Skripta: Numerička Matematika

Tin Švagelj

27. studenoga 2023.

Sadržaj

1 Greške

Greške su neizbježne, prilikom izračuna je cilj smanjiti njihov utjecaj na krajnji rezultat.

1.1 Vrste grešaka

- **Greška modela** nastaje zamjenom nekog složenog sustava ili procesa jednostavnijim.
  - Neki sustavi su pre složeni da bi se opisali i rješili poznatim matematičkim alatima.
  - Primjer: otpor zraka u balistici
- **Greška metode** nastaje zamjenom procesa (obično beskonačnog) konačnom aproksimacijom.
  - Primjer: određivanje sume beskonačnog reda
- **Greška diskretizacije** nastaje kada umjesto podataka koje nije moguće potpuno ispravno pohraniti ili predstaviti koristimo njima bliske podatke (aproksimacije).
  - Primjeri:
    - $\pi, e, ...$
    - Zamjena kontinuuma konačnim diskretnim skupom
    - Zamjena beskonačno male veličine s malom, konkretnom veličinom
- **Ulazna pogreška** nastaje zbog pogrešaka u ulaznim podacima nastalih prethodnom obradom podataka ili greškom mjerenja.
  - Primjer: akceleracija sile teže.

Zaokruživanjem ili odbacivanjem decimala u prikazu broja a u računalu dobivamo aproksimaciju realnog broja (*engl.* floating point approximation) od a koju označavamo s $\mathrm{fl}(a)$ i za koju vrijedi:

$$\mathrm{fl}(a) = a(1 + \varepsilon)$$

gdje je ε **greška aproksimacija**.

1.2 Zapis realnog broja u računalu

Za zapis realnih brojeva se generalno koristi IEEE-754 standard.

U IEEE-754 standardu su brojevi pohranjeni s tri komponente:

Ime	Opis
Predznak	`1` ukoliko se radi o negativnom broju
Eksponent	eksponent na bazi pohranjenog broja
Mantisa/realni dio	decimalni dio realnog broja

Prilikom pohrane se podrazumjeva da matisa ima prefiksni bit `1` koji se ne piše jer nema smisla pohraniti broj sa vodečim nulama.

IEEE-754 navodi raspored bitova za realne brojeve različitih veličina. Predznak je uvijek veličine jednog bita, a ostali karakteristike su navedene u sljedećoj tablici:

Podatak	binary16	binary32	binary64	binary128	binary256
Eksponent	5b	8b	11b	15b	19b
Mantisa	10b	23b	52b	122b	236b
Ukupna veličina	16b	32b	64b	128b	256b
ulp^*	2^{-11}	2^{-24}	2^{-53}	2^{-123}	2^{-237}
exp_{min}	-14	-126	-1022	-16383	-262142
exp_{max}	+15	+127	+1023	+16384	+262143
val_{min} (normalan)	$6.10 \cdot 10^{-5}$	$1.18 \cdot 10^{-38}$	$2.23 \cdot 10^{-308}$	$3.36 \cdot 10^{-4932}$	$2.48 \cdot 10^{-78913}$
val_{min} (subnormalan**)	$5.96 \cdot 10^{-8}$	$1.40 \cdot 10^{-45}$	$4.94 \cdot 10^{-324}$	$6.48 \cdot 10^{-4966}$	$2.25 \cdot 10^{-78984}$
val_{max}	65504	$3.40 \cdot 10^{38}$	$1.80 \cdot 10^{308}$	$1.19 \cdot 10^{4932}$	$1.61 \cdot 10^{78193}$

* **ulp**: jedinična strojna greška, također se zove i *strojni epsilon*.

** subnormalne IEEE-754 vrijendosti su spremljene manjom decimalnom preciznošću, tj. uz veču jediničnu strojnu grešku

1.2.1 Posebne vrijednosti

IEEE-754 brojevi također imaju posebne vrijednosti kojima simboliziraju da se prilikom operacije dogodilo prekoračnje minimalne/maksimalne vrijednosti (`+inf` i `-inf`), te kada je provedena nevaljana operacija nad brojem `NaN` poput djeljenja s nulom.

Ime	Binarni zapis		
`+inf`	`0`	`1...1`	`0...0`
`-inf`	`1`	`1...1`	`0...0`
`NaN`	`_`	`1...1`	`_..._`

Zbog toga što `NaN` može biti predstavljen sa mnogo različitih binarnih kombinacija, jednačenje realnih brojeva na računalu nije smisleno rješivo. Različiti programski jezici se nose s tim problemom na različite načine:

- tretiraju sve `NaN` vrijednosti kao istu,
- tretiraju svaku (čak i one čiji binarni zapisi se podudaraju) kao različitu,
- bacaju grešku prilikom usporedbe,
- imaju koncepte *djelomično jednačivih/usporedivih* vrijednosti.

2 Apsolutna i relativna greška

Razliku $a - a^*$ između *stvarne veličine* a i *njene aproksimacije* a^* nazivamo **greška aproksimacije**. Apsolutnu vrijednost greške aproksimacije nazivamo **apsolutna greška aproksimacije** i označavamo je s Δa^*. Vrijedi:

$$\Delta a^* = |a - a^*|$$

Omjer između apsolutne greške Δa^* i apsolutne vrijednosti $|a|$ nazivamo **relativna greška** i označavamo je s δa^*. Vrijedi:

$$\delta a^* = \frac{\Delta a^*}{|a|}, \qquad a \neq 0$$

S obzirom da u praksi često nije poznata točna vrijednost a, koristi se **aproksimacija relativne greške**:

$$\delta a^* \approx \frac{\Delta a^*}{|a^*|}, \qquad a \neq 0$$

- Redoslijed izvršavanja operacija na računalu je bitan jer utječe na veličinu greške zbog ograničene pohrane decimalnih brojeva (floating point error).
  - Zbrajanje i množenje nisu asocijativni
  - Množenje prema zbrajanju nije distributivno

2.1 Katastrofalno kraćenje

Oduzimanje brojeva x i y koji su istog predzanaka može izazvati proizvoljno velike greške kada je $|x - y| \ll |x|, |y|$
Za katastrofalno kraćenje vrijedi:

$$(x - y)(1 + \varepsilon) = x(1 + \varepsilon_x) - y(1 + \varepsilon_y)$$
$$|\varepsilon| \leq \left|\frac{x}{x - y}\right| |\varepsilon_x| + \left|\frac{y}{x - y}\right| |\varepsilon_y|$$

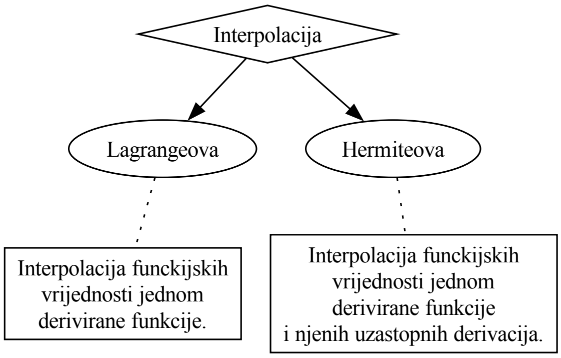
Jedna od metoda izbjegavanja kraćenja je racionalizacija korijenu koja daje puno točniji rezultat za x_1:

$$x_1 = -p + \sqrt{p^2 + q} \cdot \frac{-p - \sqrt{p^2 + q}}{-p - \sqrt{p^2 + 2}} = \frac{q}{p + \sqrt{p^2 + q}}$$

3 Metode interpolacije

Interpolacija je postupak kojim od zadanih točaka neke funkcije $f(x)$ dolazimo do funkcije $g(x)$ koja prolazi tim točkama i *blisko prati* $f(x)$.

Interpolacija uključuje pogreške mjerenja tako da je netočno reči $g(x) = f(x)$, no polinom ili po dijelovima funkcija koju dobijemo interpolacijom će dati dobar uvid u međuvrijednosti zadanih točaka.



Teorem 1: egzistencija i jedinstvenost polinoma

Neka je $n \in \mathbb{N}_0$. Za zadane točke (x_k, y_k), gdje je $y_k := f(x_k), k \in [0, n]$ i $x_i \neq x_j$ za $i \neq j$ postoji jedinstveni interpolacijski polinom najviše stupnja n

$$p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

za kojeg vrijedi

$$p_n(x_k) = y_k, \qquad k \in [0, n]$$

Dokaz:

Neka je polinom $p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ polinom stupnja najviše n.
Uvjete interpolacije zapisujemo u obliku

$$\begin{gathered} p_n(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0 \\ p_n(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n = y_1 \\ \vdots \\ p_n(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_2 \end{gathered}$$

Dobiven je sustav od $n+1$ linearne jednadžbe s $n+1$ nepoznanicom (koeficijenti $a_i, i \in [0, n]$ su nepoznanice).

