

# ZH I.

## Lebegőpontos számok, Matlab alapok

### 1. Feladat

#### Összeg:

Írjon egy Matlab függvényt, amely tetszőleges  $n$  esetén megadja az alábbi összeg értékét

$$s = \sum_{k=1}^n z$$

$$1 \leq n \leq 33$$

#### megoldás:

A képlet mindig tartalmaz egy összeget, amiben van egy  $z$  képletet

```
function fun(n)
    k = 1:n;
    s = sum(z);
end
```

#### Szorzat:

Írjon egy Matlab függvényt, amely tetszőleges  $n$  esetén megadja az alábbi szorzat értékét

$$s = \prod_{k=1}^n z$$

$$1 \leq n \leq 33$$

#### megoldás:

A képlet mindig tartalmaz egy szorzatot, amiben van egy  $z$  képletet

```
function fun(n)
    k = 1:n;
    s = prod(z);
end
```

## 2. Feladat

Ábrázolni szeretnénk az

$$f(x) = z$$

függvényt a  $[a, b]$  intervallumon. Ehhez meghatározzuk az intervallum  $n$  darab, egyenlő lépésközű pontját, ezt az  $x$  vektorban tároljuk. Ezután ezekben a pontokban kiszámítjuk a függvény értékét, ezeket az  $y$  vektorba tesszük.

### **megoldás:**

Megadjuk a kapott intervallumot `linspace` segítségével, majd felírjuk a kapott függvényt.

```
function fun(n)
    x = linspace(a,b,n);
    y = z;
end
```

## 3. Feladat

Adottak az  $x$  és  $y$  sorvektorok, továbbá az  $n$  természetes szám.

$a$  az a  $2n$  elemű sorvektor legyen, aminek páros sorszámú elemei az  $x$  vektor első  $n$  eleme, a páratlan sorszámú elemei pedig az  $1, 2, \dots, n$  számok.

$b$  az a sorvektor legyen, amit úgy kapunk, hogy elhagyjuk az  $y$  vektor 2., 4. és 5. elemét.

### **megoldás:**

```
function fun(x,y,n)
    a=zeros(1,2*n);
    a(2:2:end)=x(1:n);
    a(1:2:end)=1:n;
    b=y;
    b([2,4,5])=[];
end
```

## **4. Feladat**

$a = 2, t = 4, k_- = -3, k_+ = 4$  esetén mi lesz a 0.125 lebegőpontos szám jobboldali szomszédja?

### **megoldás:**

Átírjuk a számot 2-es számrendszerbe majd megnézzük mennyi a  $t$ , hogy egyszerűbb legyen számolni normalizáljuk, majd a szomszéd típusától függően megnézzük a legelső 1-estől  $t$  számjegyet és a legutolsó számjegy helyére hozzáadunk/kivonunk egyet

$$0.125 = 0.001$$

$$\text{normalizálás : } 0.001 \Rightarrow 0.1000 \cdot 2^{-2}$$

$$t = 4 \Rightarrow 1000$$

$$\text{jobb oldali szomszéd} \Rightarrow 0.1001$$

A kapott szám 0.001001, vagyis 0.140625

## **5. Feladat**

$a = 2, t = 3, k_- = -3, k_+ = 3$  esetén mi lesz a 0.4375 normalizált alakja?

### **megoldás:**

10-es számrendszerbeli szám átírása 2-es számrendszerbe:

$$\text{pl: } 0.4375_{(10)} = 0.0111_{(2)}$$

```

0|.4375 *2  = 0.0111
0|.8750
1|.750
1|.50
1|.0

```

A tizedesvesszőt az első 1-es számjegy elé eltoljuk,  $t$  darab számjegynek kell a tizedesvessző után állnia, majd ez után a kapott számot megszorozzuk  $x$  annyiadik hatványára emelve, ahány jeggyel a tizedesvessző eltolásra került.

Ha a tizedesvessző  $z$  hellyel balra tolódott, akkor  $x^z$ , ha jobbra akkor  $x^{-z}$

```

0.0111 -> 0.111
*** - t számjegy

```

Az így kapott szám:  $2^{-1} \cdot 0.111$

## 6. Feladat

Az  $F = [a = 2, k_- = -6, k_+ = 6, t = 5]$  rendszerben a(z)  $\frac{221}{576}$  szám normalizálva, szabályos kerekítéssel:

### megoldás:

10-es számrendszerbeli szám átírása 2-es számrendszerbe, majd normalizáljuk azt. Megnézzük hogy  $t$  esetén mi a  $t + 1$ -ik számjegye, ha 0 akkor nem változik a szám 1-es esetén hozzáadunk a  $t$ -edikhez 1-et

```

0.11001|000111000111

```

Mivel a  $t + 1$ -edik számjegy 0, nem csinálunk vele semmit.

Majd levágjuk  $t$ -n felüli elemeket

Az így kapott szám:  $2^{-1} \cdot 0.11001$

# Mátrixok, lineáris egyenletrendszerek

## 1. Feladat

$$x = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix},$$

**megoldás:**

$$\|x\|_1 = |a| + |b| + |c| + |d|$$

$$\|x\|_\infty = \max(|a|, |b|, |c|, |d|)$$

$$\|x\|_2 = (|a|^2 + |b|^2 + |c|^2 + |d|^2)^{\frac{1}{2}}$$

## 2. Feladat

Írjon egy függvényt, mely adott t (n elemű) vektor esetén az alábbi A mátrixszal tér vissza

$$A = \begin{bmatrix} 1 & a(t_1) & b(t_1) \\ 2 & a(t_2) & b(t_2) \\ \vdots & & \\ n & a(t_n) & b(t_n) \end{bmatrix}$$

**megoldás:**

```
function fun(t)
    n = 1:numel(t);
    t = reshape(t,1,numel(t));
    A = [n; a(t); b(t)]';
end
```

## 3. Feladat

$$A = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

### **megoldás:**

```
function fun(A)
    B = A;
    B(A>0) = 0;           % Pozitív számokat 0-ra cseréli
    B(A<0) = 0;           % Negatív számokat 0-ra cseréli
    B = [A sum((A<0),2)]; % A sorok végére a negatív számok darabszámát írja
end
```

## **4. Feladat**

Írjon egy függvényt, mely adott  $n$  esetén kiszámolja az  $x = (x_1, \dots, x_n)$  vektor **adott** normáját, ahol  $x_k = y$  függvény, ha  $k = 1, \dots, n$

### **megoldás:**

$\|x\|_1$  esetén `h = sum(abs(x));`

$\|x\|_\infty$  esetén `h = max(abs(x));`

$\|x\|_2$  esetén `h = (sum(abs(x).^2))^(1./2);`

```
function fun(n)
    k = 1:n;
    x = y;
    h = normál forma;
end
```