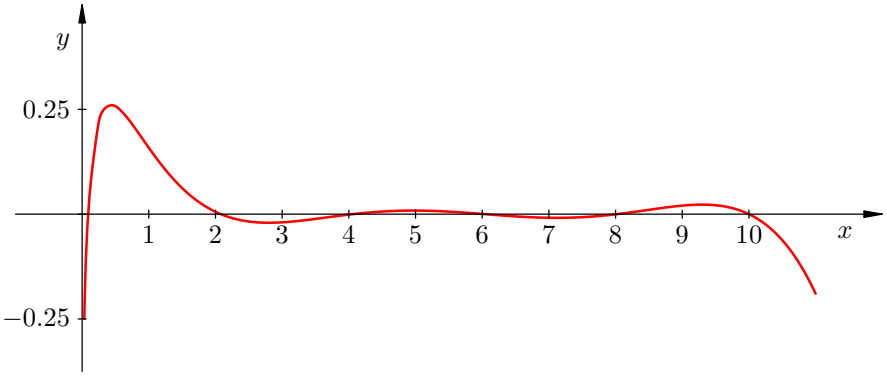
Ekvidistantna mreža, interpolacioni polinom stupnja 4.



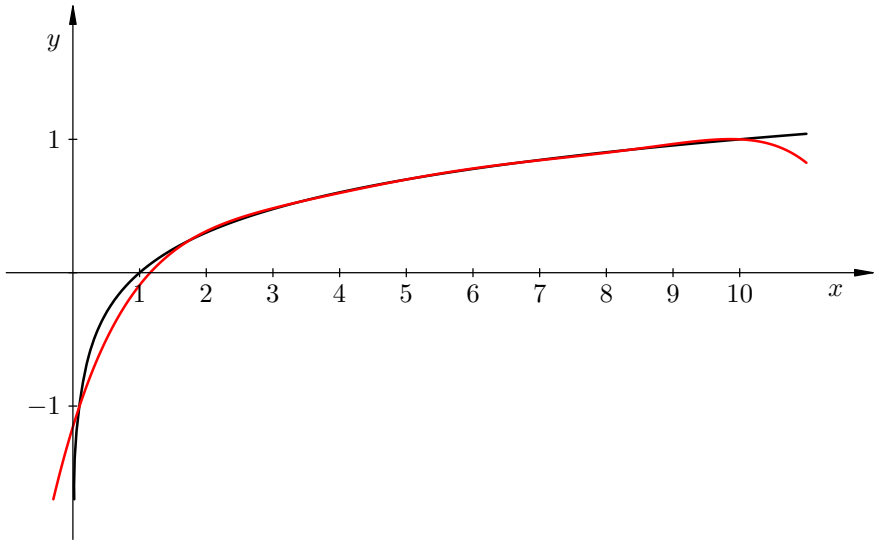
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 4.



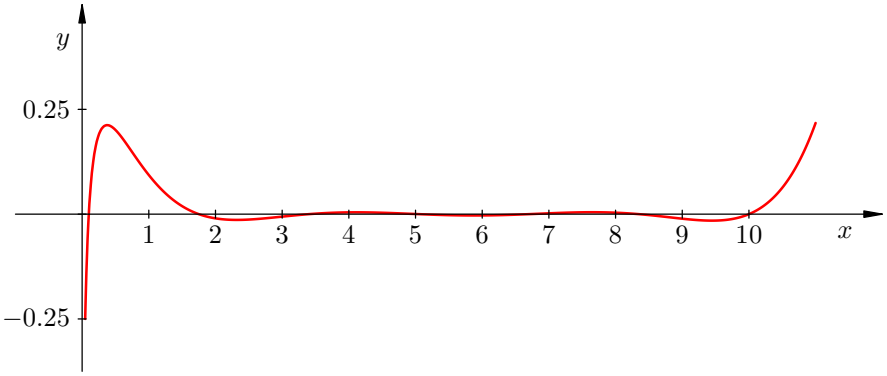
Ekvidistantna mreža, interpolacioni polinom stupnja 5.



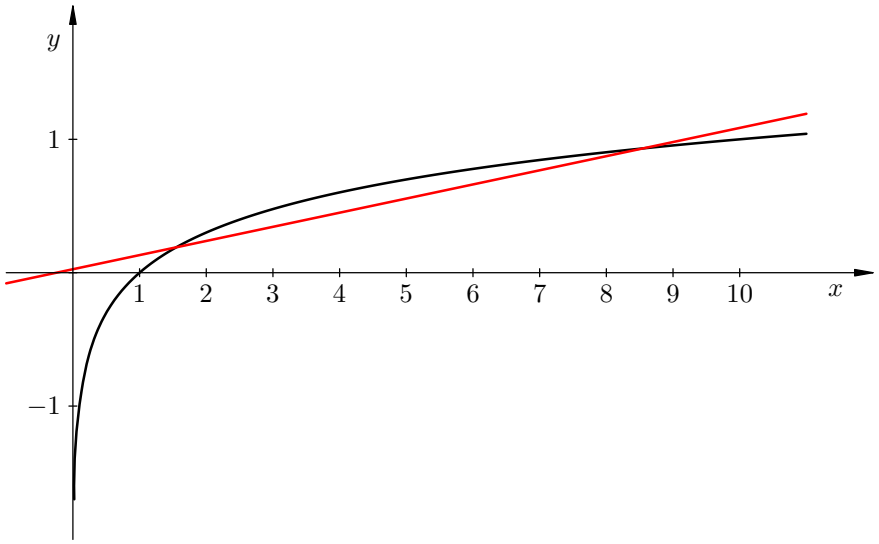
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 5.



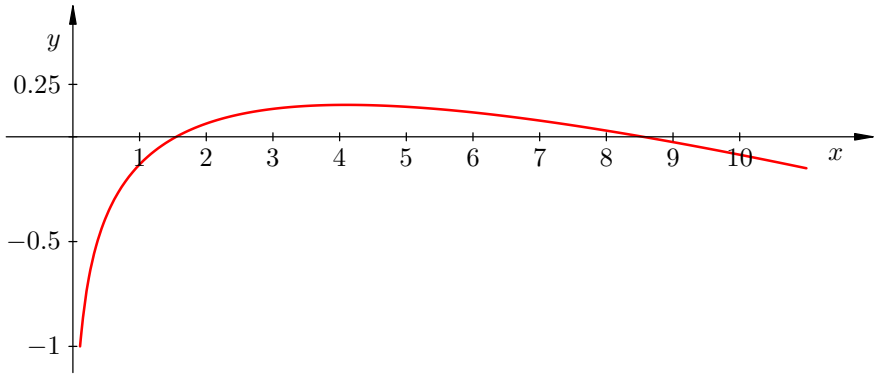
Ekvidistantna mreža, interpolacioni polinom stupnja 6.



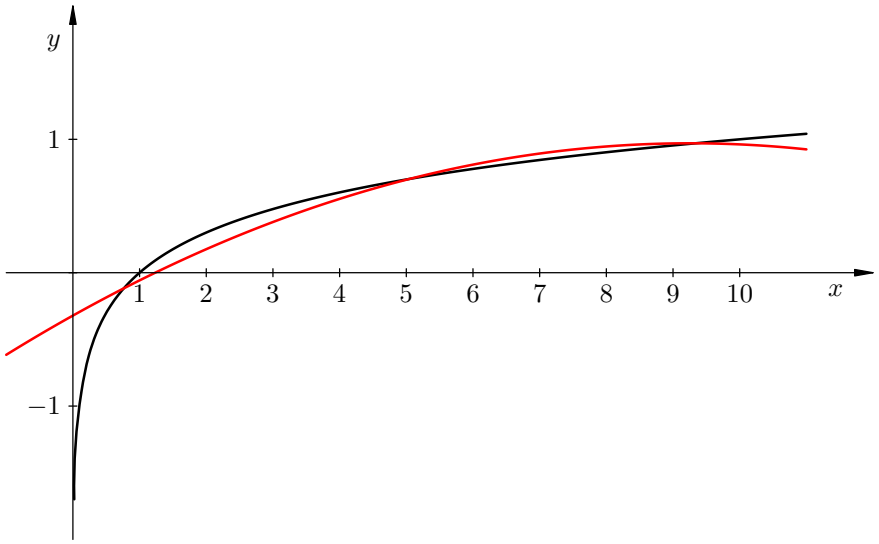
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 6.



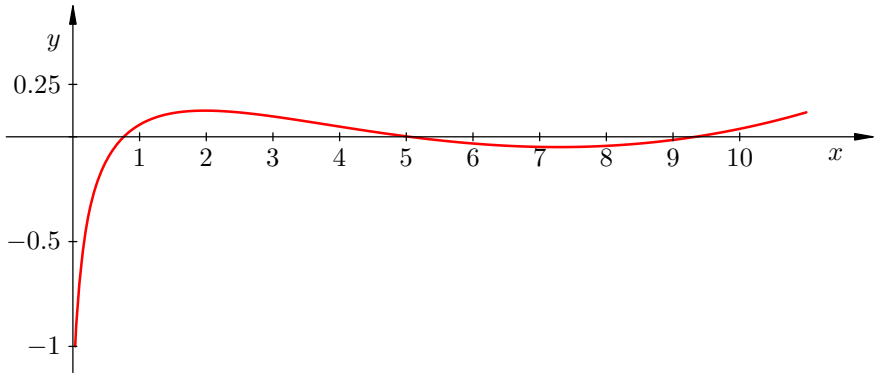
Čebiševljeva mreža, interpolacioni polinom stupnja 1.



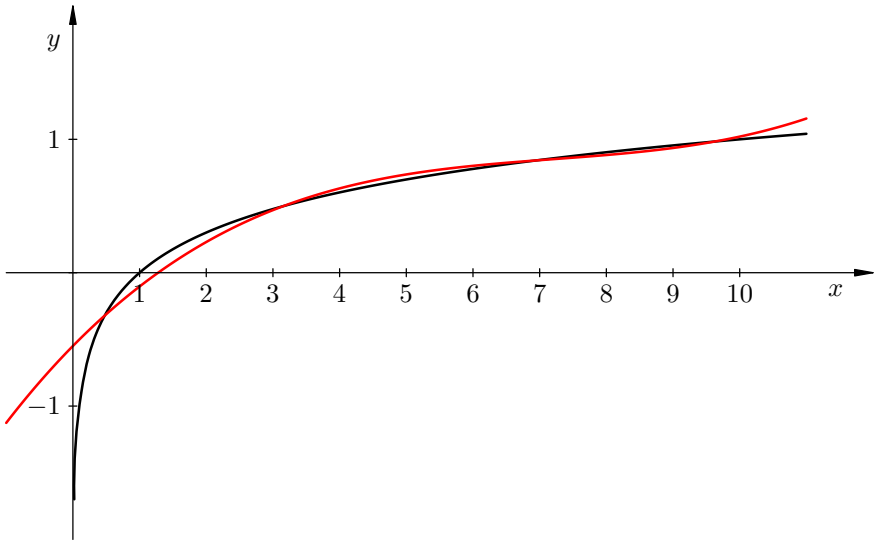
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 1.



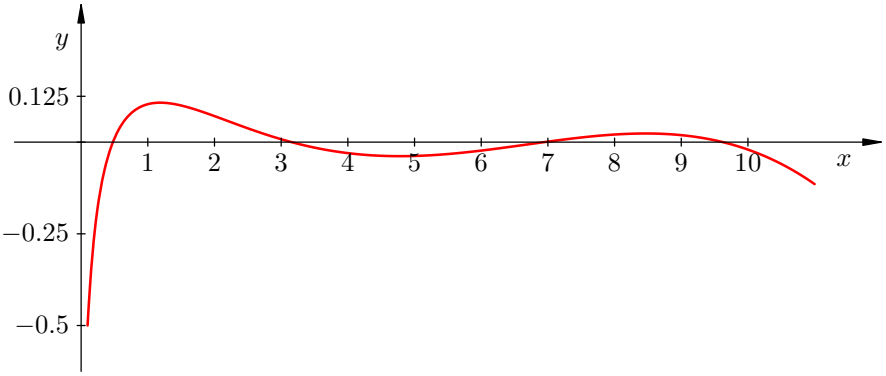
Čebiševljeva mreža, interpolacioni polinom stupnja 2.



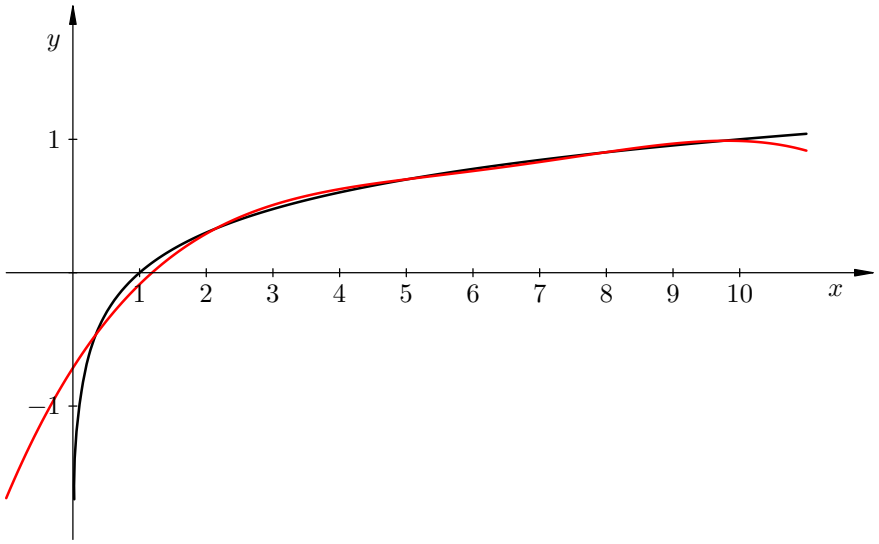
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 2.



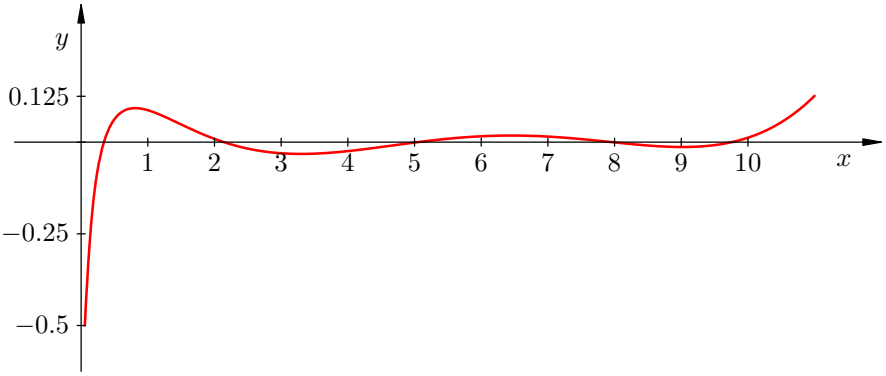
Čebiševljeva mreža, interpolacioni polinom stupnja 3.



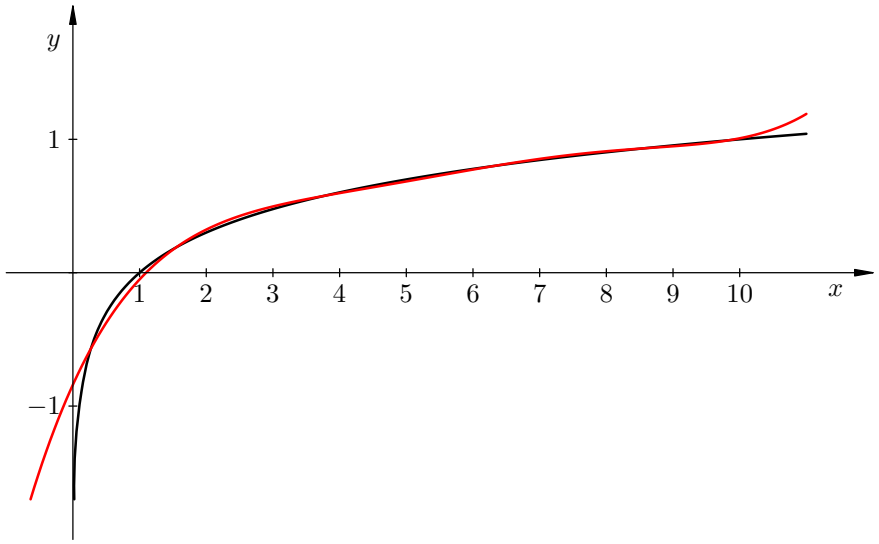
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 3.



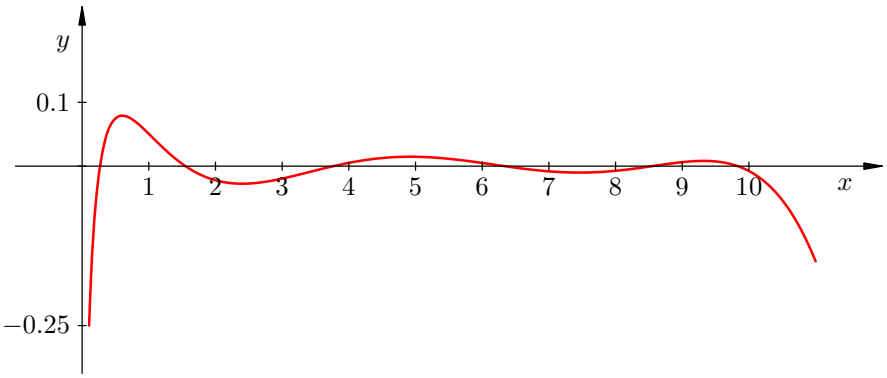
Čebiševljeva mreža, interpolacioni polinom stupnja 4.



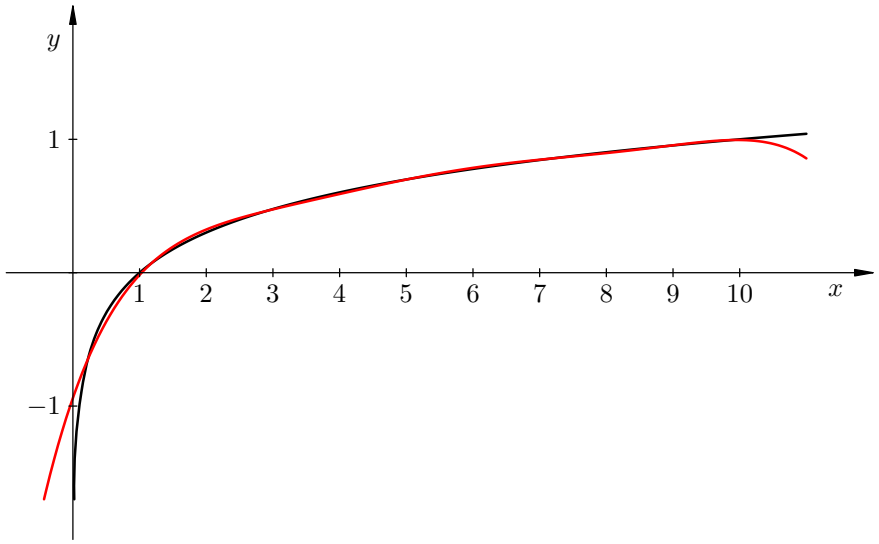
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 4.



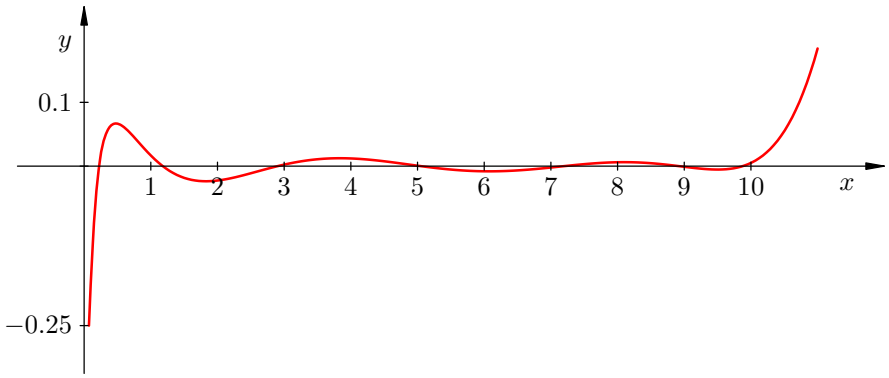
Čebiševljeva mreža, interpolacioni polinom stupnja 5.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 5.



Čebiševljeva mreža, interpolacioni polinom stupnja 6.



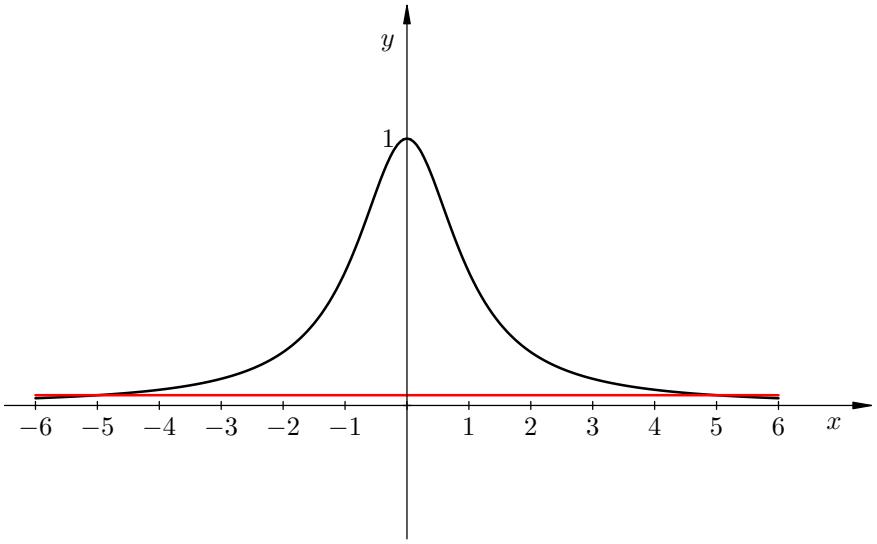
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 6.

Primjer 10.2.3. *Već smo pokazali na primjeru Runge da interpolacioni polinomi koji interpoliraju funkciju*

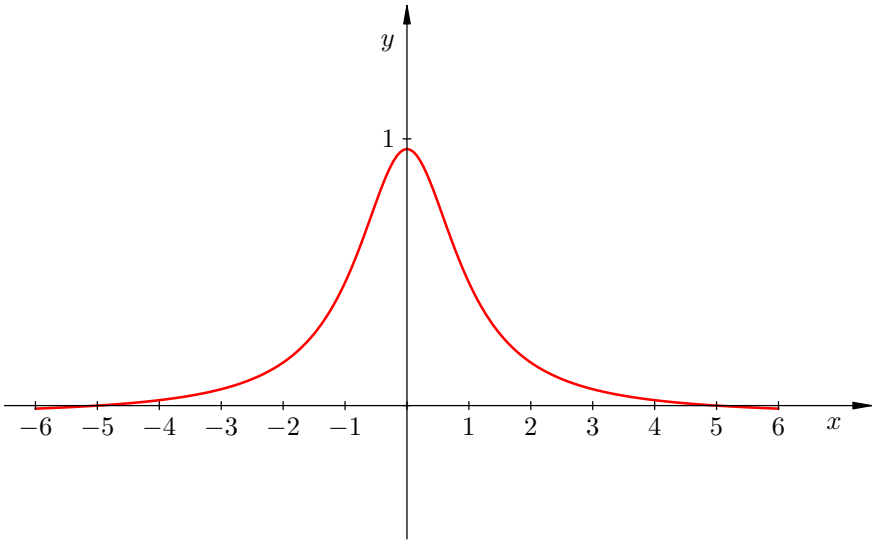
$$f(x) = \frac{1}{1+x^2} \quad za\ x \in [-5, 5]$$

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!).

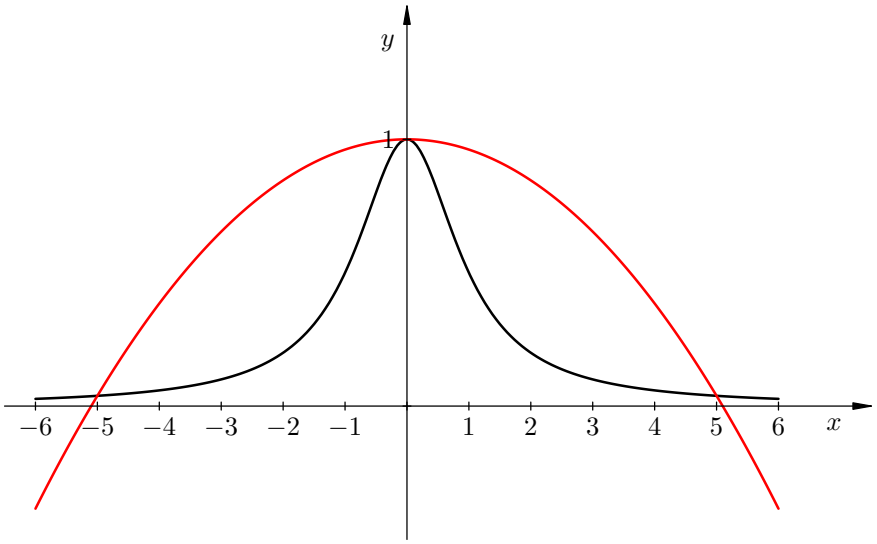
Ponovno, kao i u prošlom primjeru, grafovi su u parovima.



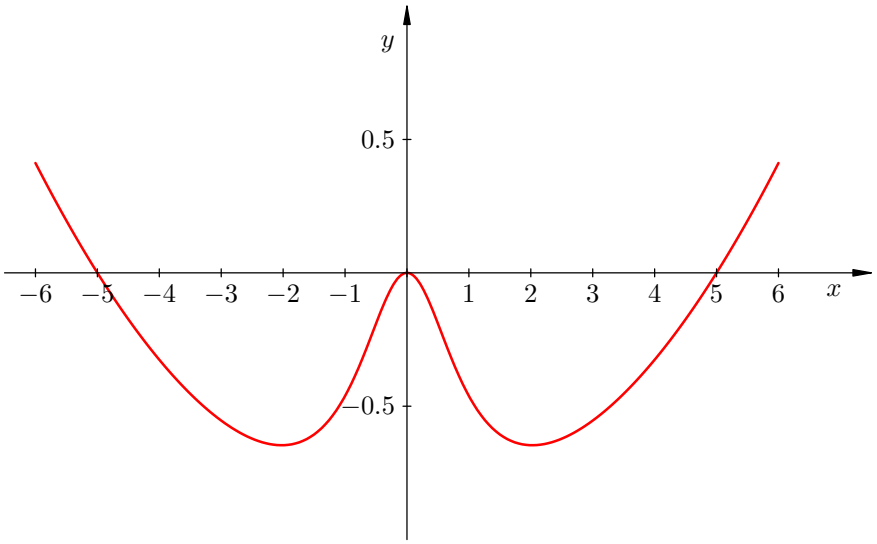
Ekvidistantna mreža, interpolacioni polinom stupnja 1.



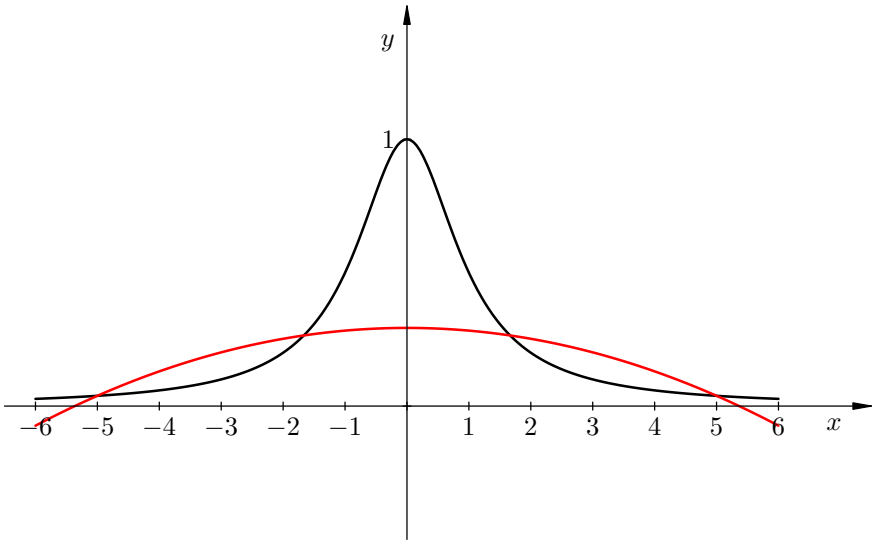
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 1.



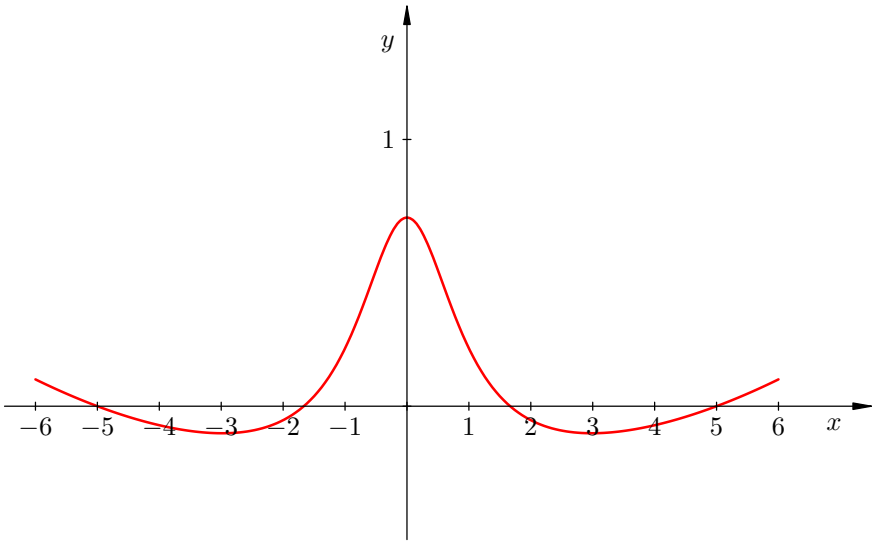
Ekvidistantna mreža, interpolacioni polinom stupnja 2.



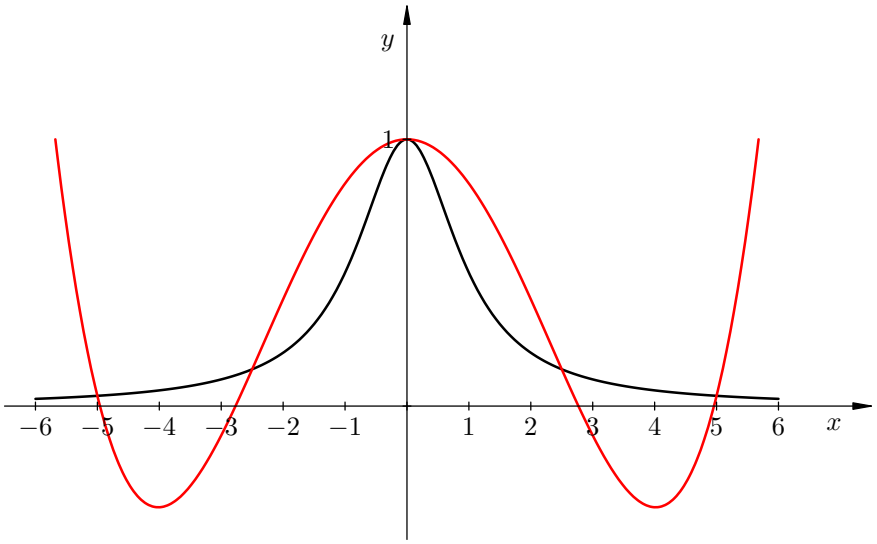
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 2.



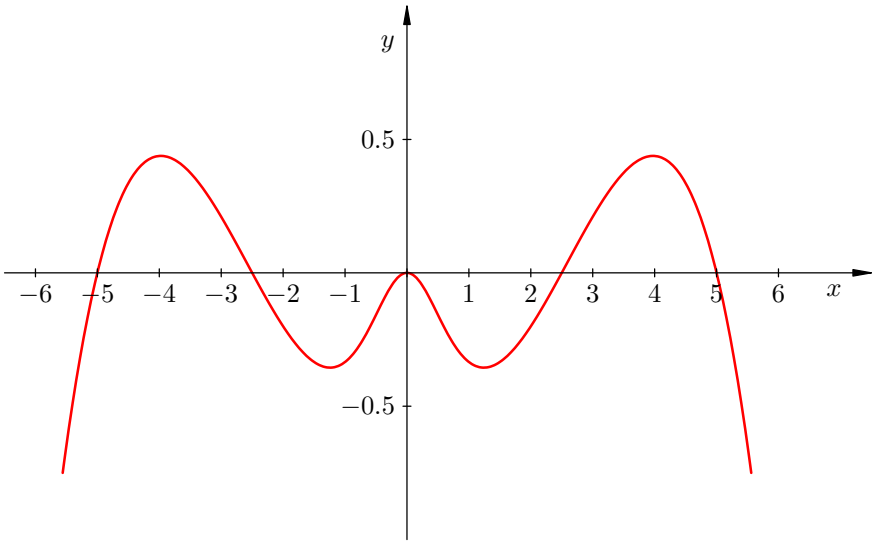
Ekvidistantna mreža, interpolacioni polinom stupnja 3.



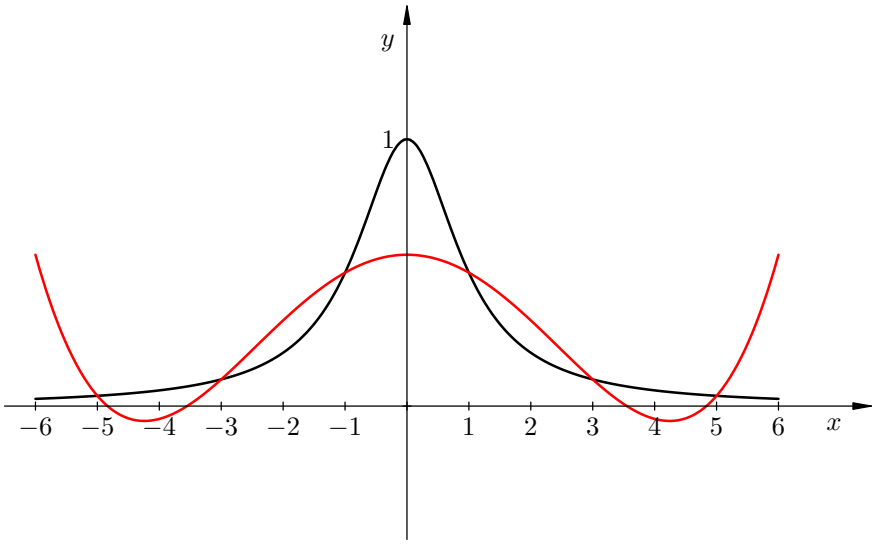
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 3.



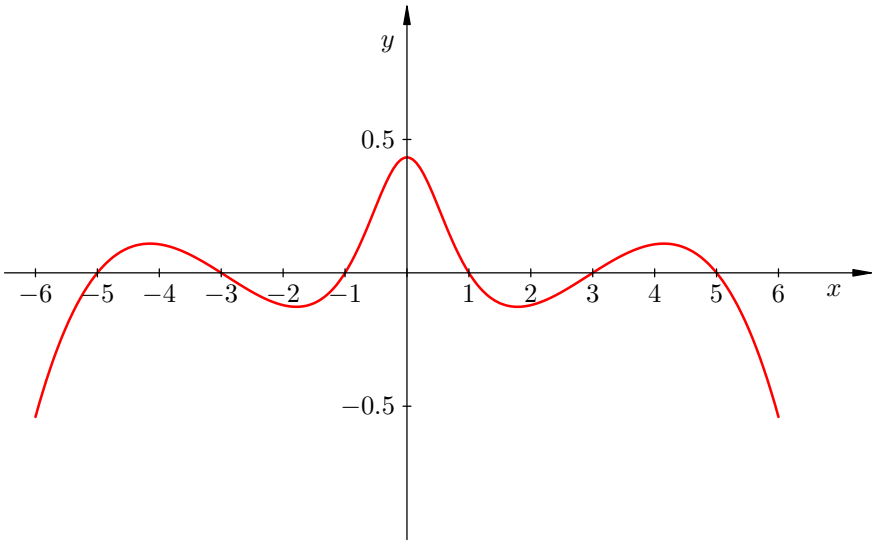
Ekvidistantna mreža, interpolacioni polinom stupnja 4.



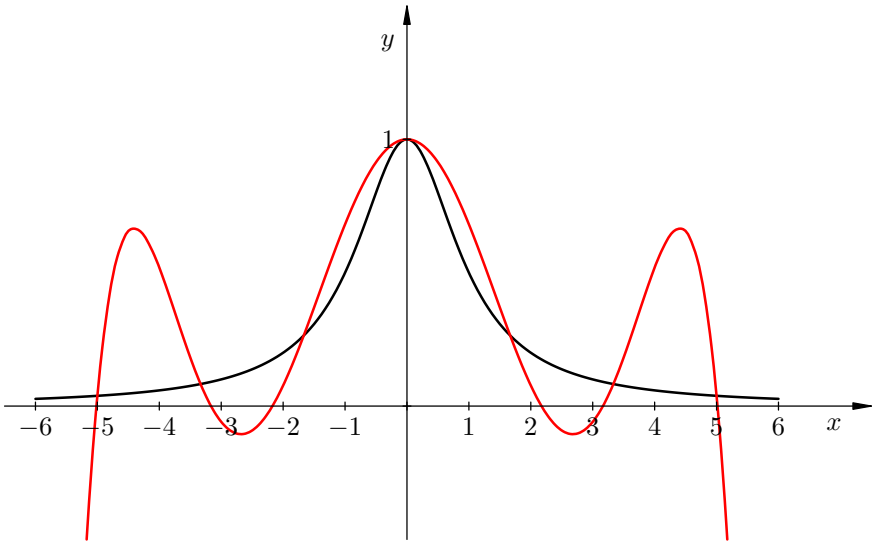
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 4.



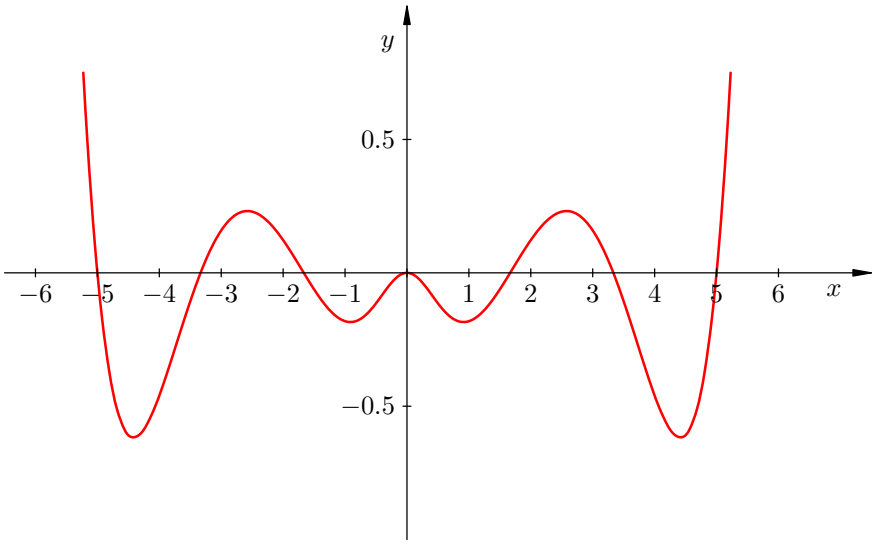
Ekvidistantna mreža, interpolacioni polinom stupnja 5.



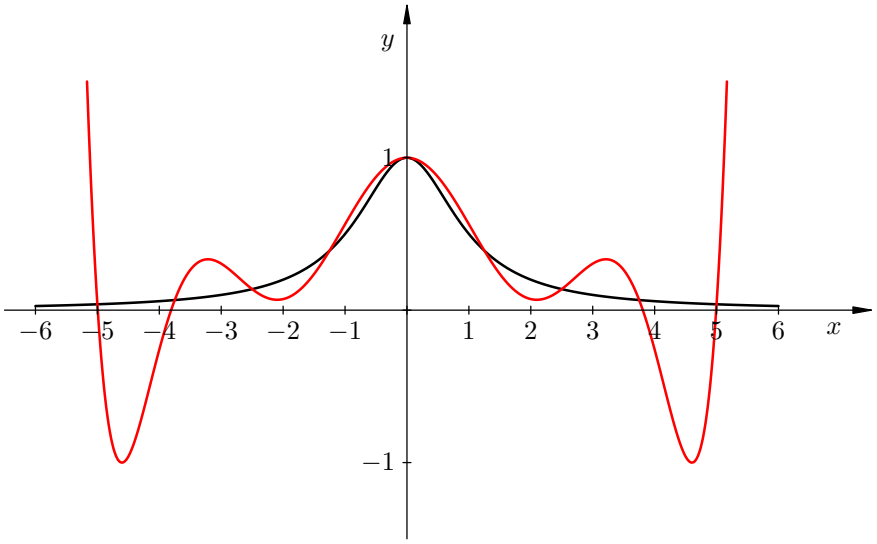
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 5.



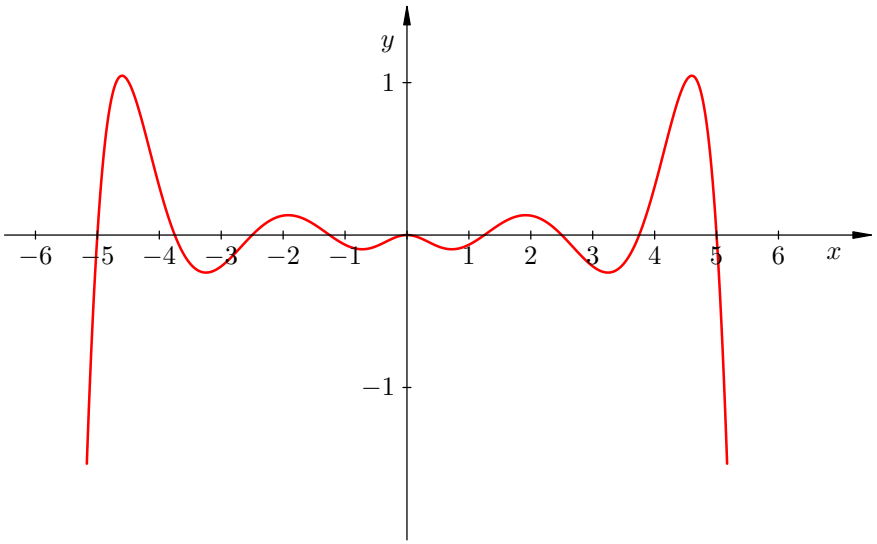
Ekvidistantna mreža, interpolacioni polinom stupnja 6.



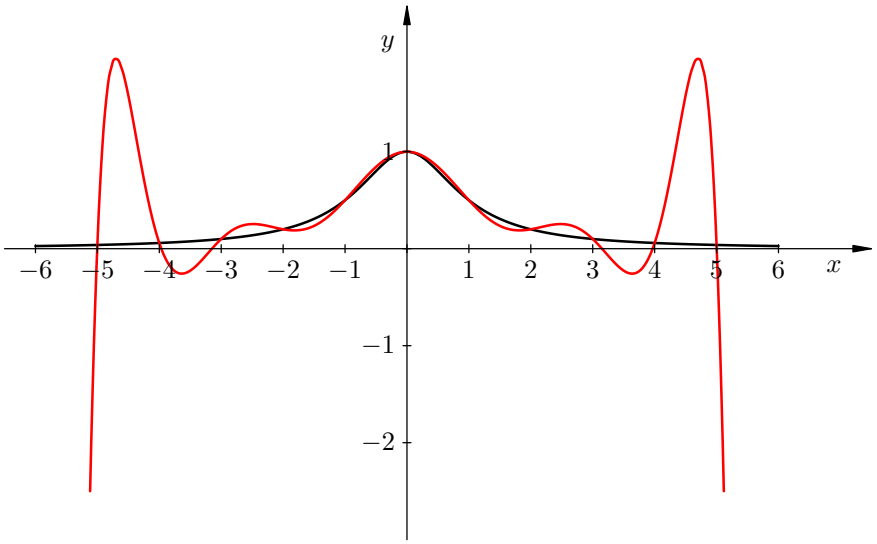
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 6.



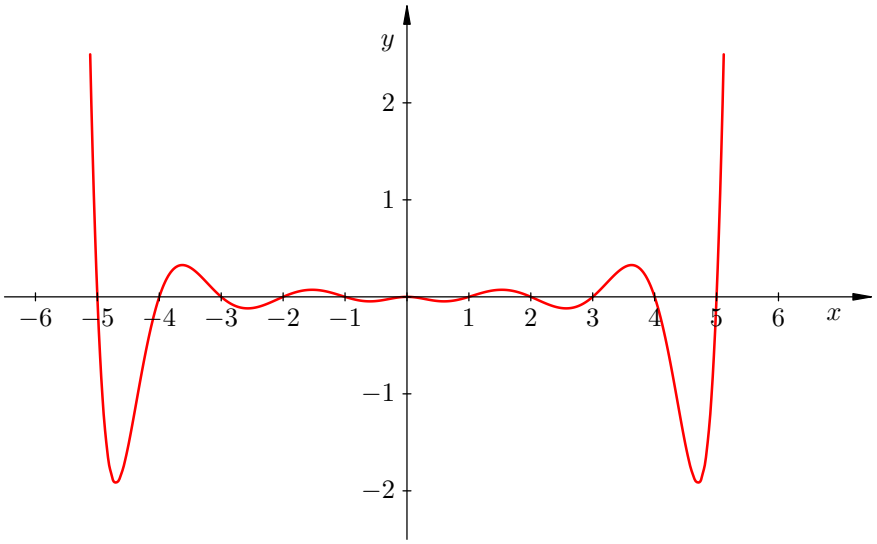
Ekvidistantna mreža, interpolacioni polinom stupnja 8.



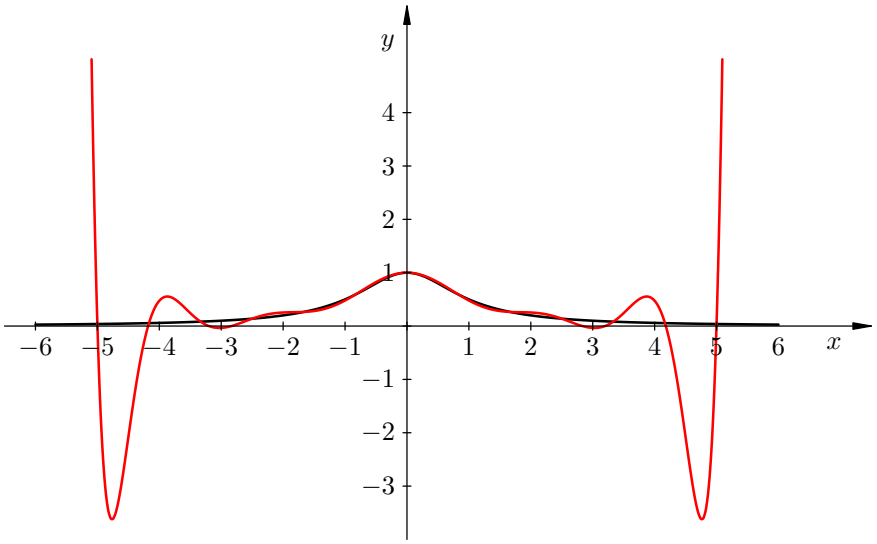
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 8.



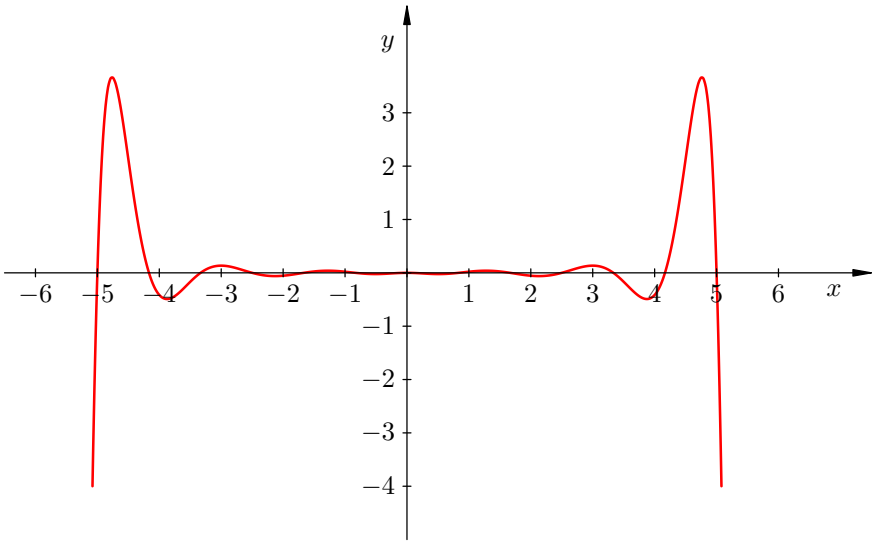
Ekvidistantna mreža, interpolacioni polinom stupnja 10.



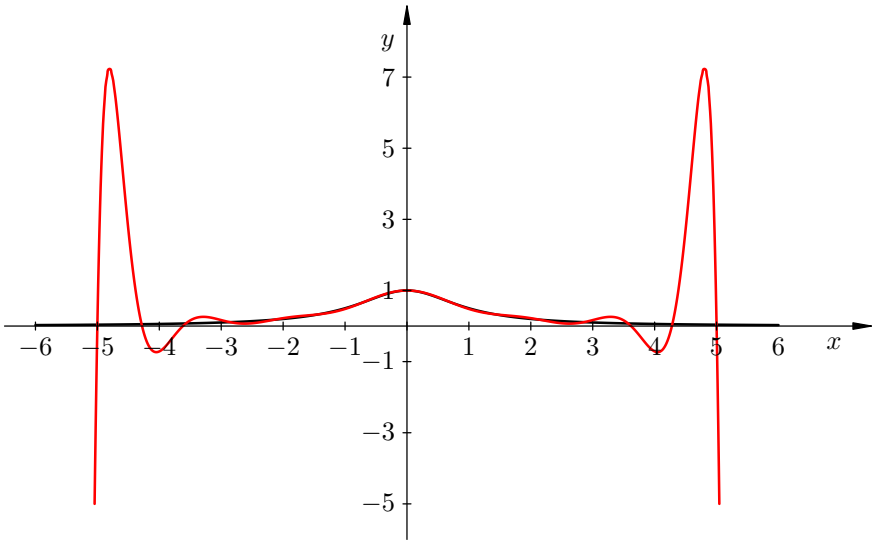
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 10.



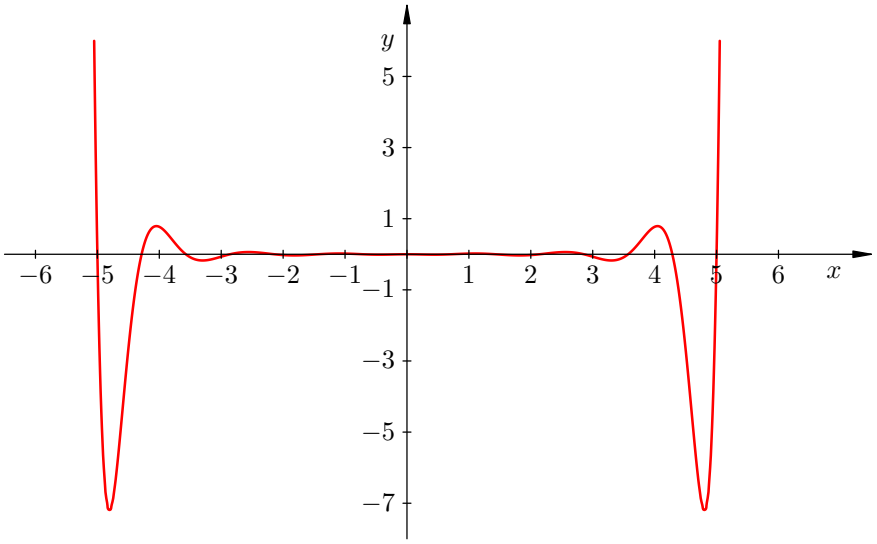
Ekvidistantna mreža, interpolacioni polinom stupnja 12.



