Računanje elementarnih funkcija u računalu

Bilogrević, Vlado

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Vlado Bilogrević

RAČUNANJE ELEMENTARNIH FUNKCIJA U RAČUNALU

Diplomski rad

Voditelj rada:
doc. dr. sc. Tina Bosner

Zagreb, rujan, 2023

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

Elementarne funkcije (sinus, kosinus, eksponencijalne, logaritamske, ...) su najčešće korištene matematičke funkcije. Pošto računalo "zna" samo osnovne računske operacije, one se računaju aproksimativno. Prilikom aproksimacije neka svojstva koja želimo imati su:

- brzina;
- točnost;
- razumna upotreba resursa;
- očuvanje važnih matematičkih svojstava kao što su monotonost i druga svojstva;
- očuvanje smjera zaokruživanja: na primjer, ako je aktivni način zaokruživanja zaokruživanje prema $-\infty$ (poglavlje 1), rezultat koji se vraća mora biti manji ili jednak točnom rezultatu. To je bitno za implementaciju intervalne aritmetike.

Ovaj diplomski rad prati knjigu [1]. U prvom poglavlju se daje pregled nekoliko elemenata računalne aritmetike kao kratak uvod u aritmetiku s pomičnom točkom koji su potrebni za razumijevanje narednih poglavlja.

Izuzimajući nekoliko slučajeva, elementarne funkcije se ne mogu izračunati točno nego se moraju aproksimirati. Većina algoritama sastoji se od evaluacije aproksimacija po dijelovima polinomnih ili po dijelovima racionalnih funkcija. U drugom poglavlju se opisuju algoritmi koji se temelje na polinomnim ili racionalnim aproksimacijama elementarnih funkcija. Teorija takvih aproksimacija seže unatrag od kraja 19. stoljeća. Jedine funkcije s jednom varijablom koje se mogu točno izračunati koristeći ograničen broj zbrajanja, oduzimanja i množenja su polinomi. Dodavanjem dijeljenja među dopuštene osnovne operacije, moguće je računati samo racionalne funkcije. Stoga je prirodno pokušati aproksimirati elementarne funkcije polinomnim ili racionalnim funkcijama. Takve aproksimacije bile su u širokoj upotrebi prije pojave suvremenih računala.

Kako bismo postigli preciznu polinomnu aproksimaciju funkcije unutar velikog intervala, često će biti potrebno koristiti polinome visokog stupnja. Na primjer, ako želimo aproksimirati funkciju $\ln(1 + x)$ unutar intervala $[-1/2, +1/2]$ s maksimalnom pogreškom

manjom od 10^{-8}, bit će potrebno koristiti polinom stupnja 12. Ovo može rezultirati produljenim vremenima računanja te potencijalnim problemima s širenjem greške zaokruživanja, s obzirom na broj aritmetičkih operacija koje su potrebne (osim ako se ne koristi veća preciznost za međuizračune). Rješenje za izbjegavanje ovih nedostataka je upotreba tablica. Upravo se o tome govori u poglavlju 3, gdje se opisuju metode temeljene na tablici.

Poglavlje 1

Aritmetika pomične točke

1.1 Formati pomične točke

Cilj ove sekcije je pružiti neke osnovne pojmove o aritmetici pomične točke (floating-point aritmetika) i definirati notacije koje se koriste kroz cijeli rad. Ovdje se uglavnom fokusiramo na standard IEEE-754. Definicije su većinom preuzete iz [1]

Definicija 1.1.1. *U sustavu pomične točke baze r, duljine mantise n i raspon eksponenata $E_{min} \dots E_{max}$, broj t je predstavljen mantisom $M_t = t_0.t_1t_2 \dots t_{n-1}$ koji je n-znamenkasti broj u bazi r, koji zadovoljava $0 \leq M_t < r$, predznakom $s_t = \pm 1$, i eksponentom $E_t, E_{min} \leq E_t \leq E_{max}$, tako da je*

$$t = s_t \times M_t \times r^{E_t}. \tag{1.1}$$

Zbog točnosti se obično zahtijeva da mantise budu veće ili jednake 1. To također zahtijeva i poseban prikaz za nulu. Najveći prikazivi konačni broj u IEEE-754 formatu dvostruke preciznosti je

$$\left(2 - 2^{-52}\right) \times 2^{1023} \approx 1.7976931348623157 \times 10^{308} \tag{1.2}$$

a najmanji pozitivni normalizirani broj[1]

$$2^{-1022} \approx 2.225073858507201 \times 10^{-308}. \tag{1.3}$$

[1]U primijenjenoj matematici, broj je normaliziran kada je zapisan u znanstvenom zapisu s jednom decimalnom znamenkom različitom od nule ispred decimalne točke. Dakle, pravi broj, kada je zapisan u normaliziranom znanstvenom zapisu, je sljedeći: $\pm d_0.d_1d_2d_3 \cdots \times r^n$ gdje je n cijeli broj, a $d_0, d_1, d_2, d_3, \dots$, su znamenke broja u bazi r, a d_0 nije nula.

Načini zaokruživanja

Definirajmo broj s pomičnom točkom kao broj koji se može točno predstaviti sustavom pomične točke. Općenito, zbroj, produkt, i kvocijent dva broja s pomičnom točkom ne mora biti broj s pomičnom točkom i rezultat takve aritmetičke operacije mora biti zaokružen.

U sustavu s pomičnom točkom pod IEEE-754 standardom korisnik može odabrati jedan od aktivnih načina zaokruživanja:

- zaokruživanje prema $-\infty$: $\nabla(x)$ je najveći broj s pomičnom točkom manji od ili jednak x;
- zaokruživanje prema $+\infty$: $\Delta(x)$ je najmanji broj s pomičnom točkom veći od ili jednak x;
- zaokruživanje prema 0: $\mathcal{Z}(x)$ je jednako $\nabla(x)$ ako je $x \geq 0$, a $\Delta(x)$ ako je $x < 0$;
- zaokruživanje na najbliži: $\mathcal{N}(x)$ je broj s pomičnom točkom koji je najbliži x (uz posebnu konvenciju ako je x točno između dva broja s pomičnom točkom: izabrani broj je "parni", tj. onaj čiji je zadnji bit mantise nula).

Ako je aktivni način zaokruživanja označen s $\diamond$, a u i v su brojevi s pomičnom točkom, onda standard IEEE-754 zahtijeva da dobiveni rezultat uvijek treba biti $\diamond(u \top v)$ kada se računa $u \top v$ ($\top$ je operacija zbrajanja, oduzimanja, množenja ili dijeljenja). Stoga se sustav mora ponašati kao da je rezultat prvi put točno izračunat, s beskonačnom preciznošću, a zatim zaokružen. Operacije koje zadovoljavaju takvo svojstvo nazivaju se "ispravno zaokružene". Postoji sličan zahtjev za kvadratni korijen. Takav zahtjev ima brojne prednosti:

- dovodi do potpune kompatibilnosti između računalnih sustava: isti program će dati iste vrijednosti na različitim računalima
- mogu se dizajnirati mnogi algoritmi koji koriste ovo svojstvo.
- može se jednostavno implementirati intervalna aritmetika, ili općenitije se mogu dobiti donje ili gornje granice točnog rezultata niza aritmetičkih operacija.

Vrlo koristan rezultat koji se može dokazati pod pretpostavkom ispravnog zaokruživanja je sljedeći algoritam

Teorem 1.1.2. *(algoritam Fast2Sum) Pretpostavimo da je baza r sustava s pomičnom točkom koji se razmatra manji ili jednak od 3, i da korištena aritmetika osigurava ispravno zaokruživanje sa zaokruživanjem na najbliži. Ovdje $\mathcal{N}(x)$ znači x zaokruženo na najbliži. Neka su a i b brojevi s pomičnom točkom i pretpostavimo da je eksponent od a veći ili*

jednak eksponentu od b. Sljedeći algoritam izračunava dva broja s pomičnom točkom s i t koji zadovoljavaju:

- *točno* $s + t = a + b$;
- *s je broj s pomičnom točkom koji je najbliži* $a + b$.

Algoritam 1 (Fast2Sum(a,b))

$$s := \mathcal{N}(a + b);$$
$$z := \mathcal{N}(s - a);$$
$$t := \mathcal{N}(b - z).$$

Subnormalni brojevi i iznimke

U standardu pomične točke IEEE-754 brojevi su normalizirani osim ako nisu jako mali. Subnormalni brojevi (koji se nazivaju i denormalizirani brojevi) su brojevi različiti od nule s nenormaliziranom mantisom i najmanjim mogućim eksponentom (tj. eksponent koji se koristi za predstavljanje nule).

U sustavu s pomičnom točkom s ispravnim zaokruživanjem i subnormalnim brojevima, vrijedi sljedeći teorem.

Teorem 1.1.3. *(**Sterbenzova lema**) U sustavu s pomičnom točkom s ispravnim zaokruživanjem i subnormalnim brojevima, ako su x i y brojevi s pomičnom točkom takvi da* $x/2 \leq y \leq 2x$*, tada će x - y biti točno izračunat.*

Taj je rezultat koristan pri izračunavanju točnih granica pogreške za neke algoritme elementarnih funkcija. Standard IEEE-754 također definira posebne prikaze za iznimke:

- NaN (Not a Number) je rezultat nevažeće aritmetičke operacije kao što je $\sqrt{-5}$, ∞/∞, $+\infty + (-\infty)$, ...;
- $\pm\infty$ može biti rezultat overflow-a ili točan rezultat dijeljenja s nula; i
- ± 0: postoje dvije nule s predznakom koje mogu biti rezultat underflow-a, ili točan rezultat dijeljenja s $\pm\infty$.

ULP-ovi

Ako se x točno može predstaviti u formatu pomične točke i nije cjelobrojna potencija baze r, izraz ulp(x) (unit in the last place) označava magnitudu zadnje znamenke mantise od x. Ako je

$$x = \pm x_0.x_1x_2\ldots x_{n-1} \times r^{E_x}$$

tada je $ulp(x) = r^{Ex-n+1}$.