Pitamo se ima li navedeni sustav rješenje i je li ono **jedinstveno**. Za to je dovoljno provjeriti je li odgovarajuća matrica sustava regularna.[1] Matrični sustav koji promatramo je

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{3.1}$$

Želimo odrediti D_n, gdje je

$$D_n = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{vmatrix}$$

[1]Matrica sustava je regularna ako joj je determinanta različita od 0.

Vandermondeova determinanta D_n je poznata. Ako je $D_n \neq 0$, tada sustav 3.1 ima jedinstveno rješenje. Definiramo pomoćnu determinantu oblika

$$V_n(x) = \begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x & x^2 & \cdots & x^n \end{vmatrix}, \qquad V_n(x_n) = D_n$$

Promatramo li $V_n(x)$ kao funkciju od x, razvojem po posljednjem retku uočavamo da je $V_n(x)$ polinom stupnja najviše n u varijabli x, a vodeći koeficijent tog polinoma je determinanta D_{n-1}. Zaista vrijedi

$$V_n(x) = (-1)^{n+1} \cdot \begin{vmatrix} x_0 & x_0^2 & \cdots & x_0^n \\ x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^n \end{vmatrix} \cdot 1 + (-1)^{n+2} \cdot \begin{vmatrix} 1 & x_0^2 & \cdots & x_0^n \\ 1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1}^2 & \cdots & x_{n-1}^n \end{vmatrix} \cdot x + \cdots$$
$$+ \underbrace{(-1)^{2n} \cdot \begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{vmatrix}}_{D_{n-1}} \cdot x^n$$

Dakle, vodeći koeficijent (koeficijent uz x^n) je

$$D_{n-1} = (-1)^{2n} \cdot \begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{vmatrix} = \begin{vmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{vmatrix}. \tag{3.2}$$

Primjetimo da, ako se u $V_n(x)$ redom uvrste $x_0, x_1, \dots, x_{n-1}$, vrijedit će

$$V_n(x_0) = V_n(x_1) = \cdots = V_n(x_{n-1}) = 0,$$

što povlači da su $x_0, x_1, \dots, x_{n-1}$ nultočke polinoma $V_n(x)$ koji je polinom n-tog stupnja. Da bi polinom n-tog stupnja bio potpuno definiran, trebaju biti poznate nultočke i njegov vodeći koeficijent (3.2). Dakle, traženi polinom je

$$V_n(x) = D_{n-1}(x - x_0)(x - x_1) \dots (x - x_{n-1}).$$

Ako uvrstimo x_n u $V_n(x)$, dobivamo rekurzivnu formulu za D_n, odnosno

$$D_n = V_n(x_n) = D_{n-1}(x_n - x_0)(x_n - x_1) \dots (x_n - x_{n-1}).$$

Trivijalno vrijedi $D_0 = 1$ te lako dobivamo

$$D_n = \prod_{0 \leq j < i \leq n} (x_i - x_j).$$

Budući da je $x_i \neq x_j$ za $i \neq j$, onda je $D_n \neq 0$, a vrijedi i obrat. Dakle, matrica sustava je regularna (jer $D_n \neq 0$) i stoga postoji jedinstveno rješenje $(a_0, a_1, \dots, a_n)$ što povlači da postoji jedinstveni interpolacijski polinom.

4 Vandermondeova determinanta

Za n zadanih točaka (x_k, y_k) gdje je $y_k := f(x_k)$, k predstavlja indeks točke $(k = 0, 1, \dots, n)$, ako ni jedne dvije točke nemaju istu x vrijednost postoji jedinstveni **interpolacijski polinom** najviše stupnja n:

$$p_n(x) := \sum_{i=0}^{n} a_i x^i = a_0 + a_1 x + \dots a_n x^n \tag{4.1}$$

Sustavi dobiveni uporabom vandermondeove determinante su nestabilni te se zbog toga najčešće koristi baza polinoma koja daje stabilnije rezultate.

Primjer 1: uporaba vandermondeove determinante

Zadane su sljedeće točke, odredi interpolacijski polinom uporabom vandermondeove determinante:

x	0	2	4
y	3	4	2

Zadane su 3 točke pa očekujemo polinom maksimalno drugog stupnja:

$$p_2(x) = a_0 + a_1 x + a_2 x^2$$

Uvršavanjem točaka dobivamo:

$$\begin{aligned} p_2(0) &= a_0 \cdot 1 + a_1 \cdot 0 + a_2 \cdot 0^2 = 3 \\ p_2(2) &= a_0 \cdot 1 + a_1 \cdot 2 + a_2 \cdot 2^2 = 4 \\ p_2(4) &= a_0 \cdot 1 + a_1 \cdot 4 + a_2 \cdot 4^2 = 2 \end{aligned}$$

Što predstavljeno kao matrična jednadžba izgleda kao:

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix}$$

$$\begin{aligned} A \cdot X &= B \\ X &= A^{-1} \cdot B \end{aligned}$$

Zatim računamo **vandermondeovu determinantu**:

$$D_n = det(A) = \begin{vmatrix} 1 & 0 & 0 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{vmatrix} = 1 \cdot \begin{vmatrix} 2 & 4 \\ 4 & 16 \end{vmatrix} = 32 - 16 = 16 \neq 0$$

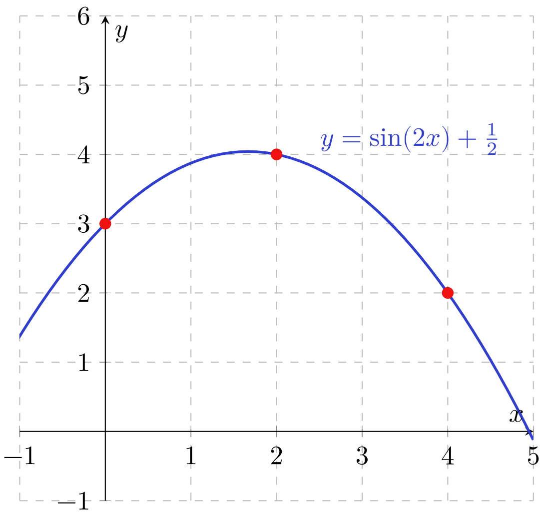
Kako bismo odredili A^{-1} trebamo provjeriti da je $det(A) \neq 0$, u suprotnom inverzna matrica ne postoji. U ovom primjeru, inverzna matrica je:

$$A^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{3}{4} & 1 & -\frac{1}{4} \\ \frac{1}{8} & -\frac{1}{4} & \frac{1}{8} \end{bmatrix}$$

$$X = A^{-1} \cdot B = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{3}{4} & 1 & -\frac{1}{4} \\ \frac{1}{8} & -\frac{1}{4} & \frac{1}{8} \end{bmatrix} \cdot \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ \frac{5}{4} \\ -\frac{3}{8} \end{bmatrix}$$

Rješavanjem sustava dobivamo $a_0 = 3$, $a_1 = \frac{5}{4}$, $a_2 = -\frac{3}{8}$, kao i interpolaciju funkcije:

$$p_2(x) = 3 + \frac{5x}{4} + -\frac{3x^2}{8}$$



5 Lagrangeov interpolacijski polinom

Formula za Lagrangeov interpolacijski polinom ($\mathbf{L}_n$) je:

$$\mathbf{I}_i(x) := \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \tag{5.1}$$

$$\mathbf{L}_n(x) := \sum_{i=0}^{n} y_i \mathbf{I}_i(x) \tag{5.2}$$

Lagrangeov interpolacijski polinom je bolji od korištenja vandermondeove determinante jer su sustavi dobiveni vandermondeovom determinantom nestabilni. Također je i postupak brži jer ne zahtjeva računanje inverzne matrice koja u računalnoj primjeni zahtjeva mnogo međukoraka i instrukcija.