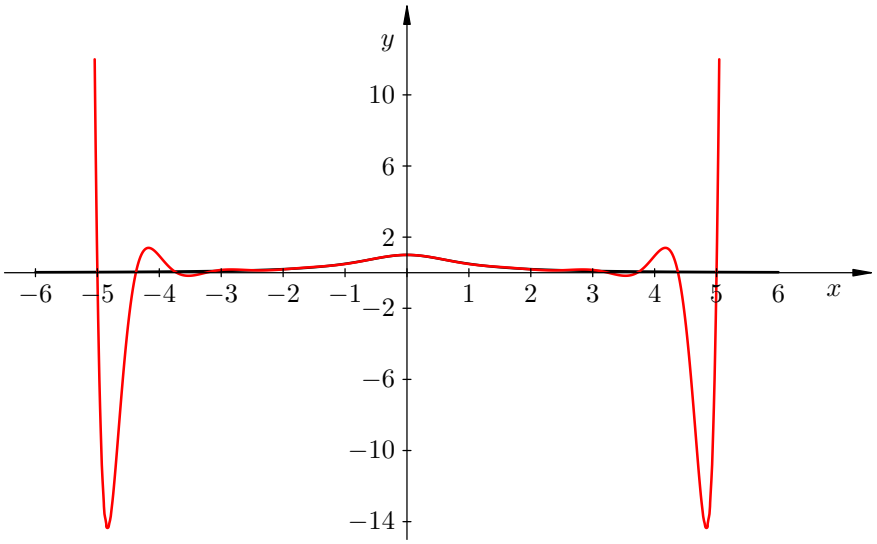
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 12.



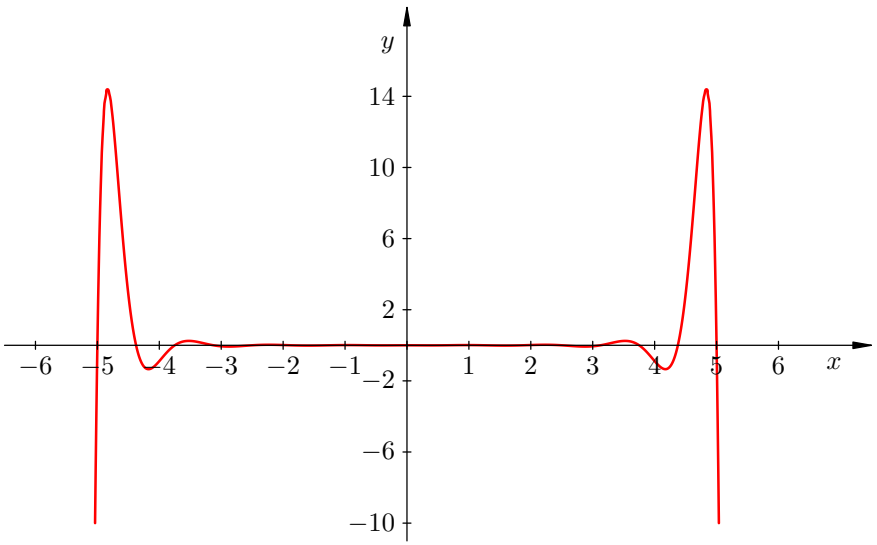
Ekvidistantna mreža, interpolacioni polinom stupnja 14.



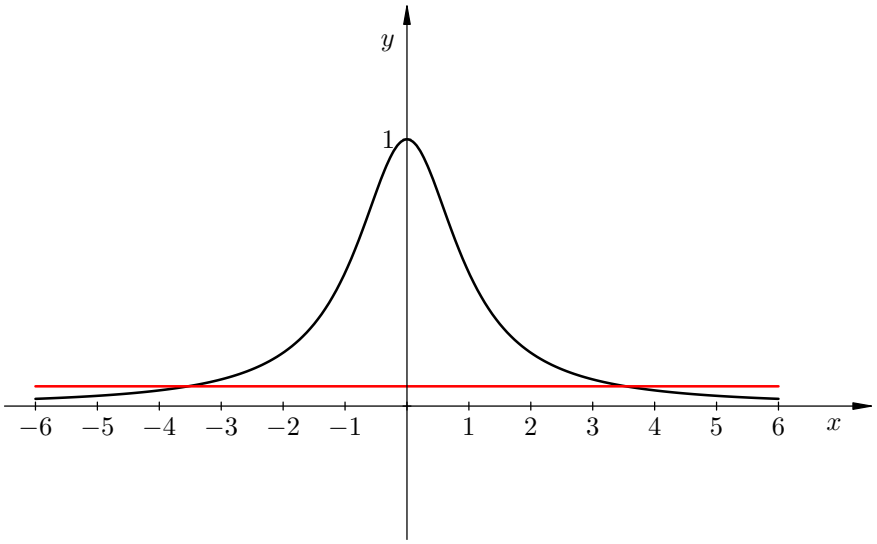
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 14.



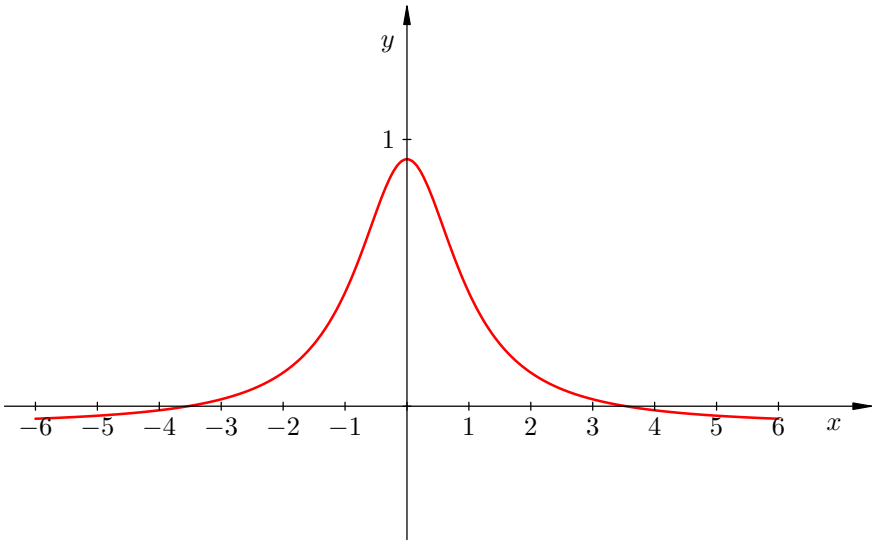
Ekvidistantna mreža, interpolacioni polinom stupnja 16.



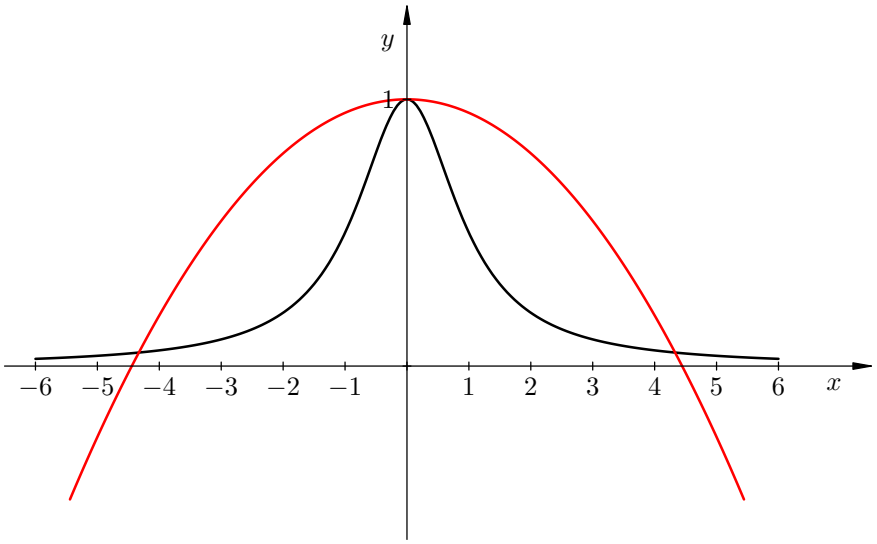
Ekvidistantna mreža, greška interpolacionog polinoma stupnja 16.



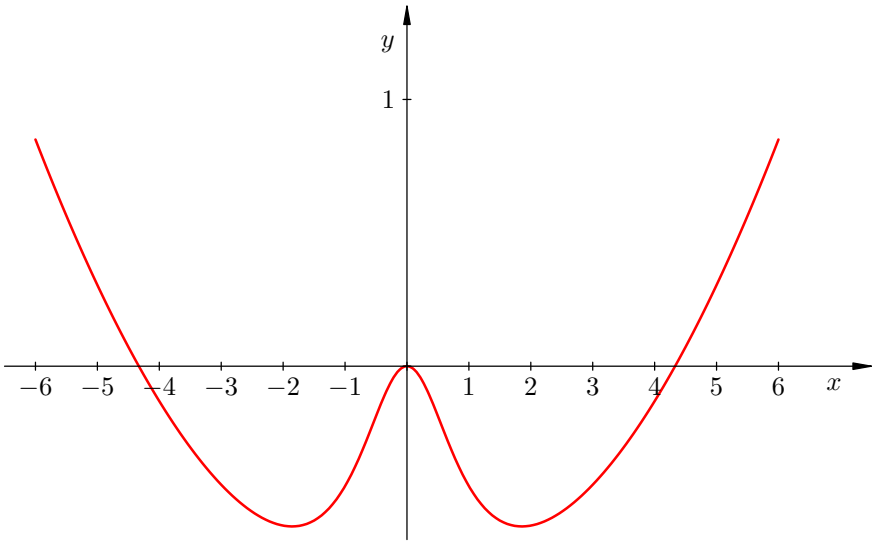
Čebiševljeva mreža, interpolacioni polinom stupnja 1.



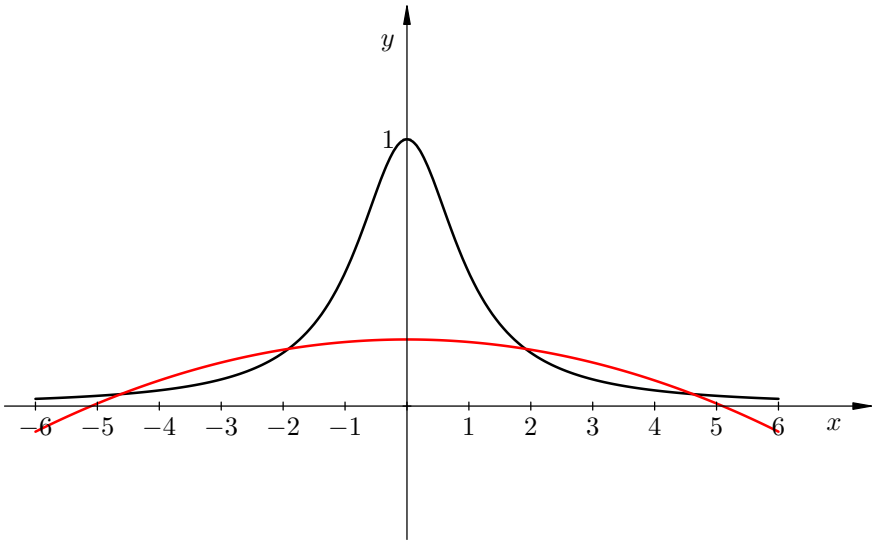
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 1.



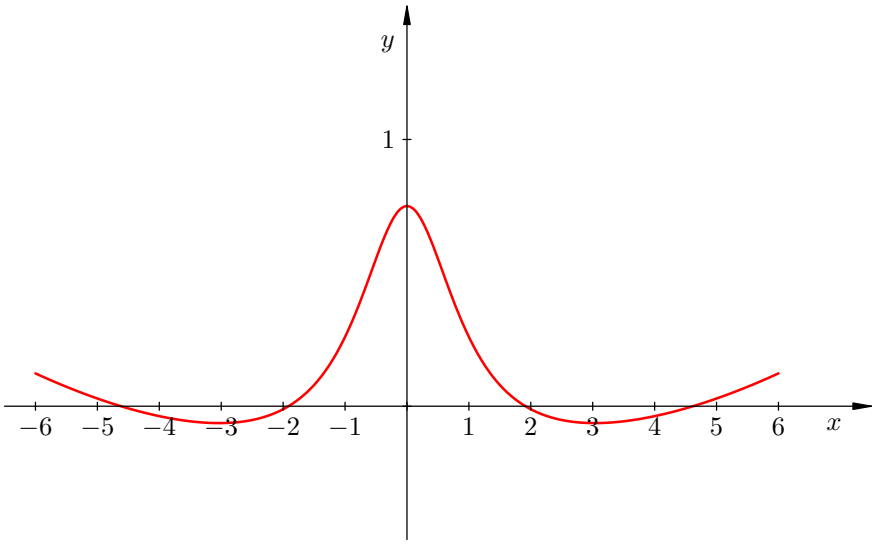
Čebiševljeva mreža, interpolacioni polinom stupnja 2.



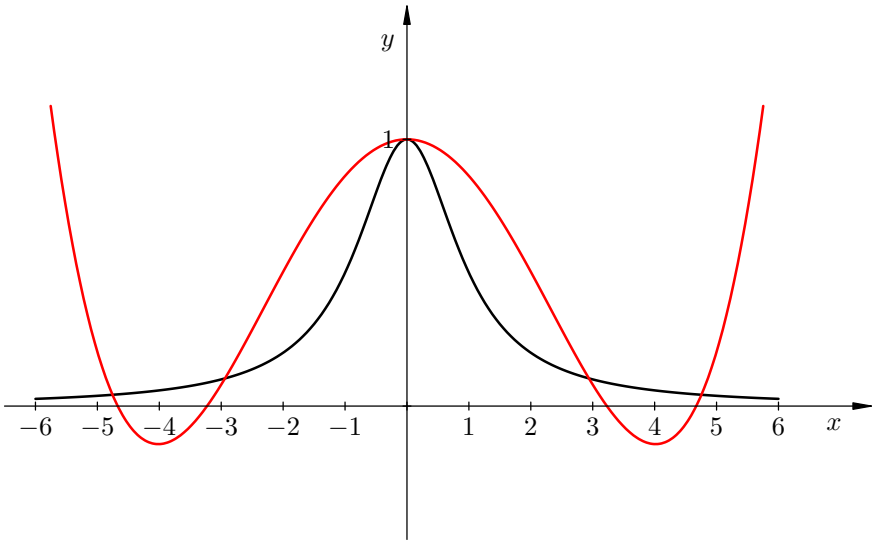
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 2.



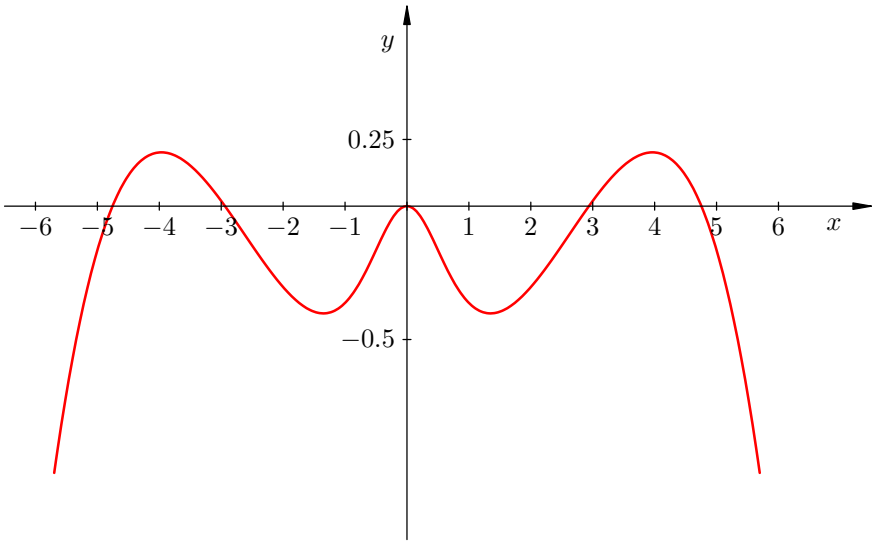
Čebiševljeva mreža, interpolacioni polinom stupnja 3.



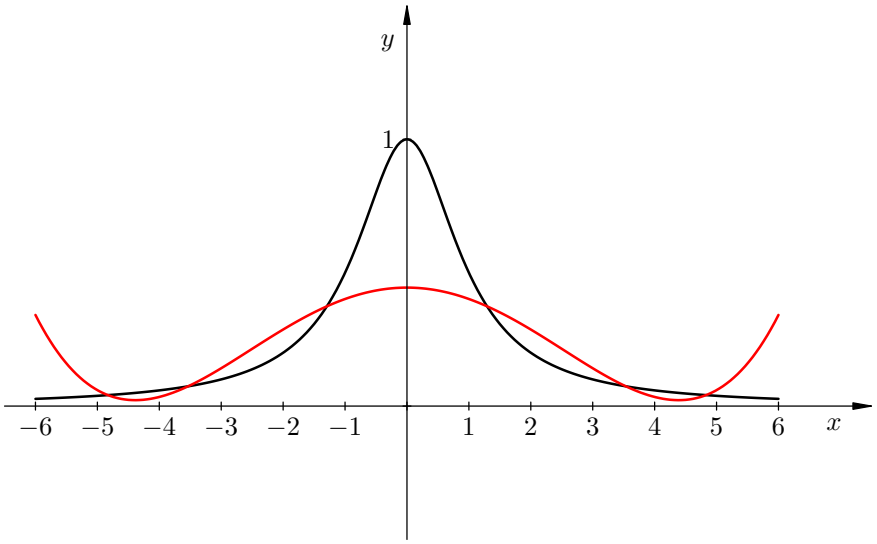
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 3.



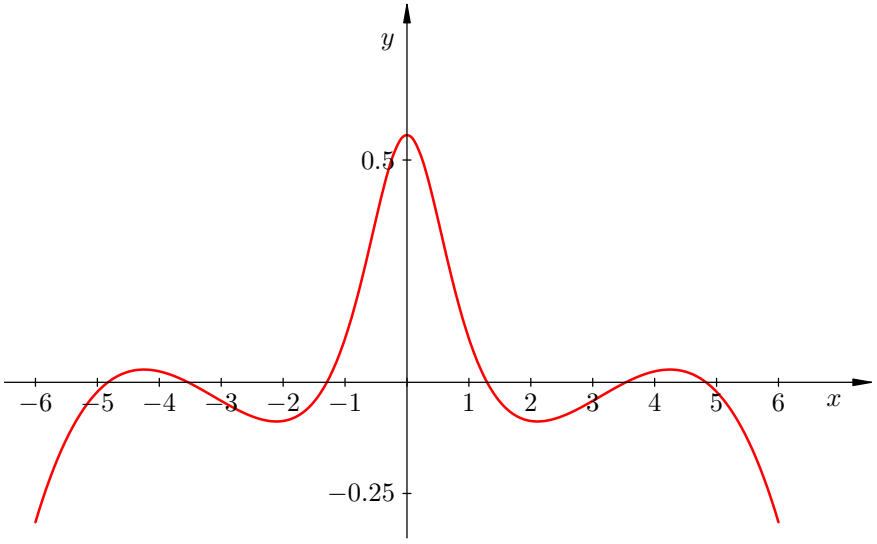
Čebiševljeva mreža, interpolacioni polinom stupnja 4.



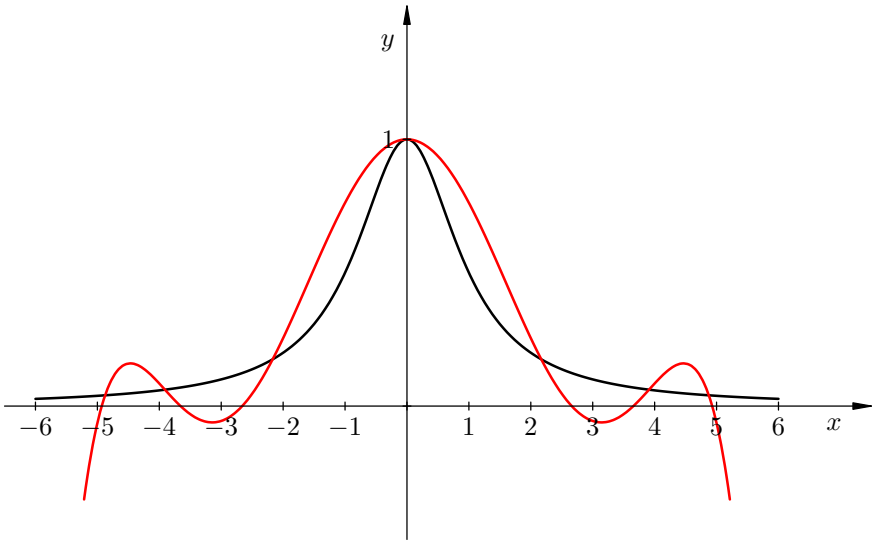
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 4.



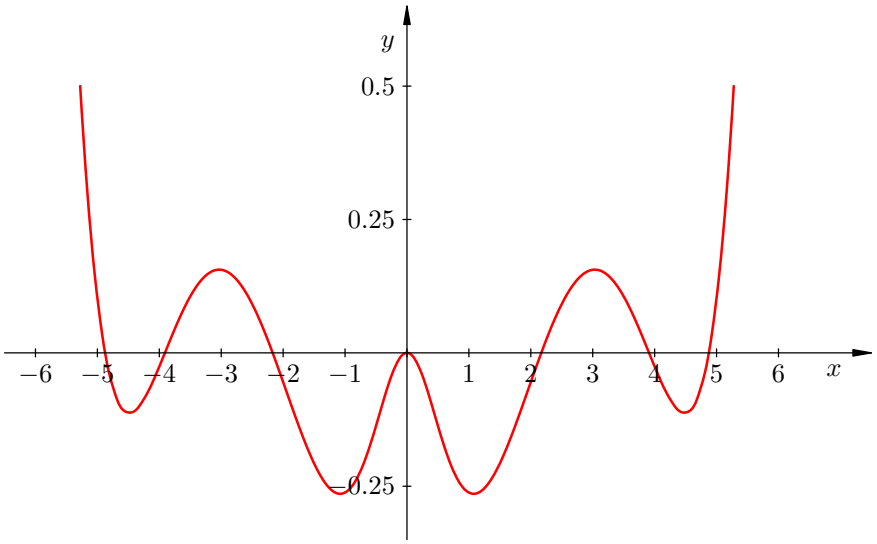
Čebiševljeva mreža, interpolacioni polinom stupnja 5.



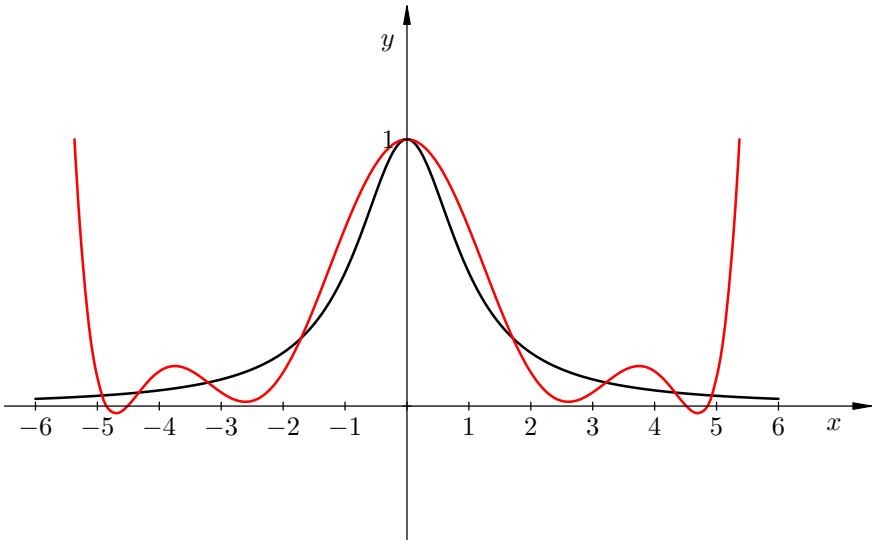
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 5.



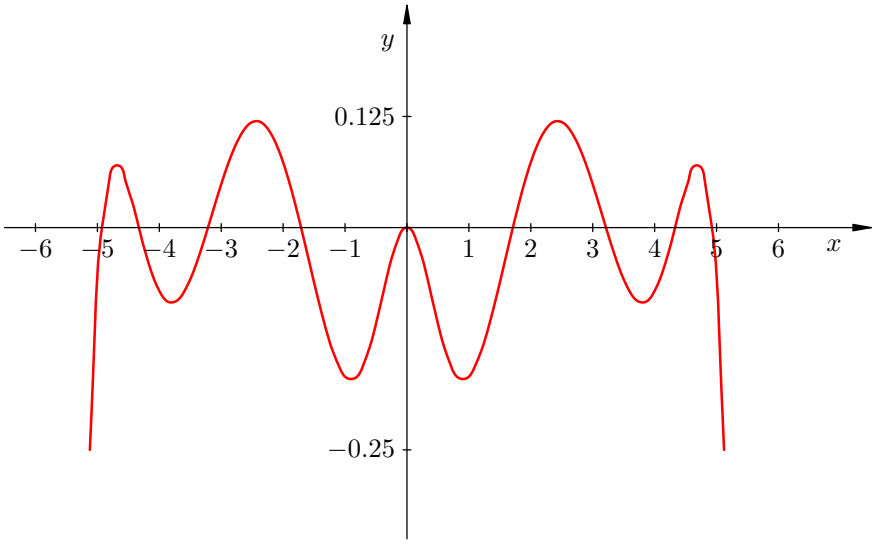
Čebiševljeva mreža, interpolacioni polinom stupnja 6.



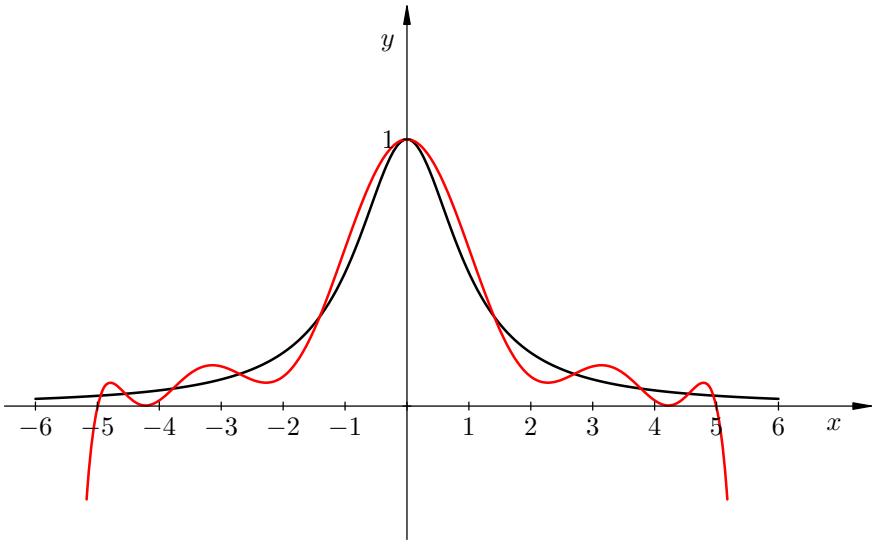
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 6.



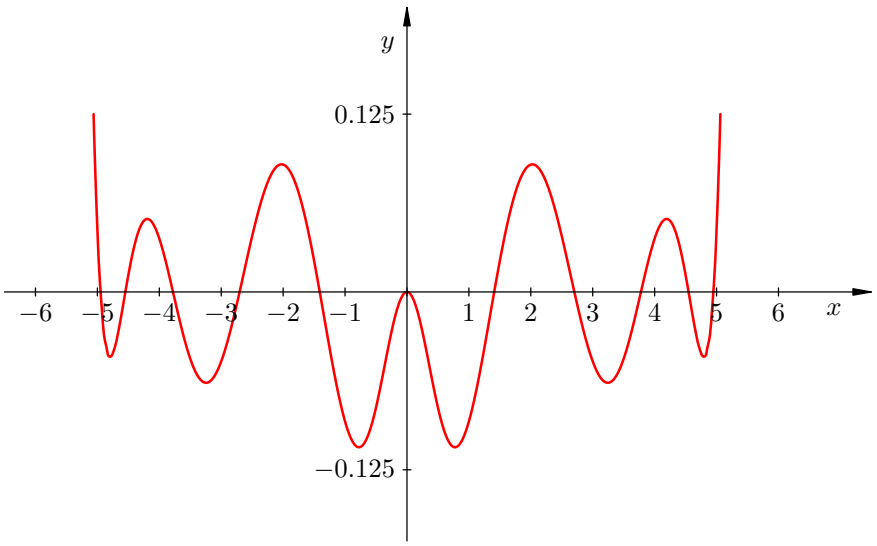
Čebiševljeva mreža, interpolacioni polinom stupnja 8.



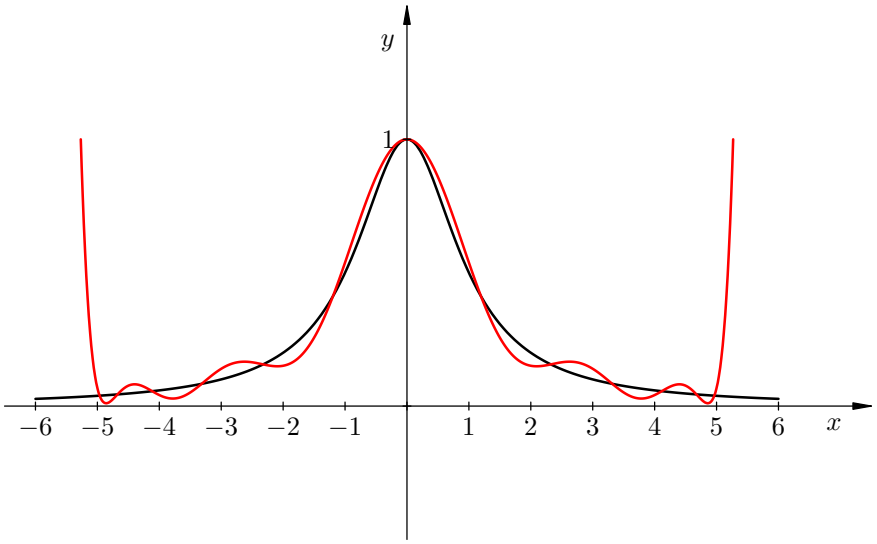
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 8.



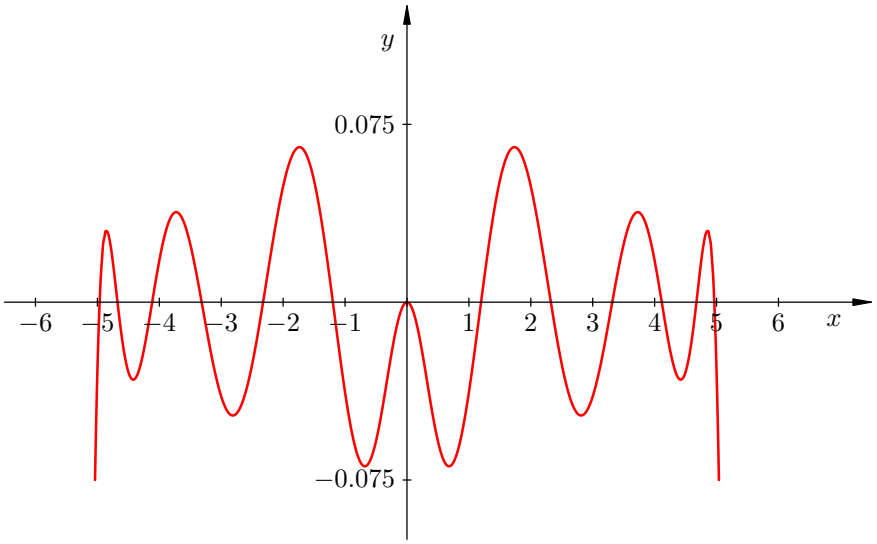
Čebiševljeva mreža, interpolacioni polinom stupnja 10.



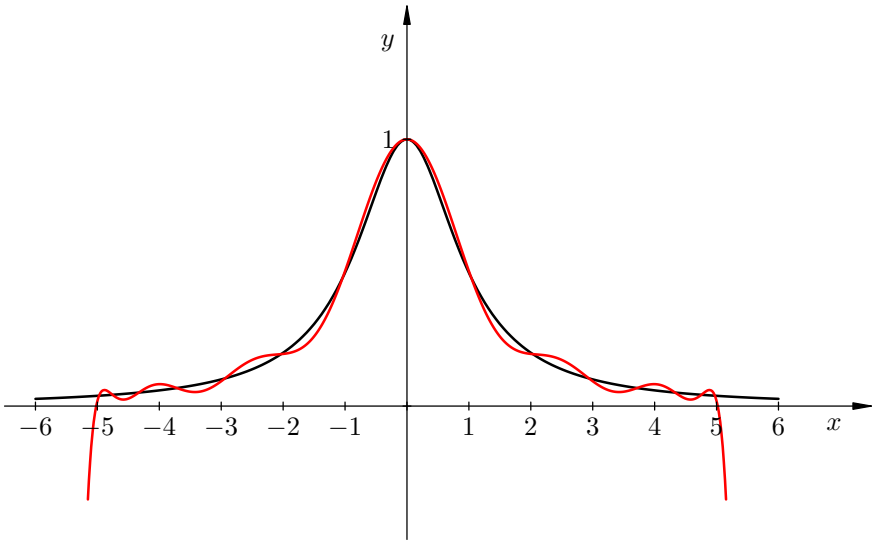
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 10.



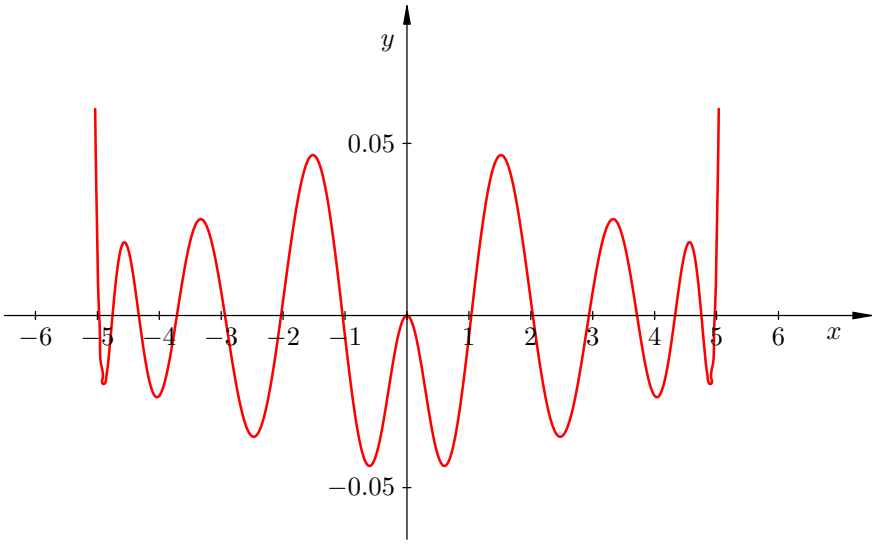
Čebiševljeva mreža, interpolacioni polinom stupnja 12.



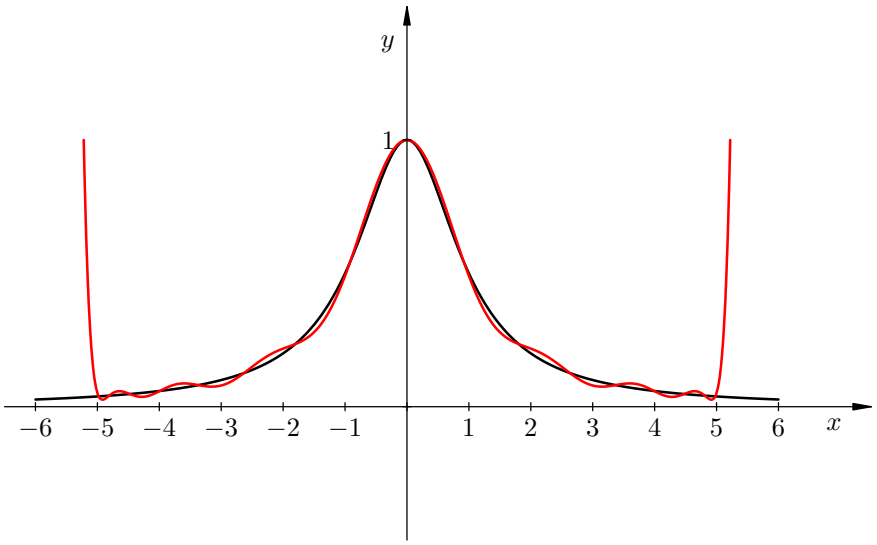
Čebiševljeva mreža, greška interpolacionog polinoma stupnja 12.



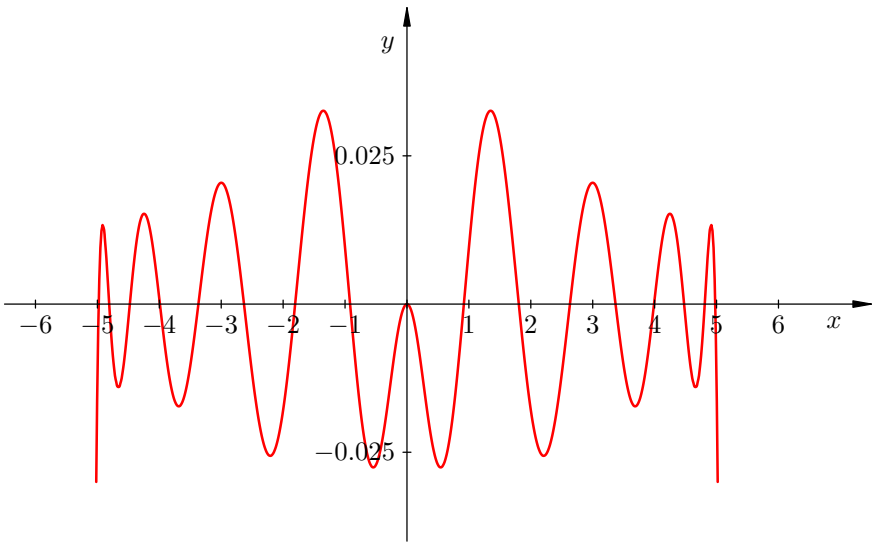
Čebiševljeva mreža, interpolacioni polinom stupnja 14.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 14.



Čebiševljeva mreža, interpolacioni polinom stupnja 16.



Čebiševljeva mreža, greška interpolacionog polinoma stupnja 16.

10.2.7. 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 10.2.4. (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 10.2.3. (Faber, 1914.) *Za svaki mogući izbor točaka interpolacije postoji neprekidna funkcija f, za čiji interpolacijski polinom $p_n(x)$ stupnja n vrijedi*

$$\|f(x) - p_n(x)\|_\infty \not\to 0.$$

10.2.8. 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

$$p_n(x_i) = f_i = f(x_i), \quad i = 0, \ldots, n. \tag{10.2.14}$$

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 10.2.1.) pokazali smo da vrijedi

$$V(x_0, \ldots, x_n) := \det \begin{pmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{pmatrix} = \prod_{0 \le i < j \le n} (x_j - x_i). \tag{10.2.15}$$

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

$$\ell_i(x_j) := \delta_{ij}, \tag{10.2.16}$$

gdje je δ_{ij} Kroneckerov simbol, tada Lagrangeov interpolacijski polinom koji udovoljava uvjetima (10.2.14) možemo zapisati kao

$$p_n(x) = \sum_{i=0}^{n} f(x_i)\ell_i(x). \tag{10.2.17}$$

Tražene polinome ℓ_i stupnja n, koji su jednoznačno određeni interpolacijskim uvjetima (10.2.16), možemo "pogoditi"

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \ne i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad i = 0, \ldots, n. \tag{10.2.18}$$

Funkcije ℓ_i zovu se funkcije **Lagrangeove baze**.

Zadatak 10.2.2. *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 10.2.4. *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

$$\begin{aligned} h_{i,0}(x) &= [1 - 2(x - x_i)\ell_i'(x_i)]\,\ell_i^2(x) \\ h_{i,1}(x) &= (x - x_i)\,\ell_i^2(x), \end{aligned} \tag{10.2.19}$$

gdje su ℓ_i funkcije Lagrangeove baze (10.2.18). Direktno možemo provjeriti da su $h_{i,0}(x)$ i $h_{i,1}(x)$ polinomi stupnja $2n + 1$ koji zadovoljavaju sljedeće relacije

$$\begin{aligned} h_{i,0}(x_j) &= \delta_{ij}, \quad & h_{i,1}(x_j) &= 0, \\ h'_{i,0}(x_j) &= 0, \quad & h'_{i,1}(x_j) &= \delta_{ij}, \end{aligned} \qquad \text{za} \quad i, j = 0, \ldots, n.$$

Ako definiramo polinom formulom

$$h_{2n+1}(x) = \sum_{i=0}^{n} \Big(f_i\, h_{i,0}(x) + f_i'\, h_{i,1}(x) \Big), \tag{10.2.20}$$

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 10.2.3. *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 10.2.4. *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 10.2.2.).

Teorem 10.2.5. *Greška kod interpolacije Hermiteovim polinomom $h_{2n+1}(x)$ (v. teorem* 10.2.4.*) 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 10.2.2.

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 10.2.5. *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 (10.2.15)*, 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 egzistencija 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 10.2.6. *Zapisom polinoma P u standardnoj bazi potencija formulirajte matrično problem proširene Hermite–Birkhoffove interpolacije.*

10.3. 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

$$(m+1) \cdot n \tag{10.3.1}$$

koeficijenata.

Interpolacioni uvjeti su

$$\varphi(x_k) = f_k, \quad k = 0, \ldots, n,$$

što za svaki polinom daje po 2 uvjeta

$$\begin{aligned} p_k(x_{k-1}) &= f_{k-1} \\ p_k(x_k) &= f_k, \end{aligned} \quad k = 1, \ldots, n, \tag{10.3.2}$$

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.

10.3.1. 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 10.3.1. (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.

10.3.2. 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

$$p_k(x) = c_{0,k} + c_{1,k}(x - x_{k-1}) + c_{2,k}(x - x_{k-1})^2 + c_{3,k}(x - x_{k-1})^3 \quad \text{za} \quad x \in [x_{k-1}, x_k], \quad k = 1, \ldots, n. \tag{10.3.3}$$

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

$$\begin{aligned} p_k(x) = f[x_{k-1}] &+ f[x_{k-1}, x_{k-1}] \cdot (x - x_{k-1}) \\ &+ f[x_{k-1}, x_{k-1}, x_k] \cdot (x - x_{k-1})^2 \\ &+ f[x_{k-1}, x_{k-1}, x_k, x_k] \cdot (x - x_{k-1})^2 (x - x_k) \end{aligned} \tag{10.3.4}$$

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 (10.3.3) i (10.3.4), dobit ćemo koeficijente $c_{i,k}$. Jednadžbu (10.3.4) 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 (10.3.4) 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 (10.3.3), 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.

10.3.3. 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.

10.3.4. 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 (10.2.5)) 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

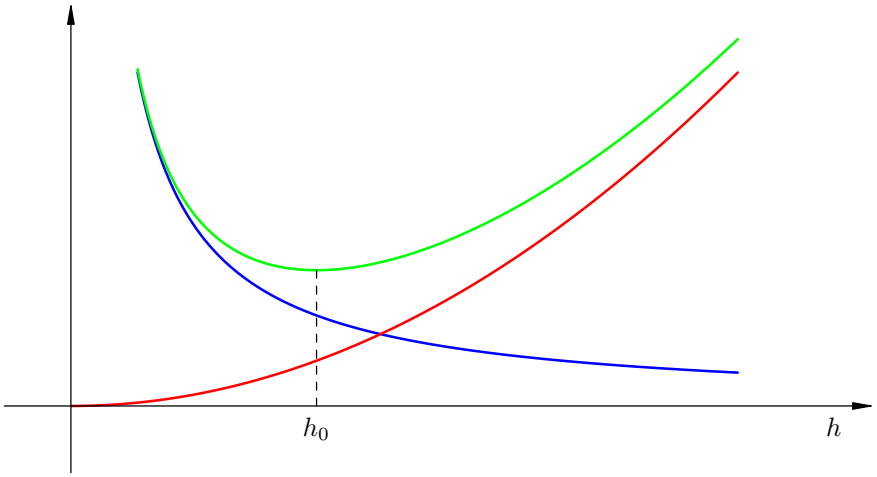
$$|err_2| \leq \frac{\varepsilon}{h} + \frac{M_3}{6}h^2, \quad M_3 = \max_{x\in[x_{-1},x_1]} |f^{(3)}(x)|.$$

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?

10.3.5. 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

$$\begin{aligned} s_n = p'_{2,n-1}(x_n) &= f[x_{n-2}, x_{n-1}](x_n - x_{n-2}) + f[x_{n-2}, x_{n-1}, x_n](x_n - x_{n-2}) \\ &\quad + f[x_{n-2}, x_{n-1}, x_n](x_n - x_{n-1}) \\ &= \frac{(h_{n-1} + 2h_n)\,f[x_{n-2}, x_{n-1}] - h_n\,f[x_{n-1}, x_n]}{h_{n-1} + h_n}. \end{aligned}$$

Dakle, za Besselovu po dijelovima kubičnu interpolaciju stavljamo

$$s_k = \begin{cases} \dfrac{(2h_1 + h_2)\,f[x_0, x_1] - h_1\,f[x_1, x_2]}{h_1 + h_2}, & \text{za } k = 0, \\ \dfrac{h_{k+1}\,f[x_{k-1}, x_k] + h_k\,f[x_k, x_{k+1}]}{h_k + h_{k+1}}, & \text{za } k = 1, \ldots, n-1, \\ \dfrac{(h_{n-1} + 2h_n)\,f[x_{n-2}, x_{n-1}] - h_n\,f[x_{n-1}, x_n]}{h_{n-1} + h_n}, & \text{za } k = n. \end{cases}$$

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.

10.3.6. 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 (10.3.3) 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

$$c_{2,k} + 3c_{3,k}(x_k - x_{k-1}) = c_{2,k+1}. \tag{10.3.5}$$

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 (10.3.5), 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 $k=1,\ldots,n-1$, 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 $s_1,\ldots s_{n-1}$. 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

$$(h_{n-1}+h_n)s_{n-1}+h_{n-1}s_n=\frac{(h_n+2(h_{n-1}+h_n))h_{n-1}\,f[x_{n-1},x_n]+h_n^2\,f[x_{n-2},x_{n-1}]}{h_{n-1}+h_{n-2}}.$$

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 10.3.1. *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 10.3.2. *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* $O(h^2)$.

10.4. 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

$$a = x_0 < x_1 < x_2 < \cdots < x_{N-1} < x_N = b. \tag{10.4.1}$$

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 10.4.1. *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 $\dim \mathcal{S}(n) = n + N$.

10.4.1. 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 (10.4.1). 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 10.4.1. *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 10.4.1. (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 (10.4.1.) greška je

$$E(x) := S_1(x) - f(x) = (1 - t)f_i + tf_{i+1} - f(x), \quad x \in [x_i, x_{i+1}]. \tag{10.4.2}$$

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 10.4.1. 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 (10.4.2) 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 (10.4.2) 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 (10.4.2) 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

$$|E(x)| \leq \frac{1}{2}\,h_i^2\,t(1-t)\,\|D^2 f\|_\infty \leq \frac{1}{8}\,\bar{h}^2\,\|D^2 f\|_\infty.$$

■

Zadatak 10.4.2. *Dokažite da se u slučajevima* (3) *i* (4) *teorema* 10.4.1. *može ocijeniti i greška u derivaciji, točnije, da vrijedi:*

(3) $\|DS_1(x) - Df(x)\|_\infty \leq \omega(Df)$;

(4) $\|DS_1(x) - Df(x)\|_\infty \leq \dfrac{\bar{h}}{2}\,\|D^2 f\|_\infty$.

Teoremi poput teorema 10.4.1. 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 10.4.3. *Pokažite da u slučaju* (4) *teorema* 10.4.1. *funkcija* $f(x) = x^2$ *igra ulogu* **ekstremale**, *tj. da vrijedi "*$=$*" umjesto "*$\leq$*", pa je ocjena ujedno i najbolja moguća.*

Zadatak 10.4.4. *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**.

10.4.2. 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 10.4.1. *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 10.2.4. na svakom podintervalu mreže $[x_i, x_{i+1}]$, lagano vidimo da vrijedi

$$H(x) = \varphi_1(t) f_i + \varphi_2(t) f_{i+1} + \varphi_3(t) h_i f_i' + \varphi_4(t) h_i f_{i+1}', \quad t = \frac{x - x_i}{h_i}, \tag{10.4.3}$$

gdje je

$$\begin{aligned} \varphi_1(t) &= (1-t)^2(1+2t), & \varphi_2(t) &= t^2(3-2t), \\ \varphi_3(t) &= t(1-t^2), & \varphi_4(t) &= -t^2(1-t). \end{aligned}$$

Napomenimo još samo da kod računanja treba prvo izračunati koeficijente A_i i B_i formulama

$$\begin{aligned} A_i &= -2\,\frac{f_{i+1} - f_i}{h_i} + (f_i' + f_{i+1}'), \\ B_i &= -A_i + \frac{f_{i+1} - f_i}{h_i} - f_i', \end{aligned} \qquad \text{za} \quad i = 0, \ldots, N-1, \tag{10.4.4}$$

i zapamtiti ih. Za zadanu točku $x \in [x_i, x_{i+1}]$, Hermiteov splajn računamo formulom

$$H(x) = f_i + (th_i)\,[f_i' + t(B_i + tA_i)]. \tag{10.4.5}$$

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 10.4.2. *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 $\xi, \eta, \ldots$ označavaju točke u $[x_i, x_{i+1}]$. Prema Lemi 10.4.1. 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

$$\begin{aligned} E(x) = &\int_{x_i}^{x} (1-t)^2\,[-th_i + (1+2t)(v-x_i)]\,D^2 f(v)\,dv \\ &+ \int_{x}^{x_{i+1}} t^2\,[-(1-t)h_i + (3-2t)(x_{i+1}-v)]\,D^2 f(v)\,dv. \end{aligned}$$

Zamjenom varijable $v - x_i = \tau h_i$ dobivamo

$$E(x) = h_i^2 \left\{ \int_0^t \psi_1(t,\tau)\,D^2 f(x_i+\tau h_i)\,d\tau + \int_t^1 \psi_2(t,\tau)\,D^2 f(x_i+\tau h_i)\,d\tau \right\}, \quad (10.4.6)$$

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.

$$\begin{aligned} \int_0^t \psi_1(t,\tau)\,D^2 f(x_i+\tau h_i)\,d\tau &= D^2 f(\xi) \int_0^{\tau^*} \psi_1(t,\tau)\,d\tau + D^2 f(\eta) \int_{\tau^*}^t \psi_1(t,\tau)\,d\tau \\ &= \frac{t^2(1-t)^2}{2(1+2t)}\,\{4t^2\,D^2 f(\eta) - D^2 f(\xi)\}. \end{aligned}$$

Analogno

$$\int_0^t \psi_2(t,\tau)\,D^2 f(x_i+\tau h_i)\,d\tau = \frac{(1-t)^2 t^2}{2(3-2t)}\,\{4(1-t^2)\,D^2 f(\bar{\xi}) - D^2 f(\bar{\eta})\}.$$

Iz (10.4.6) 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 10.4.1. 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

$$|E(x)| \leq \frac{t^2(1-t)^2}{4!}\, h_i^4\, \|D^4 f\|_\infty, \quad t \in [0,1]. \tag{10.4.7}$$

Odavde se lagano dobije ocjena za $\|E(x)\|_\infty$. ■

Zadatak 10.4.5. *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 10.4.6. *Pokušajte za prvih pet slučajeva (klasa glatkoće funkcije* f*) iz teorema* 10.4.2. *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 10.4.7. (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.*

10.4.3. 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 10.4.2. *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:

$$\begin{array}{lll}
(R1)\ DS_3(a) = Df(a), & DS_3(b) = Df(b), & \text{(potpuni rubni uvjeti);} \\
(R2)\ D^2S_3(a) = 0, & D^2S_3(b) = 0, & \text{(prirodni rubni uvjeti);} \\
(R3)\ D^2S_3(a) = D^2f(a), & D^2S_3(b) = D^2f(b); & \\
(R4)\ DS_3(a) = DS_3(b), & D^2S_3(a) = D^2S_3(b), & \text{(periodički rubni uvjeti).}
\end{array} \tag{10.4.8}$$

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 (10.4.3), S_3 možemo na svakom podintervalu napisati u obliku

$$\begin{aligned}
S_3(x) = {} & f_i\,(1-t)^2(1+2t) + f_{i+1}\,t^2(3-2t) \\
& + m_i\,h_i\,t(1-t)^2 - m_{i+1}\,h_i\,t^2(1-t),
\end{aligned} \tag{10.4.9}$$

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

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = c_i, \quad i = 1, \ldots, N-1. \tag{10.4.10}$$

Dobili smo $N-1$ jednadžbi za $N+1$ nepoznanica m_i, pa nam fale još dvije jednadžbe. Naravno, uvjetima (10.4.10) treba dodati još neke rubne uvjete.