Definicija 1.1.4. *Ako x leži između dva konačna uzastopna broja s pomičnom točkom a i b, a da nije jednak ni jednom od njih, tada je $ulp(x) = |b - a|$, inače je ulp(x) udaljenost između dva konačna broja s pomičnom točkom najbliža x*

Glavna prednost ove definicije je da u svim slučajevima, zaokruživanje na najbliži odgovara pogrešci od najviše 1/2 ulp od prave vrijednosti. Ova definicija pretpostavlja da je x realan broj.

Kombinirane operacije množenja i zbrajanja

Neki procesori imaju instrukciju kombiniranog množenja-zbrajanja (FMA- fused multiply-add), koji omogućuje izračunavanje $ax \pm b$, gdje su a, x i b brojevi s pomičnom točkom i samo s jednim konačnim zaokruživanjem. Takva instrukcija može biti od velike pomoći za dizajnere aritmetičkih algoritama.

FMA čini evaluaciju polinoma bržom i općenito točnijom: kada se koristi Hornerova shema, broj potrebnih operacija (odnosno, broj zaokruživanja) je prepolovljen. Ovo je iznimno važno za evaluaciju elementarne funkcije jer su polinomne aproksimacije često korištene za te funkcije.

Poglavlje 2

Polinomne ili racionalne aproksimacije

Korištenjem konačnog broja zbrajanja, oduzimanja, množenja i usporedbi, jedine funkcije jedne varijable koje se mogu izračunati su po dijelovima polinomi (piecewise polynomials). Ako skupu dostupnih operacija dodamo dijeljenje, jedine funkcije koje se mogu izračunati su po dijelovima racionalne funkcije (piecewise rational functions). Stoga je prirodno pokušati aproksimirati elementarne funkcije polinomima ili racionalnim funkcijama. Pitanja koja odmah padaju na pamet su:

- Kako možemo izračunati takve polinomne ili racionalne aproksimacije?
- Koji je najbolji način (u smislu točnosti i/ili brzine) za računanje vrijednosti polinoma ili racionalne funkcije?
- Konačna pogreška bit će zbroj dviju pogrešaka: pogreška aproksimacije (tj."udaljenost" između funkcije koja se aproksimira i polinoma ili racionalne funkcije), i pogreška evaluacije zbog činjenice da se polinom ili racionalna funkcija evaluiraju u konačnoj preciznosti aritmetike pomične točke. Možemo li izračunati precizne granice za te pogreške?

U ovom poglavlju s P_n označavamo skup polinoma stupnja manjeg ili jednakog od n s realnim koeficijentima, a $R_{p,q}$ skup racionalnih funkcija s realnim koeficijentima čiji brojnik i nazivnik imaju stupanj manji ili jednak p odnosno q.

Usredotočimo se prvo na problem izgradnje polinomnih aproksimacija. Naravno, ključno je izračunati koeficijente takvih aproksimacija preciznošću znatno većom od "ciljane preciznosti" (tj. preciznosti konačnog rezultata). Želimo aproksimirati funkciju f pomoću $p^* \in P_n$ na intervalu [a, b]. Ovdje se razmatraju dvije vrste aproksimacija: aproksimacije koje minimiziraju "prosječnu pogrešku", nazvanu aproksimacija najmanjih kvadrata, i aproksimacije koje minimiziraju maksimalnu pogrešku, koje zovemo minimaks aproksimacije. U oba slučaja, želimo minimizirati "udaljenost" $\|p^* - f\|$.

Za aproksimacije najmanjih kvadrata ta je udaljenost

$$\|p^* - f\|_2 = \sqrt{\int_a^b w(x)(f(x) - p^*(x))^2 dx}$$

gdje je w neprekidna, nenegativna težinska funkcija, koja se može koristiti za odabir dijelova [a, b] gdje želimo da aproksimacija bude preciznija. Za minimaks aproksimacije, udaljenost je

$$\|p^* - f\|_\infty = \max_{a \le x \le b} w(x)|p^*(x) - f(x)|.$$

2.1 Polinomne aproksimacije najmanjih kvadrata

Tražimo polinom stupnja $\le n$,

$$p^*(x) = p_n^* x^n + p_{n-1}^* x^{n-1} + \cdots + p_1^* x + p_0^*$$

koji zadovoljava

$$\int_a^b w(x)(f(x) - p^*(x))^2 dx = \min_{p \epsilon \mathcal{P}_n} \int_a^b w(x)(f(x) - p(x))^2 dx.$$

Definicija 2.1.1. *Definirajmo* $\langle f, g \rangle$ *kao*

$$\langle f, g \rangle = \int_a^b w(x) f(x) g(x) dx.$$

Aproksimacija p^* može se izračunati na sljedeći način:

- izgraditi niz $(T_m), (m \le n)$ polinoma tako da je (T_m) stupnja m, i tako da $\langle T_i, T_j \rangle = 0$ za i = j. Takvi se polinomi nazivaju ortogonalni polinomi;
- izračunati koeficijente:
$$a_i = \frac{\langle f, T_i \rangle}{\langle T_i, T_i \rangle} \tag{2.1}$$
- izračunati:
$$p^* = \sum_{i=0}^{\infty} a_i T_i \tag{2.2}$$

Neki nizovi ortogonalnih polinoma, pridruženi s jednostavnim težinskim funkcijama w, dobro su poznati, pa ih nema potrebe ponovno računati. Predstavimo sada neke od njih.

Legendreovi polinomi

- težinska funkcija: $w(x) = 1$;
- interval [a,b] = [-1,1];
- definicija:
$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ T_n(x) = \frac{2n-1}{n} x T_{n-1}(x) - \frac{n-1}{n} T_{n-2}(x); \end{cases}$$
- vrijednosti skalarnih produkta:
$$\langle T_i, T_j \rangle = \begin{cases} 0 & \text{ako je } i \neq j \\ \frac{2}{2i+1} & \text{inače.} \end{cases}$$

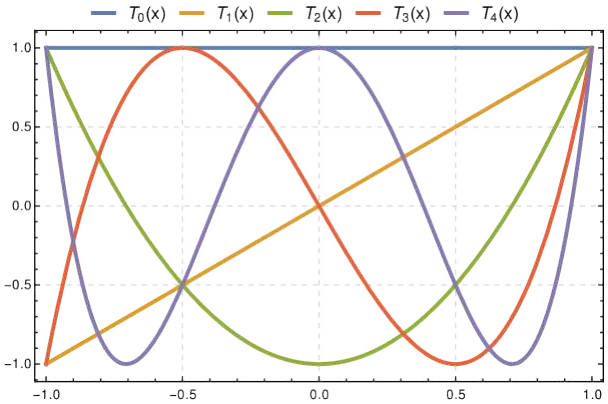
Čebiševljevi polinomi

- težinska funkcija: $w(x) = \frac{1}{\sqrt{1-x^2}}$;
- interval [a,b] = [-1,1];
- definicija:
$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ T_n(x) = 2x T_{n-1}(x) - T_{n-2}(x) = \cos(n \cos^{-1} x) \end{cases}$$
- vrijednosti skalarnih produkta:
$$\langle T_i, T_j \rangle = \begin{cases} 0 & \text{ako je } i \neq j \\ \pi & \text{ako je } i = j = 0 \\ \pi/2 & \text{inače.} \end{cases}$$

Primjer Čebiševljevih polinoma ($T_0(x)$ do $T_4(x)$) prikazan je na slici 2.1. Čebiševljevi polinomi igraju središnju ulogu u teoriji aproksimacije. Među njihovim brojnim svojstvima, sljedeća tri se često koriste.

Teorem 2.1.2. *Za $n \geq 0$, vrijedi*

$$T_n(x) = \frac{n}{2} \sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \frac{(n-k-1)!}{k!(n-2k)!} (2x)^{n-2k}.$$



Slika 2.1: Čebiševljevi polinomi na intervalu [-1,1]

Dakle, vodeći koeficijent od T_n je 2^{n-1}. T_n ima n realnih korijena, svi striktno između -1 i 1.

Teorem 2.1.3. *Postoji $n + 1$ točaka $x_0, x_1, x_2, \ldots, x_n$ koje zadovoljavaju*

$$-1 = x_0 < x_1 < x_2 < \cdots < x_n = 1$$

tako da vrijedi

$$T_n(x_i) = (-1)^{n-i} \max_{x\epsilon[-1,1]} |T_n(x)| \quad \forall i, i = 0, \ldots, n.$$

To jest, najveća apsolutna vrijednost T_n postiže se na x_i, a predznak T_n izmjenjuje se na ovim točkama.

Nazovimo polinom čiji je vodeći koeficijent 1 moničkim polinomom.

