Uvod u numeričku matematiku

M. Klaričić Bakula

Ožujak, 2009.

1 Uvod

Jedan od osnovnih zadataka numeričke matematike je izračunavanje vrijednosti funkcije u nekoj točki ili na nekom skupu točaka, tj. **izvrednjavanje funkcije**. Zašto baš to?

Efikasno možemo raditi samo s onim funkcijama za koje imamo dobar algoritam za izvrednjavanje. Pri tom moramo voditi računa o tome da aritmetika računala u stvarnosti podržava samo četiri osnovne aritmetičke operacije (odnosno, strogo gledano dvije). Korištenje raznih elementarnih funkcija (eksponencijalnih, logaritamskih, trigonometrijskih) ovisi o kvaliteti aproksimacije uporabljene u odgovarajućem programskom jeziku ili odgovarajućem sklopu računala. Nadalje, računanje s osnovnim operacijama nije egzaktno jer se prilikom svake operacije stvaraju greške zaokruživanja. Zbog toga prilikom konstrukcije nekog algoritma imamo sva ciljna uvjeta:

- **efikasnost** tj. **brzina** (što manje elementarnih operacija);
- **točnost**, tj. **stabilnost** na greške zaokruživanja.

Kod konstrukcije algoritma za izvrednjavanje oba zahtjeva postaju posebno bitna jer se izvrednjavanje obično provodi velik broj puta, pa mala poboljšanja u konačnici znače puno. Općenito očekujemo da brži algoritam ima i manju grešku jer je izvedeno manje operacija, no ovo **ne mora biti istina**.

U ovom poglavlju više će nas zanimati efikasnost nego točnost, osim u slučaju kada je ova potonja znatno ugrožena, tj. kada se javlja velika nestabilnost.

Neka je zadana funkcija $f : D \to \mathbb{R},\ D \subseteq \mathbb{R}$. Zadatak je izračunati vrijednost funkcije f u točki $x_0 \in D$. Točnije, moramo naći algoritam koji računa $f(x_0)$ za bilo koji $x_0 \in D$. Naravno, postavlja se pitanje kako se zadaje funkcija f : očito f mora biti zadana s najviše konačno mnogo podataka i ti podaci moraju jednoznačno određivati f. To je fundamentalno ograničenje i ono bitno smanjuje klasu funkcija kojima se uopće mogu algoritamski izračunati vrijednosti. Ovim pitanjima se bavi **teorija izračunljivosti** u okviru **matematičke logike** i **osnova matematike**.

U stvarnosti se nameću i bitno veća ograničenja. Naime, ako imamo na raspolaganju samo 4 aritmetičke operacije, onda su racionalne funkcije jedine funkcije kojima možemo računati vrijednosti u bilo kojoj točki domene. Takve funkcije možemo jednoznačno zadati konačnim brojem parametara, npr. konačnim brojem čvorova ili koeficijentima u nekom prikazu funkcije. Dakle, sigurno trebamo algoritme za izvrednjavanje racionalnih funkcija, a kako su ove kvocijenti dvaju polinoma zgodno je imati i algoritme za izvrednjavanje polinoma. S druge strane, polinomi su jednostavne funkcije koje imaju veliku teorijsku i praktičnu primjenu, a prilikom njihovog izvrednjavanja se ne javlja dijeljenje i definirani su za sve realne (ili kompleksne) brojeve.

Strogo govoreći, sve ostale funkcije moramo aproksimirati polinomima ili racionalnim funkcijama. U praksi možemo pretpostaviti da za neke osnovne matematičke funkcije već imamo dobre aprosimacije (tj. algoritme za izvrednjavanje) i to bilo unutar hardwarea ili unutar softwarea. U tu klasu bi spadale opće potencije, eksponencijalne, trigonometrijske, hiperbolne, arcus i area funkcije. Ovo je važno jer ih u tom slučaju možemo koristiti u našim algoritmima uz sigurnost da uzeta vrijednost funkcije ima malu relativnu grešku u odnosu na preciznost konačne aritmetike (strojnu preciznost).

Reklo bi se da je ovim sve riješeno, no nije tako. Računanje $f(x_0)$ katkad traje i puno dulje nego trajanje jedne osnovne aritmetičke operacije: ono je ponekad deset i više puta duže. Zbog toga izbjegavamo puno poziva takvih funkcija ako je to ikako moguće izbjeći (tj. efikasno i bez većeg gubitka točnosti).

U ovom poglavlju ćemo se upoznati s nekim općim algoritmima tog tipa.