Za rubne uvjete $(R1)$, $(R2)$ i $(R3)$ dobivamo linearni sustav oblika

$$\begin{aligned} 2m_0 + \mu_0^*\, m_1 &= c_0^*, \\ \lambda_i\, m_{i-1} + 2m_i + \mu_i\, m_{i+1} &= c_i, \qquad i = 1, \ldots, N-1, \\ \lambda_N^*\, m_{N-1} + 2m_N &= c_N^*. \end{aligned} \tag{10.4.11}$$

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 (10.4.10) 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

$$\begin{aligned} 2m_1 + \mu_1 m_2 + \lambda_1 m_N &= c_1, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} &= c_i, \qquad i = 2, \ldots, N-1, \\ \mu_N m_1 + \lambda_N m_{N-1} + 2m_N &= c_N. \end{aligned} \tag{10.4.12}$$

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 10.4.3. *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 10.4.1. (Interpolacijski kubični splajn)

(1) Riješi linearni sustav (10.4.11) ili (10.4.12);
(2) Binarnim pretraživanjem nađi indeks i tako da vijedi $x \in [x_i, x_{i+1})$;
(3) Hornerovom shemom (10.4.5) 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 10.4.1. sastoji od LR faktorizacije matrice, koju treba izračunati samo jednom.

Za računanje vrijednosti $S_3(x)$ obično se koriste formule (10.4.4)–(10.4.5). 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 10.4.2.) i oblika danog formulama (10.4.4)–(10.4.5). Definiciona reprezentacija splajna kao kubične funkcije na svakom podintervalu subdivizije zove se ponekad i **po dijelovima polinomna** reprezentacija, ili skraćeno PP–reprezentacija.

Zadatak 10.4.8. *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

$$\mu_i\,M_{i-1} + 2M_i + \lambda_i\,M_i = \frac{6}{h_{i-1} + h_i}\left(\frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}\right), \tag{10.4.13}$$

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 10.4.9. *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 (10.4.4)–(10.4.5) za algoritam 10.4.1.):

$$S_3(x) = f_i + t\,\{(f_{i+1} - f_i) - (x_{i+1} - x)\,[(x_{i+1} - x + h_i)\widetilde{M}_i + (h_i + x - x_i)\widetilde{M}_{i+1}]\},$$

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 10.4.4. *Neka je S_3 interpolacijski kubični splajn za funkciju f koji zadovoljava jedan od rubnih uvjeta $(R1)$–$(R4)$ u (10.4.8). 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 10.4.4. (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. (10.4.9), (10.4.3)), 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

$$|S(x) - f(x)| \le |H(x) - f(x)| + h_i\,t(1-t)\,\max_i |m_i - Df(x_i)|. \tag{10.4.14}$$

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

$$\begin{aligned} |DS(x) - Df(x)| \le |DH(x) - Df(x)| \\ + [(1-t)|1-3t| + t|2-3t|]\,\max_i |m_i - Df(x_i)|. \end{aligned} \tag{10.4.15}$$

Ocjene za $|H(x) - f(x)|$ izveli smo u teoremu 10.4.4., a ocjene za $|DH(x) - Df(x)|$ mogu se izvesti na sličan način (v. zadatke 10.4.5. i 10.4.6.). Ostaje dakle ocijeniti drugi član na desnoj strani u (10.4.14) i (10.4.15).

Za drugu derivaciju znamo da je

$$D^2S(x) = M_i\,(1-t) + M_{i+1}\,t,$$

pa zaključujemo da je

$$D^2S(x) - D^2f(x) = (1-t)\,(M_i - D^2f(x_i)) + t\,(M_{i+1} - D^2f(x_{i+1}) \\ + (1-t)\,D^2f(x_i) + t\,D^2f(x_{i+1}) - D^2f(x).$$

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 10.4.1.), možemo ju i ovako ocijeniti

$$|D^2S(x) - D^2f(x)| \le |S_1(x) - D^2f(x)| + \max_i |M_i - D^2f(x_i)|. \tag{10.4.16}$$

Slično je i za treću derivaciju

$$|D^3S(x) - D^3f(x)| \le |DS_1(x) - D^3f(x)| + \frac{2}{\min_i h_i} \max_i |M_i - D^2f(x_i)|. \tag{10.4.17}$$

Ako pogledamo nejednakosti (10.4.14), (10.4.15), (10.4.16) i (10.4.17), 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 (10.4.11) nove nepoznanice $q_i := m_i - Df(x_i)$, dobijemo sustav

$$\begin{aligned} q_0 &= 0, \\ \lambda_i\,q_{i-1} + 2q_i + \mu_i\,q_{i+1} &= \tilde{c}_i, \qquad i = 1, \ldots, N-1, \\ q_N &= 0, \end{aligned}$$

gdje su desne strane

$$\begin{aligned}\tilde{c}_i = 3\mu_i \frac{f_{i+1}-f_i}{h_i} + 3\lambda_i \frac{f_i - f_{i-1}}{h_{i-1}} \\ - \lambda_i\, Df(x_{i-1}) - 2Df(x_i) - \mu_i\, Df(x_{i+1}).\end{aligned} \tag{10.4.18}$$

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 $\lambda_i + \mu_i \le 1$ (jednakost vrijedi u svim jednadžbama, osim prve i zadnje), slijedi $\|B\|_\infty \le 1$. Sada nije teško ocijeniti $\|A^{-1}\|_\infty$

$$A = 2\Big(I + \frac{1}{2}\,B\Big) \implies \|A^{-1}\|_\infty \le \frac{1}{2}\Big(1 - \frac{1}{2}\,\|B\|_\infty\Big)^{-1} \le 1.$$

Na kraju, iz $q = A^{-1}\tilde{c}$ slijedi

$$|q_i| \le \|q\|_\infty \le \|A^{-1}\|_\infty \|\tilde{c}\|_\infty = \max_i |\tilde{c}_i|.$$

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 (10.4.18) 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

$$|\tilde{c}_i| \le 3(\lambda_i + \mu_i)\,\omega(Df) = 3\,\omega(Df),$$

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 (10.4.18) 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

$$\begin{aligned}\tilde{c}_i = 3\mu_i \Big\{ &Df(x_i) + \frac{h_i}{2}\,D^2 f(x_i) + \frac{h_i^2}{6}\,D^3 f(x_i + 0) \\ &+ \frac{1}{6h_i} \int_{x_i}^{x_{i+1}} (x_{i+1} - v)^3\, D^4 f(v)\, dv \Big\} \\ + 3\lambda_i \Big\{ &Df(x_i) - \frac{h_{i-1}}{2}\,D^2 f(x_i) + \frac{h_{i-1}^2}{6}\,D^3 f(x_i - 0) \\ &- \frac{1}{6h_{i-1}} \int_{x_i}^{x_{i-1}} (x_{i-1} - v)^3\, D^4 f(v)\, dv \Big\}\end{aligned}$$

$$
\begin{aligned}
&- \mu_i \left\{ Df(x_i) + h_i\, D^2 f(x_i) + \frac{h_i^2}{2}\, D^3 f(x_i + 0) \right. \\
&\qquad\qquad \left. + \frac{1}{2} \int_{x_i}^{x_{i+1}} (x_{i+1} - v)^2\, D^4 f(v)\, dv \right\} \\
&- 2Df(x_i) \\
&- \lambda_i \left\{ Df(x_i) - h_{i-1}\, D^2 f(x_i) + \frac{h_{i-1}^2}{2}\, D^3 f(x_i - 0) \right. \\
&\qquad\qquad \left. + \frac{1}{2} \int_{x_i}^{x_{i-1}} (x_{i-1} - v)^2\, D^4 f(v)\, dv \right\}.
\end{aligned}
$$

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. 10.4.10) 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

$$|\tilde{c}_i| \leq \frac{1}{24}\,\bar{h}^3\,\|D^4 f\|_\infty.$$

Upotrebom Taylorove formule, teorema o srednjoj vrijednosti i leme 10.4.1., 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 (10.4.13), 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 (10.4.14):

$$|S(x) - f(x)| \leq |H(x) - f(x)| + h_i\,t(1-t)\,\max_i |m_i - Df(x_i)|.$$

Prvi dio čini greška kod interpolacije Hermiteovim splajnom, za koju, prema (10.4.7), 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 $(R1)$). ■

10.5. Diskretna metoda najmanjih kvadrata

Ponovno, neka je funkcija f zadana na diskretnom skupu točaka $x_0, \ldots, x_n$. Također, pretpostavljamo da je tih točaka mnogo više nego nepoznatih parametara aproksimacione funkcije.

Aproksimaciona funkcija

$$\varphi(x, a_0, \ldots, a_m)$$

određuje se iz uvjeta da je 2-norma vektora pogrešaka u čvorovima aproksimacije najmanja moguća, tj. tako da minimiziramo

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

Ovu funkciju S (kvadrat 2-norme vektora greške) interpretiramo kao funkciju nepoznatih parametara

$$S = S(a_0, \ldots, a_m).$$

Očito je uvijek $S \geq 0$, bez obzira kakvi su parametri. Dakle, zadatak je minimizirati funkciju S kao funkciju više varijabli $a_0, \ldots, a_m$. Ako je S dovoljno glatka funkcija, a ova je (jer je funkcija u parametrima a_k), nužni uvjet ekstrema je

$$\frac{\partial S}{\partial a_k} = 0, \quad k = 0, \ldots, m.$$

Takav pristup vodi na tzv. **sustav normalnih jednadžbi**.

10.5.1. Linearni problemi i linearizacija

Ilustrirajmo to na najjednostavnijem primjeru, kad je aproksimaciona funkcija pravac.

Primjer 10.5.1. *Zadane su točke* $(x_0, f_0), \ldots, (x_n, f_n)$*, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo pravcem*

$$\varphi(x) = a_0 + a_1 x.$$

Greška aproksimacije u čvorovima koju minimiziramo je

$$S = S(a_0, a_1) = \sum_{k=0}^{n} (f_k - \varphi(x_k))^2 = \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k)^2 \to \min .$$

Nađimo parcijalne derivacije po parametrima a_0 *i* a_1*:*

$$0 = \frac{\partial S}{\partial a_0} = -2 \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k),$$
$$0 = \frac{\partial S}{\partial a_1} = -2 \sum_{k=0}^{n} (f_k - a_0 - a_1 x_k) x_k.$$

Dijeljenjem s -2 *i sređivanjem po nepoznanicama* a_0*,* a_1*, dobivamo linearni sustav*

$$a_0 (n+1) + a_1 \sum_{k=0}^{n} x_k = \sum_{k=0}^{n} f_k$$
$$a_0 \sum_{k=0}^{n} x_k + a_1 \sum_{k=0}^{n} x_k^2 = \sum_{k=0}^{n} f_k x_k.$$

Uvedemo li standardne skraćene oznake

$$s_\ell = \sum_{k=0}^{n} x_k^\ell, \quad t_\ell = \sum_{k=0}^{n} f_k x_k^\ell, \quad \ell \geq 0,$$

onda linearni sustav možemo pisati kao

$$\begin{aligned} s_0 a_0 + s_1 a_1 &= t_0 \\ s_1 a_0 + s_2 a_1 &= t_1. \end{aligned} \tag{10.5.1}$$

Nije teško pokazati da je matrica sustava regularna, što slijedi iz linearne nezavisnosti vektora

$$(1, 1, \ldots, 1)^T \quad i \quad (x_0, x_1, \ldots, x_n)^T,$$

uz uvjet da imamo barem dvije različite točke x_k *(prirodan uvjet za pravac), pa postoji jedinstveno rješenje sistema. Samo rješenje dobiva se rješavanjem linearnog sustava (10.5.1).*

Ostaje još pitanje da li smo dobili minimum, ali i to nije teško pokazati, korištenjem drugih parcijalnih derivacija (dovoljan uvjet minimuma je pozitivna definitnost Hesseove matrice). Ipak, provjera da je to minimum, može i puno lakše. Budući da se radi o zbroju kvadrata, S *predstavlja paraboloid s otvorom prema gore u varijablama* a_0*,* a_1*, pa je jasno da takvi paraboloidi imaju minimum. Zbog toga se nikad ni ne provjerava da li je dobiveno rješenje minimum za* S*.*

Za funkciju φ mogli bismo uzeti i polinom višeg stupnja,

$$\varphi(x) = a_0 + a_1 x + \cdots + a_m x^m,$$

ali postoji opasnost da je za malo veće m ($m \approx 10$) dobiveni sustav vrlo loše uvjetovan (blizak singularnom), pa dobiveni rezultati mogu biti jako pogrešni. Zbog toga se to nikada, ovako direktno, ne radi. Ako se uopće koriste aproksimacije polinomima viših stupnjeva, onda se to radi korištenjem ortogonalnih polinoma.

Linearni model diskretnih najmanjih kvadrata je potpuno primjenjiv na opću linearnu funkciju

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

gdje su $\varphi_0, \ldots, \varphi_m$ poznate (zadane) funkcije. Ilustrirajmo to ponovno na općoj linearnoj funkciji s 2 parametra.

Primjer 10.5.2. *Zadane su točke $(x_0, f_0), \ldots, (x_n, f_n)$, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo funkcijom oblika*

$$\varphi(x) = a_0\varphi_0(x) + a_1\varphi_1(x).$$

Postupak je potpuno isti kao u prošlom primjeru. Opet minimiziramo kvadrat 2-norme vektora pogrešaka aproksimacije u čvorovima

$$S = S(a_0, a_1) = \sum_{k=0}^{n} (f_k - \varphi(x_k))^2 = \sum_{k=0}^{n} (f_k - a_0\varphi_0(x_k) - a_1\varphi_0(x_k))^2 \to \min.$$

Sređivanjem parcijalnih derivacija

$$0 = \frac{\partial S}{\partial a_0} = -2\sum_{k=0}^{n} (f_k - a_0\varphi_0(x_k) - a_1\varphi_1(x_k))\,\varphi_0(x_k),$$
$$0 = \frac{\partial S}{\partial a_1} = -2\sum_{k=0}^{n} (f_k - a_0\varphi_0(x_k) - a_1\varphi_1(x_k))\,\varphi_1(x_k),$$

po varijablama a_0, a_1, uz dogovor da je

$$s_0 = \sum_{k=0}^{n} \varphi_0^2(x_k), \qquad s_1 = \sum_{k=0}^{n} \varphi_0(x_k)\varphi_1(x_k), \qquad s_2 = \sum_{k=0}^{n} \varphi_1^2(x_k),$$
$$t_0 = \sum_{k=0}^{n} f_k\varphi_0(x_k), \qquad t_1 = \sum_{k=0}^{n} f_k\varphi_1(x_k),$$

dobivamo potpuno isti oblik linearnog sustava

$$s_0 a_0 + s_1 a_1 = t_0$$
$$s_1 a_0 + s_2 a_1 = t_1.$$

Ovaj sustav ima ista svojstva kao i u prethodnom primjeru. Pokažite to!

Što ako φ nelinearno ovisi o parametrima? Dobili bismo nelinearni sustav jednadžbi, koji se relativno teško rješava. Uglavnom, problem postaje ozbiljan optimizacijski problem, koji se, recimo, može rješavati metodama pretraživanja ili nekim drugim optimizacijskim metodama, posebno prilagođenim upravo za rješavanje nelinearnog problema najmanjih kvadrata (na primjer, Levenberg–Marquardt metoda).

Postoji i drugi pristup. Katkad se jednostavnim transformacijama problem može transformirati u linearni problem najmanjih kvadrata.

Nažalost, rješenja lineariziranog problema najmanjih kvadrata i rješenja originalnog nelinearnog problema, u principu, **nisu** jednaka. Problem je u različitim mjerama za udaljenost (grešku).

Ilustrirajmo, ponovno, nelinearni problem najmanjih kvadrata na jednom jednostavnom primjeru.

Primjer 10.5.3. *Zadane su točke $(x_0, f_0), \ldots, (x_n, f_n)$, koje po diskretnoj metodi najmanjih kvadrata aproksimiramo funkcijom oblika*

$$\varphi(x) = a_0 e^{a_1 x}.$$

Greška aproksimacije u čvorovima (koju minimiziramo) je

$$S = S(a_0, a_1) = \sum_{k=0}^{n} (f_k - \varphi(x_k))^2 = \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k})^2 \to \min .$$

Parcijalnim deriviranjem po varijablama a_0 i a_1 dobivamo

$$0 = \frac{\partial S}{\partial a_0} = -2 \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k})\, e^{a_1 x_k},$$
$$0 = \frac{\partial S}{\partial a_1} = -2 \sum_{k=0}^{n} (f_k - a_0 e^{a_1 x_k}) a_0\, x_k e^{a_1 x_k},$$

što je nelinearan sustav jednadžbi.

S druge strane, ako logaritmiramo relaciju

$$\varphi(x) = a_0 e^{a_1 x},$$

dobivamo

$$\ln \varphi(x) = \ln(a_0) + a_1 x.$$

Moramo logaritmirati još i vrijednosti funkcije f u točkama x_k, pa uz supstitucije

$$h(x) = \ln f(x), \quad h_k = h(x_k) = \ln f_k, \quad k = 0, \ldots, n,$$

i

$$\psi(x) = \ln \varphi(x) = b_0 + b_1 x,$$

gdje je

$$b_0 = \ln a_0, \quad b_1 = a_1,$$

dobivamo linearni problem najmanjih kvadrata

$$\widetilde{S} = \widetilde{S}(b_0, b_1) = \sum_{k=0}^{n} (h_k - \psi(x_k))^2 = \sum_{k=0}^{n} (h_k - b_0 - b_1 x_k)^2 \to \min .$$

Na kraju, iz rješenja b_0 i b_1 ovog problema, lako očitamo a_0 i a_1

$$a_0 = e^{b_0}, \quad a_1 = b_1.$$

Uočite da ovako dobiveno rješenje uvijek daje pozitivan a_0, tj. linearizacijom dobivena funkcija $\varphi(x)$ će uvijek biti veća od 0. Nekako je odmah jasno da to nije "pravo" rješenje za sve početne podatke (x_k, f_k)! No, možemo li na ovako opisani način linearizirati sve početne podatke? Očito je **ne**, *jer mora biti $f_k > 0$ da bismo mogli logaritmirati.*

Ipak, i kad su neki $f_k \leq 0$, nije teško, korištenjem translacije svih podataka dobiti $f_k +$ translacija > 0, pa onda nastaviti postupak linearizacije. Pokušajte korektno formulirati linearizaciju!

Konačno, evo i popisa nekoliko funkcija koje su često u upotrebi i njihovih standardnih linearizacija u problemu najmanjih kvadrata.

(a) Funkcija

$$\varphi(x) = a_0 x^{a_1}$$

linearizira se logaritmiranjem

$$\psi(x) = \log \varphi(x) = \log(a_0) + a_1 \log x, \quad h_k = \log f_k, \quad k = 0, \ldots, n.$$

Drugim riječima, dobili smo linearni problem najmanjih kvadrata

$$\widetilde{S} = \widetilde{S}(b_0, b_1) = \sum_{k=0}^{n} (h_k - b_0 - b_1 \log(x_k))^2 \to \min,$$

gdje je

$$b_0 = \log(a_0), \quad b_1 = a_1.$$

U ovom slučaju, da bismo mogli provesti linearizaciju, moraju biti i $x_k > 0$ i $f_k > 0$.

(b) Funkcija

$$\varphi(x) = \frac{1}{a_0 + a_1 x}$$

linearizira se na sljedeći način

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 + a_1 x, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 x_k)^2 \to \min .$$

(c) Funkciju

$$\varphi(x) = \frac{x}{a_0 + a_1 x}$$

možemo linearizirati na više načina. Prvo, možemo staviti

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 \frac{1}{x} + a_1, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 \frac{1}{x_k} - a_1)^2 \to \min .$$

Može i ovako

$$\psi(x) = \frac{x}{\varphi(x)} = a_0 + a_1 x, \quad h_k = \frac{x_k}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 x_k)^2 \to \min .$$

(d) Funkcija

$$\varphi(x) = \frac{1}{a_0 + a_1 e^{-x}}$$

linearizira se stavljanjem

$$\psi(x) = \frac{1}{\varphi(x)} = a_0 + a_1 e^{-x}, \quad h_k = \frac{1}{f_k}, \quad k = 0, \ldots, n.$$

Pripadni linearni problem najmanjih kvadrata je

$$\widetilde{S} = \widetilde{S}(a_0, a_1) = \sum_{k=0}^{n} (h_k - a_0 - a_1 e^{-x_k})^2 \to \min .$$

10.5.2. Matrična formulacija linearnog problema najmanjih kvadrata

Da bismo formirali matrični zapis linearnog problema najmanjih kvadrata, moramo preimenovati nepoznanice, naprosto zato da bismo matricu, vektor desne strane i nepoznanice u linearnom sustavu pisali u uobičajenoj formi (standardno su nepoznanice $x_1, \ldots, x_m$, a ne $a_0, \ldots, a_m$).

Pretpostavimo da imamo skup mjerenih podataka (t_k, y_k), $k = 1, \ldots, n$, i da želimo taj model aproksimirati funkcijom oblika $\varphi(t)$. Ako je $\varphi(t)$ linearna, tj. ako je

$$\varphi(t) = x_1\varphi_1(t) + \cdots + x_m\varphi_m(t),$$

onda bismo željeli pronaći parametre x_j tako da mjereni podaci (t_k, y_k) zadovoljavaju

$$y_k = \sum_{j=1}^{m} x_j\varphi_j(t_k), \quad k = 1, \ldots, n.$$

Ako označimo

$$a_{kj} = \varphi_j(t_k), \quad b_k = y_k,$$

onda prethodne jednadžbe možemo u matričnom obliku pisati kao

$$Ax = b.$$

Ako je mjerenih podataka više nego parametara, tj. ako je $n > m$, onda ovaj sustav jednadžbi ima više jednadžbi nego nepoznanica, pa je preodređen.

Kao što smo već u uvodu rekli, postoji mnogo načina da se odredi "najbolje" rješenje, ali zbog statističkih razloga to je često metoda najmanjih kvadrata, tj. određujemo x tako da minimizira grešku $r = Ax - b$ (r se često zove rezidual)

$$\min_x \|r\|_2 = \min_x \|Ax - b\|_2, \quad A \in \mathbb{R}^{n\times m}, \quad b \in \mathbb{R}^n. \tag{10.5.2}$$

Ako je $\operatorname{rang}(A) < m$, onda rješenje x ovog problema očito **nije** jedinstveno, jer mu možemo dodati bilo koji vektor iz nul-potprostora od A, a da se rezidual ne promijeni. S druge strane, među svim rješenjima x problema najmanjih kvadrata uvijek postoji jedinstveno rješenje x najmanje norme, tj. koje još minimizira i $\|x\|_2$.

10.5.3. Karakterizacija rješenja

Prvo, karakterizirajmo skup svih rješenja problema najmanjih kvadrata.

Teorem 10.5.1. *Skup svih rješenja problema najmanjih kvadrata (10.5.2) označimo s*

$$\mathcal{S} = \{x \in \mathbb{R}^m \mid \|Ax - b\|_2 = \min\}.$$

Tada je $x \in \mathcal{S}$ ako i samo ako vrijedi sljedeća relacija ortogonalnosti

$$A^T(b - Ax) = 0. \tag{10.5.3}$$

Dokaz:

Pretpostavimo da $\hat{x}$ zadovoljava

$$A^T\hat{r} = 0, \quad \hat{r} = b - A\hat{x}.$$

Tada za bilo koji $x \in \mathbb{R}^m$ imamo

$$r = b - Ax = \hat{r} + A\hat{x} - Ax = \hat{r} - A(x - \hat{x}).$$

Ako označimo

$$e = x - \hat{x},$$

onda je

$$\|r\|_2^2 = r^T r = (\hat{r} - Ae)^T(\hat{r} - Ae) = \hat{r}^T\hat{r} + \|Ae\|_2^2,$$

što je minimizirano kad je $x = \hat{x}$.

S druge strane, pretpostavimo da je

$$A^T\hat{r} = z \neq 0$$

i uzmimo

$$x = \hat{x} + \varepsilon z.$$

Tada je

$$r = \hat{r} - \varepsilon Az$$

i

$$\|r\|_2^2 = r^T r = \hat{r}^T\hat{r} - 2\varepsilon z^T z + \varepsilon^2(Az)^T(Az) < \hat{r}^T\hat{r}$$

za dovoljno mali ε, pa $\hat{x}$ nije rješenje u smislu najmanjih kvadrata. ■

Relacija (10.5.3) često se zove sustav normalnih jednadžbi i uobičajeno se piše u obliku

$$A^T Ax = A^T b.$$

Matrica $A^T A$ je simetrična i pozitivno semidefinitna, a sustav normalnih jednadžbi je uvijek konzistentan, jer je

$$A^T b \in \mathcal{R}(A^T) = \mathcal{R}(A^T A).$$

Čak štoviše, vrijedi i sljedeći teorem.

Teorem 10.5.2. *Matrica $A^T A$ je pozitivno definitna ako i samo ako su stupci od A linearno nezavisni, tj. ako je* $\operatorname{rang}(A) = m$.

Dokaz:

Ako su stupci od A linearno nezavisni, tada za svaki $x \neq 0$ vrijedi $Ax \neq 0$ (definicija linearne nezavisnosti), pa je za takav x

$$x^T A^T A x = \|Ax\|_2^2 > 0,$$

tj. $A^T A$ je pozitivno definitna.

S druge strane, ako su stupci linearno zavisni, tada postoji $x_0 \neq 0$ takav da je $Ax_0 = 0$, pa je za takav x_0

$$x_0^T A^T A x_0 = 0.$$

Ako je x takav da je $Ax \neq 0$, onda je $x^T A^T A x > 0$, pa je $A^T A$ pozitivno semidefinitna. ■

Iz prethodnog teorema slijedi, da ako je $\text{rang}(A) = m$, onda postoji jedinstveno rješenje problema najmanjih kvadrata, koje je dano s

$$x = (A^T A)^{-1} A^T b, \quad r = b - A(A^T A)^{-1} A^T b.$$

Ako je $S \subset \mathbb{R}^n$ potprostor, onda je $P_S \in \mathbb{R}^{n \times n}$ **ortogonalni projektor** na S, ako je $\mathcal{R}(P_S) = S$ i

$$P_S^2 = P_S, \quad P_S^T = P_S.$$

Nadalje, vrijedi i

$$(I - P_S)^2 = I - P_S, \quad (I - P_S)P_S = 0,$$

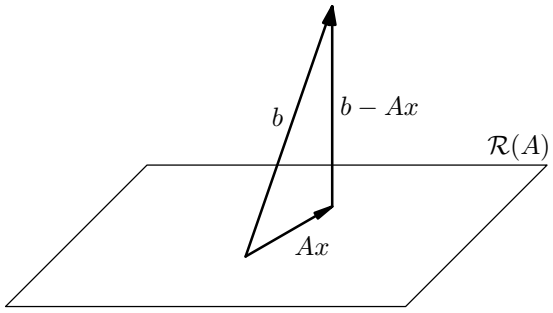
pa je $I - P_S$ projektor na ortogonalni komplement od S.

Tvrdimo da postoji jedinstveni ortogonalni projektor na S. Pretpostavimo da postoje dva ortogonalna projektora P_1 i P_2. Za sve $z \in \mathbb{R}^n$, onda vrijedi

$$\begin{aligned}\|(P_1 - P_2)z\|_2^2 &= z^T (P_1 - P_2)^T (P_1 - P_2) z = z^T (P_1^T P_1 - P_2^T P_1 - P_1^T P_2 + P_2^T P_2) z \\ &= z^T (P_1 - P_2 P_1 - P_1 P_2 + P_2) z \\ &= z^T P_1 (I - P_2) z + z^T P_2 (I - P_1) z = 0.\end{aligned}$$

Odatle odmah slijedi da je $P_1 = P_2$, tj. ortogonalni je projektor jedinstven.

Iz geometrijske interpretacije problema najmanjih kvadrata odmah vidimo da je Ax ortogonalna projekcija vektora b na $\mathcal{R}(A)$.



Također

$$r = (I - P_{\mathcal{R}(A)})b$$

i u slučaju punog ranga matrice A vrijedi

$$P_{\mathcal{R}(A)} = A(A^TA)^{-1}A^T.$$

Ako je $\text{rang}(A) < m$, onda A ima netrivijalni nul-potprostor i rješenje problema najmanjih kvadrata nije jedinstveno. Istaknimo jedno od rješenja $\hat{x}$. Skup svih rješenja $\mathcal{S}$ onda možemo opisati kao

$$\mathcal{S} = \{x = \hat{x} + z \mid z \in \mathcal{N}(A)\}.$$

Ako je $\hat{x} \perp \mathcal{N}(A)$, onda je

$$\|x\|_2^2 = \|\hat{x}\|_2^2 + \|z\|_2^2,$$

pa je $\hat{x}$ jedinstveno rješenje problema najmanjih kvadrata koje ima minimalnu 2-normu.

10.5.4. Numeričko rješavanje problema najmanjih kvadrata

Postoji nekoliko načina rješavanja problema najmanjih kvadrata u praksi. Obično se koristi jedna od sljedećih metoda:

1. sustav normalnih jednadžbi,
2. QR faktorizacija,
3. dekompozicija singularnih vrijednosti,
4. transformacija u linearni sustav.

Sustav normalnih jednadžbi

Prva od navedenih metoda je najbrža, ali je najmanje točna. Koristi se kad je A^TA pozitivno definitna i kad je njena uvjetovanost mala. Matrica A^TA rastavi se faktorizacijom Choleskog, a zatim se riješi linearni sustav

$$A^TAx = A^Tb.$$

Ukupan broj aritmetičkih operacija za računanje A^TA, A^Tb, te zatim faktorizaciju Choleskog je $nm^2 + \frac{1}{3}m^3 + O(m^2)$. Budući da je $n \geq m$, onda je prvi član dominantan u ovom izrazu, a potječe od formiranja A^TA.

Korištenje QR faktorizacije u problemu najmanjih kvadrata

Ponovno, pretpostavimo da je A^TA pozitivno definitna. Polazimo od rješenja problema najmanjih kvadrata dobivenog iz sustava normalnih jednadžbi

$$x = (A^TA)^{-1}A^Tb.$$

Zatim napišemo QR faktorizaciju matrice A

$$A = QR = Q_0R_0,$$

gdje je Q_0 ortogonalna matrica tipa (n, m), a R_0 trokutasta tipa (m, m) i uvrstimo u rješenje. Dobivamo

$$\begin{aligned} x &= (A^TA)^{-1}A^Tb = (R_0^TQ_0^TQ_0R_0)^{-1}R_0^TQ_0^Tb \\ &= (R_0^TR_0)^{-1}R_0^TQ_0^Tb = R_0^{-1}R_0^{-T}R_0^TQ_0^Tb = R_0^{-1}Q_0^Tb, \end{aligned}$$

tj. x se dobiva primjenom "invertirane" skraćene QR faktorizacije od A na b (po analogiji s rješavanjem linearnih sustava, samo što A ne mora imati inverz).

Preciznije, da bismo našli x, rješavamo trokutasti linearni sustav

$$R_0x = Q_0^Tb.$$

Na ovakav se način najčešće rješavaju problemi najmanjih kvadrata. Nije teško pokazati da je cijena računanja $2nm^2 - \frac{2}{3}m^3$, što je dvostruko više nego za sustav normalnih jednadžbi kad je $n \gg m$, a približno jednako za $m = n$.

QR faktorizacija može se koristiti i za problem najmanjih kvadrata kad matrica A nema puni stupčani rang, ali tada se koristi QR faktorizacija sa stupčanim pivotiranjem (na prvo mjesto dovodi se stupac čiji "radni dio" ima najveću normu). Zašto baš tako? Ako matrica A ima rang $r < m$, onda njena QR faktorizacija ima oblik

$$A = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

gdje je R_{11} nesingularna reda r, a R_{12} neka $r \times (m - r)$ matrica. Zbog grešaka zaokruživanja, umjesto pravog R, izračunamo

$$R' = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{bmatrix}.$$

Naravno, željeli bismo da je $\|R_{22}\|_2$ vrlo mala, reda veličine $\varepsilon\|A\|_2$, pa da je možemo "zaboraviti", tj. staviti $R_{22} = 0$ i tako odrediti rang od A. Nažalost, to nije uvijek

tako. Na primjer, bidijagonalna matrica

$$A = \begin{bmatrix} \frac{1}{2} & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \frac{1}{2} \end{bmatrix}$$

je skoro singularna ($\det(A) = 2^{-n}$), njena QR faktorizacija je $Q = I$, $R = A$, i nema niti jednog R_{22} koji bi bio po normi malen.

Zbog toga koristimo pivotiranje, koje R_{11} pokušava držati što bolje uvjetovanim, a R_{22} po normi što manjim.

Dekompozicija singularnih vrijednosti i problem najmanjih kvadrata

Vjerojatno jedna od najkorisnijih dekompozicija i s teoretske strane (za dokazivanje činjenica) i s praktične strane je dekompozicija singularnih vrijednosti (engl. "singular value decomposition") ili, skraćeno, SVD.

Teorem 10.5.3. *Neka je A proizvoljna matrica tipa $n \times m$, $n \geq m$. Tada se A može dekomponirati kao*

$$A = \widehat{U} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^* = U \Sigma V^*,$$

gdje je

$$\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_m), \quad \sigma_1 \geq \cdots \geq \sigma_m \geq 0,$$

$\widehat{U} = [U, U_0]$ je $n \times n$, a V je $m \times m$ unitarna matrica. Stupce matrice U (u oznaci u_i) zovemo lijevi singularni vektori, stupce matrice V (u oznaci v_i) desni singularni vektori, a dijagonalne elemente matrice Σ singularne vrijednosti. Ako je $n < m$, dekompozicija singularnih vrijednosti definira se za A^*. Ako je A realna, U i V su, također, realne.

Prije samog formalnog dokaza, objasnimo značenje dekompozicije. Ako o matrici A razmišljamo kao zapisu operatora koji preslikava vektor $x \in \mathbb{R}^m$ u vektor $y = Ax \in \mathbb{R}^n$, onda možemo izabrati ortogonalni koordinatni sustav u $\mathbb{R}^m$ (osi su mu jedinični vektori stupci u V) i drugi ortogonalni koordinatni sustav u $\mathbb{R}^n$ (osi su mu jedinični vektori stupci u U), takve da je zapis tog operatora u tom paru baza dijagonalna matrica.

Drugim riječima, A preslikava vektor

$$x = \sum_{i=1}^{m} \beta_i v_i$$

u

$$y = Ax = \sum_{i=1}^{m} \sigma_i \beta_i u_i,$$

tj. svaka se matrica može "dijagonalizirati" u **paru** baza, ako smo joj za domenu i sliku izabrali odgovarajuće ortogonalne koordinatne sustave (baze).

Dokaz:

Dokaz se provodi indukcijom po n i m. Pretpostavljamo da postoji dekompozicija singularnih vrijednosti od matrice dimenzije $(n-1) \times (m-1)$ i dokazujemo da tada postoji i za $n \times m$ matricu. Pretpostavljamo da je $A \neq 0$, jer u protivnom je $\Sigma = 0$, a U i V su proizvoljne unitarne matrice.

Baza indukcije je za $m = 1$, jer je $n \geq m$. Napišimo tu jednostupčanu matricu u obliku

$$A = U \Sigma V^*,$$

gdje je

$$U = \frac{A}{\|A\|_2}, \quad \Sigma = \|A\|_2, \quad V = 1.$$

Za korak indukcije, izaberemo vektor v, takav da je $\|v\|_2 = 1$ i na njemu se baš dostiže maksimum 2-norme za A, tj. vrijedi

$$\|A\|_2 = \max_{\|x\|_2 = 1} \|Ax\|_2 = \|Av\|_2.$$

Definiramo jedinični vektor

$$u = \frac{Av}{\|Av\|_2}.$$

Vektore u i v dopunimo matricama $\widetilde{U}$, odnosno $\widetilde{V}$, tako da

$$U_0 = [u, \widetilde{U}], \quad V_0 = [v, \widetilde{V}]$$

budu redom $n \times n$ i $m \times m$ unitarne matrice. Sad možemo pisati

$$U_0^* A V_0 = \begin{bmatrix} u^* \\ \widetilde{U}^* \end{bmatrix} A \, [v, \widetilde{V}] = \begin{bmatrix} u^*Av & u^*A\widetilde{V} \\ \widetilde{U}^*Av & \widetilde{U}^*A\widetilde{V} \end{bmatrix}. \tag{10.5.4}$$

Po definiciji vektora u i v je

$$u^*Av = \frac{v^*A^*}{\|Av\|_2} Av = \frac{\|Av\|_2^2}{\|Av\|_2} = \|Av\|_2 = \|A\|_2 := \sigma.$$

Nadalje, zbog ortogonalnosti stupaca unitarne matrice U_0, svi stupci matrice $\widetilde{U}$ su okomiti na vektor u, pa je $\widetilde{U}^*u = 0$. Onda je i

$$\widetilde{U}^*Av = \widetilde{U}^*u\|Av\|_2 = 0.$$

Tvrdimo i da je $u^*A\widetilde{V} = 0$. Označimo s

$$A_1 = U_0^* A V_0, \quad w^* = u^* A \widetilde{V}, \quad B = \widetilde{U}^* A \widetilde{V}.$$

Relacija (10.5.4) tada glasi

$$A_1 = \begin{bmatrix} \sigma & w^* \\ 0 & B \end{bmatrix}.$$

Zbog unitarne invarijantnosti 2-norme je

$$\sigma = \|A\|_2 = \|U_0^* A V_0\|_2 = \|A_1\|_2.$$

S druge strane, za proizvoljni vektor $z \neq 0$ vrijedi

$$\|A_1\|_2 = \max_{x \neq 0} \frac{\|A_1 x\|_2}{\|x\|_2} \geq \frac{\|A_1 z\|_2}{\|z\|_2},$$

odnosno

$$\|A_1\|_2 \|z\|_2 \geq \|A_1 z\|_2.$$

Izaberimo

$$z = \begin{bmatrix} \sigma \\ w \end{bmatrix}.$$

Onda je

$$\|A_1\|_2^2 \|z\|_2^2 = \|A_1\|_2^2 (\sigma^2 + \|w\|_2^2) \geq \|A_1 z\|_2^2 = \left\| A_1 \begin{bmatrix} \sigma \\ w \end{bmatrix} \right\|_2^2$$
$$= \left\| \begin{bmatrix} \sigma & w^* \\ 0 & B \end{bmatrix} \begin{bmatrix} \sigma \\ w \end{bmatrix} \right\|_2^2 = (\sigma^2 + w^* w)^2 + \|Bw\|_2^2 \geq (\sigma^2 + \|w\|_2^2)^2,$$

pa vidimo da je

$$\|A_1\|_2^2 (\sigma^2 + \|w\|_2^2) \geq (\sigma^2 + \|w\|_2^2)^2.$$

Dijeljenjem s $(\sigma^2 + \|w\|_2^2)$ dobivamo

$$\sigma^2 = \|A\|_2^2 = \|A_1\|_2^2 \geq \sigma^2 + \|w\|_2^2,$$

što je moguće samo za $w = 0$.

Drugim riječima, vrijedi

$$U_0^* A V_0 = \begin{bmatrix} \sigma & 0 \\ 0 & B \end{bmatrix}.$$

Sada možemo iskoristiti pretpostavku indukcije na matricu B,

$$B = U_1 \Sigma_1 V_1^*$$

pa dobivamo

$$U_0^* A V_0 = \begin{bmatrix} \sigma & 0 \\ 0 & U_1 \Sigma_1 V_1^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^*,$$

odakle odmah slijedi tvrdnja.

Ako želimo biti potpuno precizni, treba još silazno poredati singularne vrijednosti. To se postiže primjenom matrica permutacije P_1 reda m, i P_2 reda n, tako da matrica

$$\Sigma := P_2^* \begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix} P_1$$

ima silazno poredanu dijagonalu. Lako se vidi da je

$$P_2 = \begin{bmatrix} P_1 & 0 \\ 0 & I_{n-m} \end{bmatrix},$$

gdje je I_{n-m} jedinična matrica reda $n-m$. Na kraju, znamo da su P_1 i P_2 unitarne matrice, a produkt unitarnih matrica je opet unitarna matrica, pa su

$$U := U_0 \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} P_2, \qquad V := V_0 \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix} P_1,$$

unitarne matrice i vrijedi $A = U\Sigma V^*$. ■

Nabrojimo neka svojstva SVD-a. Budući da je riječ o mnogo tvrdnji koje su (uglavnom) neovisne, radi lakšeg praćenja, podijelit ćemo iskaz teorema u tvrdnje i svaku od njih odmah i dokazati.

Teorem 10.5.4. *Neka je $A = U\Sigma V^T$ dekompozicija singularnih vrijednosti (SVD) realne matrice A tipa $n \times m$, $n \geq m$.*

Tvrdnja 1. *Ako je A simetrična matrica reda m sa svojstvenim vrijednostima λ_i i ortonormalnim svojstvenim vektorima u_i, tj. ako je svojstvena dekompozicija za A oblika*

$$A = U\Lambda U^T, \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_m), \quad U = [u_1, \ldots, u_m], \quad UU^T = I,$$

onda je SVD matrice A

$$A = U\Sigma V^T,$$

gdje je $\sigma_i = |\lambda_i|$ i $v_i = \operatorname{sign}(\lambda_i) u_i$, uz dogovor da je $\operatorname{sign}(0) = 1$.

Dokaz:

Očit iz definicije SVD-a. ■

Tvrdnja 2. *Svojstvene vrijednosti simetrične matrice $A^T A$ su σ_i^2. Desni singularni vektori v_i su pripadni svojstveni vektori.*

Dokaz:

Vrijedi

$$A^TA = V\Sigma^TU^TU\Sigma V^T = V\Sigma^2V^T$$

i to je svojstvena dekompozicija od A^TA. ■

Tvrdnja 3. *Svojstvene vrijednosti matrice AA^T su σ_i^2 i još $n-m$ njih koje su jednake nula. Lijevi singularni vektori u_i su pripadni svojstveni vektori za svojstvene vrijednosti σ_i^2. Za preostale nula svojstvene vrijednosti, možemo kao svojstvene vektore uzeti bilo kojih $n-m$ vektora koji s prethodnima čine ortogonalnu matricu (dopuna do ortonormirane baze).*

Dokaz:

Uzmemo puni SVD, s kvadratnom matricom $\widehat{U}$. Onda je

$$AA^T = \widehat{U}\begin{bmatrix}\Sigma\\0\end{bmatrix}V^TV\,[\Sigma^T,0]\,\widehat{U}^T = \widehat{U}\begin{bmatrix}\Sigma^2&0\\0&0\end{bmatrix}\widehat{U}^T = [U,U_0]\begin{bmatrix}\Sigma^2&0\\0&0\end{bmatrix}[U,U_0]^T,$$

što je svojstvena dekompozicija od AA^T. ■

Tvrdnja 4. *Neka je A kvadratna matrica reda m, i neka je $A = U\Sigma V^T$ SVD od A, uz*

$$\Sigma = \operatorname{diag}(\sigma_1,\ldots,\sigma_m),\quad U = [u_1,\ldots,u_m],\quad V = [v_1,\ldots,v_m].$$

Neka je

$$H = \begin{bmatrix}0&A^T\\A&0\end{bmatrix}.$$

Tada je $2m$ svojstvenih vrijednosti od H jednako $\pm\sigma_i$, a pripadni svojstveni vektori su

$$\frac{1}{\sqrt{2}}\begin{bmatrix}v_i\\\pm u_i\end{bmatrix}.$$

Dokaz:

Uvrstimo SVD od A u formulu za H. Onda je

$$H = \begin{bmatrix}0&A^T\\A&0\end{bmatrix} = \begin{bmatrix}0&V\Sigma U^T\\U\Sigma V^T&0\end{bmatrix} = \begin{bmatrix}0&V\\U&0\end{bmatrix}\begin{bmatrix}0&\Sigma\\\Sigma&0\end{bmatrix}\begin{bmatrix}0&U^T\\V^T&0\end{bmatrix}.$$

Lako se provjerava da je matrica

$$Q = \frac{1}{\sqrt{2}}\begin{bmatrix}I&I\\-I&I\end{bmatrix}$$

ortogonalna i da vrijedi

$$Q\begin{bmatrix}0&\Sigma\\\Sigma&0\end{bmatrix}Q^T = \begin{bmatrix}\Sigma&0\\0&-\Sigma\end{bmatrix}.$$

Konačno, zaključujemo da je

$$\begin{aligned} H &= \begin{bmatrix} 0 & V \\ U & 0 \end{bmatrix} Q^T Q \begin{bmatrix} 0 & \Sigma \\ \Sigma & 0 \end{bmatrix} Q^T Q \begin{bmatrix} 0 & U^T \\ V^T & 0 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{bmatrix} \left(\frac{1}{\sqrt{2}} \begin{bmatrix} V & V \\ U & -U \end{bmatrix} \right)^T, \end{aligned}$$

što je svojstvena dekompozicija od H. ■

Tvrdnja 5. *Ako A ima puni rang, onda je rješenje problema najmanjih kvadrata*

$$\min_x \|Ax - b\|_2$$

jednako

$$x = V\Sigma^{-1}U^T b,$$

tj. dobiva se primjenom "invertiranog" skraćenog SVD-a od A na b.

Dokaz:

Vrijedi

$$\|Ax - b\|_2^2 = \|U\Sigma V^T x - b\|_2^2.$$

Budući da je A punog ranga, to je i Σ. Zbog unitarne ekvivalencije 2-norme, vrijedi

$$\begin{aligned} \|U\Sigma V^T x - b\|_2^2 = \|\widehat{U}^T(U\Sigma V^T x - b)\|_2^2 &= \left\| \begin{bmatrix} U^T \\ U_0^T \end{bmatrix} (U\Sigma V^T x - b) \right\|_2^2 \\ = \left\| \begin{bmatrix} \Sigma V^T x - U^T b \\ -U_0^T b \end{bmatrix} \right\|_2^2 &= \|\Sigma V^T x - U^T b\|_2^2 + \|U_0^T b\|_2^2. \end{aligned}$$

Prethodni izraz se minimizira ako je prvi član jednak 0, tj. ako je

$$x = V\Sigma^{-1}U^T b.$$

Usput dobivamo i vrijednost minimuma $\min_x \|Ax - b\|_2 = \|U_0^T b\|_2$. ■

Tvrdnja 6. *Neka je A kvadratna matrica reda m i pretpostavimo da je A nesingularna. Ako je σ_1 najveća, a σ_m najmanja singularna vrijednost od A, onda je*

$$\|A\|_2 = \sigma_1, \quad \|A^{-1}\|_2^{-1} = \sigma_m, \quad \kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_m}.$$

Dokaz:

Zbog unitarne ekvivalencije 2-norme, vrijedi

$$\begin{aligned} \|A\|_2 = \|U^T A V\|_2 = \|\Sigma\|_2 = \sigma_1 \\ \|A^{-1}\|_2 = \|U^T A^{-1} V\|_2 = \|\Sigma^{-1}\|_2 = \sigma_m^{-1}. \end{aligned}$$

■

Tvrdnja 7. *Pretpostavimo da je* $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_m = 0$. *Tada je* $\text{rang}(A) = r$. *Nul-potprostor od A, tj. potprostor svih vektora v za koje je* $Av = 0$, *je potprostor razapet sa stupcima* $v_{r+1}, \ldots, v_m$ *od V. Slika operatora A, tj. potprostor svih vektora oblika Aw za sve w, razapet je stupcima* $u_1, \ldots, u_r$ *od U.*

Dokaz:

Ponovno, napišimo SVD korištenjem kvadratnih matrica $\widehat{U}$ i V. Budući da su obje ortogonalne, one su nesingularne, pa je $\text{rang}(A) = \text{rang}(\Sigma) = r$, po pretpostavci. Također, v je u nul-potprostoru od A ako i samo ako je $V^T v$ u nul-potprostoru od

$$\widehat{U}^T AV = \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} := \widetilde{\Sigma},$$

jer je $Av = 0$ ako i samo ako je $\widehat{U}^T AV(V^T v) = 0$. Ali, nul-potprostor matrice $\widetilde{\Sigma}$ je, očito razapet stupcima $(r+1)$ do m jedinične matrice I_m, pa je nul-potprostor od A razapet sa V puta ti stupci, tj. sa stupcima $(r+1)$ do m matrice V. Sličan argument vrijedi i za ostatak dokaza. ■

Tvrdnja 8. *Neka je* S^{m-1} *jedinična sfera u* $\mathbb{R}^m$,

$$S^{m-1} = \{x \in \mathbb{R}^m \mid \|x\|_2 = 1\}.$$

Neka je $A \cdot S^{m-1}$ *slika od* S^{m-1} *kad se jedinična sfera preslika operatorom A,*

$$A \cdot S^{m-1} = \{Ax \mid x \in \mathbb{R}^m,\ \|x\|_2 = 1\}.$$

Ta slika $A \cdot S^{m-1}$ *je elipsoid sa središtem u ishodištu od* $\mathbb{R}^m$ *i glavnim osima* $\sigma_i u_i$, *za* $i = 1, \ldots, m$.

Dokaz:

Radi jednostavnosti, pretpostavimo da je A kvadratna i nesingularna. Matrica V je ortogonalna, pa ona preslikava jedinične vektore u neke druge jedinične vektore, tj. vrijedi $V^T S^{m-1} = S^{m-1}$. Budući da je $v \in S^{m-1}$ ako i samo ako je $\|v\|_2 = 1$, onda je $w \in \Sigma S^{m-1}$ ako i samo ako je $\|\Sigma^{-1} w\|_2 = 1$ ili

$$\sum_{i=1}^{m} \left(\frac{w_i}{\sigma_i}\right)^2 = 1.$$

Time je definiran elipsoid s glavnim osima $\sigma_i e_i$. Konačno, množenjem svakog $w = \Sigma v$ matricom U rotira taj elipsoid (oko ishodišta), tako da se svaki vektor e_i preslika u u_i. Dobiveni elipsoid ima glavne osi $\sigma_i u_i$.