Teorem 2.1.4. *Neka su a, b realni brojevi i $a \leq b$. Monički polinom stupnja n P koji minimizira*

$$\max_{x\epsilon[a,b]} |P(x)|$$

je

$$\frac{(b-a)^n}{2^{2n-1}} T_n\left(\frac{2x-b-a}{b-a}\right).$$

Jacobijevi polinomi

- težinska funkcija: $w(x) = (1-x)^\alpha(1+x)^\beta \quad (\alpha, \beta > 1)$;
- interval [a,b] = [-1,1];

- definicija:
$$T_n(x) = \frac{1}{2^n}\sum_{m=0}^{n}\binom{n+\alpha}{m}\binom{n+\beta}{n-m}(x-1)^{n-m}(x+1)^m;$$
- vrijednosti skalarnih produkta:
$$\langle T_i, T_j\rangle = \begin{cases} 0 & \text{ako je } i \neq j \\ h_i & \text{inače.} \end{cases}$$
gdje je
$$h_i = \frac{2^{\alpha+\beta+1}}{2i+\alpha+\beta+1}\frac{\Gamma(i+\alpha+1)\Gamma(i+\beta+1)}{i!\Gamma(i+\alpha+\beta+1)}.$$

Laguerreovi polinomi

- težinska funkcija: $w(x) = e^{-x}$;
- interval [a,b] = [0,+∞];
- definicija:
$$T_n(x) = \frac{e^x}{n!}\frac{d^n}{dx^n}(x^n e^{-x});$$
- vrijednosti skalarnih produkta:
$$\langle T_i, T_j\rangle = \begin{cases} 0 & \text{ako je } i \neq j \\ 1 & \text{inače.} \end{cases}$$

Osim Laguerreovih polinoma, koji su relevantni za interval $[a, b] = [0, +\infty]$, ortogonalni polinomi koje smo prethodno predstavili dani su za interval [-1, 1]. No, dobivanje aproksimacije za drugi interval [a, b] može se jednostavno postići:

- za $u \in [-1, 1]$, definirati:
$$g(u) = f\left(\frac{b-a}{2}u + \frac{a+b}{2}\right);$$
primijetimo da je $x = (\frac{b-a}{2})u + (\frac{a+b}{2}) \in [a, b]$;
- izračunati aproksimaciju polinomom najmanjih kvadrata q^* od g na $[-1, 1]$;
- dobiti aproksimaciju f metodom najmanjih kvadrata, recimo p^*, kao:
$$p^*(x) = q^*\left(\frac{2}{b-a}x - \frac{a+b}{b-a}\right).$$

Minimaks polinomne aproksimacije

Kao u prethodnom odjeljku, želimo aproksimirati funkciju f polinomom $p^* \epsilon P_n$ na segmentu [a, b]. Pretpostavimo da je težinska funkcija w(x) jednaka 1. U nastavku, $\|f - p\|_\infty$ označava udaljenost:

$$\|f - p\|_\infty = \max_{a \le x \le b} |f(x) - p(x)|$$

Tražimo polinom p^* koji zadovoljava:

$$\|f - p^*\|_\infty = \min_{p \epsilon P_n} \|f - p\|_\infty$$

Polinom p^* naziva se minimaks polinomnom aproksimacijom stupnja n za f na intervalu [a,b].

Godine 1885. Weierstrass je dokazao sljedeći teorem, koji pokazuje da se neprekidna funkcija može aproksimirati polinomom po volji precizno.

Teorem 2.1.5. ***(Weierstrass, 1885)*** *Neka je f neprekidna funkcija. Za bilo koji $\epsilon > 0$ postoji polinom p takav da je $\|p - f\|_\infty \le \epsilon$.*

Drugi teorem, kojeg je dao Čebisev daje karakterizaciju minimaks aproksimacije funkcije.

Teorem 2.1.6. ***(Čebišev)*** *p^* je minimaks aproksimacija stupnja n za f na [a, b] ako i samo ako postoji najmanje n + 2 vrijednosti*

$$a \le x_0 < x_1 < x_2 < \cdots < x_{n+1} \le b$$

tako da vrijedi:

$$p^*(x_i) - f(x_i) = (-1)^i [p^*(x_0) - f(x_0)] = \pm \|f - p^*\|_\infty.$$

Čebiševljev teorem pokazuje da ako je p^* minimaks polinomna aproksimacija stupnja n za f, tada se najveća pogreška aproksimacije postiže najmanje n + 2 puta, do na predznak. To nam svojstvo omogućuje izravno pronalaženje p^* u nekim posebnim slučajevima, koje ćemo pokazati u sljedećim primjerima.

Primjeri aproksimacija za e^x pomoću polinoma stupnja 2

Pretpostavimo sada da želimo izračunati polinomnu aproksimaciju stupnja 2 za eksponencijalnu funkciju na intervalu [-1, 1]. Koristit ćemo neke od prethodno predstavljenih metoda za izračunavanje i usporediti različite aproksimacije

Aproksimacija najmanjih kvadrata pomoću Legendreovih polinoma

prva tri Legendreova polinoma su:

$$T_0 = 1$$
$$T_1 = x$$
$$T_2 = \frac{3}{2}x^2 - \frac{1}{2}$$

Skalarni produkt povezan s Legendreovom aproksimacijom je:

$$\langle f, g\rangle = \int_{-1}^{1} f(x)g(x)dx$$

iz toga se lako dobije:

$$\langle e^x, T_0\rangle = e - 1/e$$
$$\langle e^x, T_1\rangle = 2/e$$
$$\langle e^x, T_2\rangle = e - 7/e$$
$$\langle T_0, T_0\rangle = 2$$
$$\langle T_1, T_1\rangle = 2/3$$
$$\langle T_2, T_2\rangle = 2/5$$

Stoga su koeficijenti od a_i po (2.1):

$$a_0 = \frac{1}{2}(e - \frac{1}{e}), \quad a_1 = \frac{3}{e}, \quad a_2 = \frac{5}{2}(e - \frac{7}{e}),$$

pa je polinom p^* po (2.2):

$$p^*(x) = \frac{15}{4}(e - \frac{7}{e})x^2 + \frac{3}{e}x + \frac{33}{4e} - \frac{3e}{4} \simeq 0.5367215x^2 + 1.103683x + 0.9962940$$

Aproksimacija najmanjih kvadrata pomoću Čebiševljevih polinoma

Prva tri Čebiševljeva polinoma su:

$$T_0 = 1$$
$$T_1 = x$$
$$T_2 = 2x^2 - 1$$

Skalarni produkt povezan s Čebiševljevom aproksimacijom je:

$$\langle f, g\rangle = \int_{-1}^{1} \frac{f(x)g(x)}{\sqrt{1 - x^2}}dx$$

Iz toga dobivamo:

$$\langle e^x, T_0\rangle = 3.977463261\cdots$$
$$\langle e^x, T_1\rangle = 1.775499689\cdots$$
$$\langle e^x, T_2\rangle = 0.426463882\cdots$$
$$\langle T_0, T_0\rangle = \pi$$
$$\langle T_1, T_1\rangle = \pi/2$$
$$\langle T_2, T_2\rangle = \pi/2$$

iz toga izračunamo polinom $p^* = a_0T_0 + a_1T_1 + a_2T_2$ koji je približno jednak

$$0.5429906776x^2 + 1.130318208x + 0.9945705392$$

Minimaks aproksimacija

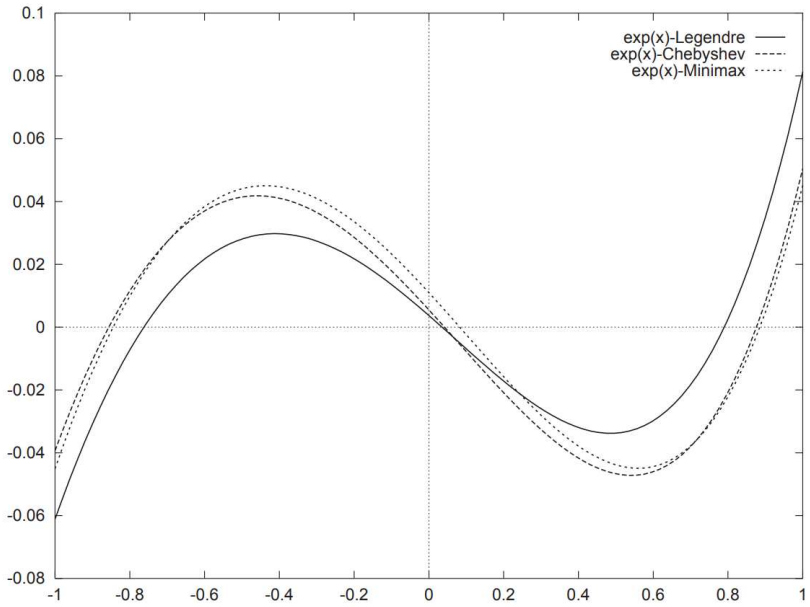
Pretpostavimo da je $p^*(x) = a_0 + a_1x + a_2x^2$ minimaks aproksimacija za e^* na [-1, 1]. Iz Teorema 2.1.6., postoje najmanje četiri vrijednosti x_0, x_1, x_2 i x_3 gdje je najveća pogreška aproksimacije ϵ postignuta s alternativnim predznacima. Konveksnost eksponencijalne funkcije implicira $x_0 = -1$ i $x_3 = +1$. Štoviše, derivacija $e^x - p^*(x)$ jednaka je nuli za $x = x_1$ i $x = x_2$. To daje sustav nelinearnih jednadžbi:

$$\begin{cases} a_0 - a_1 + a_2 - 1/e = \epsilon \\ a_0 + a_1x_1 + a_1x_1^2 - e^{x_1} = -\epsilon \\ a_0 + a_1x_2 + a_2x_2^2 - e^{x_2} = \epsilon \\ a_0 + a_1 + a_2 - e = -\epsilon \\ a_1 + 2a_2x_1 - e^{x_1} = 0 \\ a_1 + 2a_2x_2 - e^{x_2} = 0 \end{cases}$$

a rješenje tog sustava jednako je:

$$\begin{cases} a_0 = 0.98903973\cdots \\ a_1 = 1.13018381\cdots \\ a_2 = 0.55404091\cdots \\ x_1 = -0.43695806\cdots \\ x_2 = 0.56005776\cdots \\ \epsilon = 0.04501739\cdots \end{cases}$$

Stoga je najbolja minimaks polinomna aproksimacija stupnja 2 za e^x na intervalu [-1, 1] jednaka $0.98903973 + 1.13018381x + 0.55404091x^2$, a najveća pogreška aproksimacije je



Slika 2.2: Pogreške različitih aproksimacija stupnja 2 za e^x na intervalu [-1, 1]. Legendreova aproksimacija je u prosjeku bolja, a Čebiševljeva aproksimacija je blizu minimaks aproksimacije.(preuzeta iz [1])

otprilike 0.045. Na slici 2.2 nalaze se pogreške različitih aproksimacija polinoma stupnja 2 za e^x na [-1, 1] U prethodnim smo odjeljcima vidjeli da se bilo koja neprekidna funkcija može aproksimirati polinomom koliko god je to potrebno. Nažalost, da bi se postigla određena pogreška aproksimacije, stupanj potrebnog polinoma aproksimacije može biti prilično velik.