2 Hornerova shema

Polinomi su najjednostavnije algebarske funkcije. Možemo ih definirati nad bilo kojim prstenom R u obliku

$$p\left(x\right)=\sum_{i=0}^{n}a_ix^i,\quad n\in\mathbb{N}_0,$$

gdje su a_i koeficijenti iz prstena R, a x simbolička varijabla. Polinomi, kao simbolički objekti, imaju i sami strukturu prstena.

Međutim, polinome možemo interpretirati i kao funkcije koje se daju izvrednjavati u svim točkama x_0 prstena R uvrštavanjem x_0 umjesto simboličke varijable x. Dobiveni rezultat $p\left(x_0\right)$ je opet u R. Nas zanimaju efikasni algoritmi za računaje te vrijednosti. Složenost postupka očito ovisi o broju članova u sumi, a da broj članova ne bi bio umjetno prevelik uzimamo da je $p\neq 0$ i $a_n\neq 0$ (tj. $\partial p=n$).

Mi ćemo pretpostavljati da radimo isključivo s polinomoma nad $\mathbb{R}$ i $\mathbb{C}$ jer nam u praksi to najčešće i treba. Znamo da za svaki $n \in \mathbb{N}$ skup svih polinoma stupnja n nad $\mathbb{R}$ $(\mathbb{C})$ čini vektorski potprostor vektorskog prostora svih polinoma nad $\mathbb{R}$ $(\mathbb{C})$.

Polinom se obično zadaje stupnjem n i koeficijentima $a_0, a_1, \ldots, a_n$ u nekoj bazi vektorskog prostora polinoma stupnja ne većeg od n. Mi ćemo za sada koristiti standardnu bazu

$$\mathcal{B} = \{1, x, \ldots, x^n\} ,$$

a kasnije ćemo koristiti i neke druge, nestandardne baze.

2.1 Računanje vrijednosti polinoma u točki

Neka je zadan polinom stupnja $n \in \mathbb{N}$

$$p_n(x) = \sum_{i=0}^{n} a_i x^i, \quad a_n \neq 0$$

kojemu treba izračunati vrijednost u točki x_0. To se može napraviti na više načina, a najočitiji je onaj direktni po zapisu. Krenemo li od nulte potencije $x^0 = 1$, svaku sljedeću dobijemo rekurzivno iz

$$x^k = x \cdot x^{k-1}, \quad k = 1, \ldots, n.$$

Dakle, imamo li zapamćen x^{k-1}, lako izračunamo x^k pomoću samo jednog množenja.

ALGORITAM (Vrijednost polinoma s pamćenjem potencija, $p_n(x_0) = sum$)

$sum := a_0;$
$pot := 1;$
for $i := 1$ **to** n **do**
    begin
    $pot := pot * x_0;$
    $sum := sum + a_i * pot;$
    end;

Uočimo da ukupno imamo $2n$ množenja i n zbrajanja ako je riječ o relnom polinomu, dok se n udvostručava ako je riječ o kompleksnom.

Izvrednjavanje polinoma možemo izvesti i s manje množenja. Ako polinom p_n zapišemo u obliku

$$p_n(x) = (\cdots((a_n x + a_{n-1})x + a_{n-2})x + \cdots + a_1) + a_0,$$

onda imamo jednostavniji algoritam za računanje $p_n(x_0)$ koji je inače poznat pod imenom **Hornerova shema**, a koristi se od davne 1819. godine (sličan zapis polinoma koristio je Newton još 1669. godine!).

ALGORITAM (Hornerova shema, $p_n(x_0) = sum$)

$sum := a_n;$
$pot := 1;$
for $i := n-1$ **downto** 0 **do**
  begin
  $sum := sum * x_0 + a_i;$
  end;

Odmah je očito da smo broj množenja prepolovili, pa je njegova složenost n množenja i n zbrajanja ako je riječ o relnom polinomu.

2.2 Hornerova shema je optimalan algoritam

Lako se vidi da uporaba Hornerove sheme ima smisla samo kada je večina koeficijenata promatranog polinoma različita od nule. Npr., polinom zadan s

$$p_{100}(x) = x^{100} + 1$$

očigledno nema smisla izvrednjavati Hornerovom shemom. U ovom slučaju najbolje bi bilo izabrati binarno potenciranje

$$x \to x^2 \to x^4 \to x^8 \to x^{16} \to x^{32} \to x^{64}$$

$$x^{64} \cdot x^{32} \cdot x^4 = x^{100}$$

jer bismo imali ukupno 8 množenja i 1 zbrajanje.

Također, kada izvrednjavamo polinom koji ima samo parne koeficijente Hornerovu shemu trebamo modificirati tako da koristi samo parne potencije $\left(x \to x^2\right)$.

Za Hornerovu shemu se može pokazati da je optimalan algoritam.