Isti argument vrijedi i u općem slučaju $n \geq m$ i kad je $\text{rang}(A) = r \leq m$. Matrica V radi isto, a ΣS^{m-1} je elipsoid s glavnim osima $\sigma_i e_i$, za $i = 1, \ldots, r$ (za $i > r$, ostale osi $\sigma_i e_i$ su nula). Na kraju, U rotira taj elipsoid po istom pravilu. ■

Tvrdnja 9. *Zapišimo matrice U i V iz SVD-a od A u stupčanom obliku*

$$U = [u_1, \ldots, u_m], \quad V = [v_1, \ldots, v_m].$$

Onda matricu A (preko SVD-a) možemo pisati kao zbroj matrica ranga 1

$$A = U\Sigma V^T = \sum_{i=1}^{m} \sigma_i u_i v_i^T.$$

Matricu A_k, istog tipa kao i A, ranga $\operatorname{rang}(A_k) \leq k < m$, *koja je po* 2*-normi najbliža matrici A, možemo zapisati kao*

$$A_k = U\Sigma_k V^T = \sum_{i=1}^{k} \sigma_i u_i v_i^T,$$

pri čemu je

$$\Sigma_k = \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0).$$

Pritom je

$$\|A - A_k\|_2 = \sigma_{k+1}$$

najmanja udaljenost između A i svih matrica ranga najviše k.

Dokaz:

Ovo je, zapravo, tvrdnja o najboljoj aproksimaciji matrice A matricom nižeg ranga i njihovoj udaljenjosti u 2-normi. Prema konstrukciji, A_k ima rang najviše k i vrijedi

$$\|A - A_k\|_2 = \left\| \sum_{i=k+1}^{m} \sigma_i u_i v_i^T \right\|_2 = \|U \operatorname{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_m) V^T\|_2 = \sigma_{k+1}.$$

Ostaje pokazati da je to i najbliža matrica ranga najviše k matrici A. Neka je B bilo koja matrica istog tipa za koju vrijedi $\operatorname{rang}(B) \leq k$. Onda njezin nul-potprostor ima dimenziju barem $m - k$. S druge strane, potprostor razapet vektorima $v_1, \ldots, v_{k+1}$ ima dimenziju $k+1$, pa sigurno postoji netrivijalni vektor koji se nalazi u njegovom presjeku s nul-potprostorom od B. Neka je h pripadni jedinični vektor koji se nalazi u presjeku ta dva potprostora. Onda je $Bh = 0$ i

$$\begin{aligned} \|A - B\|_2 &\geq \|(A - B)h\|_2 = \|Ah\|_2 = \|U\Sigma V^T h\|_2 \\ &= \|\Sigma V^T h\|_2 \geq \sigma_{k+1} \|V^T h\|_2 = \sigma_{k+1}. \end{aligned}$$

Zadnja nejednakost je posljedica pretpostavke da je h linearna kombinacija vektora $v_1, \ldots, v_{k+1}$. ■

Ovom tvrdnjom smo završili teorem 10.5.4. o svojstvima dekompozicije singularnih vrijednosti.

Uočite da u prethodnom teoremu, u tvrdnji 5, piše rješenje problema najmanjih kvadrata kad je matrica A punog ranga. Uobičajeno se SVD primjenjuje u metodi najmanjih kvadrata i kad matrica A nema puni stupčani rang. Rješenja su istog oblika (sjetite se, više ih je), samo što moramo znati izračunati "inverz" matrice Σ kad ona nije regularna, tj. kad ima neke nule na dijagonali. Takav inverz zove se generalizirani inverz i označava sa Σ^+ ili $\Sigma^\dagger$. U slučaju da je

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix},$$

pri čemu je Σ_1 regularna, onda je

$$\Sigma^+ = \begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

Još preciznije, za problem najmanjih kvadrata tada vrijedi sljedeća propozicija.

Propozicija 10.5.1. *Neka matrica A ima rang $r < m$. Rješenje x koje minimizira $\|Ax - b\|_2$ može se karakterizirati na sljedeći način. Neka je $A = U\Sigma V^T$ SVD od A i neka je*

$$A = U\Sigma V^T = [U_1, U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1, V_2]^T = U_1 \Sigma_1 V_1^T,$$

gdje je Σ_1 nesingularna, reda r, a matrice U_1 i V_1 imaju r stupaca. Neka je

$$\sigma := \sigma_{\min}(\Sigma_1),$$

najmanja ne-nula singularna vrijednost od A. Tada se sva rješenja problema najmanjih kvadrata mogu napisati u formi

$$x = V_1 \Sigma_1^{-1} U_1^T b + V_2 z,$$

gdje je z proizvoljni vektor. Rješenje x koje ima minimalnu 2-normu je ono za koje je $z = 0$, tj.

$$x = V_1 \Sigma_1^{-1} U_1^T b,$$

i vrijedi ocjena

$$\|x\|_2 \leq \frac{\|b\|_2}{\sigma}.$$

Dokaz:

Nadopunimo matricu $[U_1, U_2]$ stupcima matrice U_3 do ortogonalne matrice reda n, i označimo je s $\widehat{U}$. Korištenjem unitarne invarijantnosti 2-norme, dobivamo

$$\|Ax - b\|_2^2 = \|\widehat{U}^T(Ax - b)\|_2^2 = \left\| \begin{bmatrix} U_1^T \\ U_2^T \\ U_3^T \end{bmatrix} (U_1 \Sigma_1 V_1^T x - b) \right\|_2^2$$

$$= \left\| \begin{bmatrix} \Sigma_1 V_1^T x - U_1^T b \\ -U_2^T b \\ -U_3^T b \end{bmatrix} \right\|_2^2 = \|\Sigma_1 V_1^T x - U_1^T b\|_2^2 + \|U_2^T b\|_2^2 + \|U_3^T b\|_2^2.$$

Očito, izraz je minimiziran kad je prva od tri norme u posljednjem redu jednaka 0, tj. ako je

$$\Sigma_1 V_1^T x = U_1^T b,$$

ili

$$x = V_1 \Sigma_1^{-1} U_1^T b.$$

Stupci matrica V_1 i V_2 su međusobno ortogonalni, pa je $V_1^T V_2 z = 0$ za sve vektore z. Odavde vidimo da x ostaje rješenje problema najmanjih kvadrata i kad mu dodamo $V_2 z$, za bilo koji z, tj. ako je

$$x = V_1 \Sigma_1^{-1} U_1^T b + V_2 z.$$

To su ujedno i sva rješenja, jer stupci matrice V_2 razapinju nul-potprostor $\mathcal{N}(A)$ (tvrdnja 7. teorema 10.5.4.). Osim toga, zbog spomenute ortogonalnosti vrijedi i

$$\|x\|_2^2 = \|V_1 \Sigma_1^{-1} U_1^T b\|_2^2 + \|V_2 z\|_2^2,$$

a to je minimalno za $z = 0$. Na kraju, za to minimalno rješenje vrijedi ocjena

$$\|x\|_2 = \|V_1 \Sigma_1^{-1} U_1^T b\|_2 = \|\Sigma_1^{-1} U_1^T b\|_2 \leq \frac{\|U_1^T b\|_2}{\sigma} = \frac{\|b\|_2}{\sigma}.$$

Primjerom se lako pokazuje da je ova ocjena dostižna. ■

Rješenje problema najmanjih kvadrata korištenjem SVD-a je najstabilnije, a može se pokazati da je, za $n \gg m$, njegovo trajanje približno jednako kao i trajanje rješenja korištenjem QR-a. Za manje n, trajanje je približno $4nm^2 - \frac{4}{3}m^3 + O(m^2)$.

Transformiranje problema najmanjih kvadrata na linearni sustav

Ako matrica A ima puni rang po stupcima, onda problem najmanjih kvadrata možemo transformirati i na linearni sustav različit od sustava normalnih jednadžbi. Simetrični linearni sustav

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

ekvivalentan je sustavu normalnih jednadžbi. Ako napišemo prvu i drugu blok-komponentu

$$r + Ax = b, \quad A^T r = 0,$$

onda uvrštavanjem r-a iz prve blok-jednadžbe u drugu dobivamo sustav

$$A^T(b - Ax) = 0.$$

Prvi sustav ima bitno manji raspon elemenata od sustava normalnih jednadžbi. Osim toga, ako je matrica A loše uvjetovana, kod tog sustava možemo lakše koristiti iterativno profinjavanje rješenja.

10.6. Opći oblik metode najmanjih kvadrata

Nakon što smo napravili osnovni oblik diskretne metode najmanjih kvadrata, na sličan način možemo riješiti i opći problem aproksimacije po metodi najmanjih kvadrata, tj. u 2-normi. Dovoljno je uočiti da je diskretna 2-norma generirana običnim euklidskim skalarnim produktom na konačno dimenzionalnim prostorima. Po istom principu, u općem slučaju, radimo na nekom unitarnom prostoru s nekim skalarnim produktom, a pripadna norma je generirana tim skalarnim produktom.

Na početku zgodno je uvesti oznake koje nam omogućavaju da diskretni i neprekidni slučaj analiziramo odjednom, u istom općem okruženju unitarnih prostora.

10.6.1. Težinski skalarni produkti

Unitarni prostor $\mathcal{U}$ je vektorski prostor na kojem je definiran skalarni produkt.

10.7. Familije ortogonalnih funkcija

Za dvije funkcije reći ćemo da su ortogonalne, ako je njihov skalarni produkt jednak 0. Na primjer, za neprekidnu ili diskretnu mjeru $d\lambda$, te funkcije u i v koje imaju konačnu normu možemo definirati skalarni produkt kao

$$\int_{\mathbb{R}} u(x)\, v(x)\, d\lambda.$$

Postoji mnogo familija ortogonalnih funkcija. Evo nekoliko primjera takvih familija (sistema).

- Ortogonalni polinomi;
- Trigonometrijski polinomi.

10.8. Neka svojstva ortogonalnih polinoma

Ortogonalni polinomi imaju još i niz dodatnih "dobrih" svojstava, zbog kojih se mogu konstruktivno primijeniti u raznim granama numeričke matematike. Sljedeći niz teorema sadrži samo neka osnovna svojstva koja ćemo kasnije iskoristiti za konstrukciju algoritama. Sva ta svojstva su direktna posljedica ortogonalnosti polinoma i ne ovise bitno o tome da li je skalarni produkt diskretan ili kontinuiran.

Međutim, na ovom mjestu je zgodno napraviti razliku između diskretnih i kontinuiranih skalarnih produkata, prvenstveno radi jednostavnosti iskaza, dokaza i kasnijeg pozivanja na ove teoreme. Pažljivije čitanje će samo potvrditi da bitne razlike nema.

Standardno ćemo promatrati neprekidni skalarni produkt

$$\langle u, v\rangle = \int_a^b w(x)u(x)v(x)\,dx$$

generiran težinskom funkcijom $w \geq 0$ na $[a, b]$. Ako svi polinomi pripadaju odgovarajućem prostoru kvadratno integrabilnih funkcija, onda postoji pripadna familija ortogonalnih polinoma koju označavamo s $\{p_n(x) \mid n \geq 0\}$. Dogovorno smatramo da je stupanj polinoma p_n baš jednak n, za svaki $n \geq 0$.

Paralelno ćemo promatrati i diskretni skalarni produkt

$$\langle u, v\rangle = \sum_{i=0}^{n} w_i u(x_i)v(x_i)$$

generiran međusobno različitim čvorovima $x_0, \ldots, x_n$ i pripadnim pozitivnim težinama $w_0, \ldots, w_n$. Pripadni unitarni prostor "funkcija" na zadanoj mreži čvorova (izomorfno) sadrži sve polinome stupnja manjeg ili jednakog n, pa sigurno postoji pripadna baza ortogonalnih polinoma koju označavamo s $\{p_k(x) \mid 0 \leq k \leq n\}$. Opet uzimamo je stupanj polinoma p_k baš jednak k, za svaki $k \in \{0, \ldots, n\}$.

Teorem 10.8.1. *Neka je $\{p_n(x) \mid n \geq 0\}$ familija ortogonalnih polinoma na intervalu $[a, b]$ s težinskom funkcijom $w(x) \geq 0$. Ako je f polinom stupnja m, tada vrijedi*

$$f = \sum_{n=0}^{m} \frac{\langle f, p_n\rangle}{\langle p_n, p_n\rangle}\, p_n.$$

Dokaz:

Prvo, pokažimo da se svaki polinom može napisati kao kombinacija ortogonalnih polinoma stupnja manjeg ili jednakog njegovom.

Dokaz ide korištenjem Gram–Schmidtove ortogonalizacije. Pokažimo, redom, da se monomi $\{1, x, x^2, \ldots\}$ mogu prikazati pomoću ortogonalnih polinoma.

Ako je stupanj ortogonalnog polinoma 0, on je nužno konstanta različita od nule, tj. vrijedi

$$p_0(x) = c_{0,0}, \quad c_{0,0} \neq 0,$$

pa se prvi monom 1 može napisati kao

$$1 = \frac{1}{c_{0,0}}\, p_0(x).$$

Za polinome stupnja jedan, konstrukcija slijedi iz Gram–Schmidtovog procesa ortogonalizacije sustava funkcija $\{1, x\}$

$$p_1(x) = c_{1,1}x + c_{1,0}p_0(x), \quad c_{1,1} \neq 0,$$

tj. vrijedi

$$x = \frac{1}{c_{1,1}}\left[p_1(x) - c_{1,0}p_0(x)\right].$$

Korištenjem indukcije u Gram–Schmidtovom procesu na $\{1, x, x^2, \ldots, x^n\}$, dobivamo

$$p_n(x) = c_{n,n}x^n + c_{n,n-1}p_{n-1}(x) + \cdots + c_{n,0}p_0(x), \quad c_{n,n} \neq 0,$$

gdje su $p_0, p_1, \ldots, p_{n-1}$ dobiveni ortogonalizacijom iz $\{1, x, \ldots, x^{n-1}\}$, pa je

$$x^n = \frac{1}{c_{n,n}}[p_n(x) - c_{n,n-1}p_{n-1}(x) - \cdots - c_{n,0}p_0(x)].$$

Neka je f bilo koji polinom stupnja (manjeg ili jednakog) m, za neki $m \in \mathbb{N}_0$. Tada se f može napisati kao linearna kombinacija monoma $\{1, x, \ldots, x^m\}$, prikazom u standardnoj bazi. Budući da se svaki monom može napisati kao linearna kombinacija ortogonalnih polinoma stupnja manjeg ili jednakog od stupnja tog monoma, odmah slijedi da se i f može napisati kao neka linearna kombinacija ortogonalnih polinoma stupnjeva manjih ili jednakih m, tj. da vrijedi

$$f = \sum_{j=0}^{m} b_j p_j.$$

Ostaje samo odrediti koeficijente b_j. Množenjem prethodne relacije težinskom funkcijom w, pa polinomom p_n, a zatim integriranjem na $[a, b]$, tj. skalarnim množenjem s p_n, dobivamo

$$\langle f, p_n\rangle = \sum_{j=0}^{m} b_j\langle p_j, p_n\rangle = b_n\langle p_n, p_n\rangle,$$

koristeći ortogonalnost p_j i p_n, za $j \neq n$. Odatle odmah slijedi da je

$$b_n = \frac{\langle f, p_n\rangle}{\langle p_n, p_n\rangle},$$

jer je $\|p_n\|^2 = \langle p_n, p_n\rangle > 0$. ■

Razvoj polinoma f stupnja m iz prethodnog teorema možemo napisati i tako da suma ide do ∞, a ne do m, samo su svi dodatni koeficijenti $b_n = 0$, za $n > m$. To je posljedica sljedeće tvrdnje.

Korolar 10.8.1. *Ako je f polinom stupnja manjeg ili jednakog $m-1$, onda je*

$$\langle f, p_m \rangle = 0,$$

tj. p_m je okomit na f. Dakle, p_m je okomit na sve polinome stupnja strogo manjeg od m.

Dokaz:

Po prethodnom teoremu, f se može razviti po ortogonalnim polinomima stupnja manjeg ili jednakog $m-1$

$$f(x) = \sum_{n=0}^{m-1} b_n p_n(x).$$

Množenjem s $w(x)p_m(x)$, te integriranjem, dobivamo da je

$$\langle f, p_m \rangle = \sum_{n=0}^{m-1} b_n \langle p_n, p_m \rangle = 0,$$

zbog svojstva ortogonalnosti ortogonalnih polinoma $\langle p_n, p_m \rangle = 0$, za $n \neq m$. ■

Teorem 10.8.2. *Neka je $\{p_n(x) \mid n \geq 0\}$ familija ortogonalnih polinoma na intervalu $[a,b]$ s težinskom funkcijom $w(x) \geq 0$. Tada svaki polinom p_n ima točno n različitih (jednostrukih) realnih nultočaka na otvorenom intervalu (a,b).*

Dokaz:

Neka su $x_1, x_2, \ldots, x_m$ sve nultočke polinoma p_n za koje vrijedi:

- $a < x_i < b$,
- $p_n(x)$ mijenja predznak u x_i.

Budući da je p_n stupnja n, po osnovnom teoremu algebre, polinom p_n ima ukupno n nultočaka, pa onih koje zadovoljavaju prethodna dva svojstva ima manje ili jednako n. Pretpostavimo da je nultočaka koje zadovoljavaju tražena dva svojstva striktno manje od n, tj. $m < n$. Pokažimo da je to nemoguće.

Definiramo polinom

$$B(x) = (x - x_1) \cdots (x - x_m).$$

Po definiciji točaka $x_1, \ldots, x_m$, polinom

$$p_n(x)B(x) = (x - x_1) \cdots (x - x_m)p_n(x)$$

ne mijenja znak prolaskom kroz točke $x_1, \ldots, x_m$, tj. čitav polinom ne mijenja znak na (a,b). Preciznije, to implicira oblik funkcije p_n

$$p_n(x) = h(x)(x - x_1)^{r_1} \cdots (x - x_m)^{r_m},$$

pri čemu moraju biti svi r_i neparni, a $h(x)$ ne smije promijeniti predznak na (a, b). Množenjem s $B(x)$, dobivamo

$$p_n(x)B(x) = h(x)(x - x_1)^{r_1+1} \cdots (x - x_m)^{r_m+1}.$$

Nadalje, vrijedi

$$\int_a^b w(x)B(x)p_n(x)\,dx \neq 0,$$

budući da je to integral nenegativne funkcije. S druge je strane taj integral skalarni produkt od B (polinom stupnja $m < n$) i sa p_n (polinom stupnja n), pa je po prethodnom korolaru

$$\int_a^b w(x)B(x)p_n(x)\,dx = \langle B, p_n\rangle = 0.$$

To je, očito kontradikcija, pa je pretpostavka o stupnju polinoma B bila pogrešna, tj. mora biti $m = n$. Budući da p_n ima točno n nultočaka $x_1, \ldots, x_n$ u kojima mijenja predznak, one moraju biti jednostruke, jer je $p_n'(x_i) \neq 0$. ■

Neka je ponovno zadana familija ortogonalnih polinoma na intervalu $[a, b]$ i neka su prva dva koeficijenta funkcije p_n jednaki

$$p_n(x) = A_n x^n + B_n x^{n-1} + \cdots.$$

Također, tada p_n možmo napisati i kao

$$p_n(x) = A_n(x - x_{n,1})(x - x_{n,2}) \cdots (x - x_{n,n}).$$

Definiramo također i

$$a_n = \frac{A_{n+1}}{A_n}, \quad \gamma_n = \langle p_n, p_n\rangle > 0.$$

Teorem 10.8.3. (tročlana rekurzija) *Neka je* $\{p_n(x) \mid n \geq 0\}$ *familija ortogonalnih polinoma na intervalu* $[a, b]$ *s težinskom funkcijom* $w(x) \geq 0$. *Tada za* $n \geq 1$ *vrijedi rekurzija*

$$p_{n+1}(x) = (a_n x + b_n)p_n(x) - c_n p_{n-1}(x),$$

pri čemu su

$$b_n = a_n\left(\frac{B_{n+1}}{A_{n+1}} - \frac{B_n}{A_n}\right), \quad c_n = \frac{A_{n+1}A_{n-1}}{A_n^2} \cdot \frac{\gamma_n}{\gamma_{n-1}}.$$

Dokaz:

Promatrajmo polinom

$$\begin{aligned} G(x) &= p_{n+1}(x) - a_n x p_n(x) \\ &= (A_{n+1}x^{n+1} + B_{n+1}x^n + \cdots) - \frac{A_{n+1}}{A_n}x(A_n x^n + B_n x^{n-1} + \cdots) \\ &= \left(B_{n+1} - \frac{A_{n+1}B_n}{A_n}\right)x^n + \cdots. \end{aligned}$$

Očito, polinom G je stupnja manjeg ili jednakog n, pa ga možemo napisati kao linearnu kombinaciju ortogonalnih polinoma stupnja manjeg ili jednakog n, tj.

$$G(x) = d_n p_n(x) + \cdots + d_0 p_0(x)$$

za neki skup konstanti d_i. Računanjem d_i izlazi

$$d_i = \frac{\langle G, p_i\rangle}{\langle p_i, p_i\rangle} = \frac{1}{\gamma_i}(\langle p_{n+1}, p_i\rangle - a_n\langle xp_n, p_i\rangle).$$

Budući da je $\langle p_{n+1}, p_i\rangle = 0$ za $i \leq n$ i da za $i \leq n-2$ vrijedi

$$\langle xp_n, p_i\rangle = \int_a^b w(x)p_n(x)xp_i(x)\,dx = 0,$$

zaključujemo da je stupanj polinoma $xp_i(x)$ manji ili jednak $n-1$. Kombiniranjem ta dva rezultata, dobivamo

$$d_i = 0 \quad \text{za } 0 \leq i \leq n-2,$$

pa je zbog toga

$$G(x) = d_n p_n(x) + d_{n-1}p_{n-1}(x)$$
$$p_{n+1}(x) = (a_n x + d_n)p_n(x) + d_{n-1}p_{n-1}(x).$$

Ostaje još samo pokazati koliki su koeficijenti d_{n-1} i d_n. Iz prve od dvije prethodne relacija, uspoređivanjem vodećih koeficijenata funkcije G i vodećih koeficijenata funcije s desne strane, dobivamo relaciju za d_n. ■

Teorem 10.8.4. (Christoffel–Darbouxov identitet) *Neka je* $\{p_n(x) \mid n \geq 0\}$ *familija ortogonalnih polinoma na intervalu* $[a, b]$ *s težinskom funkcijom* $w(x) \geq 0$. *Za njih vrijedi sljedeći identitet*

$$\sum_{k=0}^{n} \frac{p_k(x)p_k(y)}{\gamma_k} = \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{a_n\gamma_n(x-y)}.$$

Dokaz:

Manipulacijom tročlane rekurzije. ■

10.9. Trigonometrijske funkcije

Trigonometrijske funkcije

$$\{1, \cos x, \cos 2x, \cos 3x, \ldots, \sin x, \sin 2x, \sin 3x, \ldots\}$$

čine ortogonalnu familiju funkcija na intervalu $[0, 2\pi]$ uz mjeru

$$d\lambda = \begin{cases} dx & \text{na } [0, 2\pi], \\ 0 & \text{inače.} \end{cases}$$

Pokažimo da je to zaista istina. Neka su $k, \ell \in \mathbb{N}_0$. Tada vrijedi

$$\int_0^{2\pi} \sin kx \cdot \sin \ell x \, dx = -\frac{1}{2} \int_0^{2\pi} (\cos(k+\ell)x - \cos(k-\ell)x) \, dx.$$

U slučaju da je $k = \ell$, onda je prethodni integral jednak

$$-\frac{1}{2}\left[\frac{\sin(k+\ell)x}{k+\ell} - x\right]\Bigg|_0^{2\pi} = \pi.$$

Ako je $k \neq \ell$, onda je jednak

$$-\frac{1}{2}\left[\frac{\sin(k+\ell)x}{k+\ell} - \frac{\sin(k-\ell)x}{k-\ell}\right]\Bigg|_0^{2\pi} = 0.$$

Drugim riječima, vrijedi

$$\int_0^{2\pi} \sin kx \cdot \sin \ell x \, dx = \begin{cases} 0, & k \neq \ell, \\ \pi, & k = \ell, \end{cases} \qquad k, \ell = 1, 2, \ldots,$$

Na sličan način, pretvaranjem produkta trigonometrijskih funkcija u zbroj, možemo pokazati da je

$$\int_0^{2\pi} \cos kx \cdot \cos \ell x \, dx = \begin{cases} 0, & k \neq \ell, \\ 2\pi, & k = \ell = 0, \\ \pi, & k = \ell > 0, \end{cases} \qquad k, \ell = 0, 1, \ldots,$$

te, također, da je

$$\int_0^{2\pi} \sin kx \cdot \cos \ell x \, dx = 0, \quad k = 1, 2, \ldots, \quad \ell = 0, 1, \ldots,$$

Ako periodičku funkciju f osnovnog perioda duljine 2π želimo aproksimirati redom oblika

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx),$$

onda, množenjem odgovarajućim trigonometrijskim funkcijama i integriranjem, za koeficijente u redu formalno dobivamo

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \, \cos kx \, dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \, \sin kx \, dx.$$

Prethodni red poznat je pod imenom Fourierov red, a koeficijenti kao Fourierovi koeficijenti.

Posebno, ako Fourierov red odsiječemo za $k = m$ i dobijemo trigonometrijski polinom, koji je najbolja L_2 aproksimacija za f u klasi trigonometrijskih polinoma stupnja manjeg ili jednakog m, obzirom na normu

$$\|u\|_2 = \left(\int_0^{2\pi} |u(t)|^2 \, dt \right)^{1/2}.$$

10.9.1. Diskretna ortogonalnost trigonometrijskih funkcija

Umjesto neprekidne, za pripadnu mjeru možemo uzeti i diskretnu mjeru, pa umjesto integrala, dobivamo sume. Da bismo dobili željeni razvoj moramo poznavati relacije diskretne relacije ortogonalnosti.

Teorem 10.9.1. *Za trigonometrijske funkcije, na mreži od točaka* $0, 1, \ldots, N$*, uz oznaku*

$$x_k = \frac{2\pi}{N+1} kx, \quad x_\ell = \frac{2\pi}{N+1} \ell x,$$

vrijede sljedeće relacije ortogonalnosti

$$\sum_{x=0}^{N} \sin x_k \sin x_\ell = \begin{cases} 0, & k \neq \ell \ i \ k = \ell = 0, \\ (N+1)/2, & k = \ell \neq 0, \end{cases}$$

$$\sum_{x=0}^{N} \sin x_k \cos x_\ell = 0$$

$$\sum_{x=0}^{N} \cos x_k \cos x_\ell = \begin{cases} 0, & k \neq \ell, \\ (N+1)/2, & k = \ell \neq 0, \\ N+1, & k = \ell = 0, \end{cases}$$

uz uvjet da je $k + \ell \leq N$*.*

Dokaz:

Dokažimo samo prvu relaciju. Iskoristimo formulu za pretvaranje produkta dva sinusa u zbroj trigonometrijskih funkcija. Vrijedi

$$\begin{aligned} \sin x_k \cdot \sin x_\ell &= \sin\left(\frac{2\pi}{N+1} kx\right) \cdot \sin\left(\frac{2\pi}{N+1} \ell x\right) \\ &= \frac{1}{2}\left[\cos\left(\frac{2\pi}{N+1}(k-\ell)x\right) - \cos\left(\frac{2\pi}{N+1}(k+\ell)x\right)\right]. \end{aligned}$$

Ako je $k + \ell \leq N$, onda su za $x = 0, 1, \ldots, N$ onda su argumenti prvog kosinusa s desne strane redom

$$0, \frac{2\pi}{N+1}(k-\ell), \frac{4\pi}{N+1}(k-\ell), \ldots, \frac{2N\pi}{N+1}(k-\ell).$$

Ako je $k = \ell$, onda su svi argumenti prvog kosinusa 0, pa je

$$\sum_{x=0}^{N} \cos\left(\frac{2\pi}{N+1}(k-\ell)x\right) = \sum_{x=0}^{N} \cos 0 = N+1, \quad k = \ell.$$

Za slučaj $k \neq \ell$ koristimo znanje iz kompleksne analize. Članovi $\dfrac{2\pi}{N+1}(k-\ell)$ podsjećaju na argumente $(N+1)$-og korijena iz 1 ("višak" je $(k-\ell)$). Označimo s ω_j, $j = 0, \ldots, N$ sve $(N+1)$-ve korijene iz 1,

$$\omega_j = \cos\frac{2\pi j}{N+1} + i \sin\frac{2\pi j}{N+1}. \tag{10.9.1}$$

Nadalje, označimo s

$$\omega = \cos\frac{2\pi}{N+1} + i \sin\frac{2\pi}{N+1}.$$

Primijetite da se tada svi $(N+1)$-vi korijeni iz 1 mogu, korištenjem De Moivreove formule (za potenciranje kompleksnih brojeva) napisati kao

$$\omega_j = \omega^j.$$

Očito svi ω^j zadovoljavaju jednadžbu $x^{N+1} - 1 = 0$. Iskoristimo li da su ω^j svi korijeni te jednadžbe, nju možemo napisati u faktoriziranom obliku kao

$$x^{N+1} - 1 = (x - \omega^0)(x - \omega^1)\cdots(x - \omega^N).$$

Uspoređivanjem članova uz N-tu potenciju slijeva i zdesna, dobivamo

$$0 = -\sum_{j=0}^{N} \omega^j.$$

Iskoristimo li (10.9.1) dobivamo

$$0 = \sum_{j=0}^{N} \omega^j = \sum_{j=0}^{N} \cos\frac{2\pi j}{N+1} + i \sum_{j=0}^{N} \sin\frac{2\pi j}{N+1}.$$

Budući da je kompleksan broj jednak 0, nuli moraju biti jednaki i njegov realni i njegov imaginarni dio. Drugim riječima, vrijedi

$$\sum_{j=0}^{N} \cos\frac{2\pi j}{N+1} = 0, \quad \sum_{j=0}^{N} \sin\frac{2\pi j}{N+1} = 0.$$

Vratimo se na početak. Trebali smo dokazati da je

$$\sum_{j=0}^{N} \cos\frac{2\pi j}{N+1}(k-\ell) = 0.$$

Primijetimo da su argumenti kosinusa u prethodnoj formuli argumenti $(N+1)$-og korijena iz 1, pomnoženi s $(k-\ell)$, što znači da su to argumenti od $(\omega^j)^{(k-\ell)} = (\omega^{(k-\ell)})^j$. To bi odgovaralo izboru korijena

$$\widetilde{\omega} = \cos\frac{2\pi(k-\ell)}{N+1} + i\sin\frac{2\pi(k-\ell)}{N+1}$$

umjesto ω. Odmah je jasno da vrijedi

$$\sum_{j=0}^{N}\widetilde{\omega}^j = \sum_{j=0}^{N}\omega^j,$$

pa je dokazano da je

$$\sum_{j=0}^{N}\cos\frac{2\pi j(k-\ell)}{N+1} = 0.$$

Na sličan se način pokazuje da je

$$\sum_{j=0}^{N}\cos\frac{2\pi j(k+\ell)}{N+1} = 0,$$

za $j+k \neq 0$, čime je pokazana relacija ortogonalnosti za sinuse. ■

Ovo znači da restrikcije funkcija

$$\cos\frac{2\pi}{N+1}kx, \quad \sin\frac{2\pi}{N+1}kx \tag{10.9.2}$$

pri čemu su dozvoljeni $k \in \mathbb{N}_0$ za kosinuse i $k \in N$ za sinuse, na mreži $\{0,\ldots,N\}$ možemo koristiti kao ortogonalnu familiju. Linearne kombinacije funkcija (10.9.2) zvat ćemo **trigonometrijski polinom**.

Nažalost baze takvih trigonometrijskih polinoma ovise o parnosti N.

Neparan broj točaka

Neka je zadan neparan broj točaka $\mathcal{M} = \{0,1,\ldots,N=2L\}$. Za bazu se tada uzima prvih $L+1$ kosinusa (prvi je konstanta) i prvih L sinusa, a pripadna trigonometrijska aproksimacija ima oblik

$$T_N(x) = \frac{a_0}{2} + \sum_{k=1}^{L}(a_k\cos x_k + b_k\sin x_k), \tag{10.9.3}$$

pri čemu je

$$x_k = \frac{2\pi}{N+1}kx := \frac{2\pi}{2L+1}kx.$$

Koeficijenti trigonometrijskog polinoma određuju se iz relacija ortogonalnosti na uobičajeni način, množenjem lijeve i desne strane u (10.9.3) izabranom funkcijom baze uz koju je odgovarajući koeficijent. Ako trigonometrijski polinom T_N interpolira funkciju f u $x \in \mathcal{M}$, tj. ako je $T_n(x) = f(x)$ onda množemjem (10.9.3) s $\cos x_\ell$, $\ell \geq 0$, i upotrebom relacija ortogonalnosti dolazimo do koeficijenata a_ℓ

$$f(x)\cos x_\ell = \frac{a_0}{2}\cos x_\ell + \sum_{k=1}^{L} a_k \cos x_k \cos x_\ell + \sum_{k=1}^{L} b_k \sin x_k \cos x_\ell.$$

Zbrajanjem po svim x dobivamo

$$\begin{aligned}\sum_{x=0}^{2L} f(x)\cos x_\ell &= \frac{a_0}{2}\sum_{x=0}^{2L}\cos 0\cos x_\ell + \sum_{k=1}^{L} a_k \sum_{x=0}^{2L}\cos x_k \cos x_\ell + \sum_{k=1}^{L} b_k \sum_{x=0}^{2L}\sin x_k \cos x_\ell \\ &= \frac{2L+1}{2}a_\ell.\end{aligned}$$

Odatle odmah zaključujemo da je (pišući k umjesto ℓ)

$$a_k = \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\cos x_k, \quad k = 0, \ldots, L.$$

Na sličan način, množenjem sa $\sin x_\ell$, $\ell > 0$, i zbrajanjem po svim x dobivamo

$$\begin{aligned}\sum_{x=0}^{2L} f(x)\sin x_\ell &= \frac{a_0}{2}\sum_{x=0}^{2L}\cos 0\sin x_\ell + \sum_{k=1}^{L} a_k \sum_{x=0}^{2L}\cos x_k \sin x_\ell + \sum_{k=1}^{L} b_k \sum_{x=0}^{2L}\sin x_k \sin x_\ell \\ &= \frac{2L+1}{2}b_\ell.\end{aligned}$$

Slično kao kod a_k, imamo

$$b_k = \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\sin x_k, \quad k = 1, \ldots, L.$$

Dakle, u slučaju neparnog broja točaka koeficijenti u (10.9.3) su

$$\begin{aligned}a_k &= \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\cos x_k, \quad k = 0, \ldots, L, \\ b_k &= \frac{2}{2L+1}\sum_{x=0}^{2L} f(x)\sin x_k, \quad k = 1, \ldots, L.\end{aligned}$$

Zadatak 10.9.1. *Pokažite da za bilo koju točku x^*, ne nužno iz $\mathcal{M}$ vrijedi*

$$T_N(x^*) = \frac{1}{2L+1}\sum_{x=0}^{2L} f(x)\Big(\sum_{k=0}^{2L}\cos\Big(\frac{2\pi}{2L+1}k(x-x^*)\Big)\Big).$$

Paran broj točaka

Neka je zadan paran broj točaka $\mathcal{M} = \{0, 1, \ldots, N = 2L-1\}$. Za bazu se tada uzima prvih $L+1$ kosinusa (prvi je konstanta) i prvih $L-1$ sinusa, a pripadna trigonometrijska aproksimacija ima oblik

$$T_N(x) = \frac{a_0}{2} + \sum_{k=1}^{L-1}(a_k \cos x_k + b_k \sin x_k) + \frac{1}{2}a_L \cos x_L, \tag{10.9.4}$$

pri čemu je

$$x_k = \frac{2\pi}{N+1}kx := \frac{\pi}{L}kx.$$

Na sličan način kao kod neparnog broja točaka, koeficijenti u (10.9.4) su

$$a_k = \frac{1}{L}\sum_{x=0}^{2L-1} f(x)\cos x_k, \quad k = 0, \ldots, L,$$
$$b_k = \frac{1}{2L}\sum_{x=0}^{2L-1} f(x)\sin x_k, \quad k = 1, \ldots, L-1.$$

Zadatak 10.9.2. *Pokažite da i u slučaju neparnog i u slučaju parnog broja točaka, T_N ima period $N+1$. Zbog toga se jednostavno koristi za interpolaciju trigonometrijskih funkcija, a dovoljno je zadati samo točke x iz jednog perioda.*

Primjer 10.9.1. *Funkcija f ima period 3 i zadana je tablično s*

x_k	0	1	2
f_k	0	1	1

Nađimo trigonometrijski polinom koji interpolira f u svim točkama iz $\mathbb{Z}$, a zatim izračunajmo $T_N(1/2)$ i $T_N(3/2)$.

Budući da je $N = 2$, broj točaka je neparan, pa je

$$T_2(x) = \frac{1}{2}a_0 + a_1 \cos\frac{2\pi}{3}x + b_1 \sin\frac{2\pi}{3}x.$$

Prema formulama za koeficijente, dobivamo

$$a_0 = \frac{2}{3}(0\cos 0 + 1\cdot\cos 0 + 1\cdot\cos 0) = \frac{4}{3}$$
$$a_1 = \frac{2}{3}\left(0\cos 0 + 1\cdot\cos\frac{2\pi}{3} + 1\cdot\cos\frac{4\pi}{3}\right) = -\frac{2}{3}$$
$$b_1 = \frac{2}{3}\left(0\sin 0 + 1\cdot\sin\frac{2\pi}{3} + 1\cdot\sin\frac{4\pi}{3}\right) = 0.$$

Prma tome, trigonometrijski polinom koji interpolira zadane točke je

$$T_2(x) = \frac{2}{3} - \frac{2}{3}\cos\frac{2\pi}{3}x.$$

Odatle se odmah može izračunati da je

$$T_2(1/2) = \frac{2}{3} - \frac{2}{3}\cos\frac{\pi}{3} = \frac{1}{3}$$
$$T_2(3/2) = \frac{2}{3} - \frac{2}{3}\cos\pi = \frac{4}{3}.$$

Metoda majnanjih kvadrata za trigonometrijske funkcije

I za metodu najmanjih kvadrata možemo koristiti trigonometrijske polinome, jer je dovoljno uzeti podskup baze prostora. Slično kao kod interpolacije biramo početni dio baze (10.9.2). Također moramo paziti na parnost/neparnost broja točaka N i na parnost/neparnost stupnja trigonometrijskog polinoma M, $M \leq N$.

Ilustrirajmo to na slučaju $N = 2L$ paran (broj točaka neparan) i $M = 2m$ paran (neparna dimenzija potprostora). Trigonometrijski polinom odgovarajućeg stupnja je

$$T_M(x) = \frac{1}{2}A_0 = \sum_{k=1}^{m}(A_k \cos x_k + B_k \sin x_k), \tag{10.9.5}$$

gdje je

$$x_k = \frac{2\pi}{N+1}kx := \frac{2\pi}{2L+1}kx.$$

Metoda najmanjih kvadrata minimizira kvadrat greške

$$S = \sum_{x=0}^{2L}(f(x) - T_M(x))^2 \to \min.$$

Tvrdimo da je rješenje problema mininizacije trigonometrijski interpolacijski polinom kojemu je

$$A_k = a_k, \quad k = 0, \ldots, m$$
$$B_k = b_k, \quad k = 1, \ldots, m,$$

a koeficijenti a_k i b_k se računaju po formulama za interpolaciju. Primijetite da u točkama interpolacije x, $x = 0, \ldots, 2L$ interpolacijski polinom ima istu vrijednost kao funkcija f, pa je dovoljno (u točkama interpolacije) uspoređivati interpolacijski trigonometrijski polinom T_N, $N = 2L$ i trigonometrijski polinom T_M, $M = 2m$ dobiven metodom najmanjih kvadrata. Vrijedi

$$T_N(x) - T_M(x) = \frac{1}{2}(a_0 - A_0) + \sum_{k=1}^{m}\left((a_k - A_k)\cos x_k + (b_k - B_k)\sin x_k\right)$$
$$+ \sum_{k=m+1}^{L}(a_k \cos x_k + b_k \sin x_k).$$

Dakle, u točkama interpolacije x vrijedi

$$f(x) - T_M(x) = T_N(x) - T_M(x).$$

Greška S koju minimiziramo dobiva se upotrebom relacija ortogonalnosti. Izlazi

$$\begin{aligned}
S &:= \sum_{x=0}^{2L} (T_N(x) - T_M(x))^2 \\
&= \sum_{x=0}^{2L} \frac{1}{4}(a_0 - A_0)^2 + \sum_{x=0}^{2L} \sum_{k=1}^{m} \left((a_k - A_k)\cos x_k + (b_k - B_k)\sin x_k\right)^2 \\
&\quad + \sum_{x=0}^{2L} \sum_{k=m+1}^{L} (a_k \cos x_k + b_k \sin x_k)^2 \\
&= \frac{1}{4}(a_0 - A_0)^2 \cdot (2L+1) + \sum_{x=0}^{2L} \sum_{k=1}^{m} \Big[(a_k - A_k)^2 \cos^2 x_k \\
&\qquad + 2(a_k - A_k)(b_k - B_k)\cos x_k \sin x_k + (b_k - B_k)^2 \sin^2 x_k\Big] \\
&\quad + \sum_{x=0}^{2L} \sum_{k=m+1}^{L} (a_k^2 \cos^2 x_k + 2a_k b_k \cos x_k \sin x_k + b_k^2 \sin^2 x_k) \\
&= \frac{1}{4}(a_0 - A_0)^2 \cdot (2L+1) + \frac{2L+1}{2} \sum_{k=1}^{m} (a_k - A_k)^2 + (b_k - B_k)^2) \\
&\quad + \frac{2L+1}{2} \sum_{k=m+1}^{L} (a_k^2 + b_k^2).
\end{aligned}$$

Prema tome, odmah je vidljivo da je greška S minimalna ako je

$$\begin{aligned}
A_k &= a_k, \quad k = 0, \ldots, m \\
B_k &= b_k, \quad k = 1, \ldots, m,
\end{aligned}$$

i njena minimalna vrijednost jednaka je

$$S_{\min} = \frac{2L+1}{2} \sum_{k=m+1}^{L} (a_k^2 + b_k^2).$$

Ovaj oblik minimalne greške nije praktičan, jer uobičajeno ne znamo a_k, b_k za $k > m$.

Zadatak 10.9.3. *Dokažite da vrijedi*

$$S_{\min} = \sum_{x=0}^{2L} (f(x))^2 - \frac{2L+1}{4} a_0^2 - \frac{2L+1}{2} \sum_{k=1}^{m} (a_k^2 + b_k^2)$$

korištenjem relacija ortogonalnosti. Prethodni oblik greške često se koristi za detekciju stupnja trigonometrijskog polinoma, jer nagli pad greške pri dizanju stupnja trigonometrijskog polinoma znači da smo otkrili stupanj polinoma. Greška pritom ne mora biti 0*, jer je pojava mogla imati slučajne greške koje smo ionako željeli maknuti.*

Zadatak 10.9.4. *Izvedite metodu najmanjih kvadrata za tri preostala slučaja:*

1. *broj točaka paran $N = 2L - 1$, dimenzija prostora neparna $M = 2m$,*
2. *broj točaka neparan $N = 2L$, dimenzija prostora parna $M = 2m - 1$,*
3. *broj točaka paran $N = 2L - 1$, dimenzija prostora parna $M = 2m - 1$.*

Zadatak 10.9.5. *Neka je funkcija f zadana na mreži točaka $\mathcal{M} = \{0, 1, \ldots, P-1\}$, P neparan (tj. točaka je paran broj) i neka je P period funkcije f, tj.*

$$f(x + P) = f(x).$$

Pokažite da su tada

$$a_k = \frac{2}{P} \sum_{x=-L+1}^{L} f(x) \cos \frac{2\pi}{P} kx, \quad k = 0, \ldots, L$$
$$b_k = \frac{2}{P} \sum_{x=-L+1}^{L} f(x) \sin \frac{2\pi}{P} kx, \quad k = 1, \ldots, L-1.$$

Ako je f neparna funkcija $f(-x) = -f(x)$, pokažite da je

$$a_k = 0, \quad k = 0, \ldots, L$$
$$b_k = \frac{4}{P} \sum_{x=1}^{L-1} f(x) \sin \frac{2\pi}{P} kx, \quad k = 1, \ldots L-1.$$

Ako je f parna funkcija $f(-x) = f(x)$, pokažite da je

$$a_k = \frac{2}{P}(f(0) + f(L) \cos k\pi) + \frac{4}{P} \sum_{x=1}^{L-1} f(x) \cos \frac{2\pi}{P} kx, \quad k = 0, \ldots, L$$
$$b_k = 0, \quad k = 1, \ldots L-1.$$

Zadatak 10.9.6. *Riješite prethodni zadatak uz uvjet da je $P = 2L + 1$, tj. da funkcija ima neparan period.*

10.10. Minimaks aproksimacija

Neka je f neprekidna funkcija na $[a, b]$. Ako uspoređujemo polinomne aproksimacije funkcije f dobivene različitim metodama, pitamo se koja je od njih najbolja, tj. koja daje najmanju maksimalnu grešku. Označimo s $\rho_n(f)$ maksimalnu grešku aproksimacije

$$\rho_n(f) = \inf_{\deg(p) \leq n} \|f - p\|_\infty.$$

To znači de ne postoji polinom stupnja manjeg ili jedmakog n koji bi bolje od p aproksimirao funkciju f na danom intervalu. Nas, naravno interesira za koji se polinom dostiže ta greška

$$\rho_n(f) = \|f - p_n^*\|_\infty.$$

Ako je polinom p_n jedinstven, zanima nas kako ga možemo konstruirati. Polinom p_n^* zovemo minimaks aproksimacija funkcije f na intervalu $[a, b]$.

Pokažimo kako se najbolja aproksimacija ponaša na jednom jednostavnom primjeru.

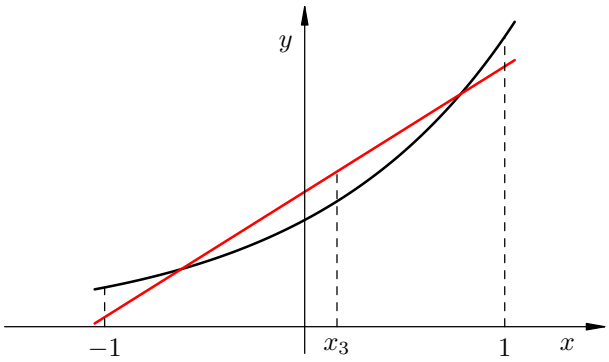
Primjer 10.10.1. *Nađimo polinom prvog stupnja $p_1^*(x) = a_0 + a_1 x$ koji je minimaks aproksimacija funkcije $f(x) = e^x$ na intervalu $[-1, 1]$, tj. da vrijedi*

$$\max_{x\in[-1,1]} |e^x - a_0 - a_1 x| \to \min .$$

Da bismo riješili problem, potrebno je malo geometrijskog zora. Nacrtajmo graf funkcije e^x i promatrajmo grešku svih polinomnih aproksimacija

$$\mathrm{err}(x) = e^x - (a_0 + a_1 x)$$

na zadanom intervalu.



Prvo, odmah je jasno da linearna minimaks aproksimacija mora sjeći graf funkcije e^x na zadanom intervalu u točno dvije točke, nazovimo ih x_1, x_2 takve da je $-1 < x_1 < x_2 < 1$. U protivnom, ako polinom ne siječe graf niti u jednoj točki, ili ako ga siječe u točno jednoj točki, može se pokazati da postoji bolja aproksimacija. Pokažite to!

Iz crteža odmah naslućujemo i rješenje. Nađimo jednadžbu pravca kroz točke $(-1, e^{-1})$ i $(1, e)$. Zatim, nađimo koeficijent a_0 takav da je taj pravac tangenta funkcije e^x u nekoj točki x_3. Rješenje zadatka je pravac paralelan s prethodna dva, jednako udaljen od oba. Odmah je jasno da će postojati točno tri točke u kojima će se dostizati maksimalne pogreške: rubovi intervala i x_3.

Pokažimo sad precizno da su naša zaključivanja ispravna. Označimo

$$\rho_1 = \max_{x\in[-1,1]} |\operatorname{err}(x)|.$$

Već smo zaključili da pogreška mora imati tri ekstrema, tj. mora vrijediti

$$\operatorname{err}(-1) = \rho_1, \quad \operatorname{err}(1) = \rho_1, \quad \operatorname{err}(x_3) = -\rho_1.$$

Budući da je $\operatorname{err}(x)$ *derivabilna, onda možemo uvjet maksimuma pogreške u* x_3 *napisati i korištenjem derivacije, tj.* $\operatorname{err}'(x_3) = 0$.

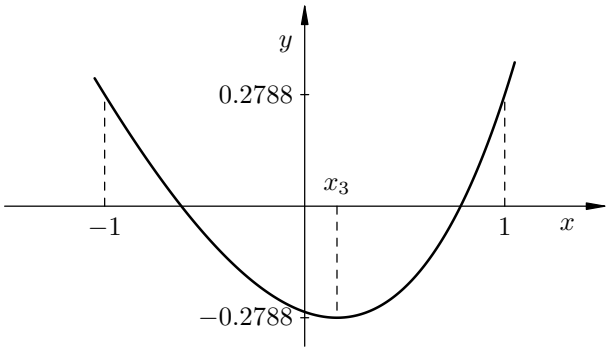
Sad možemo skupiti sve četiri jednadžbe koje trebamo zadovoljiti

$$\begin{aligned} e^{-1} - a_0 + a_1 &= \rho_1 & \qquad e^{x_3} - a_0 - a_1 x_3 &= -\rho_1 \\ e - a_0 - a_1 &= \rho_1 & \qquad e^{x_3} - a_1 &= 0. \end{aligned}$$

Rješenje te četiri jednadžbe je

$$\begin{aligned} a_1 &= \frac{e - e^{-1}}{2} \approx 1.1752 \\ x_3 &= \ln a_1 \approx 0.1614 \\ \rho_1 &= \frac{1}{2}e^{-1} + \frac{x_3}{4}(e - e^{-1}) \approx 0.2788 \\ a_0 &= \rho_1 + (1 - x_3)a_1 \approx 1.2643. \end{aligned}$$

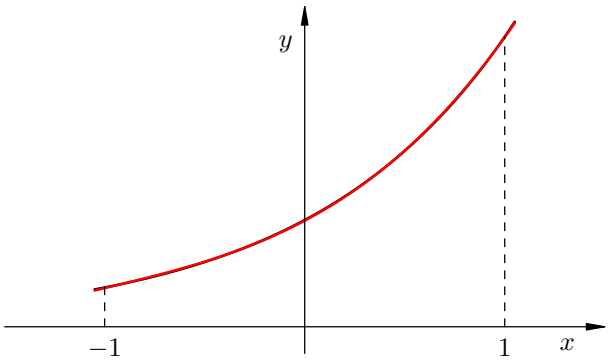
Graf pogreške ima karakterističan oscilirajući izgled.



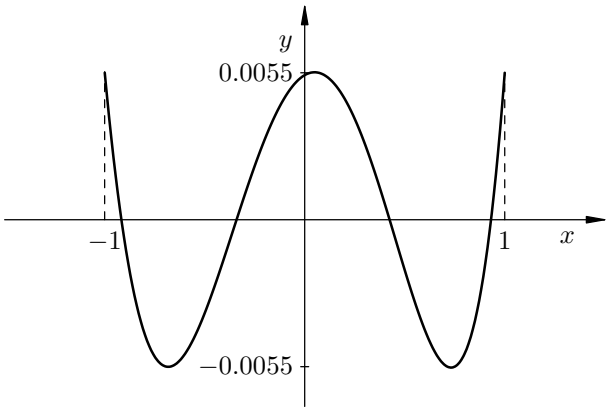
Korištenjem tzv. Remesovog algoritma možemo konstruirati i kubični polinom koji najbolje aproksimira istu funkciju. Taj polinom je

$$p_3^*(x) = 0.994579 + 0.995668x + 0.542973x^2 + 0.179533x^3.$$

Ako nacrtamo graf tog polinoma, on se na slici neće razlikovati od funkcije,



jer će greška biti iznimno mala, reda veličine 0.0055 *i opet karakteristično oscilirajuća.*



Za dobru uniformnu aproksimaciju zadane funkcije f, realno je očekivati da je greška jednako tako uniformno distribuirana na intervalu aproksimacije i da varira po predznaku. Iznijet ćemo dva vrlo važna teorema, od kojih prvi daje egzistenciju minimaks aproksimacije i važno svojstvo o oscilaciji grešaka. Drugi ocjenjuje pogrešku minimaks aproksimacije ρ_n polinoma stupnja n, bez da se sam polinom izračuna.