2.2 Remezov algoritam

Remezov algoritam je iterativni postupak koji se temelji na primjeni Čebiševljevog teorema 2.1.6 te ima svrhu konstruirati najbolju polinomnu minimaks aproksimaciju za određene funkcije f. Cilj ovog algoritma je aproksimirati funkciju f unutar intervala [a, b]. Sam Remezov algoritam sastoji se od iterativne izgradnje skupa točaka $x_0, x_1, \cdots, x_{n+1}$ prema Teoremu 2.1.6. U nastavku ćemo prikazati korake koje ovaj algoritam uključuje:

1. Polazimo od početnog skupa točaka $x_0, x_1, ..., x_{n+1}$ u [a, b].

2. Razmatramo linearni sustav jednadžbi

$$\begin{cases} p_0 + p_1 x_0 + p_2 x_0^2 + \cdots + p_n x_0^n - f(x_0) = +\epsilon \\ p_0 + p_1 x_1 + p_2 x_1^2 + \cdots + p_n x_1^n - f(x_1) = -\epsilon \\ p_0 + p_1 x_2 + p_2 x_2^2 + \cdots + p_n x_2^n - f(x_1) = +\epsilon \\ \cdots \\ p_0 + p_1 x_{n+1} + p_2 x_{n+1}^2 + \cdots + p_n x_{n+1}^n - f(x_{n+1}) = (-1)^{n+1}\epsilon \end{cases}$$

   Važno je naglasiti da vrijednost ϵ nije fiksna, već se izračunava zajedno s koeficijentima p_i.

   Ovo dovodi do sustava od n + 2 linearnih jednadžbi s n + 2 nepoznanice: $p_0, p_1, \ldots, p_n$ i ϵ. Stoga, u svim nedegeneriranim slučajevima, ovaj sustav ima jedinstveno rješenje $(p_0, p_1, ..., p_n, \epsilon)$. Rješavanjem tog sustava dobivamo polinom

$$P(x) = p_0 + p_1 x + \cdots + p_n x_n.$$

3. U nastavku postupka, izračunavamo skup točaka y_i unutar intervala [a, b], gdje razlika između polinoma P i funkcije f ima svoje lokalne ekstreme. Potom započinjemo novi ciklus (korak 2), zamjenjujući točke x_i sa vrijednostima y_i. Postupak se ponavlja sve dok greška $(P - f)$ ne postane dovoljno mala ili se ne očuva ista u sljedećem koraku.

Obično se za početni skup točaka koriste Čebiševljevi čvorovi. Ove točke su definirane kao:

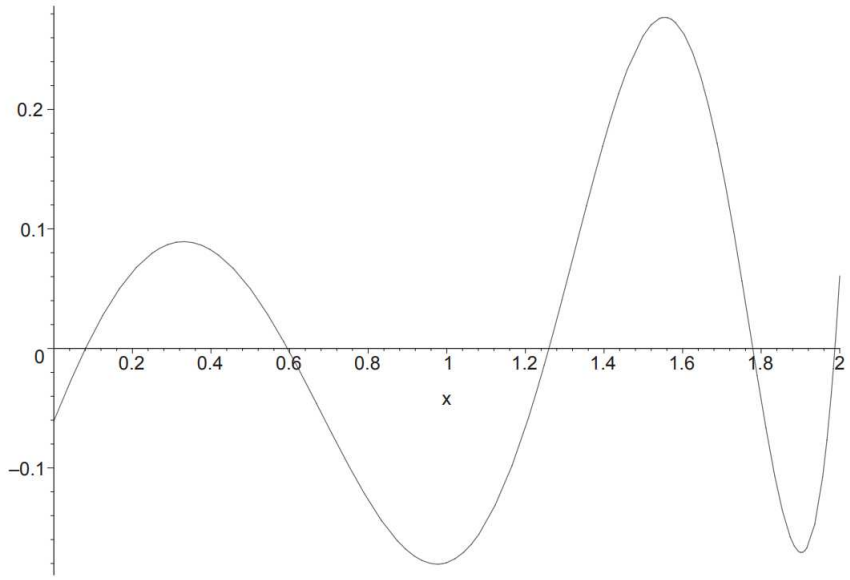
$$x_i = \frac{a+b}{2} + \frac{(b-a)}{2} \cos \frac{i\pi}{n+1}, \quad 0 \le i \le n+1$$

Ovaj odabir Čebiševljevih čvorova temelji se na zahtjevu da vrijedi $\left|T_{n+1}(\frac{(2x-b-a)}{(b-a)})\right| = 1$, gdje je T_i Čebiševljev polinom stupnja i. Ovo dolazi iz činjenice da su minimaks aproksimacija i aproksimacija pomoću Čebiševljevih polinoma vrlo bliske u većini uobičajenih slučajeva.

Primjer aproksimacije za funkciju $\sin(e^x)$ pomoću Remezovog algoritma

Da bismo demonstrirali ponašanje Remezovog algoritma, istražit ćemo izračun minimaks aproksimacije stupnja 4 za funkciju $\sin(e^x)$ unutar intervala [0, 2]. Polazimo od sljedećeg skupa točaka:

$$1 + \cos(\frac{i\pi}{5}), \quad i = 0, 1, \cdots, 5.$$



Slika 2.3: Razlika između $P^{(1)}(x)$ i $\sin(e^x)$ na intervalu $[0, 2]$.[1]

Ove vrijednosti su redom: 2, 1.809016994, 1.309016994, 0.690983005, 0.190983005 i 0, što su točke u kojima vrijedi $|T_5(x-1)| = 1$. Iz ovih početnih točaka formira se linearni sustav jednadžbi:

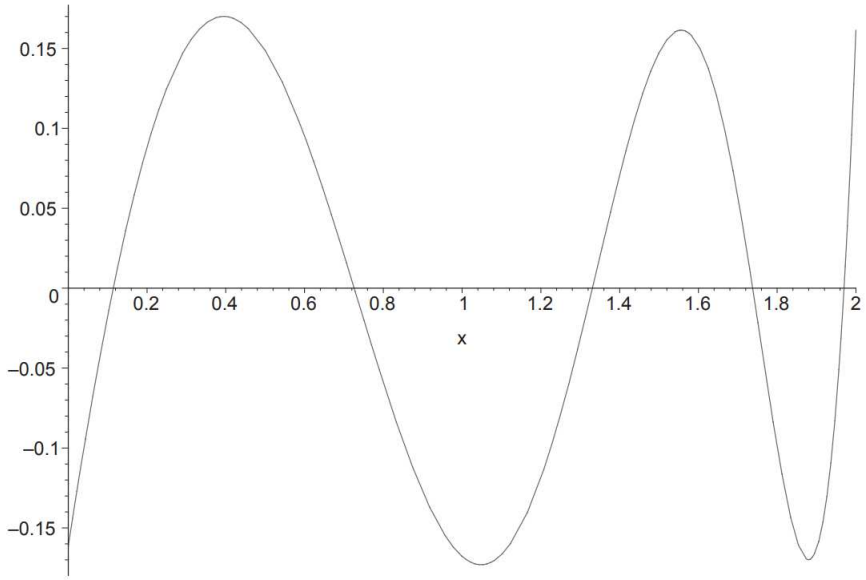
$$\begin{cases} p_0 & & & & & & -0.84147098 = \epsilon \\ p_0 + 0.19098300p_1 & + 0.036474508p_2 & + 0.006966011p_3 & + 0.001330389p_4 & - 0.93577084 = -\epsilon \\ p_0 + 0.69098300p_1 & + 0.47745751p_2 & + 0.32991502p_3 & + 0.22796567p_4 & - 0.91108820 = \epsilon \\ p_0 + 1.3090169p_1 & + 1.7135254p_2 & + 2.2430339p_3 & + 2.9361696p_4 & + 0.53198209 = -\epsilon \\ p_0 + 1.8090169p_1 & + 3.2725424p_2 & + 5.9200849p_3 & + 10.709534p_4 & + 0.17779129 = \epsilon \\ p_0 + 2p_1 & + 4p_2 & + 8p_3 & + 16p_4 & - 0.89385495 = -\epsilon. \end{cases}$$

Rješavanjem ovog sustava dobiva se sljedeći polinom:

$$P^{(1)}(x) = 0.78080774 + 1.3572109x - 0.79962767x^2 - 2.2959821x^3 + 1.1891035x^4.$$

Razlika $P^{(1)}(x) - \sin(e^x)$ prikazana je na slici 2.3. Sada izračunavamo ekstreme od $P^{(1)}(x) - \sin e^x$ u $[0, 2]$, što rezultira novim skupom točaka: 0, 0.330511, 0.975647, 1.55426, 1.9020758 i 2. Rješavanjem linearnog sustava s ovim točkama dobivamo polinom:

$$P^{(2)}(x) = 0.68008890 + 2.1440920x - 1.6313678x^2 - 2.2262202x^3 + 1.2763873x^4.$$



Slika 2.4: Razlika između $P^{(2)}(x)$ i $\sin(e^x)$ na intervalu [0, 2].[1]

Razlika $P^{(2)}(x) - \sin(e^x)$ prikazana je na slici 2.4. Primjećujemo da su ekstremne vrijednosti $|P^{(2)}(x) - \sin(e^x)|$ međusobno vrlo bliske: $P^{(2)}$ gotovo "potpuno" zadovoljava uvjet iz teorema 2.1.6. Ovaj primjer ilustrira brzu konvergenciju Remezovog algoritma jer nakon samo dvije iteracije već imamo polinom koji je izuzetno blizu minimaks polinomu. Nakon što izračunamo $P^{(3)}$ i $P^{(4)}$ koristeći istu metodu kao za prethodne polinome, dobivamo:

$$P^{(3)}(x) = 0.67517859 + 2.1238096x - 1.5488299x^2 - 2.2931470x^3 + 1.2923653x^4$$

i

$$P^{(4)}(x) = 0.67517521 + 2.1235853x - 1.5483419x^2 - 2.2934835x^3 + 1.2924400x^4$$

Primjećujemo da je omjer između najvećeg i najmanjeg lokalnog ekstrema udaljenosti $|P^{(4)}(x) - \sin(e^x)|$ manji od 1.000005. Ovaj rezultat sugerira da smo pronašli minimaks polinom, odnosno polinom koji aproksimira funkciju $\sin(e^x)$ unutar intervala [0, 2].