✎ Primjer 1: uporaba lagrangeovog polinoma

Odredi interpolacijski polinom koristeći lagrangeove polinome ako su zadane sljedeće točke:

x	-3	4	3
y	-4	2	0

Računamo $\mathrm{I}_0(x), \mathrm{I}_1(x)$, i $\mathrm{I}_2(x)$ za točke:

$$\mathrm{I}_0(x) = \frac{x - x_1}{x_0 - x_1} \cdot \frac{x - x_2}{x_0 - x_2} = \frac{x-4}{-3-4} \cdot \frac{x-3}{-3-3} = \frac{(x-4)(x-3)}{-7 \cdot -6} = \frac{x^2 - 7x + 12}{42}$$

⚠ Oprez: nedostajući član

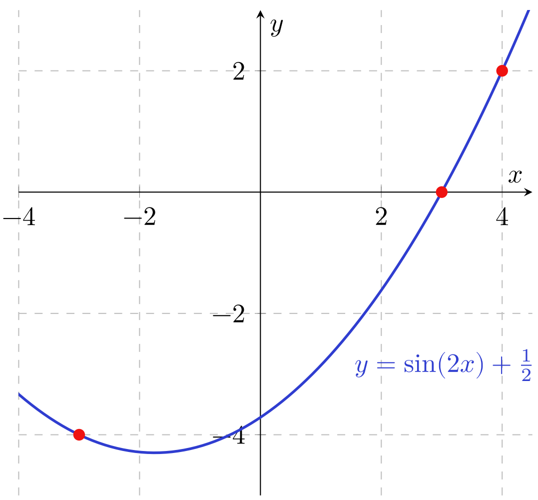
$$\frac{x - x_j}{x_0 - x_j} = \frac{x - x_0}{x_0 - x_0} = \frac{x - x_0}{0}$$

$$\mathrm{I}_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x+3)(x-3)}{7 \cdot 1} = \frac{x^2 - 9}{7}$$

$$\mathrm{I}_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x+3)(x-4)}{6 \cdot -1} = \frac{x^2 - x - 12}{-6}$$

Nakon uvrštavanja dobivamo polinom:

$$\begin{aligned}
\mathrm{L}_2(x) &= y_0 \mathrm{I}_0(x) + y_1 \mathrm{I}_1(x) + y_2 \mathrm{I}_2(x) \\
&= -4 \cdot \frac{x^2 - 7x + 12}{42} + 2 \cdot \frac{x^2 - 9}{7} + 0 \cdot \frac{x^2 - x - 12}{-6} \\
&= \frac{-2x^2 + 14x - 24 + 6x^2 - 54}{21} \\
&= \frac{4x^2}{21} + \frac{2x \cdot \cancel{7}}{3 \cdot \cancel{7}} - \frac{26 \cdot \cancel{3}}{7 \cdot \cancel{3}} \\
&\boxed{= \frac{4}{21}x^2 + \frac{2}{3}x - \frac{26}{7}}
\end{aligned}$$



5.1 Nedostaci

Definicija 1

Funkcija $g(x)$ je reda $\mathcal{O}(h(x))$ za $x \to x_0$, odnosno

$$g(x) = \mathcal{O}(h(x)), x \to x_0$$

ako postoje konstante σ, C takve da

$$|x - x_0| \leq \sigma \Rightarrow |g(x)| \leq C|h(x)|$$

5.1.1 Veliki broj aritmetičkih operacija

Interpolacija lagrangeovim polinomom zahtjeva $\mathcal{O}(n^2)$ operacija kod prvog izvrjednjavanja i $\mathcal{O}(n)$ operacija za svako sljedeće izvrjednjavanje.

Kod računalne primjene brzina izračuna ovisi u broju primjena dobivenog polinoma jer se unaprijed izračunata vandermondeova determinanta za sustav može primjeniti više puta dok je u slučaju lagrangeovih polinoma potrebno koristiti pokazivače na članove polinoma (I_n, koji su parcijalno ispunjene funkcije na gomili) što npr. za crtanje funkcije **može zahtjevati veći broj dereferenciranja sa gomile**.

Taj problem je rješiv korištenjem biblioteka za simboličku matematiku, no to povečava veličinu distribuirane binarne datoteke, iako je taj kompromis generalno ok ako se bibliteka koristi za pojednostavljivanje više izraza u toku izvođenja programa.

5.1.2 Velika zavisnost o ulaznim podacima

U slučaju izmjena ulaznih podataka (ponajviše x koordinata) je potrebno ponovo računati sve polinome interpolacije (I_n).

Kod računalne primjene, prethodno navedene parcijalno ispunjene funkcije (za I_n) je potrebno ponovo slagati - **loša memoizacija**.

5.1.3 Slaba aproksimacija i oscilacije

Ako se odabere polinom preniskog supnja, on može **slabo aproksimirati originalnu funkciju**.

Rungeov fenomen: Ako se odabere polinom previsokog stupnja, na djelovima se mogu pojaviti **prevelike oscilacije**, tj. greške na rubovima intervala aproksimacije.

6 Newtonov interpolacijski polinom

Newtonov interpolacijski polinom rješava problem loše memoizacije koji lineariziranagrangeov polinom ima prilikom dodavanja interpoliranih točaka. Rješava to jer je definiran kao **rekurzivna funkcija** kojoj se **mogu pridodati članovi** kako bi aproksimacija davala točne vrijednosti za novonastali skup:

$$\mathbf{p}_n(x) := \mathbf{p}_{n-1}(x) + \mathbf{c}(x) \tag{6.1}$$

$\mathbf{c}(x)$ nazivamo korekcijom, i ona je polinom stupnja n, oblika:

$$\mathbf{c}(x) := \mathbf{a}_n \prod_{i=0}^{n-1}(x - x_i) = \mathbf{a}_n(x - x_0)(x - x_1)\dots(x - x_{n-1}). \tag{6.2}$$

6.1 Podijeljena razlika

$\mathbf{a}_n$ je funkcija čvorova $x_i, i = 0, 1, \dots, n$ koju zovemo **podijeljena razlika n-tog reda** i označavamo:

$$\mathbf{a}_n := f[x_0, \dots, x_n]$$
$$\mathrm{a}_0 = f[x_0] := f(x_0)$$

Za *dvije točke* $(x_0,f(x_0))$, $(x_1,f(x_1))$ definiramo i označavamo **podijeljenu razliku prvog reda funkcije** f kao:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f(x_0)}{x_0 - x_1} + \frac{f(x_1)}{x_1 - x_0}$$

Za *tri točke* $(x_0,f(x_0))$, $(x_1,f(x_1))$, $(x_2,f(x_2))$ uvodimo **podjeljenu razliku drugog reda funkcije** f kao:

$$f[x_0, x_1, x_2] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{\frac{f(x_2)}{x_2 - x_1} - \frac{f(x_1)}{x_2 - x_1}}{x_2 - x_0} - \frac{\frac{f(x_1)}{x_1 - x_0} - \frac{f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$\begin{aligned}
&= \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} - \frac{f(x_1)}{(x_2 - x_0)(x_2 - x_1)} - \frac{f(x_1)}{(x_2 - x_0)(x_1 - x_0)} + \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} \\
&= \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} - \frac{f(x_1)(x_1 - x_0) + f(x_1)(x_2 - x_1)}{(x_2 - x_0)(x_1 - x_0)(x_2 - x_1)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
&= \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} - \frac{f(x_1)(\cancel{x_1} - x_0 + x_2 \cancel{- x_1})}{(x_2 - x_0)(x_1 - x_0)(x_2 - x_1)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
&= \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} - \frac{f(x_1)\cancel{(x_2 - x_0)}}{\cancel{(x_2 - x_0)}(x_1 - x_0)(x_2 - x_1)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
&= \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} - \frac{f(x_1)}{(x_1 - x_0)(x_2 - x_1)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} \\
&= \frac{f(x_0)}{(x_2 - x_0)(x_1 - x_0)} \cancel{-} \frac{f(x_1)}{(x_1 - x_0) \cdot \cancel{-1} \cdot (x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}
\end{aligned}$$

$$\boxed{= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}}$$

Zbog pojednostavljivanja rekurzivnog izračuna je poželjno zapisati podijeljene razlike u obliku tablice:

Tablica: podjeljene razlike

$$
\begin{array}{ccccccc}
x_0 & f[x_0] & & & & & \\
 & & f[x_0,x_1] & & & & \\
x_1 & f[x_1] & & f[x_0,x_1,x_2] & & & \\
 & & f[x_1,x_2] & & f[x_0,x_1,x_2,x_3] & & \\
x_2 & f[x_2] & & f[x_1,x_2,x_3] & \vdots & & \\
 & & f[x_2,x_3] & \vdots & & \cdots & f[x_0,x_1,\dots,x_n] \\
x_3 & f[x_3] & \vdots & & & & \\
\vdots & \vdots & & f[x_{n-2},x_{n-1},x_n] & & & \\
 & & f[x_{n-1},x_n] & & & & \\
x_n & f[x_n] & & & & &
\end{array}
$$

Općenita rekurzivna relacija koja vrijedi za podijeljene razlike je:

$$f[x_0, x_1, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}$$

Indukcijom iz rekurzivne relacije izvodimo **lineariziran oblik za podijeljenu razliku n-tog reda** koji nije rekurzivan:

$$f[x_a, \dots, x_b] = \sum_{i=a}^{b} \frac{f(x_i)}{\prod_{\substack{j=a \\ j\neq i}}^{b} (x_i - x_j)}, \qquad a > 0, \qquad b > a$$

Taj oblik je praktičan za primjenu prilikom ručnog izračuna jer ne zahtjeva rekurzivno raspisivanje svih članova. Iz toga također slijedi sažeti zapis newtonovog polinoma:

$$p_n(x) = \sum_{i=1}^{n} \left[\left(\sum_{j=0}^{i} \frac{f(x_j)}{\prod_{\substack{k=0 \\ k\neq j}}^{i} (x_j - x_k)} \right) \prod_{j=0}^{i-1} (x - x_j) \right]$$

Ovaj zapis je malo manje praktičan, no daje potpun uvid u polinom.