Teorem 10.10.1. (Čebiševljev teorem o oscilacijama grešaka) *Neka je* $f \in C[a,b]$ *i* $n \geq 0$. *Tada postoji jedinstven polinom* p_n^* *stupnja manjeg ili jednakog* n *za koji je*

$$\rho_n(f) = \|f - p_n^*\|_\infty.$$

Taj polinom je karakteriziran sljedećim svojstvom: postoje barem $n+2$ *točke*

$$a \leq x_0 < x_1 < \cdots < x_n < x_{n+1} \leq b$$

za koje je

$$f(x_j) - p_n^*(x_j) = \sigma(-1)^j \rho_n(f), \quad j = 0, \ldots, n+1,$$

pri čemu je $\sigma = \pm 1$ *i ovisi samo o* f *i* n.

Dokaz prethodnog teorema je tehnički, vrlo dugačak i provodi se obratom po kontrapoziciji.

Teorem 10.10.2. (de la Vallee–Poussin) *Neka je $f \in C[a,b]$ i $n \geq 0$. Pretpostavimo da polinom P stupnja manjeg ili jednakog n zadovoljava*

$$f(x_j) - P(x_j) = (-1)^j e_j, \quad j = 0, 1, \ldots, n+1$$

a e_j su različiti od 0 i istog predznaka, za x_j vrijedi

$$a \leq x_0 < x_1 < \cdots < x_n < x_{n+1} \leq b.$$

Tada je

$$\min_{0 \leq j \leq n+1} |e_j| \leq \rho_n(f) = \|f - p_n^*(x)\|_\infty \leq \|f - P\|_\infty.$$

Dokaz:

Posljednja nejednakost (gornja ograda) u prethodnoj formuli posljedica je definicije minimaks aproksimacije, tj. da polinom p_n^* ima maksimum pogreške manji ili jednak od svih ostalih polinoma P.

Donja ograda za $\rho_n(f)$ dokazuje se pretpostavljanjem suprotnog. Pretpostavimo da je

$$\rho_n(f) < \min_{0 \leq j \leq n+1} |e_j|.$$

Budući da je $\rho_n(f)$ infimum (zaboravimo načas da smo dokazali i minimum), onda sigurno postoji bar jedan polinom Q stupnja manjeg ili jednakog n koji se nalazi između $\rho_n(f)$ i minimuma $|e_j|$, tj. vrijedi

$$\rho_n(f) \leq \|f - Q\|_\infty < \min_{0 \leq j \leq n+1} |e_j|.$$

Primijetite da polinomi P i Q nisu jednaki! Definiramo

$$R(x) = P(x) - Q(x).$$

R je polinom stupnja manjeg ili jedankog n. Zbog jednostavnosti, pretpostavimo da su svi $e_j > 0$ (isti argument radit će i ako su svi manji od 0). Izračunajmo vrijednosti polinoma R u točkama x_j. Počnimo s x_0 i promotrimo predznak rezultata

$$R(x_0) = P(x_0) - Q(x_0) = (f(x_0) - Q(x_0)) - (f(x_0) - P(x_0)) = (f(x_0) - Q(x_0)) - e_0.$$

Budući da je

$$|f(x_0) - Q(x_0)| < \min |e_j| = \min e_j \leq e_0,$$

onda je

$$R(x_0) = f(x_0) - Q(x_0)) - e_0 < 0.$$

Nadalje je

$$R(x_1) = P(x_1) - Q(x_1) = (f(x_1) - Q(x_1)) - (f(x_1) - P(x_1)) = (f(x_1) - Q(x_1)) + e_1.$$

Ponovno, zbog

$$|f(x_1) - Q(x_1)| < \min |e_j| = \min e_j \leq e_1,$$

slijedi da je

$$R(x_1) = (f(x_1) - Q(x_1)) + e_1 > 0.$$

Induktivno, dobivamo da je

$$\operatorname{sign}(R(x_j)) = (-1)^{j+1}, \quad j = 0, \ldots, n+1,$$

tj. R oscilira tako da ima bar $n+2$ različita predznaka, tj. da predznak promijeni bar $n+1$ puta. Ali R je polinom stupnja n, pa predznak može mijenjati samo u nultočkama. Po osnovnom teoremu algebre znamo da polinom R stupnja $n \geq 1$ ima točno n nultočaka, a ne bar $(n+1)$-nu. Jedina mogućnost koja ostaje je da je R baš nul-polinom, tj. da je $P = Q$, što je suprotno pretpostavci. ■

Predodžbu o tome kako se ponaša najbolja aproksimacija s porastom stupnja n daje sljedeći teorem.

Teorem 10.10.3. (Jackson) *Neka funkcija f ima k neprekidnih derivacija za neki $k \geq 0$. Čak štoviše pretpostavimo da $f^{(k)}$ zadovoljava*

$$\sup_{a \leq x, y \leq b} |f^{(k)}(x) - f^{(k)}(y)| \leq M|x - y|^{\alpha}$$

za neki $M > 0$ i neki $0 < \alpha \leq 1$, tj. kaže se da f zadovoljava Hölderov uvjet s eksponentom α. Tada postoji konstanta d_k nezavisna o f i n za koju je

$$\rho_n(f) \leq \frac{M d_k}{n^{k+\alpha}}, \quad n \geq 1.$$

Ako u prethodnom teoremu želimo izbjeći Hölderov uvjet, dovoljno je pretpostaviti da f im k neprekidnih derivacija. Umjesto k-te derivacije svugdje dalje u teoremu koristimo $(k-1)$-u, stavljamo $\alpha = 1$ i

$$M = \|f^{(k)}\|_{\infty}.$$

Tada se Hölderov uvjet svede na obični teorem srednje vrijednosti, a kao rezultat dobivamo

$$\rho_n(f) \leq \frac{d_{k-1}}{n^k} \|f^{(k)}\|_{\infty}.$$

Nadalje, ako je f beskonačno puta derivabilna, tada p_n^* konvergira prema f uniformno na $[a, b]$ brže nego bilo koja potencija $1/n^k$, $k \geq 1$.

10.10.1. Remesov algoritam

Traženje minimaks aproksimacije p_n^* za f može se pronaći iterativnim algoritmom poznatijim kao drugi Remesov algoritam. Ovdje treba napomenuti da se taj algoritam može generalizirati i na racionalne funkcije i na slučaj kad funkcija f nije zadana na intervalu, nego na skupu točaka (tada se on zove diferencijalni algoritam korekcije).

Remesov algoritam koristi svojstvo oscilacije greške koje mora imati minimaks aproksimacija. Iteracije imaju tri dijela.

Prvi korak

Zadane su $n+2$ točke

$$a \leq x_0^{(0)} < x_1^{(0)} < \cdots < x_n^{(0)} < x_{n+1}^{(0)} \leq b.$$

koje određuju polinom p stupnja $\deg(p) \leq n$ iz uvjeta

$$f(x_i^{(0)}) - p(x_i^{(0)}) = (-1)^i E, \quad i = 0, \ldots, n+1,$$

pri čemu zahtjevamo da pogreška E (koju još ne znamo) oscilira s jednakim amplitudama. Prehodna ralacija vodi na linearni sustav s $n+2$ nepoznanice od kojih su $n+1$ koeficijenti polinoma p, a posljednja je E.

Drugi korak

Riješimo linearni sustav, tj. odredimo redom $a_0^{(0)}, \ldots, a_n^{(0)}$ (koeficijente polinoma p) i nađimo pirpadni E, zovimo ga E_0.

Treći korak

Tražimo novih $n+2$ točaka. Definiramo funkciju

$$h_0(x) = f(x) - \sum_{i=0}^{n} a_i^{(0)} x^i.$$

Funkcija h_0 ima u točkama $x_i^{(0)}$ vrijednosti $\pm E_0$ (to su greške) koje alterniraju po predznaku. Zbog toga, nije teško pokazati da u okolini svake točke $x_i^{(0)}$ postoji točka $x_i^{(1)}$ takva da u njoj $h_0(x)$ ima ekstrem i to istog predznaka kao što je predznak $f(x_i^{(0)}) - p(x_i^{(0)})$. Nakon toga zamjenjujemo $x_i^{(0)}$ s $x_i^{(1)}$. Naravno, rubne točke $x_0^{(1)}$ i $x_{n+1}^{(1)}$ moraju ostati u $[a, b]$.

Ove nove točke $x_i^{(1)}$ "lokalnih" ekstrema funkcije h_0 moraju sadržavati i točku u kojoj $|h_0|$ dostiže globalni maksimum na $[a, b]$. Naime, ako je $\bar{x}$ točka u kojoj $|h_0|$ poprima globalnu maksimalnu vrijednost na $[a, b]$

$$\|h_0\|_\infty = \|f - p\|_\infty = |f(\bar{x}) - p(\bar{x})|,$$

i ona nije među točakama $x_i^{(1)}$, onda treba onda treba zamijeniti jednu od točaka $x_i^{(1)}$ točkom $\bar{x}$, tako da h_0 i na tom novom skupu točaka alternira po znaku. Može se pokazati da se to uvijek može napraviti.

Primjernom teorema 10.10.2. dobivamo da je

$$m := \min_{i=0,\ldots,n+1} |f(x_i^{(1)}) - p(x_i^{(1)})| \leq \rho_n(f) \leq M := \max_{i=0,\ldots,n+1} |f(x_i^{(1)}) - p(x_i^{(1)})|.$$

Ako je omjer M/m dovoljno blizu 1, smatramo da je nađeni p dovoljno blizu polinomne minimaks aproksimacije za f.

U protivnom, treba ponoviti drugi, pa treći korak, samo na novim točkama $x_i^{(1)}$. Ovaj proces generirat će niz polinomnih aproksimacija će uniformno konvergirati prema minimaks polinomnoj aproksimaciji.

10.11. Skoro minimaks aproksimacije

Vidjeli smo da minimaks aproksimaciju nije jako lako izračunati. Zbog toga, bili bismo zadovoljni i približnom minimaks aproksimacijom. Ako se prisjetimo Čebiševljevog teorema o oscilaciji greške, možemo doći do metode koja daje dobru približnu minimaks aproksimaciju.

Vidjet ćemo da su u tu svrhu vrlo pogodni Čebiševljevi polinomi, koji zadovoljavaju sljedeću relaciju ortogonalnosti

$$\int_{-1}^{1} \frac{T_m(x)\, T_n(x)}{\sqrt{1-x^2}}\, dx = \begin{cases} 0, & \text{za } m \neq n, \\ \pi, & \text{za } m = n = 0, \\ \pi/2, & \text{za } m = n \neq 0. \end{cases}$$

Ako želimo funkciju f razviti po Čebiševljevim polinomima, potrebno je samo supstituirati podatke u opći algoritam. Želimo odrediti koeficijente c_k u razvoju funkcije f po Čebiševljevim polinomima. Ako je razvoj napišemo u obliku

$$f(x) = \frac{c_0}{2} T_0(x) + \sum_{i=1}^{\infty} c_i T_i(x), \tag{10.11.1}$$

onda ćemo, formalno gledajući, koeficijent c_j dobiti ako pomnožimo prethodnu relaciju s $T_j(x)$, zatim težinskom funkcijom w i integriramo od -1 do 1. Dobivamo formulu

$$c_j = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_j(x)}{\sqrt{1-x^2}}\, dx.$$

Označimo s f_n početni komad razvoja, tj. neka je

$$f_n(x) = \frac{c_0}{2} T_0(x) + \sum_{i=1}^{n} c_i T_i(x).$$

Ako je $f \in C[-1,1]$ tada razvoj (10.11.1) konvergira, u smislu da je

$$\lim_{n\to\infty} \int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} \Big(f(x) - f_n(x)\Big)^2 dx = 0.$$

Za uniformnu konvergenciju, možemo pokazati dosta jak rezultat

$$\rho_n(f) \le \|f - f_n\|_\infty \le \Big(4 + \frac{4}{\pi^2} \ln n\Big)\rho_n(f).$$

Kako se ponaša greška odbacivanja reda? Ako se prisjetimo da je

$$T_n(x) = \cos n\vartheta, \quad x = \cos\vartheta,$$

onda se može pokazati da vrijedi

$$f(x) - f_n(x) = \sum_{i=n+1}^{\infty} c_i T_i(x) \approx c_{n+1} T_{n+1}(x) = c_{n+1} \cos(n+1)\vartheta,$$

ako je $c_{n+1} \neq 0$ i ako koeficijenti c_i brzo konvergiraju k 0. Iz definicije T_{n+1} izlazi

$$|T_{n+1}(x)| = |\cos(n+1)\vartheta| \le 1, \quad -1 \le x \le 1.$$

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

$$x_j = \cos\Big(\frac{(2j+1)\pi}{2(n+1)}\Big), \quad j = 0, \ldots, n.$$

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

$$x_k = \cos\Big(\frac{k\pi}{(n+1)}\Big), \quad k = 0, \ldots, n+1.$$

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. Ako to iskoristimo za funkciju $c_{n+1}T_{n+1}$, onda je jasno da ona ima $n+2$ lokalna ekstrema jednakih amplituda. Po Čebiševljevom teoremu o oscilaciji grešaka, odatle odmah izlazi da je f_n skoro minimaks aproksimacija za f (ovo skoro minimaks potječe od toga što je greška odbacivanja članova reda približno jednaka $c_{n+1}T_{n+1}$).

Postoji još jedan razlog zašto se koristi razvoj po Čebiševljevim polinomima. Vrijedi sljedeći teorem.

Teorem 10.11.1. *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

$$\frac{1}{2^{n-1}}T_n(x) = x^n + \text{članovi nižeg stupnja}.$$

Budući da su točke

$$x_k = \cos\left(\frac{k\pi}{n}\right), \quad j = 0, \ldots, n,$$

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

$$\tau_n \leq \max_{-1\leq x\leq 1} |M(x)| < \frac{1}{2^{n-1}}. \tag{10.11.2}$$

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 (10.11.2) 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. ■

10.12. Interpolacija u Čebiševljevim točkama

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. Dakle izaberimo točke interpolacije $x_j \in [-1,1]$ tako da minimiziraju

$$\max_{-1\le x\le 1} |(x-x_0)\cdots(x-x_n)|.$$

Polinom u prethodnoj relaciji je stupnja $n+1$ i ima vodeći koeficijent 1. Po Teoremu 10.11.1., 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 $x_0,\dots x_n$ nultočke polinoma T_{n+1}, a njih smo već izračunali da su jednake

$$x_j = \cos\Big(\frac{(2j+1)\pi}{2n+2}\Big), \quad j = 0,\dots,n.$$

10.13. Čebiševljeva ekonomizacija

Čebiševljevi polinomi mogu se koristiti za smanjivanje stupnja interpolacionog polinoma, uz minimalni gubitak točnosti. Takav postupak zove se ekonomizacija.

Pretpostavimo da je zadan proizvoljni polinom stupnja n

$$p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0.$$

na intervalu $[-1,1]$. Taj polinom želimo zamijeniti polinomom stupnja za jedan manjeg tako da je greška koja je pritom nastala minimalna moguća

$$\max_{x\in[-1,1]} |p_n(x) - p_{n-1}(x)| \to \min.$$

Rješenje problema se neće promijeniti ako normiramo vodeći koeficijent na 1, tj. ako tražimo

$$\max_{x\in[-1,1]} \Big|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\Big| \to \min.$$

Prema Teoremu 10.11.1. o minimalnom otklanjanju od polinoma x^n, izlazi da mora biti

$$\max_{x\in[-1,1]} \Big|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\Big| \ge \frac{1}{2^{n-1}},$$

s tim da jednakost vrijedi kad je

$$\frac{1}{a_n}(p_n(x) - p_{n-1}(x)) = \frac{1}{2^{n-1}}T_n(x).$$

Drugim riječima, izbor $p_{n-1}(x)$ je

$$p_{n-1}(x) = p_n(x) - \frac{a_n}{2^{n-1}}T_n(x),$$

a s tim izborom je maksimalna greška jednaka

$$\max_{x\in[-1,1]} |p_n(x) - p_{n-1}(x)| = |a_n| \max_{x\in[-1,1]} \left|\frac{1}{a_n}(p_n(x) - p_{n-1}(x))\right| = \frac{|a_n|}{2^{n-1}}.$$

Primjer 10.13.1. *Funkciju $f(x) = e^x$ aproksimiramo na intervalu $[-1, 1]$ Taylorovim polinomom oko 0 stupnja četiri*

$$p_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24}.$$

Greška odbacivanja tog polinoma je

$$|R_4(x)| \leq \frac{M_5}{5!}|x^5|, \quad M_5 = \max_{x\in[-1,1]} |f^{(5)}(x)|.$$

Odmah se vidi da je $M_5 = e$, pa je greška odbacivanja

$$|R_4(x)| \leq \frac{e}{120}|x^5| \leq \frac{e}{120} \approx 0.023,$$

za $-1 \leq x \leq 1$. Pretpostavimo da možemo tolerirati grešku 0.05, pa pokušajmo spustiti stupanj polinoma. Nije teško naći T_4, recimo iz rekurzije

$$T_4(x) = 8x^4 - 8x^2 + 1.$$

Pokazali smo da mora biti

$$\begin{aligned} p_3(x) &= p_4(x) - \frac{a_4}{2^{4-1}}T_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} - \frac{1}{8\cdot 24}(8x^4 - 8x^2 + 1) \\ &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3. \end{aligned}$$

Tim snižavanjem stupnja, napravljena je greška

$$|p_4(x) - p_3(x)| \leq \frac{1}{24\cdot 2^3} = \frac{1}{192} \leq 0.0053.$$

Pribrojimo li tu grešku grešci odbacivanja, onda je ukupna greška manja ili jednaka $0.023 + 0.0053 = 0.0283$, što je prema uvjetima zadatka dozvoljiva greška.

Naravno, stupanj polinoma p_3 možemo pokušati još smanjiti. Budući da je

$$T_3(x) = 4x^3 - 3x,$$

dobivamo

$$\begin{aligned} p_2(x) = p_3(x) - \frac{a_3}{2^{3-1}} T_3(x) &= \frac{191}{192} + x + \frac{13}{24}x^2 + \frac{1}{6}x^3 - \frac{1}{4 \cdot 6}(4x^3 - 3x) \\ &= \frac{191}{192} + \frac{9}{8}x + \frac{13}{24}x^2. \end{aligned}$$

Pritom je napravljena greška

$$|p_3(x) - p_2(x)| = \frac{1}{6 \cdot 4} = \frac{1}{24} \approx 0.042.$$

Dodamo li tu grešku na već akumuliranu grešku 0.0283*, onda je ukupna greška* $0.0283 + 0.042 > 0.05$*, što je bila dozvoljena tolerancija.*

Postoji još jedan način ekonomizacije. Znamo da se funkcije mogu razviti u red po Čebiševljevim polinomima oblika

$$f(x) = \frac{1}{2}c_0 + \sum_{k=1}^{\infty} c_k T_k(x), \quad |x| \leq 1,$$

a koeficijenti u razvoju su integrali koji u sebi sadrže f, Čebiševljev polinom i težinsku funkciju za Čebiševljeve polinome. Koeficijente c_k u ovom razvoju možemo, i to relativno brzo, numerički izračunati, koristeći algoritme na bazi diskretne ortogonalnosti Čebiševljevih polinoma. Taj postupak opisujemo u sljedećem odjeljku.

S druge strane, ako znamo (ili lako računamo) koeficijente a_m u redu potencija

$$f(x) = \sum_{m=0}^{\infty} a_m x^m,$$

onda potencije x^m možemo razviti po Čebiševljevim polinomima, pa nakon toga primijeniti postupak ekonomizacije (odbacivanjem članova) da dobijemo ravnomjernije ponašanje pogreške.

Može se pokazati da vrijedi

$$(2x)^n = 2 \sum_{k=1}^{[n/2]} \epsilon_k \binom{n}{k} T_{n-2k}(x),$$

pri čemu je

$$\epsilon_k = \begin{cases} 1/2 & \text{za } k = n/2, \\ 1 & \text{inače.} \end{cases}$$

Taj razvoj napisan posebno za parne, a posebno za neparne potencije je

$$(2x)^{2m} = \binom{2m}{m} + 2\sum_{k=1}^{m} \binom{2m}{m-k} T_{2k}(x)$$
$$(2x)^{2m+1} = 2\sum_{k=0}^{m} \binom{2m+1}{m-k} T_{2k+1}(x).$$

Ako za polinomnu aproksimaciju uzmemo polinom stupnja n, uvrštavanjem razvoja po Čebiševljevim polinomima dobivamo

$$f_n(x) = \sum_{m=0}^{n} a_m x^m = \frac{b_0}{2} + \sum_{m=1}^{n} b_m T_m(x),$$

gdje su koeficijenti

$$b_{2k} = 2\sum_{i=k}^{2i\le n} \binom{2i}{i-k} \frac{a_{2i}}{2^{2i}}, \quad b_{2k+1} = 2\sum_{i=k}^{2i+1\le n} \binom{2i+1}{i-k} \frac{a_{2i+1}}{2^{2i}}.$$

Vrlo se često događa da su jedan ili nekoliko posljednjih koeficijenata b_m maleni, pa odbacivanje posljednjih članova bitno ne smanjuje točnost aproksimacije. Drugim riječima, smanjili smo stupanj aproksimacije. Takvo smanjivanje stupnja zove se i relaksacija stupnja aproksimacije.

10.14. Diskretne ortogonalnosti polinoma T_n

Budući da su Čebiševljevi polinomi kosinusi, onda oni zadovoljavaju diskretne relacije ortogonalnosti vrlo slične onima koje zadovoljavaju trigonometrijske funkcije.

Neka su x_α nultočke Čebiševljevog polinoma T_n, tj. neka je

$$T_n(x_\alpha) = \cos(n\vartheta_\alpha) = 0.$$

Nije teško izračunati da je tada

$$x_\alpha = \cos(\vartheta_\alpha), \quad \vartheta_\alpha = \frac{(2\alpha+1)\pi}{2n}, \quad \alpha = 0, \ldots, n-1.$$

Za Čebiševljeve polinome, u nultočkama vrijede sljedeće relacije ortogonalnosti

$$U_{j,k} = \sum_{\alpha=0}^{n-1} T_j(x_\alpha) T_k(x_\alpha) = \sum_{\alpha=0}^{n-1} \cos j\vartheta_\alpha \cos k\vartheta_\alpha,$$

gdje je

$$U_{j,k} = \begin{cases} 0 & j,k < n,\ j \neq k, \\ n/2 & j = k,\ 0 < j < n, \\ n & j = k = 0. \end{cases}$$

Sada možemo funkciju razviti po Čebiševljevim polinomima koristeći prethodnu relaciju diskretne ortogonalnosti. Može se pokazati da vrijedi sljedeći teorem.

Teorem 10.14.1. *Neka je $f_n(x)$ aproksimacija za $f(x)$,*

$$f_n(\cos\vartheta) = \frac{d_0}{2} + \sum_{k=1}^{n-1} d_k \cos k\vartheta,$$

ili

$$f_n(x) = \frac{d_0}{2} + \sum_{k=1}^{n-1} d_k T_k(x). \tag{10.14.1}$$

Tada je

$$d_k = \frac{2}{n}\sum_{\alpha=0}^{n-1} f(\cos\vartheta_\alpha)\cos k\vartheta_\alpha = \frac{2}{n}\sum_{\alpha=0}^{n-1} f(x_\alpha)T_k(x_\alpha).$$

Pretpostavimo da je f' neprekidna na $[-1,1]$, osim najviše u konačno mnogo točaka, gdje ima ograničene skokove. Tada se f može razviti u konvergentan red oblika

$$f(x) = \frac{c_0}{2} + \sum_{k=1}^{\infty} c_k T_k(x), \tag{10.14.2}$$

gdje je

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\,dx = \frac{2}{\pi}\int_0^{\pi} f(\cos\vartheta)\cos k\vartheta\,d\vartheta.$$

Osim toga, postoji veza između koeficijenata u diskretnom i kontinuiranom razvoju:

$$d_0 = c_0 + 2\sum_{r=1}^{\infty}(-1)^r c_{2rn}$$
$$d_k = c_k + \sum_{r=1}^{\infty}(-1)^r c_{2rn-k} + \sum_{r=1}^{\infty}(-1)^r c_{2rn+k}, \quad k = 1,\ldots,n-1.$$

Sljedeći teorem govori o greškama koje smo napravili aproksimacijom f_n obzirom na f.

Teorem 10.14.2. *Neka je*

$$\epsilon_n(x) = f(x) - f_n(x),$$

pri čemu su f_n i f zadani s (10.14.1) i (10.14.2). Za grešku ϵ_n tada vrijedi

$$\begin{aligned}\epsilon_n(\cos\vartheta) &= \cos n\vartheta\Big(c_n + 2\sum_{r=1}^{2n-1} c_{n+r}\cos r\vartheta + c_{3n}\cos 2n\vartheta\Big)\\ &\quad - \sin 2n\vartheta\Big(c_{3n}\sin n\vartheta + 2\sum_{r=1}^{2n-1} c_{3n+r}\sin(n+r)\vartheta + c_{5n}\sin 3n\vartheta\Big)\\ &\quad + \cos 3n\vartheta\Big(c_{5n}\cos 2n\vartheta + 2\sum_{r=1}^{2n-1} c_{5n+r}\cos(2n+r)\vartheta + c_{7n}\cos 4n\vartheta\Big) - \cdots,\end{aligned}$$

odnosno, približno

$$\epsilon_n(\cos\vartheta) \approx c_n\cos n\vartheta\Big(1 + \frac{2c_{n+1}}{c_n}\cos\vartheta\Big).$$

Posebno, vrijedi

$$\epsilon_n(\cos\vartheta_\alpha) = 0.$$

Iz prethodnih teorema uočavamo da x_α leže u unutrašnjosti intervala. U mnogim je primjenama je korisno dozvoliti aproksimaciju i u rubnim točkama ± 1 i točkama koje leže u sredini među ϑ_α. Primijetite da su to ekstremi odgovarajućeg Čebiševljevog polinoma. Sada možemo napraviti sličan niz tvrdnji kao za diskretnu ortogonalnost u nultočkama.

Neka su x_α ekstremi Čebiševljevog polinoma T_n, tj. neka je

$$x_\alpha = \cos(\psi_\alpha), \quad \psi_\alpha = \frac{\alpha\pi}{n}, \quad \alpha = 0, \ldots, n.$$

Za Čebiševljeve polinome, u ekstremima vrijede sljedeće relacije ortogonalnosti

$$\begin{aligned}V_{j,k} &= \frac{1}{2}(T_j(x_0)T_k(x_0) + T_j(x_n)T_k(x_n)) + \sum_{\alpha=1}^{n-1} T_j(x_\alpha)T_k(x_\alpha)\\ &= \frac{1}{2}(\cos j\psi_0\cos k\psi_0 + \cos j\psi_n\cos k\psi_n) + \sum_{\alpha=1}^{n-1}\cos j\psi_\alpha\cos k\psi_\alpha,\end{aligned}$$

gdje je

$$V_{j,k} = \begin{cases} 0 & j,k < n,\ j \neq k,\\ n/2 & j = k,\ 0 < j < n,\\ n & j = k = 0 \text{ ili } j = k = n.\end{cases}$$

Sada možemo funkciju razviti po Čebiševljevim polinomima koristeći prethodnu relaciju diskretne ortogonalnosti. Može se pokazati da vrijedi sljedeći teorem.

Teorem 10.14.3. *Neka je $f_n(x)$ aproksimacija za $f(x)$,*

$$f_n(\cos\psi) = \frac{e_0}{2} + \sum_{k=1}^{n-1} e_k\cos k\psi + \frac{e_n}{2}\cos n\psi,$$

ili

$$f_n(x) = \frac{e_0}{2} + \sum_{k=1}^{n-1} e_k T_k(x) + \frac{e_n}{2} T_n(x). \tag{10.14.3}$$

Tada je

$$\begin{aligned} e_k &= \frac{2}{n}\Big(\frac{f(1)+(-1)^k f(-1)}{2} + \sum_{\alpha=1}^{n-1} f(\cos\psi_\alpha)\cos k\psi_\alpha\Big) \\ &= \frac{2}{n}\Big(\frac{f(1)+(-1)^k f(-1)}{2} + \sum_{\alpha=1}^{n-1} f(x_\alpha)T_k(x_\alpha)\Big). \end{aligned}$$

Osim toga, postoji veza između koeficijenata u diskretnom i kontinuiranom razvoju:

$$\begin{aligned} e_0 &= c_0 + 2\sum_{r=1}^{\infty} c_{2rn} \\ e_n &= 2c_n + 2\sum_{r=1}^{\infty} c_{(2r+1)n} \\ e_k &= c_k + \sum_{r=1}^{\infty} c_{2rn-k} + \sum_{r=1}^{\infty} c_{2rn+k}, \quad k = 1, \ldots, n-1. \end{aligned}$$

Sljedeći teorem govori o greškama koje smo napravili aproksimacijom f_n obzirom na f.

Teorem 10.14.4. *Neka je*

$$\delta_n(x) = f(x) - f_n(x),$$

pri čemu su f_n i f zadani s (10.14.3) i (10.14.2). Za grešku δ_n tada vrijedi

$$\delta_n(\cos\psi) = -2\sin n\psi \sum_{r=1}^{\infty} c_{n+r}\sin r\psi$$

odnosno, približno

$$\delta_n(\cos\psi) \approx -2\sin n\psi \sin\psi c_{n+1}\Big(1 + \frac{2c_{n+2}}{c_{n+1}}\cos\psi\Big).$$

Posebno, vrijedi

$$\delta_n(\cos\psi_\alpha) = 0.$$

Ako se c_k iz razvoja f integrira po trapeznoj formuli (vidjeti kasnije), onda se takvom aproksimacijom dobivaju koeficijenti e_k. I d_k su koeficijenti koji se dobivaju približnom integracijom c_k (modificiranom trapeznom formulom, odnosno, tzv. "midpoint" pravilom).

10.15. Thieleova racionalna interpolacija

Racionalne funkcije bolje aproksimiraju funkcije koje imaju singularitete, nego što to mogu polinomi. Jasno je da polinomi ne mogu dobro aproksimirati funkciju u okolini točke prekida, jer ih oni sami nemaju.

Prvo, definirajmo recipročne razlike, a zatim verižni razlomak koji će interpolirati funkciju f u točkama $x_1, \ldots, x_n$ (ovdje su indeksi od 1, a ne od 0).

Recipročne razlika nultog i prvog reda definiraju se redom kao

$$\rho_0(x_0) = f(x_0), \quad \rho_1(x_0, x_1) = \frac{x_0 - x_1}{f(x_0) - f(x_1)},$$

a one viših redova rekurzivno kao

$$\rho_k(x_0, \ldots, x_k) = \frac{x_0 - x_k}{\rho_{k-1}(x_0, \ldots, x_{k-1}) - \rho_{k-1}(x_1, \ldots, x_k)} + \rho_{k-2}(x_1, \ldots, x_{k-1}), \quad k \geq 2.$$

Za računanje recipročnih razlika obično se koristi tablica vrlo slična onoj za podijeljene razlike. Kao što ćemo to pokazati kasnije, algoritam koji će koristiti recipročne razlike numerirat će točke indeksima od 1 do n (zbog toga u tablici nema x_0).

x_k	$f(x_k)$	$\rho_1(x_k, x_{k+1})$	$\rho_2(x_k, x_{k+1}, x_{k+2})$	$\cdots$	$\rho_{n-1}(x_1, \ldots, x_n)$
x_1	$f(x_1)$				
		$\rho_1(x_1, x_2)$			
x_2	$f(x_2)$		$\rho_2(x_1, x_2, x_3)$		
		$\rho_1(x_1, x_2)$		$\ddots$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\rho_{n-1}(x_1, \ldots, x_n)$
		$\rho_1(x_{n-2}, x_{n-1})$		$\iddots$	
x_{n-1}	$f(x_{n-1})$		$\rho_2(x_{n-2}, x_{n-1}, x_n)$		
		$\rho_1(x_{n-1}, x_n)$			
x_n	$f(x_n)$				

Uz recipročne razlike, često se definiraju i inverzne razlike

$$\phi_0(x_0) = f(x_0), \quad \phi_1(x_0, x_1) = \frac{x_1 - x_0}{\phi_0(x_1) - \phi_0(x_0)},$$

odnosno

$$\phi_k(x_0, \ldots, x_k) = \frac{x_k - x_{k-1}}{\phi_{k-1}(x_0, \ldots, x_{k-2}, x_k) - \phi_{k-1}(x_0, \ldots, x_{k-2}, x_{k-1})}, \quad k \geq 2.$$

Postoji i veza između inverznih i recipročnih razlika. Nije teško pokazati da vrijedi

$$\phi_0(x_0) = \rho_0(x_0), \quad \phi_1(x_0, x_1) = \rho_1(x_0, x_1),$$

odnosno za $k \geq 2$

$$\phi_k(x_0, \ldots, x_k) = \rho_k(x_0, \ldots, x_k) - \rho_{k-2}(x_0, \ldots, x_{k-2}).$$

Pokažimo da vrijedi jedan važan identitet iz kojeg ćemo izvesti Thieleovu formulu. Prvo, u formuli za recipročne razlike uzmemo x_0 kao varijablu i označimo je s x. Tvrdimo da je

$$f(x) = f(x_1) + \frac{x - x_1}{\phi_1(x_1, x_2)^+} \frac{x - x_2}{\phi_2(x_1, x_2, x_3)^+} \cdots \frac{x - x_{n-1}}{\phi_{n-1}(x_1, \ldots, x_n)^+} \frac{x - x_n}{\rho_n(x, x_1, \ldots, x_n) - \rho_{n-2}(x_1, \ldots, x_{n-1})} \quad (10.15.1)\,.$$

Iz

$$\rho_1(x, x_1) = \frac{x - x_1}{f(x) - f(x_1)}$$

slijedi da je

$$f(x) = f(x_1) + \frac{x - x_1}{\rho_1(x, x_1)}. \quad (10.15.2)$$

Zatim, iz formule

$$\rho_2(x, x_1, x_2) = \frac{x - x_2}{\rho_1(x, x_1) - \rho_1(x_1, x_2)} + \rho_0(x_1)$$

slijedi da je

$$\rho_1(x, x_1) = \rho_1(x_1, x_2) + \frac{x - x_2}{\rho_2(x, x_1, x_2) - \rho_0(x_1)}.$$

Uvrštavanjem tog izraza u (10.15.2) dobivamo

$$f(x) = f(x_1) + \frac{x - x_1}{\rho_1(x_1, x_2) + \dfrac{x - x_2}{\rho_2(x, x_1, x_2) - \rho_0(x_1)}}. \quad (10.15.3)$$

Konačno, formulu (10.15.1) dobivamo indukcijom po n, uz korištenje definicije inverznih razlika.

Pokažimo još jednu zanimljivu činjenicu vezanu uz formulu (10.15.1). Ako izbrišemo zadnji član, onda će za racionalnu funkciju (verižni razlomak)

$$R(x) = f(x_1) + \frac{x - x_1}{\phi_1(x_1, x_2)^+} \frac{x - x_2}{\phi_2(x_1, x_2, x_3)^+} \cdots \frac{x - x_{n-1}}{\phi_{n-1}(x_1, \ldots, x_n)} \quad (10.15.4)$$

vrijediti

$$R(x_i) = f(x_i), \quad i = 1, \ldots, n.$$

To se odmah vidi iz (10.15.1), ako krenemo od x_n, jer je član

$$\frac{x - x_n}{\rho_n(x, x_1, \ldots, x_n) - \rho_{n-2}(x_1, \ldots, x_{n-1})} \tag{10.15.5}$$

jednak 0 za $x = x_n$, pa je $R(x_n) = f(x_n)$. Nakon toga, gledamo $R(x_{n-1})$ i $f(x_{n-1})$. Oni su za jednu verigu kraći i to za onu verigu koja sadrži "član razlike" (10.15.5). U svakoj daljnjoj točki $x_{n-2}, \ldots, x_1$, verižni je razlomak kraći za jednu verigu od prethodne.

Formula (10.15.4) zove se Thielova interpolaciona formula. Pokažimo na nekoliko primjera koliko je dobra ta interpolacija.

Primjer 10.15.1. *Aproksimirajte*

$$\operatorname{tg} 1.565$$

korištenjem Thieleove interpolacione formule, ako znamo vrijednosti funkcije tg *u točkama*

$$x_i = 1.53 + 0.01 * i, \quad i = 0, \ldots, 4.$$

Prvo izračunajmo recipročne razlike.

x_k	$f(x_k)$	ρ_1	ρ_2	ρ_3	ρ_4
1.53	24.49841				
		0.001255851			
1.54	32.46114		−0.0308670		
		0.000640314		2.96838	
1.55	48.07848		−0.0207583		3.56026
		0.000224507		2.97955	
1.56	92.62050		−0.0106889		
		0.000008597			
1.57	1255.76557				

Thielova interpolacija daje

$$R(x) = 24.49841 + \frac{x - 1.53}{0.001255851^+} \frac{x - 1.54}{-24.5293^+} \frac{x - 1.55}{2.96713^+} \frac{x - 1.56}{3.59113}.$$

Uvrštavanjem 1.565 *dobivamo*

$$R(1.565) = 172.5208,$$

dok je prava vrijednost

$$\operatorname{tg}(1.565) = 172.5211.$$

I sumacija redova može se znatno ubrzati korištenjem racionalne ekstrapolacije. Pretpostavimo da treba izračunati

$$S = \sum_{n=0}^{\infty} a_n.$$

Označimo s S_N, N-tu parcijalnu sumu reda

$$S_N = \sum_{n=0}^{N} a_n.$$

Ove vrijednosti S_N možemo interpretirati kao vrijednosti neke funkcije f u točkama N, ili u nekim drugim točkama, na primjer, u točkama $1/N$,

$$S_N = f\Big(\frac{1}{N}\Big).$$

Očito je da vrijedi

$$S = S_\infty = f(0).$$

Ideja je $f(0)$ izračunati kao ekstrapoliranu vrijednost od

$$f(1), f\Big(\frac{1}{2}\Big), f\Big(\frac{1}{3}\Big), \ldots,$$

ili za neke više N, iz

$$f\Big(\frac{1}{N_1}\Big), f\Big(\frac{1}{N_2}\Big), \ldots, \quad N_1 < N_2 < \cdots$$

Primjer 10.15.2. *Treba izračunati*

$$S = \sum_{n=1}^{\infty} \frac{1}{n^2},$$

korištenjem racionalne ekstrapolacije.

Uzet ćemo $N = 1, 2, 4, 8, 16$ *i izračunati*

$$S_N = \sum_{n=1}^{N} \frac{1}{n^2}.$$

Shvatimo li to kao funkciju od $x = 1/N$ *i označimo* $S(x) = S_N$*, onda možemo*

formirati tablicu recipročnih razlika.

x	$S(x)$	ρ_1	ρ_2	ρ_3	ρ_4
$\frac{1}{16}$	1.584346533				
		-1.097945891			
$\frac{1}{8}$	1.527422052		-0.238678243		
		-1.204112002		4.826059143	
$\frac{1}{4}$	1.423611111		-0.166126405		0.016938420
		-1.44		9.947195880	
$\frac{1}{2}$	1.25		-0.089285214		
		-2			
1	1				

Thielova interpolacija daje

$$R(x) = 1.584346533 + \frac{x - \frac{1}{16}}{-1.097945891^+} \frac{x - \frac{1}{8}}{-1.823024776^+} \frac{x - \frac{1}{4}}{5.924005034^+} \frac{x - \frac{1}{2}}{0.255616663}.$$

Uvrštavanjem 0 *dobivamo*

$$R(0) = 1.644927974,$$

dok je prava vrijednost

$$S_\infty = \frac{\pi^2}{6} = 1.644934067.$$

Zanimljivo je spomenuti što se dobije ako samo zbrajamo članove reda i ne ekstrapoliramo. Vidjet ćemo da taj red vrlo sporo konvergira. Na primjer, dobivamo

$$S_{3000} = 1.644601, \quad S_{30000} = 1.644901, \quad S_{10000} = 1.644834, \quad S_{100000} = 1.644924.$$

11. Numerička integracija

11.1. Općenito o integracionim formulama

Zadana je funkcija $f : I \to \mathbb{R}$, gdje je I obično interval (može i beskonačan). Želimo izračunati integral funkcije f na intervalu $[a, b]$,

$$I(f) = \int_a^b f(x)\,dx. \tag{11.1.1}$$

Svi znamo da je deriviranje (barem analitički) jednostavan postupak, dok integriranje to nije, pa se integrali analitički u "lijepoj formi" mogu izračunati samo za malen skup funkcija f. Zbog toga, u većini slučajeva ne možemo iskoristiti osnovni teorem integralnog računa, tj. Newton–Leibnitzovu formulu za računanje $I(f)$ preko vrijednosti primitivne funkcije F od f u rubovima intervala

$$I(f) = \int_a^b f(x)\,dx = F(b) - F(a).$$

Drugim riječima, jedino što nam preostaje je približno, numeričko računanje $I(f)$.

Osnovna ideja numeričke integracije je izračunavanje $I(f)$ korištenjem vrijednosti funkcije f na nekom konačnom skupu točaka. Recimo odmah da postoje i integracione formule koje koriste i derivacije funkcije f, ali o tome kako se one dobivaju i čemu služe, bit će više riječi nešto kasnije.

Opća integraciona formula ima oblik

$$I(f) = I_m(f) + E_m(f),$$

pri čemu je $m+1$ broj korištenih točaka, $I_m(f)$ pripadna aproksimacija integrala, a $E_m(f)$ pritom napravljena greška. Ovakve formule za približnu integraciju funkcija jedne varijable (tj. na jednodimenzionalnoj domeni) često se zovu i **kvadraturne** formule, zbog interpretacije integrala kao površine ispod krivulje.

Ako koristimo samo funkcijske vrijednosti za aproksimaciju integrala, onda aproksimacija $I_m(f)$ ima oblik

$$I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_k^{(m)}), \tag{11.1.2}$$

pri čemu je m neki unaprijed zadani prirodni broj. Koeficijenti $x_k^{(m)}$ zovu se čvorovi integracije, a $w_k^{(m)}$ težinski koeficijenti.

U općem slučaju, za fiksni m, moramo nekako odrediti $2m + 2$ nepoznatih koeficijenata. Uobičajen način njihovog određivanja je zahtjev da su integracione formule egzaktne na vektorskom prostoru **polinoma** što višeg stupnja. Zašto baš tako? Ako postoji Taylorov red za funkciju f i ako on konvergira, onda bi to značilo da integraciona formula egzaktno integrira početni komad Taylorovog reda, tj. Taylorov polinom. Drugim riječima, greška bi bila mala, tj. jednaka integralu greške koji nastaje kad iz Taylorovog reda napravimo Taylorov polinom.

Zbog linearnosti integrala kao funkcionala

$$\int (\alpha f(x) + \beta g(x))\, dx = \alpha \int f(x)\, dx + \beta \int g(x)\, dx, \tag{11.1.3}$$

dovoljno je gledati egzaktnost tih formula na nekoj bazi vektorskog prostora, recimo na

$$\{1, x, x^2, x^3, \ldots, x^m, \ldots\},$$

jer svojstvo (11.1.3) onda osigurava egzaktnost za sve polinome do najvišeg stupnja baze.

Ako su čvorovi fiksirani, recimo ekvidistantni, onda dobivano tzv. Newton–Cotesove formule, za koje moramo odrediti $m + 1$ nepoznati koeficijent (težine). Uvjeti egzaktnosti na vektorskom prostoru polinoma tada vode na sustav linearnih jednadžbi. Kasnije ćemo pokazati da se te formule mogu dobiti i kao integrali interpolacionih polinoma stupnja m za funkciju f na zadanoj (ekvidistantnoj) mreži čvorova.

S druge strane, možemo fiksirati samo neke čvorove, ili dozvoliti da su svi čvorovi "slobodni". Ove posljednje formule zovu se formule Gaussovog tipa. U slučaju Gaussovih formula (ali može se i kod težinskih Newton–Cotesovih formula) uobičajeno je (11.1.1) zapisati u obliku

$$I(f) = \int_a^b w(x) f(x)\, dx, \tag{11.1.4}$$

pri čemu je funkcija $w \geq 0$ tzv. težinska funkcija. Ona ima istu ulogu "gustoće" mjere kao i kod metode najmanjih kvadrata. Ideja je "razdvojiti" podintegralnu

funkciju na dva dijela, tako da singulariteti budu uključeni u w. Gaussove se formule nikad ne računaju "direktno" iz uvjeta egzaktnosti, jer to vodi na nelinearni sustav jednadžbi. Pokazat ćemo da postoji veza Gaussovih formula, funkcije w i ortogonalnih polinoma obzirom na funkciju w na intervalu $[a, b]$, koja omogućava efikasno računanje svih parametara za Gaussove formule.

Na kraju ovog uvoda spomenimo još da postoje primjene u kojima je korisno tražiti egzaktnost integracionih formula na drugačijim sustavima funkcija, koji nisu prostori polinoma do određenog stupnja.

11.2. Newton–Cotesove formule

Newton–Cotesove formule zatvorenog tipa imaju ekvidistantne čvorove, s tim da je prvi čvor u točki $x_0 := a$, a posljednji u $x_m := b$. Preciznije, za zatvorenu (to se često ispušta) Newton–Cotesovu formulu s $(m + 1)$-nom točkom čvorovi su

$$x_k^{(m)} = x_0 + kh_m, \quad k = 0, \ldots, m, \quad h_m = \frac{b-a}{m}.$$

Drugim riječima, osnovni je oblik Newton–Cotesovih formula

$$\int_a^b f(x)\,dx \approx I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_0 + kh_m). \tag{11.2.1}$$

11.2.1. Trapezna formula

Izvedimo najjednostavniju (zatvorenu) Newton–Cotesovu formulu za $m = 1$.

Za $m = 1$, aproksimacija integrala (11.2.1) ima oblik

$$I_1(f) = w_0^{(1)} f(x_0) + w_1^{(1)} f(x_0 + h_1),$$

pri čemu je

$$h := h_1 = \frac{b-a}{1} = b - a,$$

pa je $x_0 = a$ i $x_1 = b$. Da bismo olakšali pisanje, kad znamo da je $m = 1$, možemo izostaviti gornje indekse u $w_k^{(1)}$, tj., radi jednostavnosti, pišemo $w_k := w_k^{(1)}$. Dakle, moramo pronaći težine w_0 i w_1, tako da integraciona formula egzaktno integrira polinome što višeg stupnja na intervalu $[a, b]$, tj. da za polinome f što višeg stupnja bude

$$\int_a^b f(x)\,dx = I_1(f) = w_0 f(a) + w_1 f(b).$$

Stavimo, redom, uvjete na bazu vektorskog prostora polinoma. Ako je f neki od polinoma baze vektorskog prostora, morat ćemo izračunati njegov integral. Zbog toga je zgodno odmah izračunati integrale oblika

$$\int_a^b x^k \, dx, \quad k \geq 0,$$

a zatim rezultat koristiti za razne k. Vrijedi

$$\int_a^b x^k \, dx = \frac{x^{k+1}}{k+1} \Bigg|_a^b = \frac{b^{k+1} - a^{k+1}}{k+1}. \tag{11.2.2}$$

Za $f(x) = 1 = x^0$ dobivamo

$$b - a = \int_a^b x^0 \, dx = w_0 \cdot 1 + w_1 \cdot 1.$$

Odmah je očito da iz jedne jednadžbe ne možemo odrediti dva nepoznata parametra, pa moramo zahtjevati da integraciona formula bude egzaktna i na polinomima stupnja 1.

Za $f(x) = x$ izlazi

$$\frac{b^2 - a^2}{2} = \int_a^b x \, dx = w_0 \cdot a + w_1 \cdot b.$$

Sada imamo dvije jednadžbe s dvije nepoznanice

$$\begin{aligned} w_0 + w_1 &= b - a \\ aw_0 + bw_1 &= \frac{b^2 - a^2}{2}. \end{aligned}$$

Pomnožimo li prvu jednadžbu s $-a$ i dodamo drugoj, dobivamo

$$(b-a)w_1 = \frac{b^2 - a^2}{2} - a(b-a) = \frac{b^2 - 2ab + a^2}{2} = \frac{(b-a)^2}{2}.$$

Budući da je $a \neq b$, dijeljenjem s $b - a$, dobivamo

$$w_1 = \frac{1}{2}(b-a) = \frac{h}{2}.$$

Drugu težinu w_0 lako izračunamo iz prve jednadžbe linearnog sustava

$$w_0 = b - a - w_1 = \frac{1}{2}(b-a) = \frac{h}{2},$$

pa je $w_0 = w_1$.

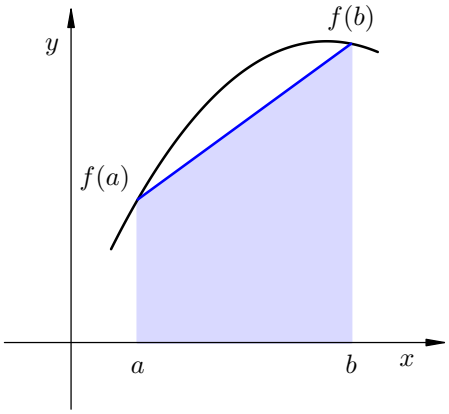
Vidimo da je integraciona formula $I_1(f)$ dobivena iz egzaktnosti na svim polinomima stupnja manjeg ili jednakog 1, i glasi

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\,(f(a) + f(b)).$$

Ta formula zove se trapezna formula. Odakle joj ime? Napišemo li je na malo drugačiji način, kao

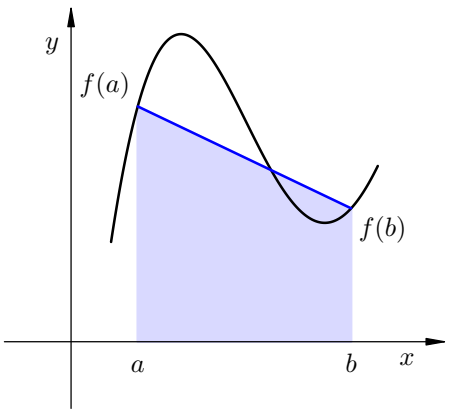
$$\int_a^b f(x)\,dx \approx \frac{f(a) + f(b)}{2}\,(b - a),$$

odmah ćemo vidjeti da je $(f(a) + f(b))/2$ srednjica, a $b - a$ visina trapeza sa slike.



Drugim riječima, površinu ispod krivulje zamijenili smo (tj. aproksimirali) površinom trapeza.

Koliko je ta zamjena dobra? Ovisi o funkciji f. Sve dok pravac razumno aproksimira oblik funkciju f, greška je mala. Na primjer, za funkciju



pravac nije dobra aproksimacija za oblik funkcije f. Da smo nacrtali funkciju f "simetričnije" oko sjecišta, moglo bi se dogoditi da je greška vrlo mala, jer bi se ono što je previše uračunato u površinu s jedne strane "skratilo" s onim što je premalo uračunato s druge strane. S numeričkog stanovišta, takav pristup je opasan.

Trapezna integraciona formula neće egzaktno integrirati sve polinome stupnja 2. To nije teško pokazati, jer već za

$$f(x) = x^2$$

vrijedi

$$\frac{b^3 - a^3}{3} = \int_a^b x^2\,dx \neq I_1(x^2) = \frac{a^2 + b^2}{2}\,(b - a).$$

Slika nas upućuje na još jednu činjenicu. Povučemo li kroz $(a, f(a))$, $(b, f(b))$ linearni interpolacioni polinom, a zatim ga egzaktno integriramo od a do b, dobivamo trapeznu formulu. Pokažimo da je to tako.