2.3 Racionalne aproksimacije

Tablica 2.1 daje različite pogreške dobivene aproksimacijom kvadratnog korijena na [0, 1] pomoću polinoma. Čak i s polinomima stupnja 12, aproksimacije su loše. Gruba procjena

pokazuje da je za aproksimaciju kvadratnog korijena na [0, 1] s apsolutnom pogreškom manjom od 10^{-7} potreban polinom stupnja 54. Slična stvar pojavljuje se ako tražimo

stupanj	pogreška
4	0.034
5	0.028
6	0.023
7	0.020
8	0.017
9	0.016
10	0.014
11	0.013
12	0.012

Tablica 2.1: Apsolutne pogreške dobivene aproksimacijom kvadratnog korijena na [0, 1] minimaks polinomom.

aproksimacije na [1/4, 1]. Minimaks polinomna aproksimacija stupnja 25 za $\sqrt{x}$ na [1/4, 1] ima pogrešku aproksimacije jednaku 0.13×10^{-14}, dok minimaks aproksimacija iste funkcije pomoću racionalne funkcije čiji nazivnik i brojnik imaju stupanj manji ili jednak od 5 daje bolju pogrešku aproksimacije, $0,28 \times 10^{-15}$. Ovo pokazuje da za neke funkcije u nekim domenama polinomne aproksimacije možda nisu prikladne. Treba pokušati s racionalnim aproksimacijama. Što se tiče racionalnih aproksimacija, postoji teorem karakterizacije racionalnih aproksimacija, sličan teoremu 2.1.6. Podsjetimo da je $R_{p,q}$ skup racionalnih funkcija s realnim koeficijentima čiji brojnik i nazivnik imaju stupanj manji ili jednaki p odnosno q. Za idući teorem nam trebaju ireducibilne funkcije koje ćemo sad definirati.

Definicija 2.3.1. *Ireducibilna racionalna funkcija je racionalna funkcija koja ne može biti dalje pojednostavljena ili rastavljena na manje faktore koji su također racionalne funkcije.*

Drugim riječima, ireducibilna racionalna funkcija je ona koja nema zajedničke faktore u brojniku i nazivniku osim konstante. Primjer ireducibilne racionalne funkcije je $f(x) = \frac{x^2+1}{x^2-3x+2}$. Ovdje nema zajedničkih faktora između brojnika i nazivnika koji bi mogli biti otklonjeni ili pojednostavljeni.

Teorem 2.3.2. ***(Čebišev)*** *Ireducibilna racionalna funkcija $R^* = P/Q$ je minimaks racionalna aproksimacija f na [a, b] među racionalnim funkcijama koje pripadaju $\mathcal{R}_{n,m}$ ako i samo ako postoje barem*

$$k = 2 + \max\{m + stupanj(P), n + stupanj(Q)\}$$

vrijednosti

$$a \leq x_0 < x_1 < x_2 < \cdots < x_{k-1} \leq b$$

tako da vrijedi:

$$R^*(x_i) - f(x_i) = (-1)^i[R^*(x_0) - f(x_0)] = \pm\|f - R^*\|_\infty$$

Čini se prilično teškim predvidjeti hoće li će se dana funkcija mnogo bolje aproksimirati racionalnim funkcijama nego polinomima. Intuitivno ima smisla pomisliti da će funkcije koje se ponašaju "vrlo nepolinomno" (konačni limesi na $\pm\infty$, polovi, beskonačne derivacije...) biti loše aproksimirane polinomima. Na primjer, minimaks polinomna aproksimacija stupnja 13 za $\tan(x)$ u $[\frac{-\pi}{4}, \frac{\pi}{4}]$ je:

$$1.00000014609x + 0.333324808x^3 + 0.13347672x^5 + 0.0529139x^7 + 0.0257829x^9 + 0.0013562x^{11} + 0.010269x^{13}$$

s apsolutnom pogreškom aproksimacije jednakom 8×10^{-9}, dok je minimaks racionalna aproksimacija s brojnikom stupnja 3 i nazivnikom stupnja 4 iste funkcije je:

$$\frac{0.9999999328x - 0.095875045x^3}{1 - 0.429209672x^2 + 0.009743234x^4}$$

s apsolutnom greškom aproksimacije jednakom 7×10^{-9}. U ovom slučaju, da bismo dobili istu točnost, potrebno je izvršiti 14 aritmetičkih operacija ako koristimo polinomnu aproksimaciju, odnosno 8 ako koristimo racionalnu aproksimaciju. Još jedna prednost racionalnih aproksimacija je njihova fleksibilnost, postoji mnogo različitih načina za pisanje iste racionalne funkcije. Na primjer, izrazi:

$$f_1(x) = \frac{3 - 9x + 15x^2 - 12x^3 + 7x^4}{1 - x + x^2}$$
$$f_2(x) = 3 - 5x + 7x^2 - \frac{x}{1 - x + x^2}$$
$$f_3(x) = 3 + x \times \frac{-6 + 12x - 12x^2 + 7x^3}{1 - x + x^2}$$

predstavljaju istu funkciju. Može se pokušati upotrijebiti ovo svojstvo kako bi se među raznim ekvivalentnim izrazima pronašao onaj koji minimizira pogrešku zaokruživanja. Pogreške su dane u tablici 2.2. Odmah vidimo da je u $[0, 1]$ izraz f_2 značajno bolji od f_1, i malo bolji od f_3.

	f_1	f_2	f_3
najgori slučaj	$0.3110887e^{-14}$	$0.1227446e^{-14}$	$0.1486132e^{-14}$
prosječni slučaj	$0.3378607e^{-15}$	$0.1847124e^{-15}$	$0.2050626e^{-15}$

Tablica 2.2: Pogreške dobivene prilikom evaluacije $f_1(x)$, $f_2(x)$ i $f_3(x)$ s dvostrukom preciznošću na 500000 pravilno raspoređenih vrijednosti između 0 i 1.

2.4 Dobivanje aproksimacija s posebnim ograničenjima

Ponekad je korisno razmotriti polinomne ili racionalne aproksimacije posebnog oblika. Na primjer, za funkciju sinus možemo koristiti izraz $x + x^3 p(x^2)$ kako bismo očuvali simetriju i smanjili broj množenja potrebnih za evaluaciju. Također, možemo razmotriti aproksimacije koje imaju fiksiranu vrijednost u nuli ili koje su dokazivo monotone. Sada ćemo proći idući primjer:

Primjer 2.4.1. *(**Sinus funkcija na intervalu** $[0, \frac{\pi}{8}]$) Pretpostavimo da želimo aproksimirati funkciju sinusa na* $[0, \frac{\pi}{8}]$*, s relativnom pogreškom ograničenom s* ϵ*, polinomom oblika:*

$$x + a_3x^3 + a_5x^5 + \cdots + a_{2n+1}x^{2n+1} = x + x^3 p(x^2)$$

gdje je $p(x) = a_3 + a_5x + a_7x^2 + \cdots + a_{2n+3}x^n$*. Želimo da je*

$$\left|\frac{\sin(x) - x - x^3 p(x^2)}{\sin(x)}\right| \leq \epsilon. \tag{2.3}$$

Podijelimo s x^3

$$\left|\frac{\frac{\sin(x)}{x^3} - \frac{1}{x^2} - p(x^2)}{\frac{\sin(x)}{x^3}}\right| \leq \epsilon.$$

Sad definirajmo $X = x^2$*. Jednadžba (2.3) jednaka je:*

$$\left|\frac{\frac{\sin(\sqrt{X})}{X^{3/2}} - \frac{1}{X} - p(X)}{\frac{\sin(\sqrt{X})}{X^{3/2}}}\right| \leq \epsilon.$$

Dakle, naš problem svodi se na pronalaženje polinomne aproksimacije $p(X)$ *koja minimizira maksimalnu razliku (minimaks) od funkcije*

$$\frac{\sin(\sqrt{X})}{X^{3/2}} - \frac{1}{X}$$

s težinskom funkcijom $X^{3/2}/\sin(\sqrt{X})$ *za* $X \in [0, \frac{\pi^2}{64}]$*. Koristeći Taylorov red:*

$$\frac{\sin(\sqrt{X})}{X^{3/2}} - \frac{1}{X} = -\frac{1}{6} + \frac{X}{120} - \frac{X^2}{5040} + \cdots + \frac{(-1)^{2n+1}X^n}{(2n+3)!} + \cdots,$$

možemo pronaći minimaks polinomnu aproksimaciju $p(X)$ stupnja 2 korištenjem Remezovog algoritma na prvih 6 članova gornjeg Taylorovog reda:

$$p(x) = -0.16666666480509 + 0.0083332602856X - 0.000197596738X^2$$

s greškom aproksimacije 0.14363×10^{-10}. *Sada kad u* $x + x^3 p(x^2)$ *ubacimo naš izračunati* $p(X)$ *dobivamo polinom*

$$x - 0.16666666480509x^3 + 0.0083332602856x^5 - 0.000197596738x^7$$

koji aproksimira $\sin x$ *na* $[0, \frac{\pi}{8}]$

2.5 Algoritmi i arhitekture za evaluaciju polinoma

U prethodnim odjeljcima proučavali smo kako se funkcija može aproksimirati polinomom ili racionalnom funkcijom. Kada pristupimo praktičnoj implementaciji ovih aproksimacija, ključno je odabrati metodu za evaluaciju polinoma kojom se smanjuje greška i/ili optimizira brzina postupka.

