

Osnovi računarstva II

Čas 8

Miloš Daković

Elektrotehnički fakultet – Podgorica

7. april 2025.

Polinomi

- Polinom stepena N je potpuno određen svojim koeficijentima. Obilježimo nezavisno promjenljivu sa x i koeficijente uz odgovarajući stepen x -a sa $a_0, a_1, \dots, a_N$. Dobijamo:

$$P(x) = a_N x^N + a_{N-1} x^{N-1} + \dots + a_2 x^2 + a_1 x + a_0$$

- Polinom se u Octave/MATLAB okruženju zapisuje kao niz koeficijenata:

$$P = [a_N, a_{N-1}, \dots, a_2, a_1, a_0]$$

- Na primjer, polinom $P(x) = 7x^5 - 2x^4 + 6x^3 + 5x^2 + 8$ definišemo nizom od šest koeficijenata:

$$P = [7, -2, 6, 5, 0, 8]$$

Uočite da je prvi koeficijent u nizu vezan za najveći stepen u polinomu, da je poslednji koeficijent slobodni član polinoma i da je koeficijent a_k jednak nuli ukoliko se x^k ne pojavljuje u posmatranom polinomu.

Izračunavanje vrijednosti polinoma

- Vrijednost polinoma u nekoj tački dobijamo funkcijom **polyval**.
- Ova funkcija ima dva argumenta: prvi argument su koeficijenti posmatranog polinoma, a drugi argument vrijednosti nezavisno promjenljive u kojima računamo vrijednost polinoma.
- Drugi argument može biti skalar, niz ili matrica. Vrijednost polinoma se računa za svaki element posebno i dobija se rezultat istih dimenzija.

- Primjeri:

polyval([1, 2, 5], 3) $\rightarrow$ 20

polyval([1, 0, 0, -1], 1:4) $\rightarrow$ [0, 7, 26, 63]

- Izračunati vrijednosti polinoma $P(x) = 2x^3 - x + 1$ za vrijednosti x od 0 do 2 sa korakom 0.1:

p = [2, 0, -1, 1]

x = 0:0.1:2

y = **polyval**(**p**, **x**)

Nule polinoma

- Nule polinoma dobijamo funkcijom **roots**. Prosljeđujemo joj koeficijente polinoma a ona nam vraća vektor kolonu sa nulama polinoma.

- Primjeri:

$$\text{roots}([1, 0, -4]) \rightarrow [-2; 2]$$

$$\text{roots}([1, -2, 2]) \rightarrow [1 + i; 1 - i]$$

- Ukoliko su nam nule polinoma poznate, koeficijente polinoma možemo dobiti komandom **poly** kojoj prosljeđujemo niz nula.

- Primjeri:

$$\text{poly}([1, 2]) \rightarrow [1, -3, 2] \quad (x-1)(x-2) = x^2 - 3x + 2$$

$$\text{poly}([-2, 2, 0, 0]) \rightarrow [1, 0, -4, 0, 0]$$

- Uočite da funkcija **poly** uvijek daje 1 kao koeficijent polinoma uz najveći stepen. Funkcije **poly** i **roots** su međusobno inverzne.

Množenje polinoma

- Množenje dva polinoma dobijamo funkcijom **conv**. Na primjer ako je $P(x) = x^2 - 3$ i $Q(x) = 2x^3 - x + 1$, tada koeficijente proizvoda $R(x) = P(x)Q(x)$ dobijamo ovako:

conv([1, 0, -3], [2, 0, -1, 1]) $\rightarrow$ [2, 0, -7, 1, 3, -3]

što znači da je rezultat množenja:

$$R(x) = 2x^5 - 7x^3 + x^2 + 3x - 3$$

- Zadatak: Za dato N odredite zbir svih koeficijenata polinoma:
 $P(x) = (1+x)(1+x+x^2)(1+x+x^2+x^3)\cdots(1+x+\dots+x^N)$ i izračunajte vrijednost polinoma $y = P(x)$ za $x = 1/\pi$.