Oprez: međupohrana

Direktan izračun gubi prednost međupohrane djelomičnih rezultata koji se mogu ponovo uporabiti prilikom dodavanja interpoliranih točaka.
Kod računalnog izračuna ima smisla pohraniti rezultate podijeljenih razlika ($f[\dots]$) u međuspremnik te po potrebi izračunati vrijednost polinoma p_n s njima.

6.2 Ekvidistantni čvorovi

U slučaju jednako udaljenih (ekvidistantnih) čvorova je podjeljena razlika jednostavnija za zapisati te se zove **konačna razlika**.

Ekvidistantne čvorove možemo zapisati u obliku:

$$x_{i+1} = x_i + h, \qquad i \in [0, n\rangle$$

gdje je h konstantan razmak (korak) između dva susjedna čvora.

Konačna razlika je oblika:

$$\Delta^k f(x_i) := \Delta^{k-1} f(x_{i+1}) - \Delta^{k-1} f(x_i), \qquad k \in \langle 0, n] \tag{6.3}$$

$$\Delta^0 f(x_i) := f(x_i)$$

Tablica: konačne razlike

$$\begin{array}{ccccccc}
x_0 & f(x_0) & & & & & \\
 & & \Delta f(x_0) & & & & \\
x_1 & f(x_1) & & \Delta^2 f(x_0) & & & \\
 & & \Delta f(x_1) & & \Delta^3 f(x_0) & & \\
x_2 & f(x_2) & & \Delta^2 f(x_1) & \vdots & & \\
 & & \Delta f(x_2) & \vdots & & \cdots & \Delta^n f(x_0) \\
x_3 & f(x_3) & \vdots & & & & \\
\vdots & \vdots & & \Delta^2 f(x_{n-2}) & & & \\
 & & \Delta f(x_{n-1}) & & & & \\
x_n & f(x_n) & & & & &
\end{array}$$

Vrijedi **lema o vezi podijeljenih i konačnih razlika**:

$$f[x_0, x_1, \dots, x_n] = \frac{\Delta^n f(x_0)}{n! h^n}, \qquad x_{i+1\_x}i = h, \qquad i \in [0, n\rangle$$

Zbog leme o vezi razlika, tablica konačnih razlika izgleda identično tablici podijeljenih razlika.

Newtonov oblik interpolacijskog polinoma na ekvidistantnim čvorovima je:

$$p_n(x) = \sum_{i=1}^{n} \left(\frac{\Delta^i f(x_0)}{i! h^i} \prod_{j=0}^{i-1} x - x_j \right) \tag{6.4}$$

$$= f(x_0) + \frac{\Delta f(x_0)}{1! h}(x - x_0) + \frac{\Delta^2 f(x_0)}{2! h^2}(x - x_0)(x - x_1) + \dots + \frac{\Delta^n f(x_0)}{n! h^n}(x - x_0)\dots(x - x_{n-1})$$

✎ Primjer 1

Funkciju $f(x) = sin(\pi x)$ na segmentu $[0,\ 1]$ aproksimirati interpolacijskim polinomom koji prolazi sljedećim točkama:

x	0	0.25	0.5	0.75	1
$f(x)$	0	0.7071	1	0.7071	0

S obzirom da su zadane točke sve podjednako udaljene ($h = 0.25$), možemo koristiti newtonovu interpolaciju za ekvidistantne točke:

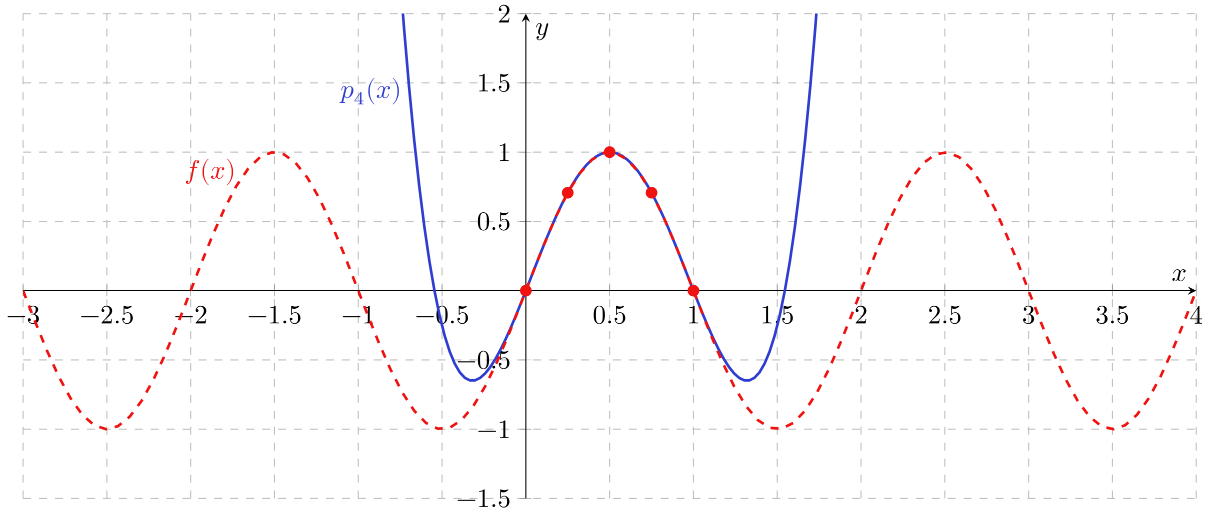
$$\begin{aligned}p_4(x) =& f(x_0) + \frac{\Delta^1 f(x_0)}{1!h^1}(x-x_0) + \frac{\Delta^2 f(x_0)}{2!h^2}(x-x_0)(x-x_1) + \\ & \frac{\Delta^3 f(x_0)}{3!h^3}(x-x_0)(x-x_1)(x-x_2) + \frac{\Delta^4 f(x_0)}{4!h^4}(x-x_0)(x-x_1)(x-x_2)(x-x_3)\end{aligned}$$

Popunjavamo tablicu konačnih razlika prema formuli 6.3:

$$\begin{array}{cccccc}
0 & 0 & & & & \\
 & & \Delta f(x_0) = 0.7071 & & & \\
0.25 & 0.7071 & & \Delta^2 f(x_0) = -0.4142 & & \\
 & & \Delta f(x_1) = 0.2929 & & \Delta^3 f(x_0) = -0.1716 & \\
0.5 & 1 & & \Delta^2 f(x_1) = -0.5858 & & \Delta^4 f(x_0) = 0.3432 \\
 & & \Delta f(x_2) = -0.2929 & & \Delta^3 f(x_1) = 0.1716 & \\
0.75 & 0.7071 & & \Delta^2 f(x_1) = -0.4142 & & \\
 & & \Delta f(x_0) = -0.7071 & & & \\
1 & 0 & & & &
\end{array}$$

Nakon što smo odredili sve konačne razlike, možemo ih uvrstiti u prethodnu formulu:

$$\begin{aligned}p_4(x) =& 0 + \frac{0.7071}{1! \cdot 0.25^1}(x-0) + \frac{-0.4142}{2! \cdot 0.25^2}(x-0)(x-0.25) + \frac{-0.1716}{3! \cdot 0.25^3}(x-0)(x-0.25)(x-0.5) \\ & + \frac{0.3432}{4! \cdot 0.25^4}(x-0)(x-0.25)(x-0.5)(x-0.75) \\ =& 2.8284x - 3.3136x(x-0.25) - 1.8304x(x-0.25)(x-0.5) + 3.6608x(x-0.25)(x-0.5)(x-0.75) \\ =& 3.6608x^4 - 7.3216x^3 + 0.576x^2 + 3.0848x\end{aligned}$$



6.3 Ocjena greške lagrangeovog i newtonovog interpolacijskog polinoma

Pod pretpostavkom da za svaki $f(x)$ postoji $f^{(n+1)}(x)$, gdje je $x \in [x_0, x_n]$ za neki $n \in \mathbb{N}_0$, i da su čvorovi interpolacijskog polinoma $p_n(x)$ za funkciju $f(x)$. Tada za svaki $x \in [x_0, x_n]$ postoji $\xi \in [x_0, x_n]$ takav da vrijedi:

$$e(x) = f(x) - p_n(x) = \frac{\omega(x)}{(n+1)!} f^{(n+1)}(\xi)$$

Ako postoji $\mathrm{M}_{n+1} = \max_{x\in[x_0,x_n]} \left|f^{(n+1)}(x)\right|$, tada je globalna ocjena greške:

$$|e(x)| = |f(x) - p_n(x)| \leq \frac{|\omega(x)|}{(n+1)!}\mathrm{M}_{n+1}$$

Primjer 2

Odrediti grešku newtonovog interpolacijskog polinoma iz prethodnog zadatka u točki $x = 0.6$ te odrediti globalnu ocjenu greške funkcije na zadanom segmentu.