U situacijama kada koeficijenti polinoma ne ispunjavaju određeno svojstvo (kao što je, primjerice, jednostavna faktorizacija) koja bi mogla ubrzati računanje, preporučljivo je primijeniti Hornerovu shemu, primjer za polinom stupnja 4:

$$(((a_4x + a_3)x + a_2)x + a_1)x + a_0$$

U slučaju kada je stupanj polinoma značajno visok, moguće je primijeniti strategiju poznatu kao "prilagodba koeficijenata", koju je razmotrio Knuth. Ovaj pristup uključuje izvođenje jednog seta transformacija nad polinomom, koji se zatim koristi za njegovu evaluaciju sa znatno manjim brojem množenja u odnosu na Hornerovu shemu. Ova metoda počiva na sljedećem teoremu:

Teorem 2.5.1. ***(Knuth)*** *Neka je $u(x)$ polinom stupnja n*

$$u(x) = u_n x_n + u_{n-1} x_{n-1} + \cdots + u_1 x + u_0$$

Neka je $m = \lfloor n/2 \rfloor - 1$. Postoje parametri $c, \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m$ i $\beta_0, \beta_1, \beta_2, \ldots, \beta_m$ tako da se $u(x)$ može izračunati uz upotrebu najviše $\lfloor n/2 \rfloor + 2$ množenja i n zbrajanja, primjenjujući sljedeći izračun:

$$\begin{aligned}
y &= x + c \\
w &= y^2 \\
z &= (u_n y + a_0)y + \beta_0 \quad \textit{ako je n paran} \\
z &= u_n y + \beta_0 \quad \textit{ako je n neparan} \\
u(x) &= (\ldots((z(w - \alpha_1) + \beta_1)(w - \alpha_2) + \beta_2)\ldots)(w - \alpha_m) + \beta_m.
\end{aligned}$$

Izraz koji smo prethodno dobili za u(x) kao funkciju parametara $c, \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m$ i $\beta_0, \beta_1, \beta_2, \ldots, \beta_m$ rezultira u nelinearnom sustavu jednadžbi. U ovom sustavu, broj nepoznatih varijabli je 1 ili 2 plus broj jednadžbi; stoga općenito postoji rješenje za većinu vrijednosti c. Dobivanje koeficijenata $c, \alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_m$ i $\beta_0, , \beta_1, \beta_2, \ldots, \beta_m$ zahtijeva prilično zahtjevan proračun. U praksi, ovaj postupak može biti dugotrajan proces isprobavanja različitih vrijednosti (većina vrijednosti c može dovesti do složenih rješenja), no taj korak se izvodi samo jednom. Na primjer, ako je $n = 8, u_i = \frac{1}{i}$ za $i \geq 1, u_0 = 1$ i $c = 1$, sustav jednadžbi ima nekoliko rješenja. Jedno od njih je:

$$\begin{aligned}
\alpha_0 &= -0.85714285714286 \\
\alpha_1 &= -1.01861477121502 \\
\alpha_2 &= 0 \\
\alpha_3 &= -4.58138522878498 \\
\beta_0 &= 1.96666666666667 \\
\beta_1 &= -6.09666666666667 \\
\beta_2 &= 20.7534008337147 \\
\beta_3 &= -94.7138478361582
\end{aligned}$$

U ovoj situaciji, primjenom ove transformacije ostvarujemo mogućnost da se polinom evaluira koristeći samo šest množenja, što je manje nego osam potrebnih koristeći Hornerovu shemu. Prilikom konstruiranja posebnog hardvera, postoji mogućnost primjene algoritama i arhitektura za evaluaciju polinoma koji su ranije u prošlosti predloženi. Razmotrimo jedan takav pristup.

Estrinova metoda

Pretpostavimo da želimo evaluirati polinom stupnja 7:

$$a_7x^7 + a_6x^6 + \cdots + a_1x + a_0$$

Ukoliko postoji mogućnost simultanog izvođenja množenja i zbrajanja, koristi se Estrinov algoritam.

Algoritam 2 (Estrin)

- ulazni podaci: $a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0$ i x.
- izlazni podaci: $p(x) = a_7x^7 + a_6x^6 + \cdots + a_1x + a_0$

1. paralelno, izračunaj $X^{(1)} = x^2$, $a_3^{(1)} = a_7x + a_6$, $a_2^{(1)} = a_5x + a_4$, $a_1^{(1)} = a_3x + a_2$ i $a_0^{(1)} = a_1x + a_0$,

2. paralelno, izračunaj $X^{(2)} = (X^{(1)})^2$, $a_1^{(2)} = a_3^{(1)}X^{(1)} + a_2^{(1)}$ i $a_0^{(2)} = a_1^{(1)}X^{(1)} + a_0^{(1)}$,

3. izračunaj $p(x) = a_1^{(2)}X^{(2)} + a_0^{(2)}$.

Ovo je primjer algoritma za polinom stupnja 7, no može se proširiti za polinom bilo kojeg stupnja.

2.6 Pogreška evaluacije pod pretpostavkom da se koristi Hornerova shema

Dosad smo se bavili najvećom greškom koja se pojavljuje pri aproksimaciji funkcije polinomom. Međutim, moramo također uzeti u obzir drugu grešku - onu koja se javlja kada se polinomna aproksimacija evaluira u okviru aritmetike s konačnom preciznošću. U takvoj situaciji, greške zaokruživanja će se manifestirati prilikom gotovo svake aritmetičke operacije, što će dovesti do grešaka u konačnoj evaluaciji. Stoga je nužno odrediti precizne granice za tu evaluacijsku grešku. U ovom kontekstu, pretpostavljamo da se izračuni izvode koristeći aritmetiku s pomičnom točkom. Neka je

$$p(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$$

polinom stupnja n. Pristupamo pretpostavci da se a_i mogu točno prikazati u formatu s pomičnom točkom. Nadalje, uzimamo u obzir da se polinom računa korištenjem Hornerove sheme. Naš cilj je čvrsto ograničiti maksimalnu moguću pogrešku evaluacije za $x \in [x_{min}, x_{max}]$.

Evaluacija pomoću zbrajanja i množenja s pomičnom točkom

Definicije

Pretpostavljamo da računamo vrijednost $p(x)$ pomoću Hornerovog pravila. Dodatno, pretpostavljamo da su temeljne operacije koje koristimo zbrajanje i množenje u sustavu pomične točke te da smo odabrali zaokruživanje prema najbližoj vrijednosti. Definirajmo

$$\begin{cases} P^*[i] = a_n x^{n-i+1} + a_{n-1}x^{n-1} + \cdots + a_i x \\ S^*[i-1] = a_n x^{n-i+1} + a_{n-1}x^{n-1} + \cdots + a_i x + a_{i-1} \end{cases}$$

Ove varijable označavaju "točne" vrijednosti koje bi bile redom izračunate ($i = n, n-1, \cdots, 0$), tijekom Hornerove evaluacije $p(x)$, za dani $x \in [x_{min}, x_{max}]$ bez pogrešaka za-

okruživanja. Točna vrijednost $p(x)$ je $S^*[0]$. Označit ćemo $P[i]$ i $S[i]$ kao izračunate vrijednosti $P^*[i]$ i $S*[i]$, koristeći relacije:

$$\begin{cases} P[i-1] = S[i-1] \otimes x \\ S[i-1] = P[i] \oplus a_{i-1} \end{cases}$$

gdje su $\oplus$ i $\otimes$ zbrajanje i množenje s pomičnim točkom. Izračunata vrijednost od $p(x)$ je $S[0]$. Izgradit ćemo donje i gornje granice za $P[i]$ i $S[i]$: $P_{max}[i]$, $P_{min}[i]$, $S_{max}[i]$, $S_{min}[i]$. Za dobivanje ovih vrijednosti, trebat će nam dodatne varijable: $\hat{P}_{max}[i]$ i $\hat{P}_{min}[i]$ će ograničavati egzaktne vrijednosti $S[i]x$, dok će $\hat{S}_{max}[i]$ i $\hat{S}_{min}[i]$ ograničiti točnu vrijednost izraza $P[i] + a_{i-1}$.

Za određivanje gornje granice greške koja se pojavljuje pri evaluaciji $p(x)$ u okviru aritmetike s pomičnim točkom, potrebno je procijeniti sljedeće granice greške:

- $\delta[i]$ je gornja granica pogreške množenja s pomičnim točkom $P[i] = S[i] \otimes x$
- $\epsilon[i-1]$ je gornja granica pogreške zbrajanja s pomičnim točkom $S[i-i] = P[i] \oplus a_{i-1}$

Definirajmo sada $err[i]$ kao gornju granicu za $|S^*[i] - S[i]|$. Naš cilj je izračunati vrijednost $err[0]$ koja predstavlja traženu granicu konačne greške evaluacije. Izračunat ćemo je iterativno, počevši od $err[n] = 0$.

Iterativno izračunavanje granica pogreške

Započinjemo sa početnim vrijednostima $S_{min}[n] = S_{max}[n] = a_n$ i postavljamo $err[n] = 0$. Sada pretpostavimo da već imamo informacije o $S_{min}[i]$, $S_{max}[i]$ i $err[i]$. Razmotrimo kako bismo iz toga mogli izvesti $S_{min}[i-1]$, $S_{max}[i-1]$ i $err[i-1]$. Prvo, jednostavno nalazimo (kroz standardno intervalno množenje)

$$\hat{P}_{min}[i] = \min\{S_{min}[i]x_{min}, S_{min}[i]x_{max}, S_{max}[i]x_{min}, S_{max}[i]x_{max}\}$$

i

$$\hat{P}_{max}[i] = \max\{S_{min}[i]x_{min}, S_{min}[i]x_{max}, S_{max}[i]x_{min}, S_{max}[i]x_{max}\}.$$

S obzirom da koristimo ispravno zaokruženo množenje, pri čemu se primjenjuje zaokruživanje prema najbližem broju, pogreška zaokruživanja koja se manifestira prilikom izračunavanja $S[i] \otimes x$ predstavlja gornju granicu je

$$\frac{1}{2} ulp(S[i]x).$$

Budući da je $ulp(t)$ rastuća funkcija od $|t|$,a također imajući na umu da je $S[i]x$ iz $[\hat{P}_{min}[i], \hat{P}_{max}[i]]$, iz toga proizlazi sljedeća ograničenja zaokruživanja greške

$$\delta[i] = \frac{1}{2} ulp\Big(\max\Big\{|\hat{P}_{min}[i]|, |\hat{P}_{max}[i]|\Big\}\Big).$$