P=1;

for k=1:N

P = conv(P, ones(1, k+1));

end

zbir = sum(P)

y = polyval(P, 1/pi)

Dijeljenje polinoma

- Dijeljenjem polinoma $P(x)$ sa polinomom $Q(x)$ dobijaju se dva polinoma: rezultat $R(x)$ i ostatak $S(x)$, takvi da vrijedi:
 $P(x) = Q(x)R(x) + S(x)$, pri čemu je stepen polinoma $S(x)$ manji od stepena polinoma $Q(x)$.
- Ovu operaciju obavlja funkcija **deconv** koja ima dva ulazna argumenta (polinomi P i Q) i dvije izlazne vrijednosti (koeficijenti polinoma R i S).

- Primjer: Naći rezultat i ostatak pri dijeljenju polinoma $4x^6 + x^5 - x^4 + 2x^3 + x^2 - 7x + 1$ sa polinomom $x^2 - 2$. Rezultat dobijamo komandom:

$[r, s] = \text{deconv}([4, 1, -1, 2, 1, -7, 1], [1, 0, -2])$

$r : [4, 1, 7, 4, 15]$

$R(x) = 4x^4 + x^3 + 7x^2 + 4x + 15$

$s : [0, 0, 0, 0, 0, 1, 31]$

$S(x) = x + 31$

- Uočite da niz koeficijenata s ima nule na početku tako da je operacija: **$\text{conv}(r, q) + s$** izvodiva.

Sabiranje i oduzimanje polinoma

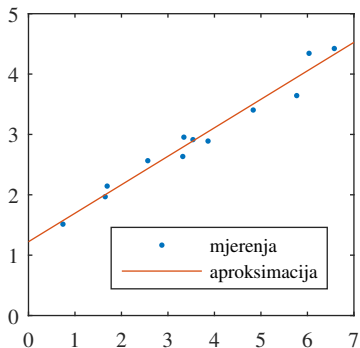
- Jednostavno saberemo (oduzmemo) odgovarajuće koeficijente. Problem se javlja kada polinomi nijesu istog stepena. U tom slučaju možemo polinomu nižeg stepena na početak dodati odgovarajući broj nula.
- Moguće rješenje:

```
function p = SabiranjePolinoma(a,b)
d = [length(a), length(b)];
n = max(d)-d;
p = [zeros(n(1),1), a]+[zeros(n(2),1), b];
```

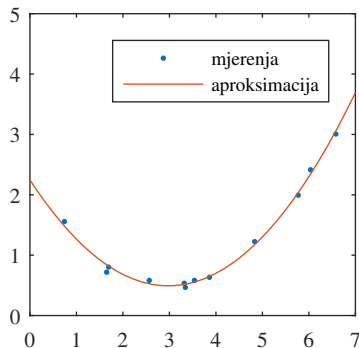
- Razmislite kako bi se iz rezultata mogli eliminisati nulti koeficijenti na početku niza. Oni će se pojaviti, na primjer, u slučaju sabiranja polinoma $x^3 - 2x^2 + 3x + 7$ i polinoma $-x^3 + 2x^2 + x + 1$.
- Pokušajte napraviti drugačiji algoritam za sabiranje polinoma.

