

Laborator 3

Lucrul cu polinoame și cu șiruri de caractere în Matlab.

1. Obiective urmărite.

- Familiarizarea cu operațiile folosite în aplicațiile cu polinoame și cu șiruri de caractere.
- Prezentarea unor aplicații semnificative

2. Prezentarea funcțiilor de lucru cu polinoame.

În Matlab un polinom se reprezintă printr-un vector linie, al coeficienților, în ordinea descrescătoare a puterilor.

Exemplu: polinomul $p(x) = x^5 - 3x^2 + 2$ se reprezintă prin: $p = [1 \ 0 \ 0 \ -3 \ 0 \ 2]$

Funcțiile predefinite de lucru cu polinoame sunt:

- $p = \text{poly}(x)$ – creează polinomul p (vector linie) pe baza rădăcinilor sale x (vector coloană).
Funcția calculează coeficienți polinomului rezultat din dezvoltarea:
 $(x-x_1)(x-x_2) \dots (x-x_n)$
- $x = \text{roots}(p)$ – calculează în vectorul coloană x , rădăcinile polinomului p
- $c = \text{conv}(a, b)$ – calculează polinomul produs $c(x) = a(x) * b(x)$
- $[q, r] = \text{deconv}(a, b)$ – calculează polinoamele cât și rest obținute din împărțirea polinoamelor $a(x) / b(x)$
- $\text{polyval}(p, x)$ – calculează valoarea polinomului în punctul scalar x
- $\text{polyvalm}(p, A)$ – calculează valoarea polinomului p în argumentul matricial A
- $c = \text{polyder}(a)$ – calculează polinomul c , derivata polinomului a
- $c = \text{polyder}(a, b)$ – calculează derivata polinomului produs $a(x) * b(x)$
- $[q, r] = \text{polyder}(a, b)$ – calculează $\frac{q(x)}{r(x)} = \left(\frac{a(x)}{b(x)} \right)$

3. Exemple cu polinoame

1. Scrieți o funcție care calculează cel mai mare divizor comun a două polinoame, folosind algoritmul lui Euclid.

```
function c = cmmdc(a, b)
% calcul cmmdc folosind algoritmul lui Euclid
while 1
    [q, r] = deconv(a, b);
    a = b;
    b = r;
    while length(b) > 1 && b(1) > 0
        b = b(2 : length(b));
    end
    if ~any(b)
        break;
    end
end
c = a;
```

Observație: A fost necesar să se elimine zerourile de la începutul restului, pentru a putea face împărțirea la pasul următor (un vector care începe cu 0 nu este acceptat ca un polinom valid).

2. Scrieți o funcție care calculează polinomul Cebâșev de gradul n , cunoscând relația de recurență:
 $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$, $T_0(x) = 1$, $T_1(x) = x$.

```
function c = ceby(n)
a = 1;
b = [1 0];
if n == 0
    c = a;
else
    if n == 1
        c = b;
    else
        for i=2:n
            c = conv([2 0], b) - [0 0 a];
            a = b;
            b = c;
        end
    end
end
```

Observație: Atenție la adunarea și scăderea polinoamelor, vectorii trebuie să aibă aceeași lungime.

3. Calculați coeficienții polinomului de interpolare Lagrange, cunoscând că:

$$p_{n-1}(x) = \sum_{i=1}^n \left(\prod_{j=1, j \neq i}^n (x - x_j) / \prod_{j=1, j \neq i}^n (x_j - x_i) \right)$$

Considerăm matricea: $A = \begin{bmatrix} 1 & x_1 - x_2 & \cdots & x_1 - x_n \\ x_2 - x_1 & 1 & \cdots & x_2 - x_n \\ \cdots & \cdots & \cdots & \cdots \\ x_n - x_1 & x_n - x_2 & \cdots & 1 \end{bmatrix}$. Produsele elementelor pe coloane

prod(A) reprezintă chiar termenii numitorului în expresia polinomului Lagrange.

Aceasta poate fi formată astfel:

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} + \begin{bmatrix} x_1 - x_1 & x_1 - x_2 & \cdots & x_1 - x_n \\ x_2 - x_1 & x_2 - x_2 & \cdots & x_2 - x_n \\ \cdots & \cdots & \cdots & \cdots \\ x_n - x_1 & x_n - x_2 & \cdots & x_n - x_n \end{bmatrix}$$

$$A = I_n + \begin{bmatrix} x_1 & x_1 & \cdots & x_1 \\ x_2 & x_2 & \cdots & x_2 \\ \cdots & \cdots & \cdots & \cdots \\ x_n & x_n & \cdots & x_n \end{bmatrix} - \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_1 & x_2 & \cdots & x_n \\ \cdots & \cdots & \cdots & \cdots \\ x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

$$A = I_n + \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & x_n \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & x_n \end{bmatrix}$$

```
function p = lagr(x, y)
%calcul coeficienți polinom de interpolare Lagrange
n = length(x);
s = poly(x);
A = eye(n,n) + diag(x)*ones(n,n)-ones(n,n)*diag(x);
z = prod(A);
p = zeros(1,n);
for i=1:n
    [q, r] = deconv(s, [1 -x(i)]);
    p = p + q.*y(i)./z(i);
end
```

4. Utilizarea șirurilor în Matlab.

Un șir este un tablou de caractere. Fiecare caracter este reprezentat intern printr-o valoare numerică pe 16 biți în codificare *Unicode*.