Koristeći interpolacijski polinom iz prethodnog zadatka dobivamo $p_4(0.6) = 0.95121$ dok je $f(0.6) = 0.95105$. Uvrštavanjem u formulu za absolutnu pogrešku dobivamo:

$$|f(0.6) - p_4(0.6)| = 1.53483 \cdot 10^{-4}$$

Prema teoremu o ocjeni lagrangeovog/newtonovog polinoma dobivamo:

$$|f(x) - p_4(x)| \leq \frac{\pi^5}{5!}|x(x-0.25)(x-0.5)(x-0.75)(x-1)| \tag{6.5}$$

$$\mathrm{M}_5 = \max_{x\in[0,1]} (\sin(\pi x))^{(5)} \tag{6.6}$$

Odnosno preciznije, jer vrijedi $\sin(\pi x)^{(5)} = \pi^5 \cos(\pi x)$, M_5 je:

$$\mathrm{M}_5 = \max_{x\in[0,1]} \left|\pi^5 \cos(\pi x)\right|$$

Na intervalu $[0, 1], \cos(\pi x)$ zauzima najveću vrijednost (1) za $x = 0$ te time slijedi:

$$\mathrm{M}_5 = \left|\pi^5 \cos(\pi \cdot 0)\right| = \left|\pi^5 \cdot 1\right| = \pi^5$$

6.4 Nedostaci

Newtonova interpolacija pati od istog nedostatka kao i lagrangeova gdje se s večim brojem interpoliranih točaka pojavljuju sve veće **oscilacija na rubovima interpoliranih točaka**.

Složenost newtonovog polinoma je ista lagrangeovom, no naknadne izmjene u praktičnoj primjeni zahtjevaju znatno manji broj ponovnih izračuna.

7 Linearni interpolacijski spline

Prilikom pokušaja interpolacije velikog broja točaka pojavljuje se problem sa oscilacijama na rubovima interpoliranih točaka gdje vrijedi:

$$\lim_{n\to\infty} \max_{x\in[-1,1]} |f(x) - \mathbf{p}_n(x)| = +\infty$$

Iz tog razloga se vrlo rijetko za interpolaciju koriste polinomi visokog stupnja (> 5) jer ona ima loša svojstva.

Jedna od efikasnih metoda interpolacije je *po dijelovima polinomna interpolacija* (engl. piecewise polynomial interpolation) gdje se na svim podsegmentima inicijalnog segmenta korise polinomi niskog stupnja:

$$p_i \in \mathcal{P}_m$$

gdje je $\mathcal{P}_m$ oznaka za **prsten polinoma** stupnja m.

Neka su zadani čvorovi interpolacije i **interpolicijski polinom dogovorenog i niskog stupnja** (ϕ) na podsegmentima $[x_i, x_{i+1}]$ za $i \in [0, n\rangle$:

$$\begin{aligned} (x_i, y_i), &\qquad i \in [0, n] \\ \phi_{[x_i, x_{i+1}]} = p_i(x), &\qquad i \in [0, n\rangle \end{aligned} \tag{7.1}$$

Uvjeti

Iz uvijeta interpolacije $\phi(x_i) = y_i$ za $i \in [0, n]$:

$$p_{i-1}(x_i) = p_i(x_i) = y_i, \qquad i \in [1, n\rangle \tag{7.2}$$

Dobivamo sljedećih $2n$ uvijeta kojima se osigurava neprekidnost funkcije $\phi(x)$:

$$p_0(x_0) = y_0 \tag{7.3}$$

$$\vdots$$

$$p_{n-2}(x_{n-1}) = p_{n-1}(x_n) = y_{n-1} \tag{7.4}$$

$$p_{n-1}(x_n) = y_n \tag{7.5}$$

Uvijeti 7.2, 7.3, 7.4 i 7.5 opisuju neprekidnost polinoma među interpoliranim podsegmentima.

Ako odaberemo da su polinomi $p_i(x) \in \mathcal{P}_1$ (polinomi prvog stupnja), tada imamo dovoljno uvjeta iz kojih možemo jedinstveno odrediti sve koeficijente linearnog interpolacijskog splinea.

7.1 Linearni spline

Svakom podsegmentu $[x_i, x_{i+1}]$ pridužimo jedinstveni polinom $p_i(x)$ kojeg jednostavno zapisujemo u obliku lagrangeovog ili newtonovog interpolacijskog polinoma:

$$\text{Lagrangeov polinom:} \qquad p_i(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} y_i + \frac{x - x_i}{x_{i+1} - x_i} y_{i+1}$$

$$\text{Newtonov polinom:} \qquad p_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i)$$

Linerani spline se na kraju definira kao funkcija po dijelovima gdje je svaki dio definiran za pojedini segment kao pripadni polinom:

$$\Phi(x) = \begin{cases} p_0(x), & x \in [x_0, x_1\rangle \\ p_1(x), & x \in [x_1, x_2\rangle \\ \vdots \\ p_n(x), & x \in [x_n, x_{n+1}] \end{cases} \tag{7.6}$$

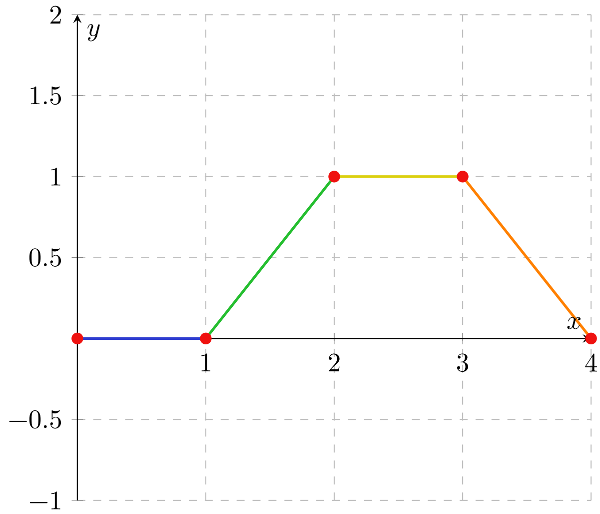
Primjer 1: određivanje linearnog splinea

Odredi linearni spline ako su zadane sljedeće točke:

x	0	1	2	3	4
y	0	0	1	1	0

Određujemo Newtonove interpolacijske polinome za sve tražene podsegmente:

$$
\begin{aligned}
p_0(x) &= y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) = 0 + \frac{0-0}{1-0}(x-0) = 0 \\
p_1(x) &= y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) = 0 + \frac{1-0}{2-1}(x-1) = x - 1 \\
p_2(x) &= y_2 + \frac{y_3 - y_2}{x_3 - x_2}(x - x_2) = 1 + \frac{1-1}{3-2}(x-2) = 1 \\
p_3(x) &= y_3 + \frac{y_4 - y_3}{x_4 - x_3}(x - x_3) = 1 + \frac{0-1}{4-3}(x-3) = 4 - x
\end{aligned}
$$



Te iz toga dobivamo formulu za linearni interpolacijski spline:

$$
\Phi(x) = \begin{cases} 0, & x \in [0,1\rangle \\ x-1, & x \in [1,2\rangle \\ 1, & x \in [2,3\rangle \\ 4-x, & x \in [3,4] \end{cases}
$$

7.2 Ocjena greške

Ocjena greške linearnog splinea slijedi iz ovjene greške interpolacijskog polinoma.

Neka je $f(x) \in \mathcal{C}^2[a,b]$. Tada je ocjena greške interpolacije funkcije f linearnim splineom:

$$
|f(x) - \phi(x)| \leq \frac{\omega(x) \max_{x \in [x_i, x_{i+1}]} |f''(x)|}{2!}
$$

Uvođenjem oznaka $h = \max_{i \in [1,n]} |x_i - x_{i-1}|$ i $M = \max_{x \in [a,b]} |f''(x)|$ tada je $\omega(x) \leq \frac{h^2}{4}$ pa za ocjenu greške linearnim splineom vrijedi:

$$
|f(x) - \phi(x)| \leq \frac{\mathrm{M}h^2}{8}
$$

Odnosno možemo pisati:

$$
f(x) = \phi(x) + \mathcal{O}(h^2), \qquad h \to 0
$$

Kako bi se postigla zadovoljavajuća točnost ovom interpolacijom je potreban veliki broj podsegmenata, no zbog jednostavnosti se ipak vrlo često koristi kod računalne grafike.