Interpolacioni polinom stupnja 1 koji prolazi kroz zadane točke je

$$p_1(x) = f(a) + f[a, b]\,(x - a).$$

Njegov integral na $[a, b]$ je

$$\begin{aligned}\int_a^b p_1(x)\,dx &= \left(f(a)x - a\,f[a, b]x + f[a, b]\,\frac{x^2}{2}\right)\Bigg|_a^b \\ &= (b - a)f(a) + \frac{(b - a)^2}{2}\,f[a, b] = (b - a)\,\frac{f(a) + f(b)}{2}.\end{aligned}$$

Ovaj nam pristup omogućava i ocjenu greške integracione formule, preko ocjene greške interpolacionog polinoma, uz uvjet da možemo ocijeniti grešku interpolacionog polinoma (tj. ako f ima dovoljan broj neprekidnih derivacija).

Neka je funkcija $f \in C^2[a, b]$. Greška interpolacionog polinoma stupnja 1 koji funkciju f interpolira u točkama $(a, f(a))$, $(b, f(b))$ na intervalu $[a, b]$ jednaka je

$$e_1(x) = f(x) - p_1(x) = \frac{f''(\xi)}{2}\,(x - a)\,(x - b).$$

Drugim riječima, vrijedi

$$E_1(f) = \int_a^b \frac{f''(\xi)}{2}\,(x - a)\,(x - b)\,dx.$$

Ostaje samo izračunati $E_1(f)$. Iskoristit ćemo generalizaciju teorema srednje vrijednosti za integrale. Ako su funkcije g i w integrabilne na $[a,b]$ i ako je $w(x) \geq 0$ na $[a,b]$, a

$$m = \inf_{x\in[a,b]} g(x), \quad M = \sup_{x\in[a,b]} g(x),$$

onda vrijedi

$$m \int_a^b w(x)\,dx \leq \int_a^b w(x)g(x)\,dx \leq M \int_a^b w(x)\,dx.$$

Prethodna formula lako se dokazuje, jer je

$$m \leq g(x) \leq M \implies mw(x) \leq g(x)w(x) \leq Mw(x),$$

pa je

$$m \int_a^b w(x)\,dx \leq \int_a^b w(x)g(x)\,dx \leq M \int_a^b w(x)\,dx. \tag{11.2.3}$$

Digresija za nematematičare. inf (čitati infimum) je minimum funkcije koji se ne mora dostići. Na primjer, funkcija

$$g(x) = x \quad \text{na} \quad (0,1) \tag{11.2.4}$$

nema minimum, ali je

$$\inf_{x\in(0,1)} x = 0.$$

Slično vrijedi i za sup (čitati supremum). Supremum je maksimum funkcije koji se ne mora dostići. Na primjer, funkcija iz relacije (11.2.4) nema ni maksimum, ali je

$$\sup_{x\in(0,1)} x = 1.$$

■

Korištenjem relacije (11.2.3), lako dokazujemo integralni teorem srednje vrijednosti s težinama.

Teorem 11.2.1. *Neka su funkcije g i w integrabilne na $[a,b]$ i neka je*

$$m = \inf_{x\in[a,b]} g(x), \quad M = \sup_{x\in[a,b]} g(x).$$

Nadalje, neka je $w(x) \geq 0$ na $[a,b]$. Tada postoji broj μ, $m \leq \mu \leq M$ takav da vrijedi

$$\int_a^b w(x)g(x)\,dx = \mu \int_a^b w(x)\,dx.$$

Posebno, ako je g neprekidna na $[a,b]$, onda postoji broj ζ takav da je

$$\int_a^b w(x)g(x)\,dx = g(\zeta)\int_a^b w(x)\,dx.$$

Dokaz:

Ako je

$$\int_a^b w(x)\,dx = 0,$$

onda je po (11.2.3) i

$$\int_a^b w(x)g(x)\,dx = 0,$$

pa za μ možemo uzeti proizvoljan realan broj. Ako je

$$\int_a^b w(x)\,dx > 0,$$

onda dijeljenjem formule (11.2.3) s prethodnim integralom dobivamo

$$m \le \frac{\int_a^b w(x)g(x)\,dx}{\int_a^b w(x)\,dx} \le M,$$

pa za μ možemo uzeti

$$\mu = \frac{\int_a^b w(x)g(x)\,dx}{\int_a^b w(x)\,dx}.$$

Posljednji zaključak teorema slijedi iz činjenice da neprekidna funkcija na segmentu postiže sve vrijednosti između minimuma i maksimuma, pa mora postići i μ. Drugim riječima, postoji ζ takav da je $\mu = g(\zeta)$. ■

Prisjetite se, već smo pokazali da je

$$E_1(f) = \int_a^b \frac{f''(\xi)}{2}\,(x-a)\,(x-b)\,dx.$$

Primijetite da je funkcija

$$\frac{(x-a)\,(x-b)}{2} \le 0 \quad \text{na} \quad [a,b],$$

pa možemo uzeti

$$w(x) = -\frac{(x-a)\,(x-b)}{2}, \quad g(x) = -f''(\xi).$$

Po generaliziranom teoremu srednje vrijednosti, ako je $f \in C^2[a,b]$, (što znači da je $f'' \in C^0[a,b]$), vrijedi da je

$$E_1(f) = -f''(\zeta) \int_a^b -\frac{(x-a)\,(x-b)}{2}\,dx.$$

Ovaj se integral jednostavno računa. Integriranjem dobivamo

$$\int_a^b \frac{(x-a)\,(x-b)}{2}\,dx = -\frac{(b-a)^3}{12} = -\frac{h^3}{12},$$

pa je

$$E_1(f) = -f''(\zeta)\frac{h^3}{12}.$$

11.2.2. Simpsonova formula

Izvedimo sljedeću (zatvorenu) Newton–Cotesovu formulu za $m = 2$, poznatu pod imenom Simpsonova formula.

Za $m = 2$, aproksimacija integrala (11.2.1) ima oblik

$$I_2(f) = w_0^{(2)} f(x_0) + w_1^{(2)} f(x_0 + h_2) + w_2^{(2)} f(x_0 + 2h_2),$$

pri čemu je

$$h := h_2 = \frac{b-a}{2}.$$

Ponovno, da bismo olakšali pisanje, kad znamo da je $m = 2$, možemo, radi jednostavnosti, izostaviti gornje indekse u $w_k := w_k^{(2)}$. Oprez, to nisu isti w_k i h kao u trapeznoj formuli! Kad uvrstimo značenje h u aproksimacionu formulu, dobivamo

$$I_2(f) = w_0 f(a) + w_1\, f\Big(\frac{a+b}{2}\Big) + w_2 f(b).$$

Stavimo uvjete na egzaktnost formule na vektorskom prostoru polinoma što višeg stupnja. Moramo postaviti najmanje tri jednadžbe, jer imamo tri nepoznata koeficijenta. Za $f(x) = 1$ dobivamo

$$b - a = \int_a^b x^0\,dx = w_0 \cdot 1 + w_1 \cdot 1 + w_2 \cdot 1.$$

Za $f(x) = x$ izlazi

$$\frac{b^2 - a^2}{2} = \int_a^b x\,dx = w_0 \cdot a + w_1 \frac{a+b}{2} + w_2 \cdot b.$$

Konačno, za $f(x) = x^2$ dobivamo

$$\frac{b^3 - a^3}{3} = \int_a^b x^2\,dx = w_0 \cdot a^2 + w_1 \frac{(a+b)^2}{4} + w_2 \cdot b^2.$$

Sada imamo linearni sustav s tri jednadžbe i tri nepoznanice

$$\begin{aligned} w_0 + w_1 + w_2 &= b - a \\ aw_0 + \frac{a+b}{2} w_1 + bw_2 &= \frac{b^2 - a^2}{2} \\ a^2 w_0 + \frac{(a+b)^2}{4} w_1 + b^2 w_2 &= \frac{b^3 - a^3}{3}. \end{aligned}$$

Rješavanjem ovog sustava, dobivamo

$$w_0 = w_2 = \frac{h}{3} = \frac{b-a}{6}, \quad w_1 = \frac{4h}{3} = \frac{4(b-a)}{6}.$$

Drugim riječima, integraciona formula $I_2(f)$ dobivena je iz egzaktnosti na svim polinomima stupnja manjeg ili jednakog 2, i glasi

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\Big(f(a) + 4f\Big(\frac{a+b}{2}\Big) + f(b)\Big).$$

Simpsonova formula ima još jednu prednost. Iako je dobivena iz uvjeta egzaktnosti na vektorskom prostoru polinoma stupnja manjeg ili jednakog 2, ona egzaktno integrira i sve polinome stupnja 3. Dovoljno je pokazati da egzaktno integrira

$$f(x) = x^3.$$

Egzaktni integral jednak je

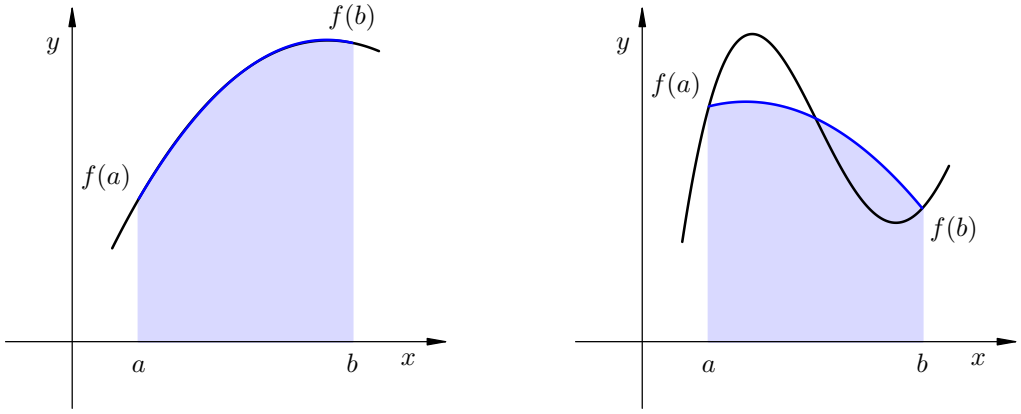
$$\int_a^b x^3\,dx = \frac{b^4 - a^4}{4},$$

a po Simpsonovoj formuli, za $f(x) = x^3$ dobivamo

$$\begin{aligned} I_2(x^3) &= \frac{b-a}{6}\Big(a^3 + 4\Big(\frac{a+b}{2}\Big)^3 + b^3\Big) \\ &= \frac{b-a}{4}\,(a^3 + a^2 b + ab^2 + b^3) = \frac{b^4 - a^4}{4}. \end{aligned}$$

Ponovno, nije teško pokazati da je i ova formula interpolaciona. Ako povučemo kvadratni interpolacioni polinom kroz $(a, f(a))$, $(\frac{a+b}{2}, f(\frac{a+b}{2}))$ i $(b, f(b))$, a zatim ga egzaktno integriramo od a do b, dobivamo Simpsonovu formulu.

Ako pogledamo kako ona funkcionira na funkcijama koje smo već integrirali trapeznom formulom, vidjet ćemo da joj je greška bitno manja. Posebno, na prvom primjeru, kvadratni interpolacioni polinom tako dobro aproksimira funkciju f, da se one na grafu ne razlikuju.



Grešku Simpsonove formule računamo slično kao kod trapezne, integracijom greške kvadratnog interpolacionog polinoma

$$e_2(x) = f(x) - p_2(x) = \frac{f'''(\xi)}{6}\,(x-a)\Big(x - \frac{a+b}{2}\Big)(x-b).$$

Dakle, za grešku Simpsonove formule vrijedi

$$E_2(f) = \int_a^b e_2(x)\,dx.$$

Nažalost, funkcija

$$(x-a)\Big(x - \frac{a+b}{2}\Big)(x-b)$$

nije više fiksnog znaka na $[a,b]$, pa ne možemo direktno primijeniti generalizirani teorem srednje vrijednosti. Pretpostavimo da je $f \in C^4[a,b]$. Označimo

$$c := \frac{a+b}{2}$$

i definiramo

$$w(x) = \int_a^x (t-a)\,(t-c)\,(t-b)\,dt.$$

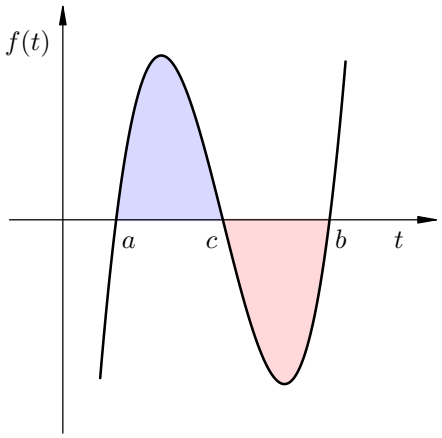
Tvrdimo da vrijedi

$$w(a) = w(b) = 0, \quad w(x) > 0, \quad x \in (a, b). \tag{11.2.5}$$

Skiciramo li funkciju

$$f(t) = (t - a)(t - c)(t - b)$$

odmah vidimo da je ona centralno simetrična oko srednje točke



pa će integral rasti od 0 do svog maksimuma (plava površina), a zatim padati (kad dođe u crveno područje) do 0.

Ostaje samo još napisati grešku interpolacionog polinoma kao podijeljenu razliku. To smo pokazali općenito u poglavlju o Newtonovom interpolacionom polinomu, a posebno za $n = 3$ vrijedi

$$f[a, b, c, x] = \frac{f'''(\xi)}{6}.$$

Uz oznaku (11.2.5), grešku Simpsonove formule, onda možemo napisati kao

$$E_2(f) = \int_a^b w'(x) f[a, b, c, x]\, dx.$$

Parcijalnom integracijom ovog integrala dobivamo

$$E_2(f) = w(x) f[a, b, c, x] \Big|_a^b - \int_a^b w(x) \frac{d}{dx} f[a, b, c, x]\, dx.$$

Prvi član je očito jednak 0, jer je $w(a) = w(b) = 0$. Ostaje još "srediti" drugi član. Kod splajnova smo objašnjavali da je podijeljena razlika s dvostrukim čvorom jednaka derivaciji funkcije. Na sličan je način derivacija treće podijeljene razlike

$f[a,b,c,x]$ po x, četvrta podijeljena razlika s dvostrukim čvorom x. Prema tome, dobivamo formulu greške u obliku

$$E_2(f) = -\int_a^b w(x) f[a,b,c,x,x]\,dx.$$

Sad je funkcija w nenegativna i možemo primijeniti generalizirani teorem srednje vrijednosti. Izlazi

$$E_2(f) = -f[a,b,c,\eta,\eta]\int_a^b w(x)\,dx,$$

gdje je $a \le \eta \le b$. Napišemo li $f[a,b,c,\eta,\eta]$ kao derivaciju, dobivamo

$$E_2(f) = -\frac{f^{(4)}(\zeta)}{4!}\int_a^b w(x)\,dx.$$

Ostaje još samo integrirati funkciju w. Vrijedi

$$\begin{aligned} w(x) &= \int_a^x (t-a)\,(t-c)\,(t-b)\,dt = \text{zamjena varijable } y = t-c \\ &= \int_{-h}^{x-c} (y-h)y(y+h)\,dy = \int_{-h}^{x-c} (y^3 - h^2 y)\,dy \\ &= \left(\frac{y^4}{4} - h^2\frac{y^2}{2}\right)\Bigg|_{-h}^{x-c} = \frac{(x-c)^4}{4} - h^2\frac{(x-c)^2}{2} + \frac{h^4}{4}. \end{aligned}$$

Nadalje je

$$\begin{aligned} \int_a^b w(x)\,dx &= \int_a^b \left(\frac{(x-c)^4}{4} - h^2\frac{(x-c)^2}{2} + \frac{h^4}{4}\right)dx = \text{zamjena varijable } y = x-c \\ &= \int_{-h}^{h} \left(\frac{y^4}{4} - h^2\frac{y^2}{2} + \frac{h^4}{4}\right)dy = \left(\frac{y^5}{20} - h^2\frac{y^3}{6} + \frac{h^4 y}{4}\right)\Bigg|_{-h}^{h} \\ &= 2\left(\frac{h^5}{20} - \frac{h^5}{6} + \frac{h^5}{4}\right) = \frac{4}{15}\,h^5. \end{aligned}$$

Kad to uključimo u formulu za grešku, dobivamo

$$E_2(f) = -\frac{f^{(4)}(\zeta)}{24}\cdot\frac{4}{15}\,h^5 = -\frac{h^5}{90}f^{(4)}(\zeta).$$

Primijetite, greška je za red veličine bolja no što bi po upotrijebljenom interpolacionom polinomu trebala biti.

11.2.3. Produljene formule

Nije teško pokazati da su sve Newton–Cotesove formule integrali interpolacionih polinoma na ekvidistantnoj mreži. Ako ne valja dizanje stupnjeva interpolacionih polinoma na ekvidistantnoj mreži, onda neće biti dobri niti njihovi integrali.

Pokažimo to na primjeru Runge. Prava vrijednost integrala je

$$\int_{-5}^{5} \frac{dx}{1+x^2} = 2\,\mathrm{arctg}\,5 \approx 2.74680153389003172.$$

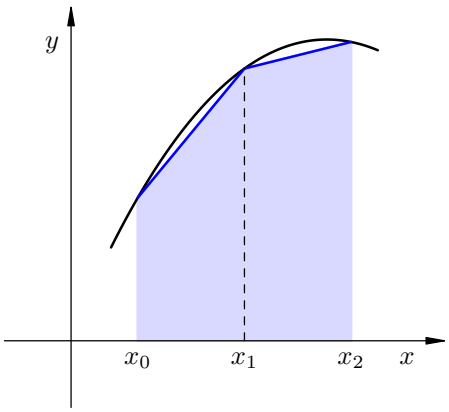
Sljedeća tablica pokazuje aproksimacije integrala izračunate Newton–Cotesovim formulama raznih redova i pripadne greške.

Red formule m	Aproksimacija integrala	Greška
1	0.38461538461538462	2.36218614927464711
2	6.79487179487179487	−4.04807026098176315
3	2.08144796380090498	0.66535357008912674
4	2.37400530503978780	0.37279622885024392
5	2.30769230769230769	0.43910922619772403
6	3.87044867347079978	−1.12364713958076805
7	2.89899440974837875	−0.15219287585834703
8	1.50048890712791179	1.24631262676211993
9	2.39861789784183472	0.34818363604819700
10	4.67330055565349876	−1.92649902176346704
11	3.24477294027858525	−0.49797140638855353
12	−0.31293651575343889	3.05973804964347061
13	1.91979721683238891	0.82700431705764282
14	7.89954464085193082	−5.15274310696189909
15	4.15555899270655713	−1.40875745881652541
16	−6.24143731477308329	8.98823884866311501
17	0.26050944143760372	2.48629209245242800
18	18.87662129010920670	−16.12981975621917490
19	7.24602608588196936	−4.49922455199193763
20	−26.84955208882447960	29.59635362271451140

Očito je da aproksimacije **ne** konvergiraju prema pravoj vrijednosti integrala. Potpunije opravdanje ovog ponašanja dajemo nešto kasnije.

I što sad? Ne smijemo dizati red formula, jer to postaje opasno. Rješenje je vrlo slično onome što smo primijenili kod interpolacije. Umjesto da dižemo red

formule, podijelimo interval $[a, b]$ na više dijelova, recimo, jednake duljine, i na svakom od njih primijenimo odgovarajuću integracionu formulu niskog reda. Tako dobivene formule zovu se **produljene** formule. Na primjer, za funkciju koju smo već razmatrali, produljena trapezna formula s 2 podintervala izgledala bi ovako.



Općenito, produljenu trapeznu formulu dobivamo tako da cijeli interval $[a, b]$ podijelimo na n podintervala oblika $[x_{k-1}, x_k]$, za $k = 1, \ldots, n$, s tim da je

$$a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b,$$

i na svakom od njih upotrijebimo "običnu" trapeznu formulu. Znamo da je tada

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} f(x)\,dx,$$

pa na isti način zbrojimo i "obične" trapezne aproksimacije u produljenu trapeznu aproksimaciju.

Najjednostavniji je slučaj kad su točke x_k ekvidistantne, tj. kad je svaki podinterval $[x_{k-1}, x_k]$ iste duljine h. To znači da je

$$x_k = a + kh, \quad k = 0, \ldots, n, \quad h = \frac{b-a}{n}.$$

Aproksimacija produljenom trapeznom formulom je

$$\int_a^b f(x)\,dx = h\Big(\frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n\Big) + E_n^T(f),$$

pri čemu je $E_n^T(f)$ greška produljene formule. Nju možemo zapisati kao zbroj grešaka osnovnih trapeznih formula na podintervalima

$$E_n^T(f) = \sum_{k=1}^{n} -f''(\zeta_k)\frac{h^3}{12}.$$

Greška ovako napisana nije naročito lijepa i korisna, pa ju je potrebno napisati malo drugačije

$$E_n^T(f) = -\frac{h^3 n}{12}\Big(\frac{1}{n}\sum_{k=1}^{n} f''(\zeta_k)\Big).$$

Izraz u zagradi je aritmetička sredina vrijednosti drugih derivacija u točkama ζ_k. Taj se broj sigurno nalazi između najmanje i najveće vrijednosti druge derivacije funkcije f na intervalu $[a,b]$. Budući da je f'' neprekidna na $[a,b]$, onda je broj u zagradi vrijednost druge derivacije u nekoj točki $\xi \in [a,b]$, pa formulu za grešku možemo pisati kao

$$E_n^T(f) = -\frac{h^3 n}{12} f''(\xi) = -\frac{(b-a)h^2}{12} f''(\xi).$$

Iz ove formule izvodimo važnu ocjenu za broj podintervala potrebnih da se postigne zadana točnost za produljenu trapeznu metodu

$$|E_n^T(f)| \le \frac{(b-a)h^2}{12} M_2 = \frac{(b-a)^3}{12n^2} M_2, \quad M_2 = \max_{x\in[a,b]} |f''(x)|.$$

Želimo li da je $|E_n^T(f)| \le \varepsilon$, onda je dovoljno tražiti da bude

$$\frac{(b-a)^3}{12n^2} M_2 \le \varepsilon,$$

odnosno da je

$$n \ge \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}}, \quad n \text{ cijeli broj.}$$

Na sličan se način izvodi i produljena Simpsonova formula. Primijetite, osnovna Simpsonova formula ima 3 točke, tj. 2 podintervala, pa produljena formula mora imati, također, paran broj podintervala. Pretpostavimo stoga da je n paran broj. Ograničimo se samo na ekvidistantni slučaj. Onda je ponovno

$$h = \frac{b-a}{n}, \quad x_k = a + kh, \quad k = 0, \ldots, n.$$

Aproksimaciju integrala produljenom Simpsonovom formulom dobivamo iz

$$\int_a^b f(x)\,dx = \sum_{k=1}^{n/2} \int_{x_{2k-2}}^{x_{2k}} f(x)\,dx,$$

tako da na svakom podintervalu $[x_{2k-2}, x_{2k}]$, duljine $2h$, primijenimo običnu Simpsonovu formulu, za $k = 1, \ldots, n/2$. Zbrajanjem izlazi

$$\int_a^b f(x)\,dx = \frac{h}{3}\Big(f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 4f_{n-1} + f_n\Big) + E_n^S(f),$$

pri čemu je $E_n^S(f)$ greška produljene formule. Nju možemo zapisati kao zbroj grešaka osnovnih Simpsonovih formula na podintervalima

$$E_n^S(f) = \sum_{k=1}^{n/2} -f^{(4)}(\zeta_k)\frac{h^5}{90}.$$

Opet je grešku korisno napisati malo drugačije

$$E_n^S(f) = -\frac{h^5(n/2)}{90}\Big(\frac{2}{n}\sum_{k=1}^{n/2} f^{(4)}(\zeta_k)\Big).$$

Sličnim zaključivanjem kao kod trapezne formule, izraz u zagradi možemo zamijeniti s $f^{(4)}(\xi)$, $\xi \in [a,b]$, pa dobivamo

$$E_n^S(f) = -\frac{h^5 n}{180}f^{(4)}(\xi) = -\frac{(b-a)h^4}{180}f^{(4)}(\xi).$$

Ponovno, iz ove formule izvodimo ocjenu za broj podintervala potrebnih da se postigne zadana točnost za Simpsonovu metodu

$$|E_n^S(f)| \le \frac{(b-a)h^4}{180}M_4 = \frac{(b-a)^5}{180n^4}M_4, \quad M_4 = \max_{x\in[a,b]}|f^{(4)}(x)|.$$

Želimo li da je $|E_n^S(f)| \le \varepsilon$, onda je dovoljno tražiti da bude

$$\frac{(b-a)^5}{180n^4}M_4 \le \varepsilon,$$

odnosno da je

$$n \ge \sqrt[4]{\frac{(b-a)^5 M_4}{180\varepsilon}}, \quad n \text{ paran cijeli broj.}$$

11.2.4. Primjeri

Primjer 11.2.1. *Izračunajte vrijednost integrala*

$$\int_1^2 xe^{-x}\,dx$$

korištenjem (produljene) Simpsonove formule tako da greška bude manja ili jednaka 10^{-6}. *Nađite pravu vrijednost integrala i pogreške. Koliko je podintervala potrebno za istu točnost korištenjem (produljene) trapezne formule?*

Prmo, moramo ocijeniti pogrešku za produljenu trapeznu i produljenu Simpsonovu formulu. Za to su nam potrebni maksimumi apsolutnih vrijednosti druge i četvrte derivacije na zadanom intervalu. Derivacije su redom

$$f^{(1)}(x) = (1-x)e^{-x}, \quad f^{(2)}(x) = (x-2)e^{-x}, \quad f^{(3)}(x) = (3-x)e^{-x},$$
$$f^{(4)}(x) = (x-4)e^{-x}, \quad f^{(5)}(x) = (5-x)e^{-x}.$$

Nađimo maksimume apsolutnih vrijednosti derivacija na zadanom intervalu.

Prvo ocijenimo grešku za produljenu trapeznu formulu. Na intervalu $[1,2]$ *je* $f^{(3)}(x) > 0$*, što znači da* $f^{(2)}$ *raste. Uočimo još da je na zadanom intervalu* $f^{(2)}(x) \leq 0$*, pa je maksimum apsolutne vrijednosti druge derivacije u lijevom rubu, tj.*

$$M_2 = \max_{x\in[1,2]} |f^{(2)}(x)| = |f^{(2)}(1)| = e^{-1} \approx 0.367879441171.$$

Broj podintervala n_T *za produljenu trapeznu formulu je*

$$n_T \geq \sqrt{\frac{(b-a)^3 M_2}{12\varepsilon}} = \sqrt{\frac{e^{-1}}{12\cdot 10^{-6}}} \approx 175.09,$$

pa je najmanji broj podintervala $n_T = 176$*.*

Sada ocijenimo grešku za produljenu Simpsonovu formulu. Na intervalu $[1,2]$ *je* $f^{(5)}(x) > 0$*, što znači da* $f^{(4)}$ *raste. Također je i* $f^{(4)}(x) < 0$*, što znači da je njen maksimum po apsolutnoj vrijednosti ponovno u lijevom rubu, tj.*

$$M_4 = \max_{x\in[1,2]} |f^{(4)}(x)| = |f^{(4)}(1)| = 3\cdot e^{-1} \approx 1.103638323514.$$

Za grešku produljene Simpsonove formule imamo

$$n_S \geq \sqrt[4]{\frac{(b-a)^5 M_4}{180\varepsilon}} = \sqrt[4]{\frac{3\cdot e^{-1}}{180\cdot 10^{-6}}} \approx 8.85,$$

tj. treba najmanje $n_S = 10$ *podintervala.*

Sad možemo upotrijebiti produljenu Simpsonovu formulu s 10 *podintervala (*11

čvorova). Imamo

k	x_k	$f(x_k)$
0	1.0	0.3678794412
1	1.1	0.3661581921
2	1.2	0.3614330543
3	1.3	0.3542913309
4	1.4	0.3452357495
5	1.5	0.3346952402
6	1.6	0.3230344288
7	1.7	0.3105619909
8	1.8	0.2975379988
9	1.9	0.2841803765
10	2.0	0.2706705665

Sada je

$$\begin{aligned}
&S_0 = f(x_0) + f(x_{10}) = 0.63855000765,\\
&S_1 = 4(f(x_1) + f(x_3) = f(x_5) + f(x_7) + f(x_9)) = 6.5995485226,\\
&S_2 = 2(f(x_2) + f(x_4) + f(x_6) + f(x_8)) = 2.6544824628.
\end{aligned}$$

Vrijednost integrala po Simpsonovoj formuli je

$$I_s = \frac{0.1}{3}(S_0 + S_1 + S_2) = 0.3297526998.$$

U ovom konkretnom slučaju možemo bez puno napora izračunati i egzaktnu vrijednost integrala. Jedina korist od toga je da vidimo koliko je zaista ocjena za Simpsonovu metodu bliska sa stvarnom greškom. Parcijalna integracija daje

$$\begin{aligned}
\int_1^2 xe^{-x}\,dx &= \left\{ \begin{array}{ll} u = x, & du = dx \\ dv = e^{-x}\,dx, & v = -e^{-x} \end{array} \right\} = -xe^{-x}\Big|_1^2 + \int_1^2 e^{-x}\,dx \\
&= e^{-1} - 2e^{-2} - e^{-x}\Big|_1^2 = e^{-1} - 2e^{-2} - e^{-2} + e^{-1} \\
&= 2e^{-1} - 3e^{-2} \approx 0.3297530326.
\end{aligned}$$

Drugim riječima, prava pogreška je

$$I - I_S = 0.3297530326 - 0.3297526998 = 3.328 \cdot 10^{-7},$$

tj. ocjena greške nije daleko od prave pogreške.

11.2.5. Midpoint formula

Ako u Newton–Cotesovim formulama ne interpoliramo (pa onda niti ne integriramo) jednu ili obje rubne točke, dobili smo otvorene Newton–Cotesove formule. Ako definiramo $x_{-1} := a$, $x_{m+1} := b$ i

$$h_m = \frac{b-a}{m+2},$$

onda otvorene Newton–Cotesove formule imaju oblik

$$\int_a^b f(x)\,dx \approx I_m(f) = \sum_{k=0}^{m} w_k^{(m)} f(x_0 + kh_m). \tag{11.2.6}$$

Vjerojatno najkorištenija i najpoznatija otvorena Newton–Cotesova formula je ona najjednostavnija za $m = 0$, poznata pod imenom "midpoint formula" (formula srednje točke).

Dakle za bismo odredili midpoint formulu, moramo naći koeficijent $w_0 := w_0^{(0)}$ takav da je

$$\int_a^b f(x)\,dx = w_0 f\Big(\frac{a+b}{2}\Big)$$

egzaktna na vektorskom prostoru polinoma što višeg stupnja.

Za $f(x) = 1$, imamo

$$b - a = \int_a^b 1\,dx = w_0,$$

odakle odmah slijedi da je

$$\int_a^b f(x)\,dx = (b-a) f\Big(\frac{a+b}{2}\Big).$$

Greška te integracione formule je integral greške interpolacionog polinoma stupnja 0 (konstante), koji interpolira funkciju f u srednjoj točki. Ako definiramo

$$w(x) = \int_a^x (t-c)\,dt, \quad c := \frac{a+b}{2},$$

onda koristeći istu tehniku kao kod izvoda greške za Simpsonovu formulu, izlazi da je greška midpoint formule

$$E_0(f) = \int_a^b e_0(x)\,dx = f''(\xi)\frac{(b-a)^3}{24}.$$

Da bismo izveli produljenu formulu, podijelimo interval $[a, b]$ na n podintervala i na svakom upotrijebimo midpoint formulu. Tada vrijedi

$$I_n(f) = h(f_1 + \cdots + f_n) + E_n^M(f), \quad h = \frac{b-a}{n}, \quad x_k = a + \Big(k - \frac{1}{2}\Big)h,$$

pri čemu je $E_n^M(f)$ ukupna greška koja je jednaka

$$E_n^M(f) = \sum_{k=1}^{n} f''(\xi_k)\frac{h^3}{24} = \frac{h^3 n}{24}\Big(\frac{1}{n}\sum_{k=1}^{n} f''(\xi_k)\Big) = \frac{h^3 n}{24} f''(\xi) = \frac{h^2(b-a)}{24} f''(\xi).$$

11.3. Rombergov algoritam

Pri izvodu Rombergovog algoritma koristimo se sljedećim principima:

- udvostručavanjem broja podintervala u produljenoj trapeznoj metodi,
- eliminacijom člana greške iz dvije susjedne produljene formule. Ponovljena primjena ovog principa zove se Richardsonova ekstrapolacija.

Asimptotski razvoj ocjene pogreške za trapeznu integraciju daje Euler–Mac-Laurinova formula.

Teorem 11.3.1. *Neka je $m \geq 0$, $n \geq 1$, m, n cijeli brojevi. Definiramo ekvidistantnu mrežu s n podintervala na $[a, b]$, tj.*

$$h = \frac{b-a}{n}, \quad x_k = a + kh, \quad k = 0, \ldots, n.$$

Pretpostavimo da je $f \in C^{(2m+2)}[a, b]$. Za pogrešku produljene trapezne metode vrijedi

$$E_n(f) = \int_a^b f(x)\,dx - I_n^T(f) = \sum_{i=1}^{m} \frac{d_{2i}}{n^{2i}} + F_{n,m},$$

gdje su koeficijenti

$$d_{2i} = -\frac{B_{2i}}{(2i)!}\,(b-a)^{2i}\,(f^{(2i-1)}(b) - f^{(2i-1)}(a)),$$

a ostatak je

$$F_{n,m} = \frac{(b-a)^{2m+2}}{(2m+2)!n^{2m+2}} \cdot \int_a^b \overline{B}_{2m+2}\Big(\frac{x-a}{h}\Big)\, f^{(2m+2)}(x)\,dx.$$

Ovdje su B_{2i} Bernoullijevi brojevi,

$$B_i = -\int_0^1 B_i(x)\,dx, \quad i \geq 1,$$

a $\overline{B}_i$ je periodičko proširenje običnih Bernoullijevih polinoma

$$\overline{B}_i(x) = \begin{cases} B_i(x), & za\ 0 \leq x \leq 1, \\ \overline{B}_i(x-1), & za\ x \geq 1. \end{cases}$$

Ovo je jedan od klasičnih teorema numeričke analize i njegov se dokaz može naći u mnogim knjigama.

Umjesto dokaza, nekoliko objašnjenja. Bernoullijevi polinomi zadani su implicitno funkcijom izvodnicom

$$\frac{t(e^{xt}-1)}{e^t-1} = \sum_{i=0}^{\infty} B_i(x)\frac{t^i}{i!}.$$

Prvih nekoliko Bernullijevih polinoma su:

$$B_0(x) = 0 \qquad B_1(x) = x \qquad B_2(x) = x^2 - x$$
$$B_3(x) = x^3 - \frac{3x^2}{2} + \frac{x}{2} \qquad B_4(x) = x^2(1-x)^2.$$

Uvijek vrijedi $B_i(0) = 0$ za $i \geq 0$. Rekurzivne relacije su

$$B_i'(x) = \begin{cases} iB_{i-1}(x), & \text{za } i \text{ paran i } i \geq 4, \\ i(B_{i-1}(x) + B_{i-1}), & \text{za } i \text{ neparan i } i \geq 3. \end{cases}$$

Iz prethodne se formule integracijom mogu dobiti $B_i(x)$, jer je slobodni član jednak 0.

Bernoullijevi brojevi također su definirani implicitno

$$\frac{t}{e^t-1} = \sum_{i=0}^{\infty} B_i\,\frac{t^i}{i!},$$

odakle se integracijom na $[0,1]$ po x u rekurziji za $B_i(x)$ dobiva

$$B_i = -\int_0^1 B_i(x)\,dx, \quad i \geq 1.$$

Prvih nekoliko Bernoullijevih brojeva:

$$B_0 = 1, \qquad B_1 = -\frac{1}{2}, \quad B_2 = \frac{1}{6}, \qquad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42},$$
$$B_8 = -\frac{1}{30}, \quad B_{10} = \frac{5}{66}, \quad B_{12} = -\frac{691}{2730}, \quad B_{14} = \frac{7}{6}, \qquad B_{16} = -\frac{3617}{510}$$

i dalje vrlo brzo rastu po apsolutnoj vrijednosti

$$B_{60} \approx -2.139994926 \cdot 10^{34}.$$

Napomena 11.3.1. *U literaturi se može naći i malo drugačija definicija Bernoullijevih polinoma, označimo ih s $B_i^*(x)$. Oni su zadani implicitno funkcijom izvodnicom*

$$\frac{te^{xt}}{e^t-1} = \sum_{i=0}^{\infty} B_i^*(x)\frac{t^i}{i!}.$$

Veza između jednih i drugih Bernoullijevih polinoma je $B_i^*(x) = B_i(x)+B_i$, za $i \geq 0$.

Rombergov algoritam dobivamo tako da eliminiramo član po član iz reda za ocjenu greške na osnovu vrijednosti integrala s duljinom koraka h i $h/2$.

Za podintegralne funkcije koje nisu dovoljno glatke, također, se može (uz blage pretpostavke) asimptotski dobiti razvoj pogreške. Posebno to vrijedi za funkcije s algebarskim (x^α) i/ili logaritamskim ($\ln x$) singularitetima.

Izvedimo sad Rombergov algoritam. Označimo s $I_n^{(0)}$ trapeznu formulu s duljinom intervala $h = (b-a)/n$. Iz Euler–MacLaurinove formule, ako je n paran, za asimptotski razvoj greške imamo

$$\begin{aligned} I - I_n^{(0)} &= \frac{d_2^{(0)}}{n^2} + \frac{d_4^{(0)}}{n^4} + \cdots + \frac{d_{2m}^{(0)}}{n^{2m}} + F_{n,m} \\ I - I_{n/2}^{(0)} &= \frac{4d_2^{(0)}}{n^2} + \frac{16d_4^{(0)}}{n^4} + \cdots + \frac{2^{2m}d_{2m}^{(0)}}{n^{2m}} + F_{n/2,m}. \end{aligned}$$

Ako prvi razvoj pomnožimo s 4 i oduzmemo mu drugi razvoj, skratit će se prva greška s desne strane $d_2^{(0)}$, tj. dobit ćemo

$$4(I - I_n^{(0)}) - (I - I_{n/2}^{(0)}) = -\frac{12d_4^{(0)}}{n^4} - \frac{60d_6^{(0)}}{n^6} + \cdots.$$

Izlučivanjem članova koji imaju I na lijevu stranu, a zatim dijeljenjem, dobivamo

$$I = \frac{4I_n^{(0)} - I_{n/2}^{(0)}}{3} - \frac{4d_4^{(0)}}{n^4} - \frac{20d_6^{(0)}}{n^6} + \cdots.$$

Prvi član zdesna možemo uzeti kao bolju, popravljenu aproksimaciju integrala, u oznaci

$$I_n^{(1)} = \frac{4I_n^{(0)} - I_{n/2}^{(0)}}{3}, \quad n \text{ paran}, n \geq 2.$$

Niz $I_n^{(2)}$, $I_n^{(4)}$, $I_n^{(6)}$ je novi integracijski niz. Njegova je greška

$$I - I_n^{(1)} = \frac{d_4^{(1)}}{n^4} + \frac{d_6^{(1)}}{n^6} + \cdots,$$

gdje je

$$d_4^{(1)} = -4d_4^{(0)}, \quad d_6^{(1)} = -20d_6^{(0)}.$$

Nađimo eksplicitnu formulu za $I_n^{(1)}$. Zbog podjele na odgovarajući broj podintervala, ako je h duljina podintervala za $I_n^{(0)}$, onda je $h_1 := 2h$ duljina podintervala za $I_{n/2}^{(0)}$, pa vrijede sljedeće formule

$$I_n^{(0)} = \frac{h}{2}(f_0 + 2f_1 + \cdots 2f_{n-1} + f_n)$$
$$I_{n/2}^{(0)} = \frac{h_1}{2}(f_0 + 2f_2 + \cdots 2f_{n-2} + f_n).$$

Uvrštavanjem u $I_n^{(1)}$, dobivamo

$$\begin{aligned} I_n^{(1)} &= \frac{4h}{3}\Big(\frac{1}{2}f_0 + 2f_1 + \cdots 2f_{n-1} + \frac{1}{2}f_n\Big) - \frac{2h}{3}\Big(\frac{1}{2}f_0 + 2f_1 + \cdots 2f_{n-1} + \frac{1}{2}f_n\Big) \\ &= \frac{h}{3}(f_0 + 4f_2 + 2f_4 + \cdots 4f_{n-2} + f_n), \end{aligned}$$

što je Simpsonova formula s n podintervala.

Sličan argument kao i prije možemo upotrijebiti i dalje. Vrijedi

$$I - I_{n/2}^{(1)} = \frac{16d_4^{(1)}}{n^4} + \frac{64d_6^{(1)}}{n^6} + \cdots.$$

Tada je

$$16(I - I_n^{(1)}) - (I - I_{n/2}^{(1)}) = \frac{-48d_6^{(1)}}{n^6} + \cdots,$$

odnosno

$$I = \frac{16I_n^{(1)} - I_{n/2}^{(1)}}{15} - \frac{-48d_6^{(1)}}{15n^6} + \cdots.$$

Ponovno, prvi član s desne strane proglasimo za novu aproksimaciju integrala

$$I_n^{(2)} = \frac{16I_n^{(1)} - I_{n/2}^{(1)}}{15}, \quad n \text{ djeljiv s } 4, \quad n \geq 4.$$

Induktivno, ako nastavimo postupak, dobivamo Richardsonovu ekstrapolaciju

$$I_n^{(k)} = \frac{4^k I_n^{(k-1)} - I_{n/2}^{(k-1)}}{4^k - 1}, \quad n \geq 2^k,$$

pri čemu je greška jednaka

$$E_n^{(k)} = I - I_n^{(k)} = \frac{d_{2k+2}^{(k)}}{n^{2k+2}} + \cdots = \beta_k(b-a)h^{2k+2}f^{(2k+2)}(\xi), \quad a \leq \xi \leq b.$$

Sada možemo definirati Rombergovu tablicu

$$\begin{array}{llll} I_1^{(0)} & & & \\ I_2^{(0)} & I_2^{(1)} & & \\ I_4^{(0)} & I_4^{(1)} & I_4^{(2)} & \\ \vdots & \vdots & \vdots & \ddots \end{array}.$$

Ako pogledamo omjere grešaka članova u stupcu, uz pretpostavku dovoljne glatkoće, onda dobivamo

$$\frac{E_n^{(k)}}{E_{2n}^{(k)}} = 2^{2k+2},$$

tj. omjeri pogrešaka u stupcu se moraju ponašati kao

$$\begin{array}{llll} 1 & & & \\ 4 & 1 & & \\ 4 & 16 & 1 & \\ 4 & 16 & 64 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{array} \quad \ddots.$$

Pokažimo na primjeru da prethodni omjeri pogrešaka u stupcu vrijede samo ako je funkcija dovoljno glatka.

Primjer 11.3.1. *Rombergovim algoritmom s točnošću* 10^{-12} *nađite vrijednosti integrala*

$$\int_0^1 e^x\,dx, \quad \int_0^1 x^{3/2}\,dx, \quad \int_0^1 \sqrt{x}\,dx$$

i pokažite kako se ponašaju omjeri pogrešaka u stupcima.

Pogledajmo redom funkcije. Eksponencijalna funkcija ima beskonačno mnogo neprekidnih derivacija, pa bi se računanje integrala morala ponašati po predviđanju. Kao vrijednost, nakon 2^5 *podintervala u trapeznoj formuli, dobivamo umjesto prave vrijednosti integrala* I*, približnu vrijednost*

$$\begin{aligned} I_5 &= 1.71828182845904524 \\ I &= e - 1 = 1.71828182845904524 \\ I - I_5 &= 0. \end{aligned}$$

Pokažimo omjere pogrešaka u stupcima,

```
0 1.0000
1 3.9512  1.0000
2 3.9875 15.6517  1.0000
3 3.9969 15.9913 62.4639   1.0000
4 3.9992 15.9777 63.6087 249.7197    1.0000
5 3.9998 15.9944 63.9017 254.4010 1000.5738 1.0000
```

a zatim samo eksponente omjera pogrešaka (eksponenti od 2*, koji bi ako je funkcija glatka morali biti* $2k+2$*).*

```
0 1.0000
1 1.9823 1.0000
2 1.9955 3.9682 1.0000
3 1.9989 3.9920 5.9650 1.0000
4 1.9997 3.9980 5.9912 7.9642 1.0000
5 1.9999 3.9995 5.9978 7.9910 9.9666 1.0000
```

Što je s drugom funkcijom? Funkciji $f(x) = x^{3/2}$ *puca druga derivacija u* 0*, pa bi zanimljivo ponašanje moralo početi veću drugom stupcu (za trapez je funkcija dovoljno glatka za ocjenu pogreške). Kao vrijednost, nakon* 2^{15} *podintervala u trapeznoj formuli, dobivamo umjesto prave vrijednosti integrala* I*, približnu vrijednost*

$$\begin{aligned} I_{15} &= 0.40000000000004512 \\ I &= 2/5 = 0.40000000000000000 \\ I - I_{15} &= -0.00000000000004512. \end{aligned}$$

Primijetite da je broj intervala poprilično velik! Što je s omjerima pogrešaka?

```
 0  1.0000
 1  3.7346 1.0000
 2  3.8154 5.4847 1.0000
 3  3.8721 5.5912 5.6484 1.0000
 4  3.9112 5.6331 5.6559 5.6566 1.0000
 5  3.9381 5.6484 5.6568 5.6568 5.6569 1.0000
 6  3.9567 5.6539 5.6568 5.6569  ···  5.6569 1.0000
 ⋮     ⋮      ⋮                          ⋱     ⋱
15  3.9981 5.6569  ···                   ···  5.6569 1.0000
```

Primjećujemo da su se nakon prvog stupca omjeri pogrešaka stabilizirali. Bit će nam mnogo lakše provjeriti što se događa ako napišemo samo eksponente omjera

pogrešaka.

$$
\begin{array}{cccccccc}
0 & 1.0000 & & & & & & \\
1 & 1.9010 & 1.0000 & & & & & \\
2 & 1.9318 & 2.4554 & 1.0000 & & & & \\
3 & 1.9531 & 2.4832 & 2.4978 & 1.0000 & & & \\
4 & 1.9676 & 2.4939 & 2.4998 & 2.4999 & 1.0000 & & \\
5 & 1.9775 & 2.4978 & 2.5000 & 2.5000 & 2.5000 & 1.0000 & \\
6 & 1.9843 & 2.4992 & 2.5000 & 2.5000 & 2.5000 & 2.5000 & 1.0000 \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 1.9993 & 2.5000 & \cdots & & & \cdots & 2.5000 \quad 1.0000
\end{array}
$$

Primijetite da su eksponenti omjera pogrešaka od drugog stupca nadalje točno za 1 *veći od eksponenta same funkcije (integriramo!).*

Situacija s funkcijom $f(x) = \sqrt{x}$ *mora biti još gora, jer njoj puca prva derivacija u* 0*. Nakon* 2^{15} *podintervala u trapeznoj formuli (što je ograničenje zbog veličine polja u programu), ne dobivamo željenu točnost*

$$
\begin{aligned}
I_{15} &= 0.66666665510837633 \\
I &= 2/3 = 0.66666666666666667 \\
I - I_{15} &= 0.00000001155829033.
\end{aligned}
$$

Omjeri pogrešaka u tablici su:

$$
\begin{array}{cccccccc}
0 & 1.0000 & & & & & & \\
1 & 2.6408 & 1.0000 & & & & & \\
2 & 2.6990 & 2.8200 & 1.0000 & & & & \\
3 & 2.7393 & 2.8267 & 2.8281 & 1.0000 & & & \\
4 & 2.7667 & 2.8281 & 2.8284 & 2.8284 & 1.0000 & & \\
5 & 2.7854 & 2.8284 & \cdots & \cdots & 2.8284 & 1.0000 & \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 2.8271 & 2.8284 & \cdots & & & \cdots & 2.8284 \quad 1.0000
\end{array}
$$

Pripadni eksponenti su

$$
\begin{array}{cccccccc}
0 & 1.0000 & & & & & & \\
1 & 1.4010 & 1.0000 & & & & & \\
2 & 1.4324 & 1.4957 & 1.0000 & & & & \\
3 & 1.4538 & 1.4991 & 1.4998 & 1.0000 & & & \\
4 & 1.4681 & 1.4998 & 1.5000 & 1.5000 & 1.0000 & & \\
5 & 1.4779 & 1.5000 & \cdots & \cdots & 1.5000 & 1.0000 & \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots \\
15 & 1.4993 & 1.5000 & \cdots & & & \cdots & 1.5000 \quad 1.0000
\end{array}
$$

Ipak, u ova dva jednostavna primjera, može se Rombergovom algoritmu "pomoći" tako da supstitucijom u integralu dobijemo glatku funkciju. U oba slučaja, ako stavimo supstituciju $x = t^2$, podintegralna će funkcija imati beskonačno mnogo neprekidnih derivacija, pa će se algoritam ponašati po ocjeni pogreške.

U literaturi postoji i malo drugačija oznaka za aproksimacije integrala u Rombergovoj tablici

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}.$$

Sama tablica ima oblik

$$\begin{array}{llll} T_0^{(0)} & & & \\ T_0^{(1)} & T_1^{(0)} & & \\ T_0^{(2)} & T_1^{(1)} & T_2^{(0)} & \\ \vdots & \vdots & \vdots & \ddots \end{array}.$$

Pokažimo sad nekoliko primjera kako treba, odnosno ne treba koristiti Rombergov algoritam.

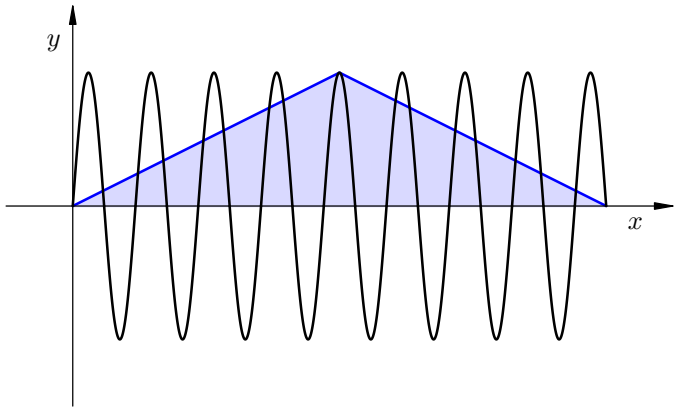
Primjer 11.3.2. *Izračunajte korištenjem Rombergovog algoritma približnu vrijednost integrala*

$$\int_0^1 \sin(17\pi x)\, dx$$

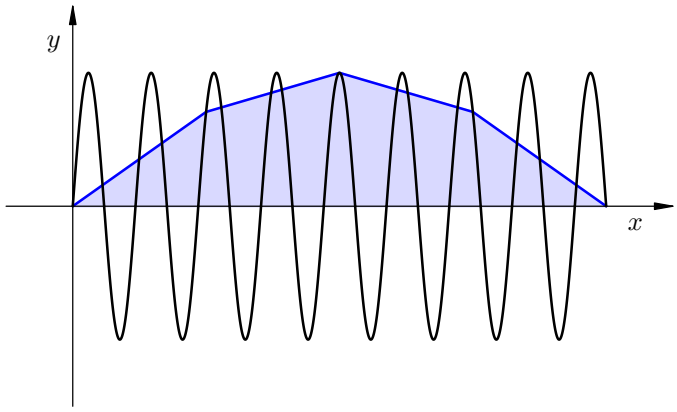
Tako da greška bude manja ili jednaka 10^{-4}. *Napišimo tablicu (samo prvih par decimala, ostale pamtimo u računalu, ali nemamo prostora za ispis)*

0	0.00000							
1	0.50000	0.66667						
2	0.60355	0.63807	0.63616					
3	0.62841	0.63671	0.63661	0.63662				
4	−0.00616	−0.21768	−0.27464	−0.28910	−0.29273			
5	0.02832	0.03982	0.05698	0.06225	0.06362	0.06397		
6	0.03525	0.03756	0.03741	0.03710	0.03700	0.03697	0.03697	
7	0.03690	0.03745	0.03745	0.03745	0.03745	0.03745	0.03745	0.03745

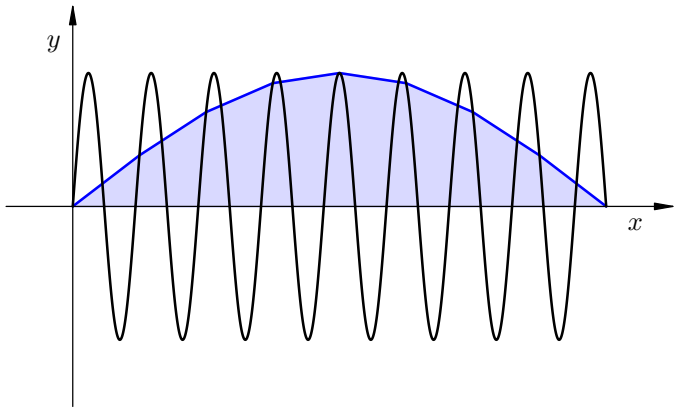
Što je razlog stabilizacije oko jedne, pa oko druge vrijednosti? Nedovoljan broj podintervala u trapezu, koji ne opisuju dobro ponašanje funkcije.