O constantă șir de caractere se reprezintă între apostrofi.

```
s = 'Pop';
```

Afișarea unui singur element sau a unui șir se face cu funcția `disp`.

```
x=1:2:10;
disp(x=''), disp(x)
x=
     1     3     5     7     9
```

Citirea de la tastatură poate fi însoțită de afișarea unui mesaj în funcția `input`:

```
A=input('Matricea A=')
Matricea A=[1 2; 3 4]
A =
     1     2
     3     4
```

Concatenarea a două șiruri se face:

- cu operatorul de concatenare `[]` :

```
[s , 'escu']
```

- cu funcția `strcat` :

```
strcat(s, 'escu')
```

Obținerea reprezentărilor numerice ale caracterelor se face aplicând șirului de caractere funcția `double`:

```
s1 = double(s)
s1 =
     80     111     112
```

crează un vector linie cu 3 întregi.

Conversia vectorului numeric `s1` într-un vector de caractere se face cu funcția `char`:

```
s2 = char(s1)
```

Mai multe șiruri de caractere vor fi păstrate într-o matrice, ceea ce presupune că toate șirurile vor avea aceeași lungime, cele mai scurte fiind completate cu spații libere până la lungimea celui mai mare șir.

Pentru a păstra șiruri de caractere de lungimi diferite se folosesc *tabele de celule* – **cell array**.

```
nume = {'Ilie'; 'Grigore'; 'Vasile'}
nume =
     'Ilie'
    'Grigore'
    'Vasile'
```

`nume` este un tabel de celule, creat folosind constructorii `{ și }`.

Clasificarea caracterelor dintr-un șir se poate face aplicând acestuia funcțiile `isletter` sau `isspace`.

Funcția crează un vector de zerouri și unități corespunzând apartenenței caracterelor la categoria testată.

```
isletter('a bc')
ans =
     1     0     1     1
```

Comparația șirurilor se face folosind funcția `strcmp` (cu variantele `strncmp` și `stricmp`), care spre deosebire de funcția din C întoarce 0 dacă șirurile diferă și 1 dacă sunt egale.

Conversia din numere în caractere se face cu funcțiile `int2str` și `num2str`. Funcția `int2str` rotunjește argumentul matrice la valori întregi pe care le convertește în șiruri.

```
A=[1.2 3.5; 2.4 8.2];
```

```
disp(int2str(A))
```

```
1      4
2      8
```

Funcția `num2str` convertește argumentul numeric în șir de caractere

```
disp(num2str(A))
```

```
1.2    3.5
2.4    8.2
```

Funcția `length` determină numărul de caractere al șirului.

Compararea șirurilor cu aceeași lungime se poate face cu operatorul de comparație `==`.

Rezultatul este o secvență de zerouri și unități care marchează caracterele egale din cele două șiruri.

Conversia între diversele baze se face folosind funcțiile: `bin2dec`, `dec2bin`, `hex2dec`, `dec2hex`, `base2dec`

Căutarea și înlocuirea într-un șir s a unui subșir s1 prin alt subșir s2 se face prin:

```
t= strrep(s, s1, s2)
```

Funcția `findstr` determină prima apariție a unui subșir `s1` într-un șir `s` (dacă `s1` nu apare, funcția întoarce []).

```
findstr('ab', 'nabab')
ans =
     2
```

Funcția `strtok`. extrage primul subșir (atom) dintr-un șir `s`, cuprins între spații libere.

```
s1='Unul pentru toti.';
strtok(s1)
ans =
    Unul
```

Funcția permite specificarea unui al doilea parametru de intrare format din separatori și a unui al doilea parametru de ieșire cu restul șirului rămas prin scoaterea atomului.

```
[atom, rest] = strtok(sir, delimitatori)
```

Într-un text cuvintele sunt separate prin delimitatori din șirul ' .,;:-'. Scrieți o funcție având ca parametru un șir, care întoarce un tablou de șiruri, în care în fiecare linie avem câte un cuvânt din text.

```
function cuvinte=sepcuv(text)
%separa cuvintele dintr-un text si le plaseaza intr-un tablou de șiruri
delim = ' .,;:-';
rest = text;
cuvinte = '';
while(any(rest))
    [atom, rest] = strtok(rest, delim);
    cuvinte = strvcat(cuvinte, atom); %concatenare verticala
end
```

5. Probleme de studiu.

1. Scrieți o funcție care calculează coeficienții polinomului ortogonal Laguerre de gradul n , știind că aceste polinoame satisfac relația de recurență:

$$G_{n+1}(x) - (2n+1-x)G_n(x) + n^2G_{n-1}(x) = 0, \quad G_0(x) = 1, \quad G_1(x) = 1-x$$

2. Scrieți o funcție care extrage dintr-un șir de caractere constantele valide și le convertește în baza 10, punându-le într-un vector de întregi. Constantele sunt separate cu spații, virgule, punct virgule sau tab. Constantele octale încep cu 0 și conțin cifre între 0 și 7, cele hexazecimale încep cu 0x sau 0X și conțin cifre hexazecimale, cele binare încep cu B sau b și conțin 0 și 1. (conversiile din octal și hexazecimal pot fi realizate cu funcția `sprintf`)