Kada je potrebno približno odrediti vrijednost $f(\xi), \xi \in [a,b]$, prvo je potrebno odrediti u kojem se intervalu nalazi ξ. Da bi odredili i takav da je $x_i \leq \xi \leq x_{i+1}$ najčešće se koristi binarno pretraživanje koje koristi $\mathcal{O}(\log(n))$ operacija.

$$
f(\xi) \approx p_i(\xi)
$$

Primjer 2: aproksimacija s točnošću

Aproksimirati funkciju $f(x) = \ln(x)$ na segmentu $[1, 100]$ po dijelovima linearnom interpolacijom. Fiksirajmo traženu točnost $\varepsilon = 10^{-4}$ koju zahtijevamo na cijelom segmentu. Nači broj čvorova kako bi se postigla zadana točnost na:

1. ekvidistantnoj mreži s korakom h na cijelom segmentu $[1, 100]$
2. podijeljenim podsegmentima $[1, 2]$, $[2, 7]$, $[7, 100]$ gdje svaki od navedenih podsegmenata dijelimop ekvidistantnom mrežom čvorova s koracima h_1, h_2, h_3, respektivno.

1) Određujemo:

$$f'(x) = \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}, \text{ te} \qquad |f''(x)| = \left|\frac{1}{x^2}\right| = \frac{1}{x^2}.$$

Promatrana funkcija uvijek postiže svoj maksimum u lijevom kraju segmenta $[1, 100]$.

Koristeći ocjenu greške za linearni spline, dobivamo da je na segmentu $[1, 100]$ uz ekvidistantnu mrežu s korakom h greška interpolacije

$$|\Phi(x) - f(x)| \leq \frac{1 \cdot h^2}{8}.$$

U našem slučaju mora vrijediti $\dfrac{h^2}{8} \leq 10^{-4}$ što povlači $h \leq 0.02828$. S obzirom da je

$$h = \frac{b-a}{n} = \frac{100-1}{n} = \frac{99}{n},$$

dobivamo da je $n = 3500.17$, a budući da je $n \in \mathbb{N}$, tada u formulu nije uključen posljednji rubni čvor, zaključujemo da nam je za zadanu točnost potrebno barem $3501 + 1 = 3502$ čvorova.

2) Na svakom zadanom podsegmentu greška interpolacije mora biti manja od $\varepsilon = 10^{-4}$.

- Za prvi podsegment $[1, 2]$ vrijedi $\mathrm{M}_2(f) = 1$ pa je $\frac{h_1^2}{8} \leq 10^{-4}$ što povlači $h_1 \leq 0.02828$, a analognim postupkom kao u prethodnom slučaju dobivamo da je $n_1 \geq 35.35$, odnosno $n_1 = 36$.
- Na drugom podsegmentu $[2, 7]$ vrijedi $\mathrm{M}_2(f) = \frac{1}{4}$ pa je $\frac{1}{4} \cdot \frac{h_2^2}{8} \leq 10^{-4}$ odnosno $h_2 \leq 0.05656$ pa je $n_2 \geq 88.38$, odnosno $n_2 = 89$.
- Na trećem podsegmentu $[7, 100]$ vrijedi $\mathrm{M}_2(f) = \frac{1}{49}$ pa je $\frac{1}{49} \cdot \frac{h_3^2}{8} \leq 10^{-4}$ odnosno $h_3 \leq 0.19798$ pa je $n_3 \geq 469.73$, odnosno $n_3 = 470$.

7.3 Nedostaci

Dok je aproksimacija danih podataka linearnim splineom može biti dovoljna u nekim slučajevima, prijelazi između segmenata mogu biti pre nagli i primjetni za druge primjene. Na primjer:

- brzina kretanja tijela u prostoru se može doimati *neprirodnom*,
- razlike u brzini porasta/pada amplitude zvuka su osjetne, ...

8 Kubični interpolacijski spline

Kako bi se izbjegao prethodno navedeni problem, poželjno je konstruirati spline višeg reda. **Kvadratni spline** može imati skokove prve derivacije u točkama mreže čvorova pa se on ne primjenjuje toliko često. **Kubični spline** se iznimno često koristi u praksi te nema prethodno navedene nedostatke.

Kada imamo zadane diskretne vrijednosti funkcije:

$$y_i = f(x_i), \qquad i \in [0, n]$$

i čvorove zadane tako da vrijedi:

$$a \leq x_0 < x_1 < \cdots < x_n \leq b$$

Kubični spline je funkcija $s(x) \in \mathcal{C}^2(\langle x_0, x_n \rangle)$ koja je:

- na svakom podintervalu $[x_i, x_{i+1}\rangle, i \in [0, n\rangle$ polinom najviše trećeg stupnja,
- u čvorovima interpolacije zadovoljava uvjete interpolacije te
- ima neprekidnu drugu derivaciju

Na svakom podintervalu koriste se polinomi:

$$\phi[x_i, x_{i+1}] = p_i(x), \qquad i \in [0, n], \qquad p_i(x) \in \mathcal{P}_3$$

Kubični polinom $p_i(x)$ ima četiri koeficijenta, i može se zapisati u obliku:

$$p_i(x) = a_0 + a_1(x - x_i) + a_2(x - x_i)^2 + a_3(x - x_i)^3$$

Uvjeti: određivanje koeficijenata kubičnog splinea

$$\begin{aligned} p_0(x_0) &= y_0 \\ p_{i-1}(x_i) = p_i(x_i) &= y_i, \qquad i \in [1, n-1] \\ p_{n-1}(x_n) &= y_n \end{aligned} \tag{8.1}$$

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}) = y_i, \qquad i \in [0, n-2] \tag{8.2}$$

$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}) = y_i, \qquad i \in [0, n-2] \tag{8.3}$$

$$\begin{aligned} s_0 &= f'(x_0) \\ s_n &= f'(x_n) \end{aligned} \tag{8.4}$$

- Uvjet 8.1 osigurava neprekidnost interpolacije i daje $2n$ uvjeta.
- Uvjet 8.2 osigurava neprekidnost prve derivacije na unutarnjim čvorovima interpolacije i daje $n-1$ uvjeta.
- Uvjet 8.3 osigurava neprekidnost druge derivacije na unutarnjim čvorovima interpolacije i daje dodatnih $n-1$ uvjeta.
- Uvjet 8.4 osigurava da je nagib interpolacije na rubovima segmenata jednak nagibu funkcije.

8.4 je jedan on različih načina za zadati dodatne uvjete kako bi dobili jedinstvena rješenja kubičnog splinea, tj. moguće je postići jedinstvenost rješenja i na druge načine.

Označimo derivacije u unutarnjim čvorovima

$$s_i = \phi'(x_i), \qquad i \in [1, n-1]$$

Polinome $p_i(x)$ u ovom slučaju je najpraktičnije zapisati u Newtonovom obliku interpolacijskog polinoma

$$\begin{array}{ccccc}
x_i & f[x_i] & & & \\
 & & s_i & & \\
x_i & f[x_i] & & f[x_i, x_i, x_{i+1}] & \\
 & & f[x_{i+1}, x_i] & & f[x_i, x_i, x_{i+1}, x_{i+1}] \\
x_{i+1} & f[x_{i+1}] & & f[x_i, x_{i+1}, x_{i+1}] & \\
 & & s_i & & \\
x_{i+1} & f[x_{i+1}] & & &
\end{array}$$

Gdje su $s_i = f'(x_i) = f[x_i, x_i]$, a $s_{i+1} = f'(x_{i+1}) = f[x_{i+1}, x_{i+1}]$.
Iz definicije Newtonovog polinoma slijedi:

$$\begin{aligned}
p_i(x) = f[x_i] + f[x_i, x_i](x - x_i) + f[x_i, x_i, x_{i+1}](x - x_i)^2 \\
+ f[x_i, x_i, x_{i+1}, x_{i+1}](x - x_i)^2(x - x_{i+1})
\end{aligned}$$

Isti polinom se može zapisati i u bazi $(x - x_i)^k$ kao:

$$p_i(x) = \sum_{k=0}^{3} a_k (x - x_i)^k$$

$$a_0 = f[x_i] \qquad\qquad a_1 = s_i$$

$$a_2 = \frac{f[x_i, x_{i+1}] - s_i}{h_i} \qquad\qquad a_3 = \frac{s_{i+1} + s_i - 2f[x_i, x_{i+1}]}{h_i^2}$$

pri čemu je $h_i = x_{i+1} - x_i$.
Iz uvjeta 8.1-8.4 proizlazi sustav:

$$h_i s_{i-1} + 2(h_{i-1} + h_i) s_i + h_{i-1} s_{i+1} = 3(h_i f[x_{i-1}, x_i] + h_{i-1} f[x_i, x_{i+1}]), \qquad i \in [1, n-1]$$

Dobiveni sustav je sastavljen od $n-1$ jednadžbi s $n-1$ nepoznanica i strogo je dijagonalno dominantan po retcima pa zaključujemo da je determinanta sustava različita od 0, odnosno sustav ima jedinstveno rješenje $\{s_1, \ldots, s_{n-1}\}$ čime je potpuno određen kubični spline.