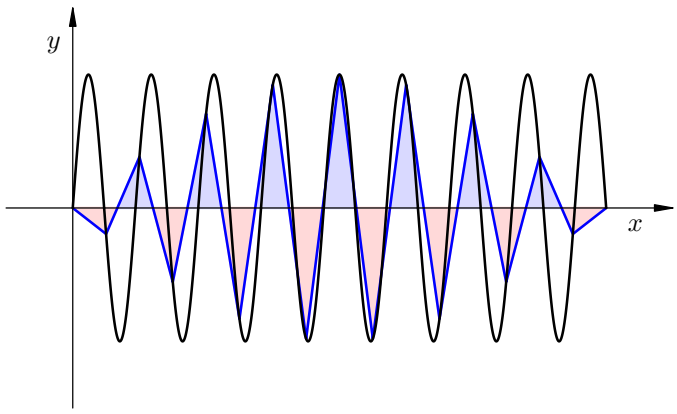
Produljena trapezna formula s 2 *podintervala.*



Produljena trapezna formula s 4 *podintervala.*



Produljena trapezna formula s 8 *podintervala.*



Produljena trapezna formula sa 16 *podintervala.*

11.4. Težinske integracione formule

Dosad smo detaljno analizirali samo nekoliko osnovnih Newton–Cotesovih integracionih formula s malim brojem točaka i pripadne produljene formule. U ovom odjeljku napravit ćemo opću konstrukciju i analizu točnosti za neke klase integracionih formula, uključujući opće Newton–Cotesove i Gaussove formule.

Želimo (približno) izračunati vrijednost integrala

$$I_w(f) = \int_a^b f(x)w(x)\,dx, \tag{11.4.1}$$

gdje je w pozitivna (ili barem nenegativna) "težinska" funkcija za koju pretpostavljamo da je integrabilna na (a, b), s tim da dozvoljavamo da w nije definirana u rubovima a i b. Interval integracije može biti konačan, ali i beskonačan. Drugim riječima, promatramo opći problem jednodimenzionalne integracije zadane funkcije f po zadanoj neprekidnoj mjeri $d\lambda$ generiranoj težinskom funkcijom w na zadanoj domeni. Katkad koristimo i skraćenu oznaku $I(f)$, umjesto $I_w(f)$, za integral u (11.4.1), ako je $w(x) = 1$ na cijelom $[a, b]$, ili kad je težinska funkcija jasna iz konteksta, da skratimo pisanje.

Kao i ranije, ovaj integral aproksimiramo "težinskom" sumom funkcijskih vrijednosti funkcije f na konačnom skupu točaka, Za razliku od ranijih oznaka, ovdje je zgodnije točke numerirati od 1, a ne od 0. Dakle, opća težinska integraciona ili kvadraturna formula za aproksimaciju integrala $I_w(f)$ ima oblik

$$I_n(f) = \sum_{k=1}^{n} w_k^{(n)} f(x_k^{(n)}), \tag{11.4.2}$$

gdje je n prirodni broj. Kao i prije, gornje indekse (n) za čvorove i težine često ne pišemo, ako su očiti iz konteksta, ali ne treba zaboraviti na ovisnost o n.

Dakle, sasvim općenito možemo pisati

$$I_w(f) = \int_a^b f(x)w(x)\,dx = I_n(f) + E_n(f), \tag{11.4.3}$$

gdje je $E_n(f)$ greška aproksimacije.

Osnovnu podlogu za konstrukciju integracionih formula i ocjenu greške $E_n(f)$ daje sljedeći rezultat.

Teorem 11.4.1. *Ako je $I_w(f)$ iz* (11.4.1) *Riemannov integral, i ako je $\widehat{f}$ bilo koja druga funkcija za koju postoji $I_w(\widehat{f})$, onda vrijedi ocjena*

$$|I_w(f) - I_w(\widehat{f})| \le \|w\|_1 \|f - \widehat{f}\|_\infty, \tag{11.4.4}$$

i postoji funkcija $\widehat{f}$ za koju se ova ocjena dostiže.

Dokaz:

Prvo uočimo da w ne mora biti nenegativna, jer je riječ o Riemannovom integralu, ali zato treba pretpostaviti da je $|w|$ integrabilna.

Ocjena izlazi direktno iz osnovnih svojstava Riemannovog integrala jer podintegralne funkcije moraju biti ograničene. Dobivamo

$$\begin{aligned} |I_w(f) - I_w(\widehat{f})| &= \Big| \int_a^b f(x)w(x)\,dx - \int_a^b \widehat{f}(x)w(x)\,dx \Big| \\ &\le \int_a^b |w(x)| \cdot |f(x) - \widehat{f}(x)|\,dx. \end{aligned}$$

Iskoristimo ocjenu

$$|f(x) - \widehat{f}(x)| \le \sup_{x\in[a,b]} |f(x) - \widehat{f}(x)| = \|f - \widehat{f}\|_\infty, \quad \forall x \in [a,b],$$

i definiciju L_1 norme funkcije w (koja je apsolutno integrabilna po pretpostavci)

$$\|w\|_1 = \int_a^b |w(x)|\,dx,$$

pa dobivamo traženu ocjenu. Ako za perturbiranu funkciju $\widehat{f}$ uzmemo

$$\widehat{f}(x) := f(x) + c\,\mathrm{sign}(w(x)),$$

gdje je $c > 0$ bilo koja konstanta, onda u ocjeni (11.4.4) dobivamo jednakost, uz $\|f - \widehat{f}\|_\infty = c$. ■

U ovoj formulaciji, za klasični Riemannov integral, domena $[a, b]$ integracije mora biti konačna. Teorem onda kaže da je apsolutni broj uvjetovanosti za $I_w(f)$ upravo jednak $\|w\|_1$ i ne ovisi o f, već samo o I_w.

Ovaj rezultat može se proširiti i na neprave Riemannove integrale (beskonačna domena, singulariteti funkcija), i tada više ne vrijedi zaključak o broju uvjetovanosti. Međutim, trenutno nam to nije bitno, već je ključna malo drugačija interpretacija ocjene (11.4.4).

Zamislimo da je $\widehat{f}$ neka aproksimacija (a ne perturbacija) funkcije f, koju želimo iskoristiti za približno računanje integrala. Onda (11.4.4) daje ocjenu (apsolutne) pogreške u integralu, preko greške aproksimacije funkcije u uniformnoj (L_∞) normi na $[a, b]$.

Ono što stvarno želimo dobiti je **niz** aproksimacija integrala koji konvergira prema $I_w(f)$. Jedan od puteva da to postignemo je izbor odgovarajućeg niza aproksimacija $\widehat{f}_n$, $n \in \mathbb{N}$, za funkciju f. Prethodna ocjena upućuje na to da, u ovisnosti o n, za aproksimacione funkcije $\widehat{f}_n$ treba uzimati takve funkcije za koje znamo da možemo postići po volji dobru **uniformnu** aproksimaciju funkcije f, jer tada

$$\|f - \widehat{f}_n\|_\infty \to 0 \Longrightarrow |I_w(f) - I_w(\widehat{f}_n)| \to 0, \quad n \to \infty.$$

Uočimo da ove aproksimacije, naravno, ovise o konkretnoj funkciji f. Da ne bismo za svaki novi f posebno konstruirali odgovarajući niz aproksimacija, poželjno je da bilo koju funkciju f, za koju postoji integral $I_w(f)$, možemo dovoljno dobro aproksimirati nekim prostorom funkcija. Tj. umjesto niza pojedinačnih aproksimacija, koristimo niz vektorskih prostora aproksimacionih funkcija V_n, a za svaki pojedini f nađemo pripadnu aproksimaciju $\widehat{f}_n \in V_n$.

Weierstraßov teorem o uniformnoj aproksimaciji neprekidnih funkcija polinomima na konačnom intervalu $[a, b]$ sugerira da treba uzeti V_n kao prostor polinoma $\mathcal{P}_d$ stupnja manjeg ili jednakog d, gdje d ovisi o n (i raste s n). Kao što ćemo vidjeti, korisno je dozvoliti da bude $d \neq n$.

Isti princip koristimo i za beskonačne domene, samo treba osigurati da su polinomi integrabilni s težinom w. To postižemo dodatnim zahtjevom na težinsku funkciju w, tako da pretpostavimo da svi momenti težinske funkcije

$$\mu_k := \int_a^b x^k w(x)\, dx, \quad k \in \mathbb{N}_0, \tag{11.4.5}$$

postoje i da su konačni. U nastavku pretpostavljamo da težinska funkcija w zadovoljava ovu pretpostavku. Takve težinske funkcije obično zovemo (polinomno) dopustivima.

Napomenimo odmah da se ovaj pristup može generalizirati i na bilo koji drugi sustav funkcija aproksimacionih funkcija $\{\widehat{f}_n \mid n \in \mathbb{N}\}$ koji je gust u prostoru $C[a,b]$ neprekidnih funkcija na $[a,b]$. Pripadni prostori V_n generirani su početnim komadima ovog sustava funkcija (kao linearne ljuske).

Za praktičnu primjenu ovog pristupa moramo moći efektivno izračunati integral $I_w(\widehat{f}_n)$ aproksimacione funkcije, i to za bilo koju funkciju f. To se najlakše postiže tako da konstruiramo pripadnu integracionu formulu I_n koja je egzaktna na cijelom prostoru $V_n = \mathcal{P}_d$ aproksimacionih funkcija. Dakle, uvjet egzaktnosti za I_n je

$$I_w(f) = I_n(f) \quad \text{ili} \quad E_n(f) = 0, \quad \text{za sve } f \in V_n.$$

Iz relacija (11.4.3) i (11.4.4) odmah dobivamo i ocjenu greške pripadne integracione formule $I_n(f)$, za bilo koji f

$$|E_n(f)| = |I_w(f) - I_n(f)| = |I_w(f) - I_w(\widehat{f}_n)| \leq \|w\|_1 \|f - \widehat{f}_n\|_\infty.$$

11.5. Gaussove integracione formule

Kao što smo već rekli, Gaussove formule imaju dvostruko više slobodnih parametara nego Newton–Cotesove, pa bi zbog toga trebale egzaktno integrirati polinome približno dvostruko većeg stupnja od Newton–Cotesovih.

Za razliku od Newton–Côtesovih formula, **Gaussove integracijske formule** su oblika

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

u kojima točke integracije x_i nisu unaprijed poznate, nego se izračunaju tako da greška takve formule bude najmanja. Motivirani praktičnim razlozima, promatrat ćemo malo općenitije integracijske formule oblika

$$\int_a^b w(x)\, f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

gdje je w **težinska funkcija**, pozitivna na otvorenom intervalu (a,b). Koeficijente w_i zovemo **težinski koeficijenti** ili, skraćeno, **težine** integracione formule. Gornji specijalni slučaj u kojem je $w \equiv 1$ čine formule koje se zovu **Gauss–Legendreove**. Težinska funkcija u općem slučaju utječe na težine i točke integracije, ali se ne pojavljuje eksplicitno u Gaussovoj formuli.

Bitno je znati da se za neke težinske funkcije na određenim intervalima, čvorovi

i težine standardno tabeliraju u priručnicima. To su

težinska funkcija w	interval	formula Gauss–
1	$[-1,1]$	Legendre
$\dfrac{1}{\sqrt{1-x^2}}$	$[-1,1]$	Čebišev
$\sqrt{1-x^2}$	$[-1,1]$	Čebišev 2. vrste
e^{-x}	$[0,\infty)$	Laguerre
e^{-x^2}	$(-\infty,\infty)$	Hermite

Glavni rezultat je sljedeći: ako zahtijevamo da formula integrira egzaktno polinome što je moguće većeg stupnja, onda su točke integracije x_i nultočke polinoma koji su ortogonalni na intervalu (a,b) obzirom na težinsku funkciju w, a težine w_i mogu se eksplicitno izračunati po formuli

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx, \quad i = 1,\ldots,n.$$

Pritom je ℓ_i poseban polinom Lagrangeove baze kojeg smo razmatrali u poglavlju o polinomnoj interpolaciji, definiran uvjetom $\ell_i(x_j) = \delta_{ij}$ (v. (10.2.16)). Primijetimo samo da je kod numeričke integracije zgodnije čvorove numerirati od x_1 do x_n, (za razliku od numeracije x_0 do x_n u poglavlju o interpolaciji), pa je i ℓ_i polinom stupnja $n-1$.

Kao što se Newton–Côtesove formule mogu dobiti integracijom Lagrangeovog interpolacijskog polinoma, tako se i Gaussove formule mogu dobiti integracijom Hermiteovog interpolacijskog polinoma. Takav pristup ekvivalentan je s pristupom u kojem zahtijevamo da Gaussove formule integriraju egzaktno polinome što je moguće višeg stupnja, tj. da vrijedi

$$\int_a^b w(x)\,x^j\,dx = \sum_{i=1}^{n} w_i x_i^j, \quad j = 0,1,\ldots,2n-1.$$

Mogli bismo iskoristiti ovu relaciju da napišemo $2n$ jednadžbi za $2n$ nepoznanica x_i i w_i, međutim nepoznanice x_i ulaze u sistem nelinearno, pa je ovakav pristup teži. Čak i dokaz da taj nelinearni sistem ima jedinstveno rješenje nije jednostavan.

Napišimo još jednom formulu za Hermiteov interpolacijski polinom h_{2n-1}, stupnja $2n-1$, koji u čvorovima integracije x_i interpolira vrijednosti $f_i = f(x_i)$

i $f_i' = f'(x_i)$, za $i = 1, \ldots, n$. Iz relacija (10.2.19) i (10.2.20) dobivamo

$$\begin{aligned} h_{2n-1}(x) &= \sum_{i=1}^{n} \Big(h_{i,0}(x)\, f_i + h_{i,1}(x)\, f_i' \Big) \\ &= \sum_{i=1}^{n} \Big([1 - 2(x - x_i)\ell_i'(x_i)]\, \ell_i^2(x)\, f_i + (x - x_i)\, \ell_i^2(x)\, f_i' \Big). \end{aligned}$$

Integracijom dobijemo

$$\int_a^b w(x)\, h_{2n-1}(x)\, dx = \sum_{i=1}^{n} \Big(A_i f_i + B_i f_i' \Big), \tag{11.5.1}$$

gdje su

$$\begin{aligned} A_i &= \int_a^b w(x)\, [1 - 2(x - x_i)\ell_i'(x_i)]\, \ell_i^2(x)\, dx, \\ B_i &= \int_a^b w(x)\, (x - x_i)\, \ell_i^2(x)\, dx. \end{aligned} \tag{11.5.2}$$

Integraciona formula (11.5.1) sliči na Gaussovu integracionu formulu, osim što ima dodatne članove $B_i f_i'$, koji koriste i derivacije funkcije f u čvorovima integracije.

Kad bi, kao u Newton–Cotesovim formulama, čvorovi x_i bili unaprijed zadani, iz uvjeta egzaktne integracije polinoma trebalo bi odrediti $2n$ parametara — težinskih koeficijenata A_i, B_i. Zato očekujemo da ovakva formula egzaktno integrira polinome do stupnja $2n-1$ (dimenzija prostora je $2n$). No, za upotrebu ove formule trebamo znati ne samo funkcijske vrijednosti $f(x_i)$ u čvorovima, već i vrijednosti derivacije $f'(x_i)$ funkcije u tim čvorovima.

Zato je ideja da probamo izbjeći korištenje derivacija, tako da izborom čvorova x_i **poništimo** koeficijente B_i uz derivacije f_i'. Točnost integracione formule mora ostati ista (egzaktna integracija polinoma stupnja do $2n - 1$), ali tako dobivena formula koristila bi samo funkcijske vrijednosti u čvorovima, tj. postala bi Gaussova integraciona formula.

Zaista, odgovarajućim izborom čvorova x_i može se postići da težinski koeficijenti B_i uz derivacije budu jednaki nula. Da bismo to dokazali, uvodimo posebni "polinom čvorova" (engl. "node polynomial") ω_n, koji ima nultočke u svim čvorovima integracije

$$\omega_n := (x - x_1)(x - x_2) \cdots (x - x_n).$$

Taj polinom smo već susreli u poglavlju o Lagrangeovoj interpolaciji. Sljedeći rezultat govori o tome kako treba izabrati čvorove.

Lema 11.5.1. *Ako je* $\omega_n(x) = (x - x_1) \cdots (x - x_n)$ **ortogonalna s težinom** w *na sve polinome nižeg stupnja, tj. ako vrijedi*

$$\int_a^b w(x)\, \omega_n(x)\, x^k\, dx = 0, \quad k = 0, 1, \ldots, n-1, \tag{11.5.3}$$

onda su svi koeficijenti B_i *u* (11.5.2) *jednaki nula.*

Dokaz:

Lagano provjerimo identitet

$$(x - x_i)\ell_i(x) = \frac{\omega_n(x)}{\omega_n'(x_i)}. \tag{11.5.4}$$

Supstitucijom u izraz (11.5.2) za B_i slijedi

$$B_i = \frac{1}{\omega_n'(x_i)} \int_a^b w(x)\, \omega_n(x)\, \ell_i(x)\, dx.$$

Kako je ℓ_i polinom stupnja $n-1$, i po pretpostavci je ω_n ortogonalna s težinom w na sve takve polinome, tvrdnja slijedi. ■

Lako se vidi da vrijedi i obrat ove tvrdnje, tj. da su svi koeficijenti $B_i = 0$ u (11.5.1), ako i samo ako je polinom čvorova ω_n ortogonalan na sve polinome nižeg stupnja (do $n-1$), s težinskom funkcijom w. Razlog tome je što su funkcije ℓ_i, $i = 1, \ldots, n$, Lagrangeove baze zaista baza prostora $\mathcal{P}_{n-1}$ (zadatak 10.2.2.).

Iz ranijih rezultata o ortogonalnim polinomima znamo da ortogonalni polinom stupnja n obzirom na w postoji i jednoznačno je određen do na (recimo) vodeći koeficijent. Da bismo dobili Gaussovu integracionu formulu u (11.5.1), polinom čvorova ω_n mora biti ortogonalni polinom s vodećim koeficijentom 1, tj. ω_n postoji i jedinstven je.

Nadalje, uvjet ortogonalnosti (11.5.3) **jednoznačno** određuje raspored čvorova za Gaussovu integraciju. Iz teorema 10.8.2. slijedi da ω_n ima n jednostrukih nultočaka u otvorenom intervalu (a, b) (što nam baš odgovara za integraciju). Njegove nultočke $x_1, \ldots, x_n$ možemo samo permutirati (drugačije indeksirati), a uz standardni dogovor $x_1 < \cdots < x_n$, one su jednoznačno određene.

Time smo dokazali da postoji jedinstvena Gaussova integraciona formula oblika

$$\int_a^b w(x)\, f(x)\, dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

Čvorovi integracije x_i su nultočke ortogonalnog polinoma stupnja n na $[a, b]$ s težinskom funkcijom w, a težinske koeficijente možemo izračunati iz (11.5.2), budući da je tada $w_i = A_i$, za $i = 1, \ldots, n$.

Iskoristimo li pretpostavku ortogonalnosti iz leme 11.5.1., možemo pojednostavniti i izraze za koeficijente $w_i = A_i$ u (11.5.2). Sasvim općenito, koristeći relaciju za B_i, koeficijent A_i možemo napisati u obliku

$$A_i = \int_a^b w(x)\,[1 - 2(x - x_i)\ell_i'(x_i)]\,\ell_i^2(x)\,dx = \int_a^b w(x)\,\ell_i^2(x)\,dx - 2\ell_i'(x_i)B_i.$$

Uz uvjet ortogonalnosti (Gaussova integracija) je $B_i = 0$ i $A_i = w_i$, pa je

$$w_i = \int_a^b w(x)\,\ell_i^2(x)\,dx.$$

Podintegralna funkcija je nenegativna i ℓ_i^2 je polinom stupnja $2(n-1)$ koji nije nul-polinom, pa desna strana mora biti pozitivna. Dakle, slijedi da su svi težinski koeficijenti u Gaussovoj integraciji pozitivni, $w_i > 0$, za $i = 1, \ldots, n$, što je vrlo bitno za numeričku stabilnost i konvergenciju.

Pokažimo još da vrijedi i

$$w_i = \int_a^b w(x)\,\ell_i^2(x)\,dx = \int_a^b w(x)\,\ell_i(x)\,dx.$$

Očito, to je isto kao i dokazati

$$\int_a^b w(x)\,\ell_i^2(x)\,dx - \int_a^b w(x)\,\ell_i(x)\,dx = \int_a^b w(x)\,\ell_i(x)\,(\ell_i(x) - 1)\,dx = 0.$$

Ali polinom $\ell_i(x) - 1$ se poništava u točki $x = x_i$, po definiciji polinoma ℓ_i, jer je $\ell_i(x_j) = \delta_{ij}$. Znači da $\ell_i(x) - 1$ mora sadržavati $x - x_i$ kao faktor, tj. možemo napisati

$$\ell_i(x) - 1 = (x - x_i)q(x),$$

gdje je q neki polinom stupnja $n-2$, za jedan manje od stupnja polinoma ℓ_i. Dakle,

$$\ell_i(x)\,(\ell_i(x) - 1) = \frac{\omega_n(x)}{\omega_n'(x_i)(x - x_i)}\,(\ell_i(x) - 1) = \frac{1}{\omega_n'(x_i)}\,\omega_n(x)\,q(x),$$

pa je zbog ortogonalnosti ω_n na sve polinome nižeg stupnja

$$\int_a^b w(x)\,\ell_i(x)\,(\ell_i(x) - 1)\,dx = \frac{1}{\omega_n'(x_i)} \int_a^b w(x)\,\omega_n(x)\,q(x)\,dx = 0.$$

■

Pokazali smo da Gaussovu integracionu formulu možemo dobiti kao integral Hermiteovog interpolacijskog polinoma, uz odgovarajući izbor čvorova, a za težinske koeficijente vrijedi

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx. \tag{11.5.5}$$

Primijetimo da je ova formula za koeficijente ista kao i ona u Newton–Côtesovim formulama, što je ovdje posljedica pretpostavke o ortogonalnosti. U oba slučaja do integracionih formula dolazimo interpolacijom funkcije u čvorovima.

Pokažimo i primjerom da ortogonalnost produkta korijenskih faktora, tj. funkcije $\omega_n(x)$ na sve polinome nižeg stupnja zapravo određuje točke integracije x_i.

Primjer 11.5.1. *Neka je $w(x) = 1$ i $n = 3$. Odredimo točke integracije iz uvjeta ortogonalnosti. Uobičajeno je da za interval integracije uzmemo $(-1, 1)$, budući da integrale na drugim intervalima možemo lagano računati, ako podintegralnu funkciju transformiramo linearnom supstitucijom. Problem se dakle svodi na to da odredimo nultočke kubične funkcije $\omega_3(x) = a + bx + cx^2 + x^3$ za koju vrijedi*

$$\int_{-1}^{1} \omega_3(x)\,x^k\,dx = 0, \quad k = 0, 1, 2.$$

Nakon integracije dobivamo sustav jednadžbi za koeficijente a, b, c

$$2a + \frac{2}{3}c = 0, \qquad \frac{2}{3}b + \frac{2}{5} = 0, \qquad \frac{2}{3}a + \frac{2}{5}c = 0,$$

odakle nađemo $a = c = 0$ i $b = -3/5$. Dobivamo

$$\omega_3(x) = x^3 - \frac{3}{5}x = \left(x + \sqrt{\frac{3}{5}}\right)x\left(x - \sqrt{\frac{3}{5}}\right),$$

odakle slijedi da su točke integracije $x_i = -\sqrt{3/5},\ 0,\ \sqrt{3/5}$.

Teorijski, ovaj pristup možemo iskoristiti za sve moguće intervale integracije i razne težinske funkcije. Za veće n potrebno je odrediti nule polinoma visokog stupnja, što je egzaktno nemoguće, a numerički u najmanju ruku neugodno. Stoga je potrebno za specijalne težine i intervale integracije doći do dodatnih informacija o ortogonalnim polinomima. Na kraju, bilo bi dobro izračunati formulom i težinske faktore w_i u Gaussovim formulama. Analitički je moguće doći do ovakvih rezultata za mnoge specijalne težine $w(x)$ koje se pojavljuju u primjenama. Riješimo na početku važnu situaciju $w \equiv 1$, $a = -1$, $b = 1$. Pripadne formule nazvali smo Gauss–Legendreovima; u gornjem primjeru izračunali smo točke integracije za Gauss–Legendreovu formulu reda 3.

Zadatak 11.5.1. *Iz uvjeta egzaktnosti i poznatih točaka integracije za $n = 3$ izračunajte težinske koeficijente w_i. Primijetite da je sustav jednadžbi linearan, pa stoga računanje ovih faktora ne predstavlja veće probleme.*

11.5.1. Gauss–Legendreove integracione formule

Prepostavimo u daljnjem da je $w \equiv 1$ na intervalu $(-1, 1)$ i izvedimo specijalnu Gaussovu formulu, tj. Gauss–Legendre-ovu formulu

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i).$$

Kao što znamo, Legendreov polinom stupnja n definiran je **Rodriguesovom formulom**

$$P_n(x) = \frac{1}{2^n n!}\,\frac{d^n}{dx^n}(x^2-1)^n.$$

Tako definirani polinomi čine **ortogonalnu bazu** u prostoru polinoma stupnja n, tj. oni su linearno nezavisni i ortogonalni obzirom na skalarni produkt

$$\langle P, Q\rangle := \int_{-1}^{1} P(x)\,Q(x)\,dx. \tag{11.5.6}$$

Pojavljuju se prirodno u parcijalnim diferencijalnim jednadžbama, kod metode separacije varijabli za Laplaceovu jednadžbu u kugli. Za nas je bitno samo jedno specijalno svojstvo, iz kojeg slijede sva ostala:

Lema 11.5.2. *Legendreov polinom stupnja n ortogonalan je na sve potencije x^k nižeg stupnja, tj. vrijedi*

$$\int_{-1}^{1} x^k P_n(x)\,dx = 0, \quad \text{za} \quad k = 0, 1, \ldots, n-1,$$

i vrijedi

$$\int_{-1}^{1} x^n P_n(x)\,dx = \frac{2^{n+1}(n!)^2}{(2n+1)!}.$$

Dokaz:

Uvrštavanjem Rodriguesove formule, nakon k $(k < n)$ parcijalnih integracija dobivamo

$$\int_{-1}^{1} x^k \frac{d^n}{dx^n}(x^2-1)^n\,dx = \underbrace{x^k \frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n\Big|_{-1}^{1}}_{=0} - \int_{-1}^{1} kx^{k-1}\frac{d^{n-1}}{dx^{n-1}}(x^2-1)^n\,dx$$

$$= \cdots = (-1)^k k! \int_{-1}^{1} \frac{d^{n-k}}{dx^{n-k}}(x^2-1)^n\,dx = 0,$$

pa smo dokazali prvu formulu. Za $k = n$, na isti način imamo

$$\int_{-1}^{1} x^n \frac{d^n}{dx^n}(x^2-1)^n\,dx = (-1)^n n! \int_{-1}^{1} (x^2-1)^n\,dx = 2n! \int_{0}^{1} (1-x^2)^n\,dx$$
$$= \{x = \sin t\} = 2n! \int_{0}^{\pi/2} \cos^{2n+1} t\,dt.$$

Za zadnji integral parcijalnom integracijom izlazi

$$\int_{0}^{\pi/2} \cos^{2n+1} t\,dt = \underbrace{\frac{\cos^{2n} t\,\sin t}{2n+1}\Big|_{0}^{\pi/2}}_{=0} + \frac{2n}{2n+1} \int_{0}^{\pi/2} \cos^{2n-1} t\,dt$$
$$= \cdots = \frac{2n(2n-2)\cdots 2}{(2n+1)(2n-1)\cdots 3} \int_{0}^{\pi/2} \cos t\,dt,$$

pa je stoga

$$\int_{-1}^{1} x^n \frac{d^n}{dx^n}(x^2-1)^n\,dx = 2n!\,\frac{2n(2n-2)\cdots 2}{(2n+1)(2n-1)\cdots 3}.$$

Pomnožimo li brojnik i nazivnik s $2n(2n-2)\cdots 2 = 2^n n!$, a zatim, zbog definicije Legendreovog polinoma P_n, sve podijelimo s $2^n n!$, slijedi

$$\int_{-1}^{1} x^n P_n(x)\,dx = \frac{1}{2^n n!}\,2n!\,\frac{2^n n! \cdot 2^n n!}{(2n+1)!} = \frac{2^{n+1}(n!)^2}{(2n+1)!}.$$

■

Lema 11.5.3. *Legendreovi polinomi su ortogonalni na intervalu* $(-1, 1)$ *obzirom na skalarni produkt* (11.5.6)

$$\int_{-1}^{1} P_m(x)\,P_n(x)\,dx = 0, \quad \text{za} \quad m \neq n.$$

Norma Legendreovog polinoma je

$$\|P_n\|^2 := \int_{-1}^{1} [P_n(x)]^2\,dx = \frac{2}{2n+1}.$$

Dokaz:

Prva tvrdnja je direktna posljedica dokazane ortogonalnosti na potencije nižeg

stupnja. Druga tvrdnja slijedi iz

$$\int_{-1}^{1} [P_n(x)]^2\,dx = \int_{-1}^{1} \left[\frac{1}{2^n n!}\,\frac{(2n)!}{n!}\,x^n + \cdots\right] P_n(x)\,dx.$$

Potencije manje od x^n ne doprinose integralu, pa druga tvrdnja leme 11.5.2. povlači

$$\int_{-1}^{1} [P_n(x)]^2\,dx = \frac{(2n)!}{2^n (n!)^2}\,\frac{2^{n+1}(n!)^2}{(2n+1)!} = \frac{2}{2n+1}.$$

■

Lema 11.5.4. *Legendreovi polinomi P_n imaju n nultočaka, koje su sve realne i različite, i nalaze se u otvorenom intervalu $(-1, 1)$.*

Dokaz:

Dokaz ide iz definicije Legendreovih polinoma

$$P_n(x) = \frac{1}{2^n n!}\,\frac{d^n}{dx^n}(x^2 - 1)^n,$$

induktivnom primjenom Rolleovog teorema. Polinom $(x^2 - 1)^n$ je stupnja $2n$ i ima višestruke (n-terostruke) nultočke u rubovima intervala ± 1. Prema Rolleovom teoremu, prva derivacija ima jednu nultočku u intervalu $(-1, 1)$. Međutim, prva derivacija je, također, nula u ± 1, pa ukupno mora imati tri nultočke u zatvorenom intervalu $[-1, 1]$. Druga derivacija stoga ima dvije unutarnje nule po Rolleovom teoremu, i dvije u ± 1, pa ima ukupno četiri nule u $[-1, 1]$. I tako redom, vidimo da $n - 1$-a derivacija ima $n - 1$ unutarnju nultočku i još dvije u ± 1. Na kraju zaključimo da n-ta derivacija, koja je do na multiplikativni faktor jednaka P_n, ima n unutarnjih nultočaka. ■

Na taj način smo zapravo našli točke integracije u Gauss–Legendreovoj formuli i bez eksplicitnog rješavanja nelinearnog sistema jednadžbi za w_i i x_i, iz uvjeta egzaktne integracije potencija najvećeg mogućeg stupnja. Taj rezultat rezimiran je u sljedećem teoremu.

Teorem 11.5.1. *Čvorovi integracije u Gauss–Legendreovoj formuli reda n su nultočke Legendreovog polinoma P_n, za svaki n.*

Dokaz:

Znamo da su točke integracije x_i nultočke polinoma ω_n po konstrukciji. Zbog uvjeta ortogonalnosti (11.5.3) polinom ω_n, s vodećim koeficijentom 1, proporcionalan je Legendreovom polinomu P_n. Vodeći koeficijent u P_n lako izračunamo iz Rodriguesove formule, odakle je

$$\omega_n(x) = \frac{2^n (n!)^2}{(2n)!}\,P_n(x),$$

pa vidimo da su sve nultočke polinoma ω_n zapravo nultočke od P_n (lema 11.5.4.).■

Primjer 11.5.2. *Iz Rodriguesove formule možemo izračunati nekoliko prvih Legendreovih polinoma.*

$$
\begin{aligned}
P_0(x) &= 1,\\
P_1(x) &= \frac{1}{2}\,\frac{d}{dx}(x^2-1) = x,\\
P_2(x) &= \frac{1}{8}\,\frac{d^2}{dx^2}(x^2-1)^2 = \frac{1}{2}\,(3x^2-1),\\
P_3(x) &= \frac{1}{48}\,\frac{d^3}{dx^3}(x^2-1)^3 = \frac{1}{2}\,(5x^3-3x),\\
P_4(x) &= \frac{1}{16\cdot 24}\,\frac{d^4}{dx^4}(x^2-1)^4 = \frac{1}{8}\,(35x^4-30x^2+3),\\
P_5(x) &= \frac{1}{8}\,(63x^5-70x^3+15x),\\
P_6(x) &= \frac{1}{16}\,(231x^6-315x^4+105x^2-5),\\
P_7(x) &= \frac{1}{16}\,(429x^7-693x^5+315x^3-35x),\\
P_8(x) &= \frac{1}{128}\,(6435x^8-12012x^6+6930x^4-1260x^2+35).
\end{aligned}
$$

Vidimo, na primjer, da su nultočke od P_3 identične s točkama integracije koje smo dobili u primjeru 11.5.1.*, direktno iz uvjeta ortogonalnosti.*

Računanje nultočaka Legendreovih polinoma (na mašinsku točnost!) nije jednostavan problem, budući da egzaktne formule postoje samo za male stupnjeve. Napomenimo za sad samo toliko, da postoje specijalni algoritmi, te da je dovoljno tabelirati te nultočke jednom, pa brzina algoritma nije važna, nego samo preciznost. Tabelirane nultočke (kao i težine w_i) moguće je naći u gotovo svim standardnim knjigama i tablicama iz područja numeričke analize.

Postoji lakši način za računanje $P_n(x)$, zasnovan na činjenici da Legendreovi polinomi zadovoljavaju tročlanu rekurziju, čiji se koeficijenti mogu eksplicitno izračunati. Ova rekurzivna formula igra važnu ulogu i u konstrukciji spomenutog specijalnog algoritma za traženje nultočaka.

Lema 11.5.5. *Legendreovi polinomi zadovoljavaju rekurzivnu formulu*

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad n \geq 1,$$

s početnim vrijednostima $P_0(x) = 1$, $P_1(x) = x$.

Dokaz:

Kako je $xP_n(x)$ polinom stupnja $n+1$ i $\{P_i\}_{i=0}^{n+1}$ baza za prostor polinoma stupnja do $n+1$, postoje koeficijenti c_i tako da vrijedi

$$xP_n(x) = \sum_{i=0}^{n+1} c_i P_i(x).$$

Pomnožimo li obje strane s $P_k(x)$ i integriramo od -1 do 1, zbog ortogonalnosti (lema 11.5.3.) slijedi

$$\int_{-1}^{1} xP_k(x)\, P_n(x)\, dx = c_k \int_{-1}^{1} P_k^2(x)\, dx. \tag{11.5.7}$$

Ali za $k < n-1$ je $xP_k(x)$ polinom stupnja manjeg ili jednakog $n-1$, pa je $P_n(x)$ ortogonalan na njega (lema 11.5.2.). Stoga je $c_k = 0$ za $k < n-1$, a u sumi za $xP_n(x)$ ostaju samo zadnja tri člana

$$xP_n(x) = c_{n+1}P_{n+1}(x) + c_nP_n(x) + c_{n-1}P_{n-1}(x). \tag{11.5.8}$$

Treba još izračunati koeficijente c_{n+1}, c_n i c_{n-1}. Kako je

$$P_n(x) = \frac{(2n)!}{2^n(n!)^2}\,\omega_n(x) = \frac{(2n)!}{2^n(n!)^2}\,x^n + \text{niže potencije od } x,$$

usporedimo li koeficijente uz x^{n+1} u (11.5.8), dobivamo da je

$$\frac{(2n)!}{2^n(n!)^2} = c_{n+1}\,\frac{(2n+2)!}{2^{n+1}[(n+1)!]^2},$$

odakle slijedi da je

$$c_{n+1} = \frac{n+1}{2n+1}.$$

Lagano se vidi (iz Rodriguesove formule) da se u Legendreovim polinomima pojavljuju samo alternirajuće potencije, tj. P_{2n} je linearna kombinacija parnih potencija x^{2k}, $k = 0, \ldots, n$, a P_{2n+1} je linearna kombinacija neparnih potencija x^{2k+1}, $k = 0, \ldots, n$. Iz rekurzije (11.5.8) na osnovu toga zaključimo da je $c_n = 0$, pa preostaje samo izračunati c_{n-1}. Za $k = n-1$, iz (11.5.7) imamo da je

$$\int_{-1}^{1} xP_{n-1}(x)\, P_n(x)\, dx = c_{n-1} \int_{-1}^{1} P_{n-1}^2(x)\, dx.$$

Zbog

$$xP_{n-1}(x) = \frac{(2(n-1))!}{2^{n-1}[(n-1)!]^2}\,x^n + \text{niže potencije od } x$$

i ortogonalnosti P_n na sve niže potencije od x, dobivamo

$$\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2}\int_{-1}^{1} x^n P_n(x)\,dx = c_{n-1}\int_{-1}^{1} P_{n-1}^2(x)\,dx.$$

Ovi integrali su poznati (lema 11.5.2. i lema 11.5.3.), pa slijedi

$$c_{n-1} = \frac{n}{2n+1}.$$

Tako smo našli sve nepoznate koeficijente u linearnoj kombinaciji (11.5.8), odakle odmah slijedi tročlana rekurzija. Primijetimo da smo usput dokazali i formulu

$$\int_{-1}^{1} xP_{n-1}(x)\,P_n(x)\,dx = \frac{n}{2n+1}\,\frac{2}{2n-1} = \frac{2n}{4n^2-1}. \tag{11.5.9}$$

■

Zadatak 11.5.2. *Budući da Legendreovi polinomi zadovoljavaju tročlanu rekurziju, moguće je napisati algoritam za brzu sumaciju parcijalnih suma redova oblika*

$$\sum_{n=0}^{\infty} a_n P_n(x),$$

poznat pod nazivom **generalizirana Hornerova shema**. *Koristeći rekurziju iz leme* 11.5.5., *napišite eksplicitno taj algoritam. Razvoji po Legendreovim polinomima pojavljuju se često kod rješavanja Laplaceove jednadžbe u sfernim koordinatama.*

Sljedeće dvije leme korisne su za dobivanje ekplicitnih formula za težine u Gauss–Legendreovim formulama.

Lema 11.5.6. (Christoffel–Darbouxov identitet) *Za Legendreove polinome P_n vrijedi*

$$(t-x)\sum_{k=0}^{n}(2k+1)P_k(x)P_k(t) = (n+1)\,[P_{n+1}(t)P_n(x) - P_n(t)P_{n+1}(x)].$$

Dokaz:

Pomnožimo li rekurziju iz leme 11.5.5. (uz zamjenu $n \mapsto k$) s $P_k(t)$, dobijemo

$$(2k+1)xP_k(x)P_k(t) = (k+1)P_{k+1}(x)P_k(t) + kP_{k-1}(x)P_k(t).$$

Zamijenimo li x i t imamo

$$(2k+1)tP_k(t)P_k(x) = (k+1)P_{k+1}(t)P_k(x) + kP_{k-1}(t)P_k(x).$$

Odbijanjem prve relacije od druge, slijedi

$$(2k+1)(t-x)P_k(x)P_k(t) = (k+1)\left[P_{k+1}(t)P_k(x) - P_k(t)P_{k+1}(x)\right] \\ - k\left[P_k(t)P_{k-1}(x) - P_{k-1}(t)P_k(x)\right].$$

Sumiramo li po k od 1 do n, sukcesivni članovi u sumi na desnoj strani se krate, pa ostaju samo prvi i zadnji

$$(t-x)\sum_{k=1}^{n}(2k+1)P_k(x)P_k(t) = (n+1)\left[P_{n+1}(t)P_n(x) - P_n(t)P_{n+1}(x)\right] - (t-x).$$

Zadnji član možemo prebaciti na lijevu stranu kao nulti član u sumi, a to je baš Christoffel–Darbouxov identitet. ■

Lema 11.5.7. *Derivacija Legendreovih polinoma može se rekurzivno izraziti pomoću samih Legendreovih polinoma, formulom*

$$(1-x^2)P_n'(x) + nxP_n(x) = nP_{n-1}(x), \quad n \geq 1.$$

Dokaz:

Polinom $(1-x^2)P_n' + nxP_n$ je očito stupnja manjeg ili jednakog od $n+1$. Napišimo P_n kao linearnu kombinaciju potencija od x (pojavljuje se samo svaka druga potencija)

$$P_n(x) = a_n x^n + a_{n-2}x^{n-2} + \cdots,$$

pa je

$$P_n'(x) = na_n x^{n-1} + (n-2)a_{n-2}x^{n-3} + \cdots.$$

No, onda je

$$(1-x^2)P_n'(x) + nxP_n(x) = (-na_n + na_n)x^{n+1} + O(x^{n-1}),$$

tj. polinom $(1-x^2)P_n' + nxP_n$ je zapravo stupnja $n-1$. Kao i u dokazu rekurzivne formule, moraju postojati koeficijenti c_i takovi da vrijedi

$$(1-x^2)P_n'(x) + nxP_n(x) = \sum_{i=0}^{n-1} c_i P_i(x).$$

Pomnožimo ovu relaciju s $P_k(x)$ i integriramo od -1 do 1. Zbog ortogonalnosti, na desnoj strani ostaje samo jedan član

$$\frac{2}{2k+1}\,c_k = \int_{-1}^{1}(1-x^2)\,P_n'(x)\,P_k(x)\,dx + n\int_{-1}^{1} xP_n(x)\,P_k(x)\,dx.$$

Prvi integral integriramo parcijalno, pa kako se faktor $(1-x^2)$ poništava na granicama integracije, slijedi

$$\frac{2}{2k+1}\,c_k = -\int_{-1}^{1} P_n(x)\,\frac{d}{dx}[(1-x^2)P_k(x)]\,dx + n\int_{-1}^{1} xP_n(x)\,P_k(x)\,dx.$$

Za $k < n-1$, oba integranda su oblika $P_n(x) \times$ (polinom stupnja najviše $n-1$), pa su svi ovi integrali jednaki nula (lema 11.5.2.), tj. $c_k = 0$ za $k < n-1$. Za $k = n-1$ treba izračunati dva integrala u prethodnoj relaciji. Drugi je jednostavan

$$n\int_{-1}^{1} xP_n(x)\,P_{n-1}(x)\,dx = (11.5.9) = \frac{2n^2}{4n^2-1}.$$

U prvom integralu

$$-\int_{-1}^{1} P_n(x)\,\frac{d}{dx}[(1-x^2)P_{n-1}(x)]\,dx,$$

zbog prve tvrdnje u lemi 11.5.2. (ortgonalnost), doprinos daje samo vodeći član u $(1-x^2)P_{n-1}(x)$, pa je taj integral jednak

$$\int_{-1}^{1} P_n(x)\,\frac{d}{dx}\Big\{x^2\,\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2}\,x^{n-1}\Big\}\,dx,$$

a zbog druge tvrdnje u lemi, integral se svodi na

$$\frac{(2n-2)!}{2^{n-1}[(n-1)!]^2}\,(n+1)\,\frac{2^{n+1}(n!)^2}{(2n+1)!} = \frac{2n(n+1)}{(2n+1)(2n-1)}.$$

Na kraju je

$$c_{n-1} = \frac{2n-1}{2}\left[\frac{2n(n+1)}{(2n+1)(2n-1)} + \frac{2n^2}{(2n+1)(2n-1)}\right] = n,$$

što smo i htjeli dokazati. ■

Lema 11.5.8. *Težinski faktori u Gauss–Legendreovim formulama mogu se eksplicitno izračunati formulama*

$$w_i = \frac{2(1-x_i^2)}{n^2[P_{n-1}(x_i)]^2},$$

gdje su x_i, $i = 0, \ldots, n$, nultočke Legendreovog polinoma P_n.

Dokaz:

Neka je x_i nultočka polinoma P_n. Stavimo li $t = x_i$ u Christoffel–Darbouxov identitet (lema 11.5.6.), dobivamo

$$\frac{(n+1)P_{n+1}(x_i)P_n(x)}{x - x_i} = -\sum_{k=0}^{n}(2k+1)P_k(x)P_k(x_i).$$

Kad integriramo ovu jednakost od -1 do 1 i uzmemo u obzir da je k-ti Legendreov polinom ortogonalan na konstantu $P_k(x_i)$, na desnoj strani preostane samo član za $k = 0$

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx = \frac{-2}{(n+1)P_{n+1}(x_i)}.$$

Tročlana rekurzija iz leme 11.5.5. u nultočki x_i Legendreovog polinoma P_n ima oblik $(n+1)P_{n+1}(x_i) = -nP_{n-1}(x_i)$, pa je stoga

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx = \frac{2}{nP_{n-1}(x_i)}.$$

Za težinske koeficijente w_i vrijede relacije (11.5.5) i (11.5.4)

$$w_i = \int_{-1}^{1} \ell_i(x)\,dx = \int_{-1}^{1} \frac{\omega_n(x)}{\omega_n'(x_i)(x - x_i)}\,dx = \int_{-1}^{1} \frac{P_n(x)}{P_n'(x_i)(x - x_i)}\,dx,$$

pa je dakle

$$w_i = \frac{2}{nP_n'(x_i)\,P_{n-1}(x_i)}. \tag{11.5.10}$$

Primijetimo da je Christoffel–Darbouxov identitet potreban jedino zato da se izračuna neugodan integral

$$\int_{-1}^{1} \frac{P_n(x)}{(x - x_i)}\,dx,$$

u kojem podintegralna funkcija ima uklonjivi singularitet.

Na kraju, iskoristimo rekurzivnu formulu za derivacije Legendreovog polinoma iz leme 11.5.7. u specijalnom slučaju kada je $x = x_i$. Dobivamo da vrijedi

$$(1 - x_i^2)P_n'(x_i) = nP_{n-1}(x_i).$$

Uvrstimo li taj rezultat u (11.5.10), tvrdnja slijedi. ■

U dokazu prethodne leme 11.5.8. pokazali smo (usput) da u nultočki x_i Legendreovog polinoma P_n vrijedi

$$(1 - x_i^2)P_n'(x_i) = nP_{n-1}(x_i) = -(n+1)P_{n+1}(x_i).$$

Ovu relaciju možemo iskoristiti na različite načine u (11.5.10), što daje pet raznih formula za težinske koeficijente u Gauss–Legendreovim formulama

$$\begin{aligned} w_i &= \frac{2(1-x_i^2)}{[nP_{n-1}(x_i)]^2} = \frac{2(1-x_i^2)}{[(n+1)P_{n+1}(x_i)]^2} \\ &= \frac{2}{nP_n'(x_i)\,P_{n-1}(x_i)} = -\frac{2}{(n+1)P_n'(x_i)\,P_{n+1}(x_i)} \\ &= \frac{2}{(1-x_i^2)\,[P_n'(x_i)]^2}. \end{aligned} \tag{11.5.11}$$

Sljedeći teorem rezimira prethodne rezultate, i ujedno daje ocjenu greške za Gauss–Legendreovu integraciju.

Teorem 11.5.2. *Za funkciju $f \in C^{2n}[-1,1]$ Gauss–Legendreova formula integracije glasi*

$$\int_{-1}^{1} f(x)\,dx = \sum_{i=1}^{n} w_i f(x_i) + E_n(f),$$

gdje su x_i nultočke Legendreovog polinoma P_n i koeficijenti w_i dani u (11.5.11). *Za grešku $E_n(f)$ vrijedi*

$$E_n(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi), \quad \xi \in (-1,1).$$

Dokaz:

Treba samo dokazati formulu za ocjenu greške. Kako je Gauss–Legendreova formula zapravo integral Hermiteovog interpolacijskog polinoma, treba integrirati grešku kod Hermiteove interpolacije, koju smo procijenili u teoremu 10.2.5., i uvrstiti odgovarajući ω_n. Integracijom i primjenom teorema srednje vrijednosti za integrale, dobivamo

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_{-1}^{1} \omega_n^2(x)\,dx,$$

za neki $\xi \in (-1,1)$. Kako je

$$\omega_n(x) = \frac{2^n(n!)^2}{(2n)!}\, P_n(x),$$

zbog poznatog kvadrata norme Legendreovog polinoma (lema 11.5.3.), imamo

$$E_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \left[\frac{2^n(n!)^2}{(2n)!}\right]^2 \frac{2}{2n+1} = \frac{2^{2n+1}(n!)^4}{(2n+1)\,[(2n)!]^3}\, f^{(2n)}(\xi).$$

■

Navedeni izraz za grešku nije lagano primijeniti, budući da je potrebno naći neku ogradu za vrlo visoku derivaciju funkcije f (red derivacije je dva puta veći nego kod Newton–Côtesovih formula). Član uz $f^{(2n)}(\xi)$ vrlo brzo pada s porastom n. Na primjer, za $n = 6$, greška je oblika

$$1.6 \cdot 10^{-12} f^{(12)}(\xi).$$

Da ocjena greške za Gaussove formule može biti previše pesimistična, pokazuje sljedeći primjer.

Primjer 11.5.3. *Primijenimo Gauss–Legendreovu formulu na integral*

$$\int_0^{\pi/2} \log(1+t)\,dt = \left(1+\frac{\pi}{2}\right)\left[\log\left(1+\frac{\pi}{2}\right)-1\right]+1.$$

Zamjena varijable $t = \pi(x+1)/4$ *prebacuje integral na standardnu formu*

$$\int_{-1}^{1} \frac{\pi}{4}\,\log\left(1+\frac{\pi(x+1)}{4}\right)dx.$$

U ovom slučaju možemo lagano izračunati bilo koju derivaciju podintegralne funkcije, koja raste s faktorijelima. Zapravo, sve ocjene greške formula za numeričku integraciju pokazuju slično ponašanje (usporedite, na primjer, trapeznu i Simpsonovu formulu), ali Gaussove formule naročito, budući da uključuju visoke derivacije. Tako je, na primjer, osma derivacija, koja je potrebna za Gaussovu formulu s četiri točke jednaka

$$\left(\frac{\pi}{4}\right)^9 \cdot \frac{-7!}{(1+t)^8},$$

pa je greška 7! *puta veća nego da smo, recimo, integrirali trigonometrijsku funkciju* sin *ili* cos*, koje imaju ograničene derivacije. Ipak, lagano vidimo da već sa šest točaka dobivamo* 6 *znamenaka točno, iako ocjena greške uključuje faktor od* 11!*. Simpsonovoj formuli treba* 64 *točke za istu točnost. Možemo slutiti, da je za analitičke funkcije moguća bolja ocjena greške.*

Korolar 11.5.1. (Uvjeti egzaktnosti) *Gauss–Legendreova formula egzaktno integrira polinome stupnja* $2n-1$*.*

Dokaz:

Očito, budući da se greška, koja uključuje $2n$-tu derivaciju, poništava na takvim polinomima. ■

Svojstvo iz gornjeg korolara može se upotrijebiti za alternativni dokaz teorema 11.5.2., kao što smo napomenuli na početku. Hermiteova interpolacija poslužila je kao "trik", da izbjegnemo rješavanje nelinearnog sistema koji proizilazi iz uvjeta egzaktnosti.