TEOREM (Borodin, Muro) Za izvrednjavanje općeg polinoma n-tog stupnja potrebno je barem n aktivnih množenja (tj. množenja između a_i i x).

Napomenimo da se rezultat ovog teorema može poboljšati samo ako izvrednjavamo isti polinom u puno točaka. U tom se slučaju koeficijenti polinoma **adaptiraju** tako da bismo kasnije imali što manje operacija po svakoj pojedinoj točki. No u to nećemo ulaziti.

Zanimljivo je da je Hornerova shema optimalna za polinome stupnja ne većeg od tri čak i ako računamo vrijednost polinoma u više točaka. No pogledajmo jedan primjer adaptiranja koeficijenata za polinom stupnja 4.

PRIMJER. Neka je zadan opći polinom p_4 stupnja 4 s

$$p_4(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

i promatrajmo sljedeću shemu računanja koja ima 3 množenja i 5 zbrajanja

$$\begin{aligned} y &= (x + c_0)\, x + c_1, \\ p_4(x) &= ((y + x + c_2)\, y + c_3)\, c_4. \end{aligned}$$

Prvo treba odrediti c_i, pa zato raspišemo p_4 po potencijama. Dobijemo

$$\begin{aligned} p_4(x) &= c_4 x^4 + (2c_0c_4 + c_4)\, x^3 + \left(c_0^2 + 2c_1 + c_0c_4 + c_2c_4\right) x^2 \\ &\quad + (2c_0c_1c_4 + c_1c_4 + c_0c_2c_4)\, x + \left(c_1^2c_4 + c_1c_2c_4 + c_3c_4\right). \end{aligned}$$

Nakon izjednačavanja polinoma i rješavanja sustava dobijemo

$$\begin{array}{cc} c_4 = a_4 & c_1 = a_1/a_4 - c_0 b \\ c_0 = (a_3/a_4 - 1)/2 & c_2 = b - 2c_1 \\ b = a_2/a_4 - c_0(c_0 + 1) & c_3 = a_0/a_4 - c_1(c_1 + c_2) \end{array}.$$

Ovo zahtjeva dosta računanja, no to radimo samo jednom: nakon toga će svako izvrednjavanje zahtijevati manje vremena nego Hornerova shema na polaznom polinomu.

Razlog je to što zbrajanje troši manje računalnog vremena nego množenje. Štoviše, vrijedi sljedeći teorem.

TEOREM (Pan) Za bilo koji realni polinom p_n stupnja $n \geq 3$ postoje realni brojevi $c, d_i, e_i,\ i = 0, \ldots, \lceil n/2 \rceil - 1$, takvi da se $p_n(x)$ može izračunati korištenjem $\lfloor n/2 \rfloor + 2$ množenja i n zbrajanja po sljedećoj shemi

$$\begin{aligned}
y &= x + c \\
w &= y^2 \\
z &= \begin{cases} (a_n y + d_0)\, y + e_0, & n \text{ paran} \\ \quad a_n y + e_0, & n \text{ neparan} \end{cases} \\
z &= z\,(w - d_i) + e_i, \quad i = 0, \ldots, \lceil n/2 \rceil - 1.
\end{aligned}$$

Uočimo da nije rečeno koliko je operacija potrebno za izračunati c, d_i, e_i.

TEOREM (Motzkin, Belaga) Slučajno odabrani polinom stupnja n ima vjerojatnost 0 da ga se može izvredniti za strogo manje od $\lceil (n+1)/2 \rceil$ množenja ili za strogo manje od n zbrajanja.

Posljedica ovog teorema je to da je Hornerova shema optimalna za izvrednjavanje gotovo svih polinoma.

2.3 Dijeljenje polinoma linearnim faktorom

Pogledajmo kako se zapisuje Hornerova shema kada se radi "na ruke". U gornji redak $2 \times (n+2)$ tablice se upišu redom koeficijenti polinoma p_n, dok se donji red upiše najprije x_0 a zatim brojevi $c_{n-1}, \ldots, c_0, r_0$ izračunati kako slijedi

$$\begin{aligned} c_{n-1} &= a_n; \\ c_{i-1} &= c_i \cdot x_0 + a_{i-1}, \; i = n-1, \ldots, 1. \end{aligned}$$

Tablica, dakle, izgleda ovako:

	a_n	a_{n-1}	$\cdots$	a_1	a_0
x_0	c_{n-1}	c_{n-2}	$\cdots$	c_0	r_0

.

Na kraju izračunati r_0 je vrijednost polinoma p_n u točki x_0.

PRIMJER. Neka je zadan polinom p_5 s

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1.$$

Treba izračunati $p_5(-1)$.