8.1 Drugа derivacija na rubu segmenta

Uvjeti: druga derivacija ruba

Kao što je prethodno navedeno uvjet 8.4 nije jedini način za postizanje jedinstvenih rješenja. Možemo ih postići i sa:

$$\begin{aligned}
\phi_0'' &= f''(x_0) \\
\phi_n'' &= f''(x_n)
\end{aligned} \tag{8.5}$$

Ovdje uvijet 8.5 osigurava da je stopa promjene nagiba ("savijenost") na rubovima interpoliranih segmenata jednaka nagibu funkcije.

Uvijeti 8.1, 8.2, 8.3 i 8.5 dovode do linearnog sustava od $n+1$ jednadžbe s $n+1$ nepoznanicom, odnosno

$$\begin{aligned}
p_1''(x_0) &= f''(x_0) \\
h_i s_{i-1} + 2(h_{i-1} + h_i) s_i + h_{i-1} s_{i+1} &= 3(h_i f[x_{i-1}, x_i] + h_{i-1} f[x_i, x_{i+1}]) \\
p_n''(x_n) &= f''(x_n)
\end{aligned}$$

Dobiveni sustav je dijagonalno dominantan po retcima pa postoji jedinstveno rješenje sustava $\{s_0, \ldots, s_n\}$. Funkcija $\phi(x)$ je klase $\mathcal{C}^2[a, b]$. Ako funkcija $f(x)$ ima omeđenu četvrtu derivaciju, tada vrijedi

$$f(x) = \phi(x) + (O)(h^4), \qquad h \to 0.$$

Primjer 1: uporaba kubičnog splinea

Primjenom kubičnog splinea odrediti $\phi(2.33)$ i $\phi(4.21)$ ako su zadani podaci:

x	1	2	3	4	5
$f(x)$	1	0.5	0.333	0.25	0.2
$f'(x)$	-0.5	-0.25	-0.111	-0.0625	-0.04

U zadanoj tablici vrijedi $s_i = f'(x_i)$. Traženi polinomi su oblika $p_i(x) = \sum_{k=0}^{3} a_k(x-x_i)^k$ pa prema sljedećem određujemo vrijednosti koeficijenata a_0, a_1, a_2, a_3 za svaki podsegment posebno.

Tablica podjeljenih razlika je

$$
\begin{array}{ccc}
1 & 1 & \\
 & & -0.5 \\
2 & 0.5 & \\
 & & -0.167 \\
3 & 0.333 & \\
 & & -0.083 \\
4 & 0.25 & \\
 & & 0.05 \\
5 & 0.2 &
\end{array}
$$

- Za $p_1(x)$ na intervalu $[1,2\rangle$ vrijedi:

$$
\begin{aligned}
a_0 &= f(x_0) = 1, \qquad a_1 = s_0 = -0.5, \\
a_2 &= \frac{f[x_0,x_1]-s_0}{h} - h\cdot a_3 = -0.25, \\
a_3 &= \frac{s_0+s_1-2f[x_0,x_1]}{h^2} = 0.25, \\
p_1(x) &= 1 - 0.5(x-1) - 0.25(x-1)^2 \\
&\quad +0.25(x-1)^3
\end{aligned}
$$

- Za $p_2(x)$ na intervalu $[2,3\rangle$ vrijedi:

$$
\begin{aligned}
a_0 &= f(x_1) = 0.5, \qquad a_1 = s_1 = -0.25, \\
a_2 &= \frac{f[x_1,x_2]-s_1}{h} - h\cdot a_3 = 0.11, \\
a_3 &= \frac{s_1+s_2-2f[x_1,x_2]}{h^2} = -0.027, \\
p_2(x) &= 0.5 - 0.25(x-2) - 0.027(x-2)^2 \\
&\quad +0.11(x-2)^3
\end{aligned}
$$

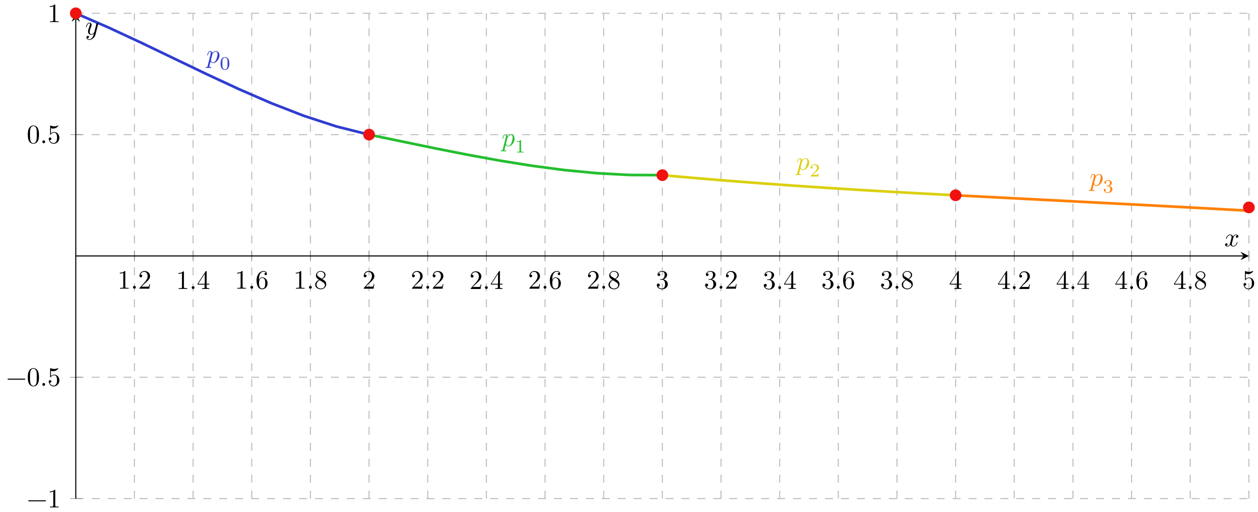
- Za $p_3(x)$ na intervalu $[3,4\rangle$ vrijedi:

$$
\begin{aligned}
a_0 &= f(x_2) = 1, \qquad a_1 = s_2 = -0.111, \\
a_2 &= \frac{f[x_2,x_3]-s_2}{h} - h\cdot a_3 = 0.0355, \\
a_3 &= \frac{s_2+s_3-2f[x_2,x_3]}{h^2} = -0.0075, \\
p_3(x) &= 0.333 - 0.111(x-3) + 0.0355(x-3)^2 \\
&\quad -0.0075(x-3)^3
\end{aligned}
$$

- Za $p_4(x)$ na intervalu $[4,5\rangle$ vrijedi:

$$
\begin{aligned}
a_0 &= f(x_3) = 0.25, \qquad a_1 = s_3 = -0.0625, \\
a_2 &= \frac{f[x_3,x_4]-s_3}{h} - h\cdot a_3 = 0.0015, \\
a_3 &= \frac{s_3+s_4-2f[x_3,x_4]}{h^2} = -0.0025, \\
p_4(x) &= 0.25 - 0.0625(x-4) + 0.0015(x-4)^2 \\
&\quad -0.0025(x-4)^3
\end{aligned}
$$

Konačno, $\Phi(2.33) = p_2(2.33) = 0.4286$.



8.2 Greška kubičnog splinea

Za kubični spline ocjena greške je

$$|f(x) - \phi(x)| \leq \frac{\mathrm{M}_4}{4!} \frac{h^4}{16} = \frac{1}{384} h^4 \mathrm{M}_4, \tag{8.6}$$

$$\mathrm{M}_4 = \max_{x \in [a,b]} |f^{(IV)}(x)|, \tag{8.7}$$

$$h = \max_{i \in [1,n]} h_i, \qquad h_i = x_{i+1} - x_i. \tag{8.8}$$

Drugim rječima, ravnomjernim povečavanjem čvorova kubičnog splinea tako da $h \to 0$, maksimalna greška te interpolacije se približava 0.

9 Regresija

Ni polinomijalna interpolacija ni kubični spline nisu dobri za aproksimaciju vrijednosti koje nisu omeđene poznatim vrijednostima.

- Kubični spline ima manje oscilacije, no i dalje se pojavljuju značajne oscilacije i odstupanja na krajnjim segmentima.
- Interpolacijom se greške u mjerenim podacima "ugrađuju" u interpolacijsku funkciju.

Regresija je metoda određivanja funkcije koja blisko prati zadane podatke na način gdje opisuje njihovo ponašanje uz **minimalno rasipanje izmjerenih vrijednosti oko poznatih segmenata**. Dakle, neće davati vrijednosti jednake izmjerenim podacima, nego njima **bliske vrijednosti**.