Rekurziju za derivacije Legendreovih polinoma iz leme 11.5.7. možemo koristiti i za računanje vrijednosti $P_n'(x)$

$$(1-x^2)P_n'(x) = n(P_{n-1}(x) - xP_n(x)), \quad n \geq 1.$$

Nažalost, ova formulu ne možemo upotrijebiti u rubnim točakama $x = \pm 1$, zbog dijeljenja s nulom. Međutim, Legendreovi polinomi zadovoljavaju i mnoge druge rekurzivne relacije. Neke od njih dane su u sljedećem zadatku.

Zadatak 11.5.3. *Dokažite da za Legendreove polinoma vrijedi $P_n(1) = 1$, za $n \geq 0$, što opravdava izbor normalizacije. Također, dokažite da za $n \geq 1$ vrijede rekurzivne relacije*

$$\begin{aligned} P_n'(x) - xP_{n-1}'(x) &= nP_{n-1}(x), \\ xP_n'(x) - P_{n-1}'(x) &= nP_n(x), \\ P_{n+1}'(x) - P_{n-1}'(x) &= (2n+1)P_n(x) \\ \int_{-1}^{x} P_n(t)\,dt &= \frac{1}{2n+1}\left(P_{n+1}(x) - P_{n-1}(x)\right). \end{aligned}$$

Na kraju, primijetimo da Gaussove formule možemo shvatiti i kao rješenje optimizacijskog problema: naći točke integracije tako da egzaktno integriramo polinom što većeg stupnja sa što manje čvorova. Rezultat su formule visoke točnosti, koje se lagano implementiraju, i imaju vrlo mali broj izvrednjavanja podintegralne funkcije. Cijenu smo platili time što ocjena greške zahtijeva vrlo glatku funkciju, ali također i time što upotreba takvih formula na "finijoj" mreži zahtijeva ponovno računanje funkcije u drugim čvorovima, koji s čvorovima formule nižeg reda nemaju ništa zajedničko. Kod profinjavanja mreže čvorova za formule Newton–Côtesovog tipa (na primjer, raspolavljanjem h), naprotiv, jedan dio čvorova ostaje zajednički, pa već izračunate funkcijske vrijednosti možemo iskoristiti (kao u Rombergovom algoritmu).

11.5.2. Druge Gaussove integracione formule

U praksi se često javljaju specijalni integrali koji uključuju težinske funkcije poput e^{-x}, e^{-x^2} i mnoge druge, na specijalnim intervalima, često neograničenim. Jednostavnom linearnom supstitucijom nije moguće takve intervale i/ili težinske funkcije prebaciti na interval $(-1, 1)$ i jediničnu težinsku funkciju — situaciju u kojoj možemo primijeniti Gauss–Legendreove formule.

Alternativa je iskoristiti odgovarajuće Gaussove formule s "prirodnom" težinskom funkcijom. Iz prethodnog odjeljka znamo da za čvorove integracije treba uzeti nultočke funkcije $\omega_n(x) = (x - x_1)\cdots(x - x_n)$, s tim da vrijede relacije ortogonalnosti (11.5.3). Težine w_i onda možemo odrediti rješavanjem linearnog sistema, a

možda u specijalnim slučajevima možemo doći i do eksplicitnih formula, kao što smo to učinili u slučaju Gauss–Legendreovih formula. Postavlja se pitanje da li možemo doći do formula za polinome koji su ortogonalni (obzirom na težinsku funkciju w) na polinome nižeg stupnja, uključivo i ostale formule na koje smo se oslanjali, poput tročlane rekurzije i slično (v. lema 11.5.2.).

U mnogim važnim slučajevima, ali ne i uvijek, moguće je analitički doći do formula sličnim onima u slučaju Gauss–Legendreove integracije. U drugim slučajevima, koji nisu pokriveni egzaktnim formulama, u principu je moguće generirati ortogonalne polinome i numerički. Poznati postupci (Stieltjesov i Čebiševljev algoritam) ne pokrivaju, međutim, sve moguće situacije, tj. nisu uvijek numerički stabilni, što ostavlja postora za daljnja istraživanja. Slučajevi tzv. **klasičnih ortogonalnih polinoma** uglavnom se mogu karakterizirati na osnovu sljedeća dva teorema, od kojih je prvi egzistencijalni, i vezan uz teoriju rubnih problema za obične diferencijalne jednadžbe.

Teorem 11.5.3. (Generalizirana Rodriguesova formula)

Na otvorenom intervalu (a, b) postoji, do na multiplikativnu konstantu, jedinstvena funkcija $U_n(x)$ koja zadovoljava diferencijalnu jednadžbu

$$D^{n+1}\left(\frac{1}{w(x)}\, D^n U_n(x)\right) = 0$$

i rubne uvjete

$$U_n(a) = DU_n(a) = \cdots = D^{n-1}U_n(a) = 0,$$
$$U_n(b) = DU_n(b) = \cdots = D^{n-1}U_n(b) = 0.$$

Ovdje opet koristimo oznaku D za operator deriviranja funkcije f jedne varijable, kad je iz konteksta očito po kojoj varijabli se derivira, jer ta oznaka znatno skraćuje zapis nekih dugih formula. Onda n-tu derivaciju funkcije f u točki x možemo pisati u bilo kojem od sljedeća tri oblika

$$D^n f(x) = \frac{d^n}{dx^n}\, f(x) = f^{(n)}(x).$$

Budući da nas interesiraju rješenja koja se mogu eksplicitno konstruirati, nećemo dokazivati ovaj teorem. U svakom konkretnom slučaju, za zadane a, b i $w(x)$, konstruirat ćemo funkciju U_n formulom. Napomenimo još da teorem 11.5.3. vrijedi i na neograničenim i poluograničenim intervalima, tj. u slučajevima $a = -\infty$ i/ili $b = \infty$.

Funkcije U_n iz prethodnog teorema generiraju familiju ortogonalnih polinoma na (a, b) s težinskom funkcijom w.

Teorem 11.5.4. *Uz pretpostavke teorema* 11.5.3., *funkcije*

$$p_n(x) = \frac{1}{w(x)}\, D^n U_n(x)$$

su polinomi stupnja n koji su ortogonalni na sve polinome nižeg stupnja na intervalu (a, b) *obzirom na težinsku funkciju* $w(x)$, *tj. vrijedi*

$$\int_a^b w(x)\, p_n(x)\, x^k\, dx = 0, \quad \text{za} \quad k = 0, 1, \ldots, n-1.$$

Dokaz:

Funkcija p_n je očito polinom stupnja n, jer je $D^{n+1} p_n(x) = 0$. Da dokažemo ortogonalnost, pretpostavimo da je $n \geq 1$. Za $k = 0$ imamo odmah po Newton–Lebnitzovoj formuli

$$\int_a^b w(x)\, p_n(x)\, dx = \int_a^b D^n U_n(x)\, dx = (n \geq 1) = D^{n-1} U_n(x)\Big|_a^b = 0,$$

zbog rubnih uvjeta $D^{n-1} U_n(a) = D^{n-1} U_n(b) = 0$.

Za $1 \leq k \leq n-1$, integriramo parcijalno k puta i iskoristimo opet rubne uvjete koje zadovoljava funkcija U_n. Dobivamo redom

$$\begin{aligned}
\int_a^b w(x) p_n(x)\, x^k\, dx &= \int_a^b x^k\, D^n U_n(x)\, dx \\
&= \underbrace{x^k\, D^{n-1} U_n(x)\Big|_a^b}_{=0} - k \int_a^b x^{k-1}\, D^{n-1} U_n(x)\, dx \\
&= -k \left(\underbrace{x^{k-1}\, D^{n-2} U_n(x)\Big|_a^b}_{=0} - (k-1) \int_a^b x^{k-2}\, D^{n-2} U_n(x)\, dx \right) \\
&= \cdots = (-1)^{k-1} k(k-1) \cdots 2 \left(\underbrace{x\, D^{n-k} U_n(x)\Big|_a^b}_{=0} - \int_a^b D^{n-k} U_n(x)\, dx \right) \\
&= (-1)^k k(k-1) \cdots 2 \cdot 1 \left(\underbrace{D^{n-k-1} U_n(x)\Big|_a^b}_{=0} \right) = 0,
\end{aligned}$$

jer je $n-k-1 \geq 0$. Primijetimo da smo za dokaz ortogonalnosti iskoristili sve rubne uvjete na funkciju U_n. ■

Ovaj teorem u mnogim slučajevima omogućava efektivnu konstrukciju ortogonalnih polinoma.

Primjer 11.5.4. *Neka je $w(x) = 1$ na intervalu $(-1, 1)$. Nađimo pripadne ortogonalne polinome. Prema teoremu* 11.5.3.*, prvi korak je rješavanje diferencijalne jednadžbe*

$$D^{n+1}(D^n U_n(x)) = D^{2n+1} U_n(x) = 0,$$

uz rubne uvjete

$$U_n(\pm 1) = DU_n(\pm 1) = \cdots = D^{n-1} U_n(\pm 1) = 0.$$

Polinom $2n$-tog stupnja koji se poništava u krajevima mora, zbog simetrije, biti oblika $U_n(x) = C_n(x^2 - 1)^n$, gdje je C_n proizvoljna multiplikativna konstanta (različita od nule). Tradicionalno, konstanta C_n uzima se u obliku

$$C_n = \frac{1}{2^n n!}.$$

Pripadni ortogonalni polinomi su tada, prema teoremu 11.5.4.*, dani formulom*

$$P_n(x) = \frac{1}{2^n n!} D^n (x^2 - 1)^n,$$

tj. dobivamo, očekivano, Legendreove polinome.

Zadatak 11.5.4. *Pokažite da je multiplikativna konstanta C_n odabrana tako da vrijedi $P_n(1) = 1$, za svako n. Također, pokažite da vrijedi $|P_n(x)| \leq 1$, za svaki $x \in [-1, 1]$ i svaki $n \geq 0$. To znači da P_n dostiže ekstreme u rubovima intervala, što je dodatno opravdanje za izbor normalizacije, jer je $\|P_n\|_\infty = 1$ na $[-1, 1]$.*

Primjer 11.5.5. *Neka je $w(x) = e^{-\alpha x}$ na intervalu $(0, \infty)$, za neki $\alpha > 0$. Nađimo pripadne ortogonalne polinome. Prema teoremu* 11.5.3.*, trebamo prvo riješiti diferencijalnu jednadžbu*

$$D^{n+1}(e^{\alpha x} D^n U_n(x)) = 0,$$

uz rubne uvjete

$$U_n(0) = DU_n(0) = \cdots = D^{n-1} U_n(0) = 0,$$
$$U_n(\infty) = DU_n(\infty) = \cdots = D^{n-1} U_n(\infty) = 0.$$

Očito je rješenje oblika

$$U_n(x) = e^{-\alpha x}(c_0 + c_1 x + \cdots + c_n x^n) + d_0 + d_1 x + \cdots + d_{n-1} x^{n-1}.$$

Rubni uvjet u točki ∞ povlači $d_0 = \cdots = d_{n-1} = 0$, a rubni uvjet u točki 0 povlači $c_0 = \cdots = c_{n-1} = 0$, pa je

$$U_n(x) = C_n x^n e^{-\alpha x}.$$

Polinomi za koje je $\alpha = 1$ i $C_n = 1$ zovu se tradicionalno **Laguerreovi polinomi***, u oznaci $\widetilde{L}_n$. Njihova Rodriguesova formula je dakle*

$$\widetilde{L}_n(x) = e^x D^n (x^n e^{-x}).$$

U općem slučaju, za $\alpha \neq 1$, uz $C_n = 1$, lagano vidimo da je $p_n(x) = \widetilde{L}_n(\alpha x)$. Tada vrijede relacije ortogonalnosti

$$\int_0^\infty e^{-\alpha x}\, \widetilde{L}_m(\alpha x)\, \widetilde{L}_n(\alpha x)\, dx = 0, \quad m \neq n.$$

Napomenimo još da oznaku L_n koristimo za ortonormirane Laguerreove polinome. Njih dobivamo izborom normalizacione konstante $C_n = 1/n!$, pa je $\widetilde{L}_n(x) = n!\, L_n(x)$.

Primjer 11.5.6. *Neka je $w(x) = e^{-\alpha^2 x^2}$ na intervalu $(-\infty, \infty)$, za neki $\alpha \neq 0$. Nađimo pripadne ortogonalne polinome. Prema teoremu* 11.5.3.*, trebamo prvo riješiti diferencijalnu jednadžbu*

$$D^{n+1}(e^{\alpha^2 x^2}\, D^n U_n(x)) = 0,$$

uz rubne uvjete

$$U_n(\pm\infty) = DU_n(\pm\infty) = \cdots = D^{n-1}U_n(\pm\infty) = 0.$$

Lagano pogodimo da je

$$U_n(x) = C_n\, e^{-\alpha^2 x^2}.$$

Odaberemo li $\alpha^2 = 1$ i multiplikativnu konstantu $C_n = (-1)^n$, dolazimo do klasičnih polinoma, koji nose ime **Hermiteovi polinomi***, u oznaci H_n, s Rodriguesovom formulom*

$$H_n(x) = (-1)^n e^{x^2}\, D^n(e^{-x^2}).$$

U općem slučaju, za $\alpha^2 \neq 1$, uz $C_n = (-\alpha)^n$, lagano vidimo da su polinomi koje tražimo oblika

$$p_n(x) = H_n(\alpha x) = (-\alpha)^n\, e^{\alpha^2 x^2}\, D^n(e^{-\alpha^2 x^2}).$$

Pripadne relacije ortogonalnosti su

$$\int_{-\infty}^\infty e^{-\alpha^2 x^2}\, H_m(\alpha x)\, H_n(\alpha x)\, dx = 0, \quad m \neq n.$$

U literaturi se ponekad može naći još jedna definicija za klasične Hermiteove polinome, koja odgovara izboru $\alpha^2 = 1/2$, uz $C_n = (-1)^n$.

Svi ortogonalni polinomi zadovoljavaju tročlane rekuzije (v. izvod Stieltjesovog algoritma uz metodu najmanjih kvadrata). Za Laguerreove i Hermiteove polinome mogu se analitički izračunati koeficijenti u rekurziji, postupkom koji je vrlo sličan onom kojeg smo u detalje proveli u slučaju Legendreovih polinoma. Primijetimo, također, da i Čebiševljevi polinomi prve vrste zadovoljavaju relacije ortogonalnosti i tročlanu rekurziju, i da smo taj slučaj do kraja proučili. Kako su čvorovi Gaussove

formule integracije reda n nultočke odgovarajućeg ortogonalnog polinoma p_n, preostaje još samo izračunati težine w_i po formuli (11.5.5). Sasvim općenito, može se pokazati da vrijedi

$$w_i = \int_a^b w(x)\,\ell_i(x)\,dx = \frac{1}{p_n'(x_i)} \int_a^b w(x)\,\frac{p_n(x)}{x - x_i}\,dx,$$

gdje su ℓ_i polinomi Lagrangeove baze, i te integrale treba naći egzaktno. Formule za težine mogu se dobiti za cijeli niz klasičnih ortogonalnih polinoma, ali njihovo računanje ovisi o specijalnim svojstvima, posebnim rekurzijama i identitetima oblika Christoffel–Darbouxovog. Obzirom na duljinu ovih izvoda, zadovoljimo se s kratkim opisom nekoliko najpoznatijih Gaussovih formula.

Gauss–Laguerreove formule

Formule oblika

$$\int_0^\infty e^{-x}\,f(x)\,dx \approx \sum_{i=1}^n w_i f(x_i)$$

zovu se **Gauss–Laguerreove formule**. Čvorovi integracije su nultočke polinoma $\widetilde{L}_n$ definiranih Rodriguesovom formulom

$$\widetilde{L}_n(x) = e^x\,\frac{d^n}{dx^n}(x^n e^{-x}),$$

a težine u Gaussovoj formuli su

$$\begin{aligned} w_i &= \frac{[(n-1)!]^2 x_i}{[n\widetilde{L}_{n-1}(x_i)]^2} = \frac{(n!)^2 x_i}{[\widetilde{L}_{n+1}(x_i)]^2} \\ &= -\frac{[(n-1)!]^2}{\widetilde{L}_n'(x_i)\,\widetilde{L}_{n-1}(x_i)} = \frac{(n!)^2}{\widetilde{L}_n'(x_i)\,\widetilde{L}_{n+1}(x_i)} \\ &= \frac{(n!)^2}{x_i[\widetilde{L}_n'(x_i)]^2}. \end{aligned}$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{(n!)^2}{(2n)!}\,f^{(2n)}(\xi), \quad \xi \in (0,\infty).$$

Gauss–Hermiteove formule

Formule oblika

$$\int_{-\infty}^\infty e^{-x^2}\,f(x)\,dx \approx \sum_{i=1}^n w_i f(x_i)$$

zovu se **Gauss–Hermiteove formule**. Čvorovi integracije su nultočke polinoma H_n definiranih Rodriguesovom formulom

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}),$$

a težine u Gaussovoj formuli su

$$\begin{aligned} w_i &= \frac{2^{n-1}(n-1)!\sqrt{\pi}}{n[H_{n-1}(x_i)]^2} = \frac{2^{n+1}n!\sqrt{\pi}}{[H_{n+1}(x_i)]^2} \\ &= \frac{2^n(n-1)!\sqrt{\pi}}{H_n'(x_i)\,H_{n-1}(x_i)} = -\frac{2^{n+1}n!\sqrt{\pi}}{H_n'(x_i)\,H_{n+1}(x_i)} \\ &= \frac{2^{n+1}n!\sqrt{\pi}}{[H_n'(x_i)]^2}. \end{aligned}$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{n!\sqrt{\pi}}{2^n(2n)!} f^{(2n)}(\xi), \quad \xi \in (-\infty, \infty).$$

Gauss–Čebiševljeve formule

Formule oblika

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

zovu se **Gauss–Čebiševljeve formule**. Čvorovi integracije su nultočke Čebiševljevih polinoma $T_n(x) = \cos(n \arccos(x))$. Izuzetno, te se nultočke mogu eksplicitno izračunati

$$x_i = \cos\left(\frac{(2i-1)\pi}{2n}\right).$$

Sve težine w_i su jednake

$$w_i = \frac{\pi}{n}.$$

Greška kod numeričke integracije dana je formulom

$$E_n(f) = \frac{\pi}{2^{2n-1}(2n)!} f^{(2n)}(\xi), \quad \xi \in (-1, 1).$$

Zadatak 11.5.5. *Neka je težinska funkcija $w(x) = (x-a)^\alpha (b-x)^\beta$ na intervalu (a, b), gdje su $\alpha > -1$ i $\beta > -1$. Nađite funkciju U_r i napišite Rodriguesovu formulu! Pridruženi ortogonalni polinomi zovu se* **Jacobijevi polinomi**. *Legendreovi i Čebiševljevi polinomi specijalni su slučaj.*

Pomoću Gaussovih formula možemo jednostavno računati neke određene integrale analitički, kao što se vidi iz sljedećih primjera.

Primjer 11.5.7. *Ako Gauss–Laguerreovom formulom reda $n = 1$ računamo integral*

$$\int_0^\infty e^{-x}\,dx,$$

imamo približnu formulu

$$\int_0^\infty e^{-x} f(x)\,dx \approx f(1),$$

budući da je $\widetilde{L}_1(x) = 1 - x$, pa je $x_1 = 1$ i $w_1 = 1/[\widetilde{L}'(1)]^2 = 1$. Kako formula egzaktno integrira konstante, za $f(x) = 1$ imamo

$$\int_0^\infty e^{-x}\,dx = f(1) = 1.$$

Slično, za $f(x) = ax + b$, budući da formula egzaktno integrira i linearne funkcije,

$$\int_0^\infty e^{-x}\,(ax + b)\,dx = f(1) = a + b.$$

Primjer 11.5.8. *Ako Gauss–Čebiševljevom formulom računamo*

$$\int_{-1}^{1} \frac{x^4}{\sqrt{1 - x^2}}\,dx$$

zgodno je upotrijebiti formulu Gauss–Čebiševa reda 3, koja zahtijeva nultočke polinoma $T_3(x) = 4x^3 - 3x$, a to su $x_1 = 0$, $x_{2,3} = \pm\sqrt{3}/2$. Formula vodi na egzaktan rezultat

$$\int_{-1}^{1} \frac{x^4}{\sqrt{1 - x^2}}\,dx = \frac{\pi}{3}\left(0 + \frac{9}{16} + \frac{9}{16}\right) = \frac{3\pi}{8}.$$

12. Metode za rješavanje običnih diferencijalnih jednadžbi

12.1. Uvod

Promatrat ćemo inicijalni (početni ili Cauchyjev) problem za običnu diferencijalnu jednadžbu

$$\frac{dy}{dx} = f(y, x), \quad y(x_0) = y_0, \tag{12.1.1}$$

pri čemu pretpostavljamo da je funkcija $f(y, x)$ neprekidna na vremenskom intervalu $x_0 \leq x \leq b$ i za $-\infty < y < \infty$.

Ideja numeričkog rješavanja je zamjena neprekidnog rješenja u vremenskom intervalu $[x_0, b]$ približnim rješenjima u konačnom skupu točaka $\{x_0, x_1, \ldots, x_N\}$. Obzirom na to iz koliko prethodnih točaka računamo novu aproksimaciju y_i u točki x_i, razlikujemo

(a) jednokoračne metode (takve su na primjer Runge–Kutta metode) – aproksimacija u y_i računa se samo iz vrijednosti aproksimacije u x_{i-1};

(b) višekoračne metode (takvi su na primjer prediktor–korektor parovi: Adams–Bashfort metoda kao prediktor, Adams–Moulton kao korektor) – aproksimacija u y_i računa se iz vrijednosti u više prethodnih točaka $x_k, x_{k+1}, \ldots, x_{i-1}$.

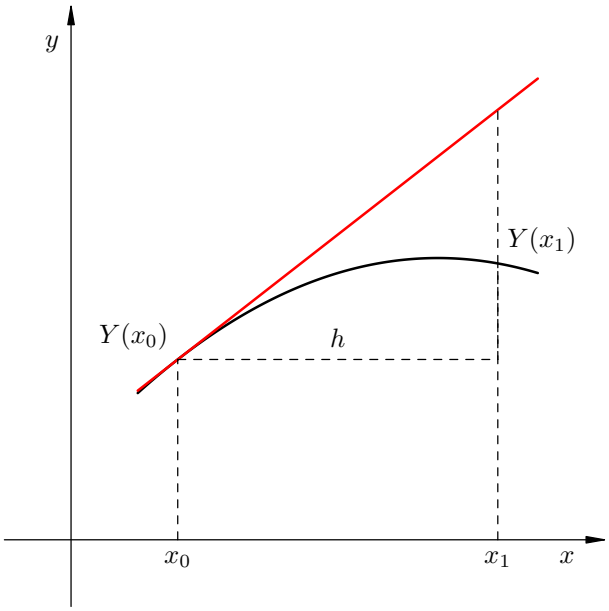
Označimo s $Y(x)$ pravo rješenje diferencijalne jednadžbe, a s $y(x)$ aproksimaciju rješenja. Bez smanjenja općenitosti, za izvod metoda možemo koristiti da su točke x_j ekvidistantne, tj. da vrijedi $h = x_j - x_{j-1}$, odnosno

$$x_j = x_0 + jh, \quad j = 0, 1, \ldots,$$

12.2. Runge–Kutta metode

Najjednostavnija metoda iz obitelji Runge–Kutta metoda je Runge–Kutta metoda prvog reda, poznatija po imenom Eulerova metoda. Izvod Eulerove metode

može se napraviti na (barem) dva načina. Prvi je lako shvatljiv, jer ima geometrijsku pozadinu, a generalizacija drugog načina dat će nam ideju za izvod Runge–Kutta metoda viših redova.



Povucimo u točki x_0 tangentu na pravo rješenje $Y(x)$. Koristeći vrijednost te tangente u x_1 dobit ćemo željenu aproksimaciju u x_1. Imamo

$$\frac{\Delta Y}{h} = \frac{Y(x_1) - Y(x_0)}{h} \approx Y'(x_0) = f(x_0, y_0),$$

ili na drugi način zapisano:

$$Y(x_1) - Y(x_0) \approx hY'(x_0) = hf(x_0, y_0).$$

Jasno je da će se na sličan način dobivati i aproksimacije za rješenja u točkama $x_2, \ldots, x_N$.

Dakle, Eulerova metoda glasi:

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0, 1, \ldots,$$

Također je jasno da ako su točke x_j neekvidistantne, u prethodnoj formuli umjesto h treba pisati h_n – varijabilna duljina koraka.

Eulerova metoda se, kao što smo već rekli može izvesti i na drugi način. Funkcija Y razvije se u Taylorov red (do prvog člana) oko točke x_n:

$$Y(x) = Y(x_n) + Y'(x_n)(x - x_n) + Y''(\xi_n)\frac{(x - x_n)^2}{2},$$

pri čemu točka ξ_n se nalazi između x_n i x. Uvrštavanjem točke x_{n+1} u prethodnu formulu dobivamo:

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + Y''(\xi_n)\frac{h^2}{2}, \quad x_n \le \xi_n \le x_{n+1}.$$

Primijetimo da odavde slijedi da je greška aproksimacije u jednom koraku Eulerove metode $\mathcal{O}(h^2)$. Primijenimo li više koraka metode, greška se "nakupi" do $\mathcal{O}(h)$ – dakle maksimalan red metode je 1 (eksponent od h).

Općenito, Runge–Kutta metode imaju oblik

$$y_{n+1} = y_n + \sum_{i=1}^{p} w_i k_i \tag{12.2.1}$$

gdje su w_i konstante, a k_i je

$$k_i = h_n f(x_n + \alpha_i h_n, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j),$$

pri čemu je $\alpha_1 = 0$. Ostali w_i, α_i, β_{ij} određuju se tako da formula što bolje aproksimira rješenje diferencijalne jednadžbe. Razvijemo li lijevu i desnu stranu u (12.2.1) u Taylorov red oko točke x_n i ako izjednačavamo prvih m koeficijenata uz h_n^r, dobivamo da ne možemo izjednačiti više od $m = p$ prvih koeficijenata. Rezultirajuća formula zove se RK (Runge–Kutta) metoda reda m. Uglavnom se promatraju RK metode do reda $m = 4$ zbog toga što je

broj računanja funkcije	1	2	3	4	5	6	7	8
maksimalan red metode	1	2	3	4	4	5	6	6

Za fiksan m može postojati više RK metoda. Tako za $m = 2$ dobivamo uvjete

$$w_1 + w_2 = 1, \quad \alpha_2 w_2 = \frac{1}{2}, \quad \beta_{21} = \alpha_2.$$

Interesantne RK–2 metode imaju $\alpha_2 = \frac{1}{2}, \frac{2}{3}$ i 1. Mi ćemo koristiti samo onu za $\alpha = 1$, koja glasi

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2), \quad n = 0, 1, 2, \ldots,$$

gdje je

$$k_1 = h_n f(x_n, y_n)$$
$$k_2 = h_n f(x_n + h_n, y_n + k_1).$$

Pogreška za jedan korak RK–2 metode je $\mathcal{O}(h^3)$, dok je za više koraka $\mathcal{O}(h^2)$.

Na sličan način, može se pokazati da postoji više RK metoda četvrtog reda, od kojih je najpoznatija

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad n = 0, 1, 2, \ldots,$$

gdje je

$$\begin{aligned} k_1 &= h_n f(x_n, y_n) \\ k_2 &= h_n f(x_n + \frac{1}{2}h_n, y_n + \frac{1}{2}k_1) \\ k_3 &= h_n f(x_n + \frac{1}{2}h_n, y_n + \frac{1}{2}k_2) \\ k_4 &= h_n f(x_n + h_n, y_n + k_3). \end{aligned}$$

Pogreška za jedan korak RK–4 metode je $\mathcal{O}(h^5)$, dok je za više koraka $\mathcal{O}(h^4)$.

12.2.1. Varijabilni korak za Runge–Kutta metode

Slično kao kod Rombergovog algoritma, promatra se RK metoda s korakom h i $h/2$. Nakon toga, napravi se 1 korak RK metode s korakom h i 2 koraka metode s korakom $h/2$ (da se dođe u istu točku). Ideja je napraviti slijedeće:

(a) ako se vrijednosti tako dobivenih aproksimacija "dosta razlikuju", onda se korak smanji na $h = h/2$ i procedura se ponovi za $h/2$ i $h/4$ (oprez od neprekidnog smanjivanja koraka!!);

(b) ako su vrijednosti tako dobivenih aproksimacije bliske, prihvaća se tako izračunata vrijednost, a iz ocjene pogreške predviđa se h za slijedeći korak metode.

Korist od varijabilnog koraka je da se za neke funkcije s puno manje računanja može dobiti rješenje na zadovoljavajuću točnost, nego kod fiksnog koraka (koji ne kontrolira točnost).

12.2.2. Runge–Kutta metode za sustave jednadžbi

Runge–Kutta metode mogu se koristiti za približno rješavanje sistema diferencijalnih jednadžbi i za približno rješavanje diferencijalnih jednadžbi viših redova.

Treba samo primijetiti da u slučaju sistema diferencijalnih jednadžbi, veličine y_n, k_i i $f(x, y)$ imaju ulogu vektora, a ostale veličine su skalari. Može se pokazati da se jednostavnim zamjenama varijabli svaka diferencijalna jednadžba višeg reda može svesti na sistem prvog reda.

Primjer 12.2.1. *Svedite na sistem diferencijalnih jednadžbi prvog reda i riješite RK–2 metodom s korakom* $h = 0.1$ *diferencijalnu jednadžbu*

$$y'' + 2y' + 3x = 5, \quad y(0) = 1, \quad y'(0) = 2$$

u točki $x = 0.1$.

Označimo s $z = y'$. *Deriviranjem i uvrštavanjem u polaznu jednadžbu dobivamo sistem diferencijalnih jednadžbi*

$$\begin{aligned} y' &= z \\ z' &= 5 - 2z - 3x \end{aligned}$$

uz početne uvjete $y(0) = 1$, $z(0) = 2$. *Rješenje zadatka dobivamo odmah u prvom koraku*

$$\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} + \frac{1}{2}\left(\begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} + \begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} \right).$$

Uočimo da je

$$\begin{bmatrix} y' \\ z' \end{bmatrix} = \begin{bmatrix} z \\ 5 - 2z - 3x \end{bmatrix}, \quad \begin{bmatrix} y(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Odatle, po formuli za RK–2 slijedi

$$\begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = 0.1 \begin{bmatrix} 2 \\ 5 - 2 \cdot 2 - 3 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}.$$

Jednako tako, imamo

$$\begin{bmatrix} y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} k_{11} \\ k_{12} \end{bmatrix} = \begin{bmatrix} 1.2 \\ 2.1 \end{bmatrix},$$

odakle izračunavamo k_2

$$\begin{bmatrix} k_{21} \\ k_{22} \end{bmatrix} = 0.1 \begin{bmatrix} 2.1 \\ 5 - 2 \cdot 2.1 - 3 \cdot 0.1 \end{bmatrix} = \begin{bmatrix} 0.21 \\ 0.05 \end{bmatrix}.$$

Sve zajedno daje

$$\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \frac{1}{2}\left(\begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix} + \begin{bmatrix} 0.21 \\ 0.05 \end{bmatrix} \right) = \begin{bmatrix} 1.205 \\ 2.075 \end{bmatrix},$$

što znači $y(0.1) \approx 1.205$ *i* $z(0.1) = y'(0.1) \approx 2.075$.

12.3. Višekoračne metode

Koriste se jer zahtijevaju manje izvrednjavanja funkcije nego RK metode. Na primjer, jedan prediktor–korektor (PC) par (samo ime mu kaže i ulogu) reda 4 je:

prediktor

$$y_{n+1}^{(0)} = y_n + \frac{h}{24}(55f(x_n, y_n) - 59f(x_{n-1}, y_{n-1}) + 37f(x_{n-2}, y_{n-2}) - 9f(x_{n-3}, y_{n-3}))$$

korektor

$$y_{n+1}^{(j)} = y_n + \frac{h}{24}(9f(x_{n+1}, y_{n+1}^{(j)}) - 19f(x_n, y_n) - 5f(x_{n-1}, y_{n-1}) + f(x_{n-2}, y_{n-2})).$$

Ako se korektor koristi jednom (očito je moguća i višestruka upotreba), onda se za novi korak računaju točno dvije funkcijske vrijednosti: $f(x_n, y_n)$ i $f(x_{n+1}, y_{n+1}^{(j)})$, a sve ostale su već morale biti izračunane u prethodnim koracima. (Uočite da odgovarajuća RK–4 ima 4 izvrednjavanja funkcije.) Neugoda korištenja višekoračnih metoda su točke potrebne za start metode. U našem slučaju PC para reda 4 za start metode potrebno je znanje funkcijskih vrijednosti u točkama x_0, x_1, x_2 i x_3.

12.4. Krute (stiff) diferencijalne jednadžbe

Najvažnija upotreba višekoračnih metoda je rješavanje (sistema) krutih diferencijalnih jednadžbi. Najpoznatija metoda poznata je kao Gearova metoda (po Williamu C. Gear-u) i sastoji se od Adams prediktora i korektora varijabilnog reda i varijabilnog koraka.

Za diferencijalnu jednadžbu reći ćemo da je kruta, ako mala perturbacija početnih uvjeta dovede do velike perturbacije u rješenju problema.

Primjer 12.4.1. *Zadana je diferencijalna jednadžba*

$$y' = 10(y - x) - 9, \quad y(0) = 1.$$

Opće rješenje ove diferencijalne jednadžbe je

$$y(x) = ce^{10x} + x + 1.$$

Partikularno rješenje za ovaj početni uvjet je

$$y = x + 1.$$

Ako malo perturbiramo početni uvjet na $y(0) = 1 + \varepsilon$, onda je partikularno rješenje te jednadžbe

$$y = \varepsilon e^{10x} + x + 1.$$

Primijetite da je prvi faktor s desne strane dominirajući za malo veće x. Što će se u računalu dogoditi s tom jednadžbom? Greške zaokruživanja i pogreške u svakom koraku metode djelovat će kao perturbacija početnih uvjeta, pa će se RK i slične metode "raspasti" – rezultati neće imati nikakve veze s pravim rješenjem.

13. Rubni problem za obične diferencijalne jednadžbe

Iz tradicionalnih ralzoga, kod rubnih problema za diferencijalne jednadžbe varijable se označavaju s x (prostorna) i t (vremenska dimenzija).

Pretpostavimo da je zadana diferencijalna jednadžba drugog reda

$$\ddot{x} - p(t)\dot{x} - r(t)x = q(t) \tag{13.0.1}$$

uz rubne uvjete

$$x(a) = x_a$$
$$x(b) = x_b.$$

13.1. Egizstencija i jedinstvenost rješenja

Za rubni problem nije osigurana niti egzistencija niti jedinstvenost rješenja. Pokažimo to na jednostavnom primjeru koji znamo egzaktno riješiti.

Primjer 13.1.1. *Zadana je diferencijalna jednadžba*

$$\ddot{x} + x = 0$$

uz rubne uvjete

$$x(0) = 1$$
$$x(\pi) = 0.$$

Opće rješenje zadane ODJ je

$$x(t) = a\cos t + b\sin t.$$

Uvrštavanjem rubnih uvjeta dobivamo kontradikciju,

$$x(0) = 1 = \quad a$$
$$x(\pi) = 0 = -a,$$

pa očito ovaj problem nema rješenja. Ako rubne uvjete promijenimo u

$$x(0) = 0$$
$$x(\pi) = 0$$

uočimo da su rješenja ovog rubnog problema oblika

$$x(t) = b \sin t, \quad b \in \mathbb{R}.$$

13.2. Metoda gađanja za linearne diferencijalne jednadžbe 2. reda

Svaku diferencijalnu jednadžbu drugog reda možemo zapisati kao sistem jednadžbi prvog reda, tako da se uvede $\dot{x} = y$. Tada jednadžba (13.0.1) glasi

$$\dot{x} = y$$
$$\dot{y} = p(t)y + r(t)x + q(t)$$

a odgovarajući rubni uvjeti su

$$x(a) = x_a$$
$$y(a) = \dot{x}(a) = ???.$$

Očito je da taj rubni uvjet treba izvući iz informacije $x(b) = x_b$.

Pretpostavimo da prvi puta za $\dot{x}(a)$ stavimo

$$\dot{x}(a) = s_1, \quad s_1 \in \mathbb{R} \quad \text{proizvoljan}$$

i da uz tu pretpostavku riješimo inicijalni problem i njegovo rješenje označimo s $v(t)$. Drugi puta za $\dot{x}(a)$ stavimo

$$\dot{x}(a) = s_2, \quad s_2 \in \mathbb{R} \quad \text{proizvoljan}$$

i da uz tu pretpostavku riješimo inicijalni problem i njegovo rješenje označimo s $w(t)$. Zbog linearnosti jednadžbe, linearna kombinacija rješenja v i w je opet rješenje, pa odaberimo takvu kombinaciju da vrijedi $x(b) = x_b$. Uočimo da oba rješenja poštuju da je $x(a) = x_a$, pa linearnu kombinaciju rješenja možemo pisati u obliku

$$x(t) = \lambda v(t) + (1 - \lambda) w(t)$$

Ako uvrstimo

$$x_b = x(b) = \lambda v(b) + (1 - \lambda) w(b)$$

dobivamo

$$\lambda = \frac{x_b - w(b)}{v(b) - w(b)}$$

i time je zadovoljen rubni uvjet $x(b) = x_b$.

13.3. Nelinearna metoda gađanja

Ako imamo nelinearnu jednadžbu 2. reda, na pr.:

$$\ddot{x} - (1 - \frac{t}{2})\dot{x}x = t \tag{13.3.1}$$

uz rubne uvjete

$$\begin{aligned} x(1) &= \quad 2 \\ x(3) &= -1 \end{aligned}$$

zapišimo je, također u obliku sistema ODJ. Ponovno, kao i kod linearne metode gađanja, pretpostavimo da je

$$\begin{aligned} \dot{x}(1) &= s_1 \quad \text{rješenje} \quad v(t) \\ \dot{x}(1) &= s_2 \quad \text{rješenje} \quad w(t). \end{aligned}$$

Zbog nelinearnosti više ne vrijedi argument linearne kombinacije rješenja, pa se do pravog rješenja može stići iterativno.

U točki 3 rješenja su za $(s_1, v(3))$ i $(s_2, w(3))$, a mi želimo da bude $(s_t, -1)$. Nakon toga, ako $v(3)$ ili $w(3)$ nije -1, napravi se linearna interpolacija kroz točke $(s_1, v(3))$ i $(s_2, w(3))$

$$y - v(3) = \frac{w(3) - v(3)}{s_2 - s_1}(s - s_1).$$

Želimo da bude $y = -1$, pa odatle izračunamo s, što je nova aproksimacija za s_t, tj. uzme se $\dot{x}(1) = s$. Ovaj proces, ako se ponovi, može, ali i ne mora konvergirati.

13.4. Metoda konačnih razlika

Pretpostavimo da je ponovno zadana ODJ (13.0.1) uz rubne uvjete

$$\begin{aligned} x(a) = x_a &= \text{ oznaka } = x_0 \\ x(b) = x_b &= \text{ oznaka } = x_n. \end{aligned}$$

Izaberimo mrežu ekvidistantnih točaka $t_0 = a, t_1, \ldots, t_n = b$ u kojima želimo aproksimirati rješenje rubnog problema. Označimo te aproksimacije s $x_0, x_1, \ldots, x_n$.

Prva derivacija aproksimira se simetričnom (centralnom) razlikom

$$\dot{x} = \frac{dx}{dt}\bigg|_{t=t_i} = \frac{x_{i+1} - x_{i-1}}{2h}$$

gdje je $h = (b - a)/n$.

Objasnimo zašto je simetrična razlika bolja nego obična podijeljena razlika. Ako su točke t_i ekvidistantne, onda Taylorov razvoj u okolini t_i glasi

$$x(t) = x(t_i) + \dot{x}(t_i)(t - t_i) + \ddot{x}(t_i)\frac{(t-t_i)^2}{2!} + x^{(3)}(\xi)\frac{(t-t_i)^3}{3!}, \tag{13.4.1}$$

gdje je ξ neka točka između t i t_i. Uvrštavanjem u taj razvoj redom $t = t_{i+1} = t_i + h$, a zatim $t = t_{i-1} = t_i - h$ dobivamo

$$\begin{aligned} x(t_{i+1}) &= x(t_i) + \dot{x}(t_i)h + \ddot{x}(t_i)\frac{h^2}{2!} + x^{(3)}(\xi)\frac{h^3}{3!} \\ x(t_{i-1}) &= x(t_i) - \dot{x}(t_i)h + \ddot{x}(t_i)\frac{h^2}{2!} - x^{(3)}(\xi)\frac{h^3}{3!}, \end{aligned}$$

pa oduzimanjem i dijeljenjem s $2h$ dobivamo

$$\frac{x(t_{i+1}) - x(t_{i-1})}{2h} = \dot{x}(t_i) + \mathcal{O}(h^2).$$

Da smo u istu formulu (13.4.1) uvrstili samo t_i i t_{i+1}, dobili bismo

$$\frac{x(t_{i+1}) - x(t_i)}{h} = \dot{x}(t_i) + \mathcal{O}(h),$$

što je za red veličine lošije!

Drugu derivaciju aproksimiramo drugom podijeljenom razlikom (podijeljena razlika dvije susjedne prve podijeljene razlike).

$$\begin{aligned} \left.\frac{dx}{dt}\right|_{t=t_i} &= \frac{x_{i+1} - x_i}{h} \\ \left.\frac{dx}{dt}\right|_{t=t_{i-1}} &= \frac{x_i - x_{i-1}}{h}. \end{aligned}$$

Tada je

$$\ddot{x} = \left.\frac{d^2x}{dt^2}\right|_{t=t_i} = \frac{\dot{x}(t=t_i) - \dot{x}(t=t_{i-1})}{h} = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2}.$$

Odavde se, uvrštavanjem u diferencijalnu jednadžbu i sređivanjem po x_{i-1}, x_i i x_{i+1} dobije trodijagonalni linearni sistem za točke t_i, $i = 1, \ldots, n-1$

$$\frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} - p(t_i)\frac{x_{i+1} - x_{i-1}}{2h} - r(t_i)x_i = q(t_i).$$

Greška metode je $c \cdot h^2$, $c \in \mathbb{R}$, dok greška metode kod metode gađanja ovisi o izabranoj RK metodi (RK–4 – greška $c \cdot h^4$).

14. Rješavanje parcijalnih diferencijalnih jednadžbi

14.1. Paraboličke PDJ — Provođenje topline

Parabolička diferencijalna jednadžba ima oblik

$$\frac{\partial^2 u}{\partial x^2} = \frac{c\rho}{k}\frac{\partial u}{\partial t} \tag{14.1.1}$$

uz rubne uvjete

$$u(0,t) = c_1(t)$$
$$u(L,t) = c_2(t)$$

i početni uvjet

$$u(x,0) = f(x) \quad \text{ili} \quad \frac{\partial u}{\partial x}(x,0) = g(x).$$

14.1.1. Eksplicitna metoda

Kao i kod rubnog problema za ODJ, aproksimirajmo derivacije na slijedeći način:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x=x_i,t=t_j} = \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2}$$
$$\left.\frac{\partial u}{\partial t}\right|_{x=x_i,t=t_j} = \frac{u_i^{j+1} - u_i^j}{\Delta t}.$$

Uvrštavanjem dobivamo

$$u_i^{j+1} = \frac{k\Delta t}{c\rho(\Delta x)^2}(u_{i+1}^j + u_{i-1}^j) + \left(1 - \frac{2k\Delta t}{c\rho(\Delta x)^2}\right)u_i^j.$$

Rješenje u $j+1$–om trenutku računa se eksplicitno iz onog u j–tom.

Označimo s

$$r = \frac{k\Delta t}{c\rho(\Delta x)^2}.$$

Zbog stabilnosti rješenja dif. jednadžbe nužno je uzeti $r \leq 1/2$. Ovime se određuje omjer vremenskih i prostornih koraka. Ako je početni uvjet glatka funkcija, može se pokazati da je pogreška najmanja ako se uzma $r = 1/6$.

Primjer 14.1.1. *Pretpostavimo da dvije velike željezne ploče debele* 1 cm *s linearno rasprostranjenom temperaturom od* $0°$–$100°$ *naglo spojimo toplijim krajevima, a rubove tih ploča držimo na* $0°$. *Što će se s pločom događati tokom vremena?*

U početnom trenutku ploče su zagrijane tako da je raspodjela temperature (po debljini)

$$f(x,0) = \begin{cases} 100x & za\ 0 \leq x \leq 1, \\ 100(2-x) & za\ 1 \leq x \leq 2. \end{cases}$$

Na krajevima ploča vrijede rubni uvjeti $u(0,t) = 0$ *i* $u(2,t) = 0$.

Fizikalno je jasno da će se na krajevima toplina gubiti, a temperatura polako ravnomjerno raspoređivati širom ploče. Ovakvo stanje ploče opisivat će jednadžba provođenja (14.1.1). Veličine c, ρ *i* k *su konstante željeza:* k *vodljivost,* c *toplinski kapacitet i* ρ *gustoća materijala.*

Jednadžbu ćemo riješiti eksplicitnom metodom. Budući da početni uvjet nije glatka funkcija (ima lom derivacije u 1), može se pokazati da je na pr. $r = 0.4$ *točnije rješenje nego za* $r = 1/6$.

Primjer 14.1.2. *Pretpostavimo da je* $k = c = \rho = 1$ *u (14.1.1), tj. da rješavamo paraboličku jednadžbu (14.1.1) uz uvjete uz rubne uvjete*

$$u(0,t) = 0$$
$$u(\pi,t) = 0$$

i početni uvjet

$$u(x,0) = \sin x,$$

što je glatka funkcija. Tada će najbolji r *biti* $r = 1/6$. *Osim toga, za ovu jednadžbu poznato je i pravo rješenje*

$$u(x,t) = e^{-t}\sin x$$

pa se mogu uspoređivati greške.

14.1.2. Crank–Nicolsonova metoda

Razmatra se jednadžba provođenja uz iste rubne i početne uvjete kao i prije. Ako derivacije u (14.1.1) zamijenimo na slijedeći način

$$\frac{1}{2}\Big(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} + \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2}\Big) = \frac{c\rho}{k}\Big(\frac{u_i^{j+1} - u_i^j}{\Delta t}\Big)$$

dobili smo Crank–Nicolsonovu metodu. Ako r označimo istu konstantu kao i prije, može se pokazati da je ova metoda stabilna za razne r, pa se može uzeti $r = 1$. Tada se metoda pojednostavljuje na

$$-u_{i-1}^{j+1} + 4u_i^{j+1} - u_{i+1}^{j+1} = u_{i-1}^j + u_{i+1}^j$$

što daje trodijagonalni sistem (za vremenski korak). Rješenja ovom metodom nešto lošija po točnosti od najbolje eksplicitne, ali se metoda jednostavno može poboljšati (Douglasova shema).

14.2. Hiperboličke PDJ — Valna jednadžba

Hiperbolička diferencijalna jednadžba ima oblik

$$\frac{\partial^2 u}{\partial t^2} = \frac{Tg}{w}\frac{\partial^2 u}{\partial x^2} \tag{14.2.1}$$

uz rubne uvjete

$$u(0,t) = c_1(t)$$
$$u(L,t) = c_2(t)$$

i početni položaj i početnu brzinu

$$u(x,0) = f(x)$$
$$\frac{\partial u(x,0)}{\partial t} = g(x)$$

gdje je g težina žice, T napetost i w linearna gustoća. Katkada se konstanta Tg/w označava s c^2.

14.2.1. Eksplicitna metoda

Derivacije se aproksimiraju na isti način kao i kod paraboličke PDJ. Uvrštavanjem u jednadžbu dobivamo:

$$\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\Delta t)^2} = c^2\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2}.$$

Označimo s

$$r = \frac{c^2(\Delta t)^2}{(\Delta x)^2}.$$

Može se pokazati da je ova shema stabilna, pa se može uzeti $r = 1$ (tada je $\Delta x = c\Delta t$). U tom slučaju prethodno eksplicitno rješenje se pojednostavi na

$$u_i^{j+1} = u_{i+1}^j + u_{i-1}^j - u_i^{j-1} \tag{14.2.2}$$

uz početne uvjete

$$u_i^0 = f(x_i), \quad i = 0, \ldots, n.$$

Uočite da nam je za start sistema potrebno i u_i^{-1}. To se može naći na jedan od slijedećih načina:

(a) brzina se aproksimira prvom centralnom razlikom, pa je

$$u_i^{-1} = u_i^1 - 2g(x_i)\Delta t$$

(b) zna se egzaktno rješenje jednadžbe iz početnih uvjeta (D'Alembertova formula)

$$u(x,t) = \frac{1}{2}[f(x-ct) + f(x+ct)] + \frac{1}{2c}\int_{x-ct}^{x+ct} g(\tau)\,d\tau.$$

Ako se uvaži da je $\Delta x = c\Delta t$ i da je u prvom trenutku t jednak $t = \Delta t$, može se izračunati u_i^1. Uvrštavanjem u prethodnu formulu dobiva se

$$\begin{aligned} u_i^1 = u(x_i, \Delta t) &= \frac{1}{2}[u_{i-1}^0 + u_{i+1}^0] + \frac{1}{2c}\int_{x_i-\Delta x}^{x_i+\Delta x} g(\tau)\,d\tau \\ &= \frac{1}{2}[u_{i-1}^0 + u_{i+1}^0] + \frac{1}{2c}\int_{x_{i-1}}^{x_{i+1}} g(\tau)\,d\tau. \end{aligned}$$

Posljednji integral može se aproksimirati Simpsonovom formulom. Primijetite da sada možemo startati računanje u_i^2, jer imamo u_i^0 i u_i^1. Drugim riječima, ovdje nam u_i^{-1} uopće nije potreban.

Primjer 14.2.1. *Žica bendža dugačka je* 80 cm*, teška* 1 g*. Napeta je silom jednakom težini mase od* 40 kg*. Cijelo vrijeme je učvršćena na oba kraja. U točki udaljenoj* 20 cm *od lijevog kraja iz ravnotežnog položaja povučemo žicu* 0.6 cm *prema gore. Nađite otklon žice u svakom trenutku t, nađite koliko joj je vremena potrebno za jedan kompletan period. Nađite frekvenciju titranja!*

Žica očito zadovoljava jednadžbu (14.2.1) uz uz rubne uvjete

$$\begin{aligned} u(0,t) &= 0 \\ u(80,t) &= 0 \end{aligned}$$

i početni položaj i početnu brzinu

$$\begin{aligned} u(x,0) &= \begin{cases} 0.03x & za\ 0 \le x \le 20 \\ -0.01x + 0.8 & za\ 20 \le x \le 80, \end{cases} \\ \frac{\partial u(x,0)}{\partial t} &= 0. \end{aligned}$$

Primjer 14.2.2. *Neka je $c = 2$ u jednadžbi (14.2.1). Ako žicu dugu 9 jedinica udarimo u ravnotežnom položaju brzinom (vidi dolje), dobivamo uvjete*

$$u(0,t) = 0$$
$$u(9,t) = 0$$

i

$$u(x,0) = 0$$
$$\frac{\partial u(x,0)}{\partial t} = 3\sin\frac{\pi x}{9}.$$

Ponovno, zgodno je promatrati greške u rješenju, jer se zna da je pravo rješenje jednako

$$u(x,t) = \frac{27}{8\pi}\Big(\cos\Big(\frac{\pi x}{9} - \frac{4\pi t}{9}\Big) - \cos\Big(\frac{\pi x}{9} + \frac{4\pi t}{9}\Big)\Big).$$