Odavde pronalazimo granice za $P[i]$:

$$\begin{cases} P_{min}[i] = \hat{P}_{min}[i] - \delta[i] \\ P_{max}[i] = \hat{P}_{max}[i] + \delta[i], \end{cases}$$

na sličan način dobijemo:

$$\begin{cases} \hat{S}_{min}[i-i] = \hat{P}_{min}[i] + a_{i-1} \\ \hat{S}_{max}[i-1] = \hat{P}_{max}[i] + a_{i-1}. \end{cases}$$

Na temelju ovih vrijednosti zaključujemo grešku koja nastaje tokom izračuna $S[i-1]$:

$$\epsilon[i-1] = \frac{1}{2} ulp\Big(\max\Big\{|\hat{S}_{min}[i-1]|, |\hat{S}_{max}[i-1]|\Big\}\Big),$$

pa onda opet iz toga pronalazimo granice za $S[i-1]$:

$$\begin{cases} S_{min}[i-1] = \hat{S}_{min}[i-1] - \epsilon[i-1] \\ S_{max}[i-1] = \hat{S}_{max}[i-1] + \epsilon[i-1]. \end{cases}$$

I sada napokon na kraju možemo izračunati $err[i-1]$:

$$err[i-1] = err[i] \max\Big\{|x_{min}|, |x_{max}|\Big\} + \delta[i] + \epsilon[i-1].$$

Evaluacija pomoću kombiniranih operacija množenja-zbrajanja

Definicije

Sada pretpostavimo da je dana arhitektura opremljena sa instrukcijom kombiniranog množenja-zbrajanja (FMA) i da koristimo tu instrukciju za implementaciju Hornerove sheme za evaluaciju polinoma. Ovo omogućuje evaluaciju izraza $ax + b$ sa samo jednim korakom konačnog zaokruživanja.

Kao i prije, definiramo:

$$S^*[i] = a_n x^{n-i} + a_{n-1} x^{n-i-1} + \cdots + a_i,$$

također ćemo definirati $S[i]$ kao izračunatu vrijednost, kada je x poznata, za $S^*[i]$ koristeći:

$$S[i-1] = (S[i]x + a_{i-1}) \quad \text{zaokruženo na najbliži},$$

s početnom vrijednosti $S[n] = a_n$, izračunat ćemo donje i gornje granice za $S[i]$, koje označavamo kao $S_{min}[i]$ i $S_{max}[i]$. Da bismo to postigli, koristit ćemo privremene varijable $\hat{S}_{min}[i-1]$ i $\hat{S}_{max}[i-1]$ koje ograničavaju točnu vrijednost izraza $(S[i]x + a_{i-1})$, i varijablu $\epsilon[i]$ koja ograničava grešku zaokruživanja koja se javlja pri izračunu $S[i]$ iz $S[i+1]$.

Kao što je već opisano u prethodnom odjeljku, $err[i]$ predstavlja gornju granicu za $|S^*[i] - S[i]|$. Naš cilj je izračunati $err[0]$, što je konačna greška evaluacije koju tražimo. Računat ćemo je iterativno, počevši od $err[n] = 0$.

Iterativno izračunavanje pogreške evaluacije

Iterativni proces koji daje $err[0]$ vrlo je sličan onome opisanom u prethodnom odjeljku. Krećemo od osnovnih vrijednosti: $S_{min}[n] = S_{max}[n] = a_n$ i $err[n] = 0$. Nakon toga, pretpostavljamo da već imamo informacije o $S_{min}[i]$, $S_{max}[i]$ i $err[i]$. Sada ćemo pokazati kako izračunati $S_{min}[i-1]$, $S_{max}[i-1]$ i $err[i-1]$. Definiramo:

$$\hat{S}_{min}[i-1] = a_{i-1} + \min\{S_{min}[i]x_{min}, S_{min}[i]x_{max}, S_{max}[i]x_{min}, S_{max}[i]x_{max}\}$$

i

$$\hat{S}_{max}[i-1] = a_{i-1} + \max\{S_{min}[i]x_{min}, S_{min}[i]x_{max}, S_{max}[i]x_{min}, S_{max}[i]x_{max}\}.$$

Zatim iz toga dobivamo:

$$\epsilon[i-1] = \frac{1}{2}ulp\Big(\max\Big\{|\hat{S}_{min}[i-1]|, |\hat{S}_{max}[i-1]|\Big\}\Big)$$

što daje sljedeće donje i gornje granice:

$$\begin{cases} S_{min}[i-1] = \hat{S}_{min}[i-1] - \epsilon[i-1] \\ S_{max}[i-1] = \hat{S}_{max}[i-1] + \epsilon[i-1]. \end{cases}$$

Sada imamo sve potrebne informacije za izračunavanje $err[i-1]$:

$$err[i-1] = err[i]\max\Big\{|x_{min}|, |x_{max}|\Big\} + \epsilon[i-1].$$

Poglavlje 3

Metode temeljene na tablici

U situacijama kada želimo aproksimirati funkciju na velikoj domeni primjenom tehnika opisanih u Poglavlju 2, možemo se naći u situaciji da nam trebaju polinomi ili racionalne funkcije visokog stupnja. Ovo može rezultirati dugim vremenom izračuna i izazvati probleme u kontroli numeričkih pogrešaka. Jedan prirodan pristup rješavanju ovog problema je podjela ciljanog intervala na manje podintervale. Dovoljno je za svaki od tih podintervala pohraniti koeficijente aproksimacije malog stupnja koja je valjana unutar tog intervala. Razmotrimo sljedeći primjer.

Primjer 3.0.1. *(**Sinus funkcija na intervalu** $[0, \frac{\pi}{4}]$) Želimo aproksimirati funkciju sinus u intervalu* $[0, \pi/4]$*, s pogreškom manjom od* 10^{-8}*. Prema Tablici 3.1, ako ne podijelimo interval* $[0, \pi/4]$ *i koristimo samo jednu polinomnu aproksimaciju, tada će biti potreban polinom stupnja 6. No, kako pokazuje Tablica 3.2, ako podijelimo taj interval na dva jednaka podintervala, bit će dovoljan polinom stupnja 5. Dodatno, Tablica 3.3 ukazuje da podjela intervala* $[0, \pi/4]$ *na 4 jednaka podintervala omogućava aproksimacije stupnja 4.*

interval	stupanj	pogreška
$[0, \frac{\pi}{4}]$	5	0.609×10^{-7}
	6	0.410×10^{-8}
	7	0.418×10^{-10}

Tablica 3.1: Minimaks aproksimacija za $\sin(x)$, $x \in [0, \pi/4]$, pomoću jednog polinoma. Ovdje su navedene apsolutne pogreške.

Ovaj primjer pokazuje da se značajno smanjuje vrijeme potrebno za računanje kada se domena razdijeli na manje dijelove. Posebno treba obratiti pažnju na rubnim područjima ako želimo sačuvati svojstva poput monotonosti. Za većinu uobičajenih funkcija nije potrebno ponovno izračunavati i pohranjivati novu polinomnu ili racionalnu aproksimaciju

interval	stupanj	pogreška
$[0, \frac{\pi}{8}]$	4	0.148×10^{-6}
	5	0.486×10^{-9}
	6	0.342×10^{-10}
$[\frac{\pi}{8}, \frac{\pi}{4}]$	4	0.126×10^{-6}
	5	0.138×10^{-8}
	6	0.289×10^{-10}

Tablica 3.2: Minimaks aproksimacija za $\sin(x)$, $x \in [0, \pi/4]$, pomoću dva polinoma. Ovdje su navedene apsolutne pogreške.

interval	stupanj	pogreška
$[0, \frac{\pi}{16}]$	3	0.478×10^{-7}
	4	0.472×10^{-8}
	5	0.382×10^{-11}
$[\frac{\pi}{16}, \frac{\pi}{8}]$	3	0.140×10^{-6}
	4	0.454×10^{-8}
	5	0.113×10^{-10}
$[\frac{\pi}{8}, \frac{3\pi}{16}]$	3	0.228×10^{-6}
	4	0.418×10^{-8}
	5	0.183×10^{-10}
$[\frac{3\pi}{16}, \frac{\pi}{4}]$	3	0.307×10^{-6}
	4	0.367×10^{-8}
	5	0.246×10^{-10}

Tablica 3.3: Minimaks aproksimacija za $\sin(x)$, $x \in [0, \pi/4]$, pomoću četiri polinoma. Ovdje su navedene apsolutne pogreške.

za svaki podinterval. Umjesto toga, možemo iskoristiti jednostavna algebarska svojstva kao što su $e^{a+b} = e^a e^b$. Na primjer, prilikom računanja eksponencijalne funkcije u domeni oblika $[0, a]$, s podintervalima jednake širine, dovoljno je imati aproksimaciju koja vrijedi u prvom podintervalu. U podintervalu $[a_k, a_{k+1}]$, vrijednost eksponencijalne funkcije za x je jednaka e^{a_k} pomnoženo s eksponencijalnom funkcijom za $x - a_k$, a $x - a_k$ očigledno pripada prvom podintervalu.

U ovom poglavlju proučavamo dvije različite klase metoda temeljenih na tablicama. Izbor metode ovisi o načinu implementacije (softver, hardver) i mogućoj dostupnosti "radne preciznosti" (tj. preciznosti korištene za međuizračune), koja je značajno veća od "ciljane preciznosti"(tj. izlaznog formata):

- metode koje koriste "standardnu tablicu" (tj. funkcija je tabulirana na podjednakim udaljenim vrijednostima) i polinomnu ili racionalnu aproksimaciju. Tangovi "tablicom vođeni" algoritmi pripadaju ovoj klasi metoda;
- metode koje koriste "točne tablice" (tj. funkcija je tabulirana na gotovo jednako udaljenim točkama). Galova "metoda točnih tablica" pripada ovoj klasi metoda.