Udaljenost d_i zadane točke (x_i, y_i) od točke na grafu se mjeri normom. Najčešći izbor je L_2 norma oblika

$$d_i = \sqrt{(x_i - x_i)^2 + (y_i - f(x_i))^2}.$$

Tako se kao "najbliža" funkcija smatra onom kod koje su te udaljenosti po svim točkama najmanje. Budući da su $d_i \geq 0, i \in [0, n]$, to znači da je **najbliža funkcija** ona kod koje je suma $\sum_{i=0}^{n} d_i$ najmanja. Traženje minimuma te sume može biti problematično pa se u praksi uvijek traži minimum sume

$$\sum_{i=0}^{n} d_i^2 \tag{9.1}$$

koja daje jednake rezultate, ali ih je znatno lakše odrediti.

Minimiziranje sume kvadrata udaljenosti se smatra kriterijem kod metode regresijem pa se metoda regresije također naziva i **metodom najmanjih kvadrata**.

Nemoguće je naći minimum sume 9.1 na skupu svih mogućih funkcija. Za zadane točke **prvo biramo model ili klasu funkcije** između kojih ćemo tražiti onu koja minimizira sumu 9.1.

9.1 Linearna regresija

Najjednostavniji odabir je traženje funkcije na skupu svih pravaca $f(x) = a_0 + a_1 x$ i takvu regresiju zovemo **linearna regresija**. Tako problem minimizacije 9.1 svodimo na problem određivanja koeficijenata a_0 i a_1 u sumi

$$S = \sum_{i=0}^{n} (y_i - (a_0 + a_1 x_i))^2$$

Dakle radi se o funkciji $S = S(a_0, a_1)$ s dvije varijable kojoj jednostavno nalazimo minimum.

Prirodno, zadajemo uvjete za stacionarnu točku funkcije dvije varijable

$$\partial_{a_0} S = 0, \qquad \partial_{a_1} S = 0.$$

Nakon sređivanja uvjeti daju sustav dvije linearne jednadžbe

$$a_0(n+1) + a_1 \sum_{i=0}^{n} x_i = \sum_{i=0}^{n} y_i$$
$$a_0 \sum_{i=0}^{n} x_i + a_1 \sum_{i=0}^{n} x_i^2 = \sum_{i=0}^{n} x_i y_i$$

za dvije nepoznanice a_0 i a_1.

Osim kod izuzetno nepovoljnog rasporeda točaka koji uistinu ne odgovara pravcu, ovaj **sustav ima determinantu različitu od nule** te time i jedinstveno rješenje, kada je broj točaka za koje ga rješavamo n$\geq$ 2. Budući da je $S \geq 0$, rješenje mora biti lokalni minimum te je time problem metode linearne regresije u potpunosti rješen.

Primjer 1: uporaba metode najmanjih kvadrata

Podatke zadane tablicom apriksimirati linearnom funkcijom koristeći metodu najmanjih kvadrata.

x	0	1	3
y	1	2	3

9.2 Kvadratna regresija

U slučaju kvadratne funkcije oblika

$$f(x) = a_0 + a_1 x + a_2 x^2$$

imamo sumu

$$S = \sum_{i=0}^{n} (y_i - (a_0 + a_1 x + a_2 x^2))^2$$

te potrebno zadati uvjete stacionarne točke za funkciju s 3 varijable:

$$\partial_{a_0} S = 0, \qquad \partial_{a_1} S = 0, \qquad \partial_{a_2} S = 0.$$

Sređivanjem dobivamo sustav koji određuje koeficijente funkcije:

$$\begin{aligned}
a_0(n+1) + a_1 \sum_{i=0}^{n} x_i + a_2 \sum_{i=0}^{n} x_i^2 &= \sum_{i=0}^{n} y_i, \\
a_0 \sum_{i=0}^{n} x_i + a_1 \sum_{i=0}^{n} x_i^2 + a_2 \sum_{i=0}^{n} x_i^3 &= \sum_{i=0}^{n} x_i y_i, \\
a_0 \sum_{i=0}^{n} x_i^2 + a_1 \sum_{i=0}^{n} x_i^3 + a_2 \sum_{i=0}^{n} x_i^4 &= \sum_{i=0}^{n} x_i^2 y_i.
\end{aligned}$$

Budući da se odredivanje nepoznatih koeficijenata svodi na rješavanje linearnog sustava, aproksimacija podataka kvadratnom funkcijom u smislu metode najmanjih kvadrata je opet linearan problem kada je broj točaka $n \geq 3$.

Primjer 2

Podatke zadane tablicom apriksimirati kvadratnom funkcijom koristeći metodu najmanjih kvadrata.

x	-1	-0.5	0.0	0.5	1.0
y	1.0	0.5	0.0	0.5	0.2

9.3 Polinomna regresija

I prethodnih regresija je vidljivo da je regresiju također moguće primjeniti u generalnom obliku koji je primjenjiv na sve polinomne funkcije stupnja m.

Regresija kojom se traži "najbliža" funkcija na skupu svih polinoma odabranog stupnja m:

$$f(x) = a_0 + a_1 x + \cdots + a_m x^m$$

se zove **polinomna regresija polinomom stupnja m**. Tako se nepoznavanje funkcije svodi na nepoznavanje $m+1$ koeficijenata $a_j, j \in [0, m]$.

Suma dobiva oblik

$$S = \sum_{i=0}^{n} \left(y_i - \sum_{j=0}^{m} a_j x_i^j \right).$$

Analogni polinomima nižeg stupnja, počinje se s uvjetom za stacionarnu točku funkcije više varijabli:

$$\partial_{a_j} S = 0, \qquad j \in [0, m].$$

Nakon sređivanja jednadžbe poprimaju oblik:

$$\sum_{k=0}^{m} a_k \sum_{i=0}^{n} x_i^{k+j} = \sum_{i=0}^{n} x_i^j y_i, \qquad j \in [0, m]$$

Osim u slučajevima kad je polinomni model loš izbor za zadane točke, ovaj će sustav imati jedinstveno rješenje ako vrijedi $n \geq m$.

9.4 Modeli svedeni na linearne

Ostali modeli funkcija za zadane točke u pravilu će voditi na složenije probleme traženja minimuma sume po parametrima odabranog modela funkcije. Poželjno je izbjeći aproksimaciju polinomima višeg stupnja.

Neki od modela koje se može svesti na linearne su:

- eksponencijalni model $y = ae^{bx}$ kojeg se svodi na linearni model $\ln y = \ln a + bx$ za točke $(x_i, \ln y_i), i \in [0, n]$,
- model potencije $y = ax^b$ kojeg se svodi na linearni model $\ln y = \ln a + b \ln x$ za točke $(\ln x_i, \ln y_i), i \in [0, n]$,
- model rasta do zasićenja $y = \frac{ab}{x+b}$ kojeg se svodi na linearni model $\frac{1}{y} = \frac{1}{a} + \frac{b}{ax}$ za točke $(\frac{1}{x_i}, \frac{1}{y_i}), i \in [0, n]$.

10 Primjena u Pythonu

Python 3 sadrži mnoge gotove alate za interpolaciju i regresiju.

10.1 Crtanje grafova

Za crtanje grafova se može korisiti `Matplotlib` biblioteka koja je re-exportana i kroz `pylab` biblioteku za primjenu u podatkovnoj znanosti.

```
import numpy as np
import pylab

x = np.linspace(0, 20, 1000)
y = np.sin(x)

pylab.plot(x, y)
```

10.2 Računalna algebra

`sympy` biblioteka pruža jako dobru podršku za rad sa simbolima u izračunu. Simbol je nepoznati dio izraza

10.3 Vandermondeova diskriminanta

10.4 Interpolacija

10.4.1 Lagrangeova interpolacija

10.4.2 Newtonova interpolacija

10.5 Regresija

```
from numpy import *
from scipy import interpolate
import pylab

def f(x):
    return x*cos(x)

xcvorovi=genfromtxt('tocke.txt', delimiter=',')
print(xcvorovi)
ycvorovi=f(xcvorovi)
print(ycvorovi)

lspline=interpolate.interp1d(xcvorovi, ycvorovi, kind='slinear')
kspline=interpolate.interp1d(xcvorovi, ycvorovi, kind='cubic')

print('Aps greska za lspline u x=1 je: ', abs(f(1)-lspline(1)))
print('Aps greska za kspline u x=1 je: ', abs(f(1)-kspline(1)))

xtocke=linspace(min(xcvorovi), max(xcvorovi), 100)
ylspline=lspline(xtocke)
ykspline=kspline(xtocke)
yfunkcija=f(xtocke)

pl.figure()
pl.plot(xcvorovi, ycvorovi, 'o', xtocke, ylspline, '-r',
        xtocke, ykspline, '-g', xtocke, yfunkcija, '--')
pl.legend(['cvorovi', 'lspline', 'kspline', 'funckija'])
pl.grid()
pl.show()
```