Polinomi – aproksimacija i interpolacija podataka

- Često nam je potrebno odrediti koeficijente polinoma, datog stepena, koji na najbolji mogući način aproksimira (i interpolira) dati skup mjerenja.



$$y = ax + b$$



$$y = ax^2 + bx + c$$

Aproksimacija i interpolacija podataka

- Funkcija **polyfit** se koristi za rješavanje ovakvih problema.
- Funkcija ima tri ulazna argumenta. Prvi i drugi su nizovi **x** i **y** koji moraju biti istih dimenzija a predstavljaju mjerenja. Treći argument je stepen polinoma kojim želimo aproksimirati zadata mjerenja.
- Funkcija vraća koeficijente polinoma traženog stepena koji aproksimira mjerene podatke sa najmanjom kvadratnom greškom.
- Dobijeni polinom možemo izračunati za proizvoljnu vrijednost argumenta **x** i time interpolirati skup mjerenja.

```
x = [ 1,  2,  5,  7,  15];  
y = [19, 12,  3,  7, 103];  
p = polyfit(X,Y,2)  
y3= polyval(p,3)  
x = 0:0.1:15;  
y = polyval(p,x);  
plot(x,y,X,Y,'*')
```

Izvod polinoma

- Napišite funkcijski fajl **izvod.m** koji uzima koeficijente polinoma $P(x)$ a vraća koeficijente izvoda ovog polinoma po varijabli x , $P'(x) = \frac{d}{dx}P(x)$.
- Iz matematike znamo da je izvod $a_k \cdot x^k$ jednak $k \cdot a_k \cdot x^{k-1}$, za $k > 0$. Izvod konstantnog člana a_0 je nula.
- Predlog rješenja:

```
function r = izvod(p)
d = length(p)-1;
r = p(1:d) .* (d:-1:1);
```

- Analizirajte predloženo rješenje. Provjerite ga na nekoliko primjera. Pokušajte da nađete izvod konstantnog polinoma $P(x) = 7$. Modifikujte rješenje tako da i u ovom slučaju daje očekivani rezultat.

Integral polinoma

- Napišite funkcijski fajl ***integral.m*** koji uzima koeficijente polinoma $P(x)$ a vraća koeficijente integrala ovog polinoma po varijabli x , $R(x) = \int P(x) dx + C$. Drugi ulazni argument je opcionalna integraciona konstanta, čija je podrazumijevana vrijednost nula.
- Znamo da je integral $a_k \cdot x^k$ jednak $\frac{1}{k+1} \cdot a_k \cdot x^{k+1} + \textit{konstanta}$.
- Predlog rješenja:

```
function r = integral(p, c)  
if nargin < 2  
    c = 0;  
end  
r = [p ./ (length(p):-1:1) , c];
```

- Analizirajte predloženo rješenje.

Stringovi

- Octave i MATLAB prepoznaju varijable tekstualnog tipa. Ranije smo vidjeli da se one mogu definisati korišćenjem apostrofa. Ukoliko se apostrof pojavljuje u stringu koji zadajemo potrebno ga je dva puta ponoviti.

```
str1 = 'Tekstualni tip varijable'
```

```
str2 = 'Apostrof ' ' u tekstu'
```

- Ove varijable su vektor vrste, pri čemu je svaki element vektora jedno slovo. Indeksiranjem možemo dobiti:

```
str1(1:5) početnih pet slova u stringu (ako ih ima toliko)
```

```
str1(2:2:end) svako drugo slovo u stringu
```

```
str1(end:-1:1) „naopako“ napisan string
```

```
str3 = [str1, ' ', str2] tri stringa spojena u jedan
```

```
str1(12)='T' stavljamo T na mjestu dvanaestog slova
```

Funkcije za rad sa stringovima

- **length(*s*)** dužina stringa *s*
- **blanks(*N*)** string od *N* razmaka (blanko karaktera)
- **deblank(*s*)** uklanja blanko karaktere sa kraja stringa
- **strcmp(*s1*, *s2*)** poredi dva stringa na jednakost
- **strcmpi(*s1*, *s2*)** poredi dva stringa na jednakost ne vodeći računa o malim i velikim slovima
- **num2str(*x*)** prevodi broj *x* u njegov tekstualni zapis
- **int2str(*n*)** prevodi cio broj *x* u njegov tekstualni zapis
- **str2num(*s*)** konvertuje string *s* u broj
- **eval(*s*)** tumači string *s* kao Octave/MATLAB komandu i izvršava ga. Na primjer, ako je u vrijednost varijable *n* = 3 komanda:
eval(['x', int2str(*n*), '=17'])
će dodijeliti vrijednost 17 varijabli *x3*.