3.1 Tablicom vođeni algoritmi

Tang predlaže smjernice za implementaciju osnovnih funkcija pomoću algoritama pretraživanja tablice. Za izračunavanje $f(x)$, njegovi algoritmi koriste tri osnovna koraka:

- **redukcija**: Iz ulaznog argumenta x, deduciramo varijablu y koja pripada vrlo maloj domeni, tako da se $f(x)$ lako može deducirati iz $f(y)$ (ili, eventualno, iz neke funkcije $g(y)$).
- **aproksimacija**: $f(y)$ (ili $g(y)$) se računa pomoću aproksimacije malog stupnja polinoma;
- **rekonstrukcija**: $f(x)$ se izvodi iz $f(y)$ (ili $g(y)$)

Sada ćemo razmotriti primjer gdje ćemo detaljnije analizirati algoritam koji je predložio Tang za $exp(x)$ u aritmetici pomične točke IEEE dvostruke preciznosti.

Tangov algoritam za $exp(x)$ u aritmetici pomične točke IEEE

Pretpostavimo da želimo izračunati exp(x) u aritmetici pomične točke IEEE dvostruke preciznosti. Tang prvo predlaže smanjenje ulaznog argumenta na vrijednost r unutar intervala:

$$\left[-\frac{\ln 2}{64}, +\frac{\ln 2}{64}\right],$$

zatim, da aproksimiramo $exp(r) - 1$ pomoću polinoma $p(r)$, i na kraju rekonstruiramo $exp(x)$ formulom

$$exp(x) = 2^m(2^{j/32} + 2^{j/32}p(r)),$$

gdje su j i m takvi da je

$$x = (32m + j)\frac{\ln 2}{32} + r, \quad 0 \leq j \leq 31. \tag{3.1}$$

Ovi koraci se implementiraju na sljedeći način.

redukcija: Kako bismo postigli veću preciznost u računanju, Tang predstavlja reducirani argument r kao zbroj dva broja s pomičnim točkom, r_1 i r_2, takvi da je $r_2 \ll r_1$ i $r_1 + r_2$ aproksimira r s većom preciznošću od radne preciznosti. Za to koristi tri broja s pomičnom točkom, L^{left}, L^{right} i Λ, pri čemu:

- Λ je broj $32/\ln 2$ zaokružen na dvostruku preciznost;
- L^{left} ima nekoliko nula na kraju;
- $L^{right} \ll L^{left}$, i $L^{left} + L^{right}$ aproksimiraju $\ln 2/32$ s preciznošću znatno većom od radne preciznosti.

Vrijednosti r_1 i r_2 izračunavaju se na sljedeći način. Neka je $N = x \times \Lambda$ zaokruženo na najbliži cijeli broj. Definiramo $N_2 = N \mod 32$ i $N_1 = N - N_2$. Tada izračunavamo, s radnom preciznošću:

$$r_1 = x - N \times L^{left}$$

i

$$r_2 = -N \times L^{right}.$$

Vrijednosti m i j iz izraza (3.1) su $m = N_1/32$ i $j = N_2$.

aproksimacija: $p(r)$ se računa na sljedeći način. Prvo, računamo $r = r_1 + r_2$ u radnoj preciznosti. Drugo, računamo

$$Q = r \times r \times (a_1 + r \times (a_2 + r \times (a_3 + r \times (a_4 + r \times a_5)))),$$

gdje su a_i koeficijenti minimaks aproksimacije. Konačno, dobivamo

$$p(r) = r_1 + (r_2 + Q).$$

Izraz r_2 koristi se samo u izrazu reda 1.

rekonstrukcija: Vrijednosti $s_j = 2^{j/32}, \quad j = 0, ..., 31$, prethodno su izračunate s većom preciznošću i predstavljene su s dva broja dvostruke preciznosti s_j^{left} i s_j^{right} tako da:

- $s_j^{left} \gg s_j^{right}$;
- šest zadnjih znamenki s_j^{left} su jednake nuli;
- $s_j = s_j^{left} + s_j^{right}$ do otprilike 100 bitova preciznosti.

Neka je S_j aproksimacija dvostruke preciznosti za s_j. Tada računamo

$$exp(x) = 2^m \times \left(s_j^{left} + \left(s_j^{right} + S_j \times p(r)\right)\right).$$

3.2 Galova metoda točnih tablica

Ova metoda pripisuje se Galu i bila je implementirana za strojeve tipa IBM/370. Relativno nedavna implementacija, posebno prilagođena strojevima koji koriste IEEE-754 aritmetiku s pomičnom točkom, opisana je od strane Gala i Bachelisa 1991. godine. Metoda se sastoji od tabuliranja funkcije koja se računa na gotovo jednako raspoređenim točkama koje zovemo "strojni brojevi" (tj. točno su reprezentativni u sustavu s pomičnom točkom koji se koristi), pri čemu je vrijednost funkcije vrlo blizu broju stroja. Na ovaj način simuliramo veću točnost. Razmotrimo sljedeći primjer.

Primjer 3.2.1. *Pretpostavimo da koristimo računalo s bazom 10 i mantisom od 4 znamenke te da želimo izračunati eksponencijalnu funkciju na intervalu* $[\frac{1}{2}, 1]$. *Prvo rješenje se dobije pohranjivanjem pet vrijednosti:* $e^{0.55}$, $e^{0.65}$, $e^{0.75}$, $e^{0.85}$ *i* $e^{0.95}$ *u tablicu. Zatim, na intervalu* $[\frac{i}{10}, \frac{i+1}{10}]$ *(gdje je* $i = 5, ..., 9$*), eksponencijalnu funkciju* x *aproksimiramo kao* $\exp\left(\frac{i+1/2}{10}\right)$ *(pohranjene vrijednosti) plus ili pomnoženo s polinomnom funkcijom u* $x - \frac{i+1/2}{10}$. *Vrijednosti pohranjene u tablici su:*

x	e^x	spremljena vrijednost	$\lvert greska \rvert$
0.55	$1.733253\cdots$	1.733	2.5×10^{-4}
0.65	$1.915540\cdots$	1.916	4.6×10^{-4}
0.75	$2.117000\cdots$	2.117	1.7×10^{-8}
0.85	$2.339646\cdots$	2.340	3.5×10^{-4}
0.95	$2.585709\cdots$	2.586	2.9×10^{-4}

Pogreška zaokruživanja prilikom pohrane vrijednosti iznosi 4.6×10^{-4} *u najgorem slučaju, s prosječnom vrijednošću greške od* 2.7×10^{-4}. *Sada ćemo pokušati koristiti Galovu metodu. Pohranjujemo vrijednosti eksponencijalne funkcije u točkama* X_i *koje zadovoljavaju sljedeće uvjete:*

1. *Te vrijednosti moraju biti točno reprezentirane u brojevnom sustavu koji se koristi (baza 10, 4 znamenke);*
2. *Moraju biti blizu vrijednostima koje su prethodno pohranjene;*
3. e^{X_i} *treba biti vrlo blizu broja koji je točno reprezentiran u brojevnom sustavu koji se koristi.*

Takve vrijednosti mogu se pronaći iscrpnim ili slučajnim pretraživanjem. Moguće je koristiti sljedeće vrijednosti:

X_i	e^{X_i}	spremljena vrijednost	$\|greska\|$
0.5487	$1.73100125\cdots$	1.731	1.2×10^{-6}
0.6518	$1.91899190\cdots$	1.919	8.1×10^{-6}
0.7500	$2.11700001\cdots$	2.117	1.7×10^{-8}
0.8493	$2.33800967\cdots$	2.338	9.6×10^{-6}
0.9505	$2.58700283\cdots$	2.587	2.8×10^{-6}

Uočavamo sada da pogreška zaokruživanja iznosi 9.6×10^{-6} *u najgorem slučaju, a prosječna vrijednost greške iznosi* 4.3×10^{-6}. *Stoga je ova tablica 60 puta preciznija u prosječnom slučaju i 50 puta u najgorem slučaju u usporedbi s prethodnom.*

Bibliografija

[1] J. M. Muller, *Elementary functions*, Birkhäuser, 2006.

Sažetak

U ovom radu su opisane metode i algoritmi za aproksimaciju elementarnih funkcija. Prvo se daju neke osnovne stvari i pojmovi o aritmetici pomične točke. Glavni fokus je na izgradnji različith vrsta polinomnih i racionalnih aproksimacija. Započinjemo prvo s dvije vrste polinomnih aproksimacija, aproksimacija najmanjih kvadrata i minimaks aproksimacija. Pokazujemo jedan iterativni algoritam (Remezov algoritam) koji konstruira minimaks aproksimaciju primjenom Čebiševljevog teorema za karakterizaciju minimaks aproksimacije funkcije. Nakon toga slijede racionalne aproksimacije koje su zapravo nadogradnja na polinomne aproksimacije (polinom je racionalna funkcija s nazivnikom 1). Zatim se govori o metodama za evaluaciju polinoma i greške koje nastaju prilikom aproksimacije gdje se najčešće koristi Hornerova shema. Konačno, bavimo se metodama temeljenih na tablicama koje koristimo kada trebamo aproksimirati funkciju na velikom intervalu. Razmatraju se dvije klase metoda, jedna koristi standarnu tablicu i polinomnu ili racionalnu aproksimaciju, a druga metoda točne tablice.

Summary

This thesis describes methods and algorithms for the approximation of the elementary functions. It begins by providing some fundamental concepts and principles of floating-point arithmetic. The main focus is on the construction of various types of polynomial and rational approximations. We start with two types of polynomial approximations: least squares approximation and minimax approximation. An iterative algorithm (Remez algorithm) is demonstrated, which constructs a minimax approximation using Chebyshev's characterization theorem of the minimax approximation of a function. Subsequently, rational approximations are discussed, which are essentially extensions of the polynomial approximations (a polynomial is a rational function with a denominator of 1). The thesis then explores methods for polynomial evaluation and the errors that arise during approximation, with Horner's scheme being commonly used. Finally, it delves into table-based methods that are employed when approximating a function over a large interval. Two classes of methods are considered, one using a standard table and a polynomial or rational approximation, and the other utilizing accurate tables.

Životopis

Rođen sam u Zagrebu 29. svibnja 1995. godine. U istom gradu 2014. godine završavam VII. gimnaziju. Iste godine na Prirodoslovno-matematičkom fakultetu u Zagrebu upisujem studij Matematike. Preddiplomski studij završavam 2019. godine i upisujem diplomski studij Računarstvo i matematika na istom fakultetu.