Imamo

	2	0	-1	4	0	1
-1	2	-2	1	3	-3	4

,

pa je $p_5(-1) = 4$.

Uočimo još jednu zanimljivost: podijelimo li polinom p_n polinomom q zadanim s $q(x) = x - x_0$, onda je ostatak pri djeljenju upravo r_0 (otud i oznaka), a koeficijenti c_i su koeficijenti polinoma kvocijenta. Naime, ako stavimo

$$p_n(x) = (x - x_0)\, q_{n-1}(x) + r_0$$

pri čemu je

$$q_{n-1}(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1 x + b_0$$

dobijemo

$$p_n(x) = b_{n-1}x^n + (b_{n-2} - b_{n-1}x_0)\, x^{n-2} + \cdots + (b_0 - b_1 x_0)\, x - b_0 x_0 + r_0,$$

pa je

$$b_{n-1} = a_n = c_{n-1},$$

a iz toga je lako dobiti

$$b_i = c_i,\ i = n-1, \ldots, 0.$$

PRIMJER. Neka je zadan polinom p_5 s

$$p_5\left(x\right) = 2x^5 - x^3 + 4x^2 + 1.$$

Treba podijeliti p_5 s polinomom q zadanim s $q\left(x\right) = x + 1$.

Po prethodnom primjeru imamo

	2	0	-1	4	0	1
-1	2	-2	1	3	-3	4

,

pa je

$$2x^5 - x^3 + 4x^2 + 1 = \left(x + 1\right)\Big(2x^4 - 2x^3 + x^2 + 3x - 3\Big) + 4.$$

2.4 Potpuna Hornerova shema

Sada nas zanima što se događa ako postupak dijeljenja polinoma linearnim faktorom nastavimo, tj. ponovimo više puta? Vrijedi

$$\begin{aligned} p_n(x) &= (x - x_0)\, q_{n-1}(x) + r_0 \\ &= (x - x_0)\left[(x - x_0)\, q_{n-2}(x) + r_1\right] + r_0 \\ &= (x - x_0)^2\, q_{n-2}(x) + r_1 (x - x_0) + r_0 \\ &= \cdots \\ &= r_n (x - x_0)^n + \cdots + r_1 (x - x_0) + r_0. \end{aligned}$$

Dakle, polinom p_n zapisan je razvijeno po potencijama linearnog faktora $(x - x_0)$. Koja je veza s Taylorovim redom razvijenim oko x_0? Lako se zaključi da vrijedi

$$r_i = \frac{p_n^{(i)}(x_0)}{i!},\ i = 1, \ldots, n.$$

PRIMJER. Nađimo sve derivacije polinoma p_5 u točki $x_0 = -1$ gdje je

$$p_5(x) = 2x^5 - x^3 + 4x^2 + 1.$$

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					

Odavde lako čitamo npr. $p_5^{(3)}(-1) = 19 \cdot 3! = 114$ ili $p_5^{(5)}(-1) = 2 \cdot 5! = 240.$

ALGORITAM (Taylorov razvoj oko točke x_0)

for $i := 0$ **to** n **do**
    begin
    $r_i := a_i;$
    end;
for $i := 0$ **to** n **do**
    begin
    for $j := n - 1$ **downto** $i - 1$ **do**
        begin
        $r_j := r_j + x_0 * r_{j+1};$
        end
    end

2.5 Hornerova shema za interpolacijske polinome

Kada želimo izračunati interpolacijski polinom u Newtonovoj formi treba izračunati koeficijente a_i u izrazu oblika

$$\begin{aligned} p_n(x) &= a_n(x-x_0)(x-x_1)\cdots(x-x_{n-1}) + a_{n-1}(x-x_0)(x-x_1)\cdots(x-x_{n-2}) \\ &\quad +a_2(x-x_0)(x-x_1) + a_1(x-x_0) + a_0, \end{aligned}$$

pri čemu su x_i točke interpolacije, a x točka u kojoj želimo izračunati vrijednost polinoma p_n. Algoritam koji ćemo provesti vrlo je sličan Hornerovoj shemi. Najprije p_n zapišemo u drugačijem obliku stavljajući $y_i = x - x_i,\ i = 1, \ldots, n-1,$

$$p_n(x) = (\cdots((a_n y_{n-1} + a_{n-1})\, y_{n-2} + a_{n-2})\, y_{n-3} + \cdots a_1)\, y_0 + a_0.$$

pa odmah dobijemo tzv. Hornerovu shemu za interpolacijske polinome.

ALGORITAM (Hornerova shema za interpolacijske polinome, $p_n(x) = sum$)

$sum := a_n;$
for $i := n - 1$ **downto** 0 **do**
    begin
    $sum := sum * (x - x_i) + a_i;$
    end;