Funkcije za rad sa stringovima

- **abs(*s*)** daje numeričke vrijednosti kodova svih karaktera u stringu *s*. **abs('A')** daje vrijednost 65 (ASCII kod slova 'A').
- **char(*n*)** niz numeričkih vrijednosti (kodova) pretvara u niz karaktera (string). **char(67)** je slovo 'C'.
- **strfind(*s1*, *s2*)** traži string *s2* u stringu *s1* i ako ga pronađe vraća odgovarajuću poziciju. U suprotnom vraća nulu.
- **strrep(*s1*, *s2*, *s3*)** svako pojavljivanje stringa *s2* u stringu *s1* zamjenjuje stringom *s3*.
- **regexp(*s*, *re*)** pretražuje pojavljivanja regularnog izraza *re* u stringu *s*. Na primjer **regexp(*s*, 'd+')** pronalazi cijele brojeve u stringu. Provjerite dokumentaciju za ovu komandu komandama **help** i **doc**.
- **lower(*s*)** (**upper(*s*)**) konvertuje sva slova u mala (velika) slova
- **split**, **bin2dec**, **dec2bin**, **dec2hex**, **hex2dec**, ... Komanda **help strfun** daje spisak svih funkcija za rad sa stringovima.

Primjer rada sa stringovima

- U stringu s potrebno je svako pojavljivanje slova 'c' zamijeniti slovom 'd'.
- Moguće rješenje:

```
for k=1:length(s)
    if s(k)=='c'
        s(k)='d';
    end
end
```

- Može i ovako:

```
strrep(s, 'c', 'd')
```

- Ili ovako:

```
s(s=='c')='d'
```

Zadatak sa stringovima

- String sadrži neki izraz u kojem se pojavljuju male i srednje zagrade. Napisati funkciju **provjera.m** koja će ispitati da li su, u datom stringu, zagrade „korektno“ postavljene.
- Korektnost zagrada se provjerava tako što se otvorena i zatvorena zagrada moraju poklapati po tipu.
Naime, stringovi: $'(x + [a - b]c) + 1'$ i $'(1 + (2 + [3 + (4 + 5)]))'$ imaju korektno postavljene zagrade,
dok stringovi: $'(x + 1)'$ i $'(a + (b + [c + (d + e) - g] - h) - c) - f'$ nemaju korektne zagrade.
- Funkcija vraća vrijednost **1** za korektne zagrade i **0** ako zagrade nijesu korektne.
- Ukoliko se od funkcije traže dvije izlazne vrijednosti, druga vrijednost je pozicija u stringu na kojoj je greška u postavljanju zagrada otkrivena. Za string $'[a + b) + (c + d]'$ odgovor treba biti 5.

Ostali zadaci

- Nađite zbir: $1^2 + 3^2 + \dots + 99^2$
- Izračunajte: $\sin 15^\circ + \sqrt{\frac{1}{\pi+1}}$
- Elementi kvadratne matrice A reda N su definisani kao

$$a_{i,j} = \begin{cases} 0 & \text{za } i < j \\ 1 & \text{za } i = j \\ \frac{1}{i-j} & \text{za } i > j \end{cases}$$

Izračunajte determinantu matice A za $N = 7$, $N = 12$ i $N = 100$.

- Za zadati prirodan broj n odredite sve moguće načine kako se n može zapisati u obliku zbira kvadrata tri prirodna broja pri čemu redoslijed sabiraka nije bitan. Nađite sva rješenja za $n = 386$ i $n = 383$. Odredite koliko brojeva u intervalu od 1 do 100 ne može biti zapisano na ovaj način i koji su to brojevi.