Bevezetés a MATLAB programozásba
Folyamatosan bővülő feladatgyűjtemény
2017 tavasza,
v9

1. Egyszerű szkriptek, függvények írása

Kötelező feladatok:

1.1 Határozd meg az alábbi értéket a MATLAB segítségével:

$$c = \sqrt[a]{b}$$

(a=5, b=2.4, c=1.1914)

1.2 Határozd meg az alábbi értéket a MATLAB segítségével:

$$d = a^b$$

(a=5, b=2.4, d=47.5813)

1.3 Határozd meg az alábbi értéket a MATLAB segítségével:

$$e = log_a b$$

(a=5, b=2.4, e=0.5440)

1.4 Határozd meg az alábbi értéket a MATLAB segítségével:

$$f = 5,21^{((log_b \sqrt[3]{\pi})^a)}$$

(a=5, b=2.4, e=1.0263)

1.5 Egy derékszögű háromszög két befogója *a* és *b*. Mekkorák a szögei radián és fok mértékegységben? A négy érték kerüljön be a g, h, k, l változókba!

(a=5, b=2.4, g=1.1233 (radián), h=64.3590 (fok), k=0.4475 (radián), l=25.6410 (fok))

1.6 Add meg az b^a előtti prímszámokat MATLAB beépített függvény segítségével, és tárold az m változóban!

(a=5, b=2.4, m=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79])

1.7 Számítsd ki a következő értéket MATLAB beépített függvény segítségével (faktoriális)!

$$n = (a * b)!$$

(a=5, b=2.4, n=479001600)

1.8 Készítsd el a $10 * (a * b)$ prímtényezős felbontását MATLAB beépített függvény segítségével az o változóba!

(a=5, b=2.4, o=[2, 2, 2, 3, 5])

1.9 Számold ki a b sugarú kör területét, és tárold a p változóban! Használj beépített MATLAB kulcsszót a π állandó helyén!

(b=2.4, p=18.0956)

További gyakorlófeladatok:

1.10 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$2,1\pi^{log_{3,2}(\sqrt[10]{8,3})^9}$$

(13.6864)

1.11 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$\frac{2,6j}{log_{3,4}(\sqrt[5]{2,8})^9}$$

(0.0000 + 1.7168i)

1.12 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$6,5i^{log_{4,7}(\sqrt[7]{10,2})^3}$$

(3.4557 + 5.5053i)

1.13 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$2,5\pi^{log_{4,1}(\sqrt[5]{8,3})^9}$$

(54.9670)

1.14 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$2,7\pi^{log_{4,8}(\sqrt[6]{8,1})^7}$$

(16.0268)

1.15 Határozd meg az alábbi kifejezés értékét MATLAB segítségével:

$$6,5j^{log_{3,1}(\sqrt[8]{5,3})^4}$$

(2.6095 + 5.9532i)

1.16 Készíts egy egyszerű függvényt és egy hozzá tartozó szkriptet az alábbi specifikáció szerint:

- a függvény:
  - 2 bemeneti paramétert vár és 2 visszatérési értéke van, ezek
    - bemenet: egy téglalap két oldalának hossza,
    - kimenet: a megadott oldalhosszak mellett a téglalap kerülete és területe;
- a szkript:
  - meghívja az előbbi függvényt az alábbi paraméterekkel, és minden esetben kiírja a visszatérési értékként kapott értékeket:
    - 3,4 és 5,8 *(18.4 és 19.72)*
    - 2,8 és 9,1 *(23.8 és 25.48)*
    - 4,3 és 1,2 *(11 és 5.16)*

2. Vektorok, find és logikai indexelés, 2D ábrázolás

Kötelező feladatok:

2.1 Legyen adott az `x` vektor (függvény bemenete). **Logikai indexelés segítségével** határozd meg azokat a **helyeket és vektorértékeket**, ahol `x` értéke az `[a, b]` zárt intervallumba esik.

(x=[0.5, 0.8, 1.1, 5, 10.2]; a=0.7; b=1.2; c=[0, 1, 1, 0, 0] és d=[0.8000, 1.1000])

2.2 Legyen adott az `x` vektor (függvény bemenete). A **find parancs használatával** határozd meg azokat az **indexeket**, ahol `x` értéke az `[a, b]` zárt intervallumba esik.

(x = [0.5, 0.8, 5, 1.1, 10.2]; a=0.7; b=1.2; e=[2, 4])

2.3 Legyen adott az `x` vektor (függvény bemenete).
1) beépített MATLAB függvény segítségével határozd meg a vektor elemeinek számtani átlagát;

(x = [34, 12, 41, 17, 29, 9, 43, 14]; f=24.8750)

2) **logikai indexelés segítségével** határozd meg azokat az **helyeket és vektorértékeket**, ahol az adott elem értéke nagyobb vagy egyenlő, mint az imént számolt számtani átlag, de kisebb, mint `a`.

(a=34; x = [34, 12, 41, 17, 29, 9, 43, 14]; g=[0, 0, 0, 0, 1, 0, 0, 0] és h=29)

2.4 Legyen adott az `x` vektor (függvény bemenete). **Logikai indexelés segítségével** határozd meg azokat a helyeket, ahol `x` értéke az `[a, b]` zárt intervallumba esik.

(a=25; b=50; x = [42, 11 23, 34, 64, 17, 21, 52, 49, 43, 9, 57, 44, 15];
k = [1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0])

Az elkészült helyvektor alapján másold ki ezeket az elemeket egy új változóba (vektorba)

(l=[42, 34, 49, 43, 44])

majd keresd meg egyetlen beépített MATLAB függvény hívással a legnagyobb elem értékét és indexét ebből az új vektorból.

(érték: m=49; index: n=3)

A korábban elkészült helyvektor alapján és egyetlen beépített MATLAB függvény segítségével határozd meg, hány elem felelt meg a feltételnek?

(o=5)

2.5 Egyetlen beépített MATLAB utasítással hozzunk létre egy `t` idővektort, ami `eps` és π között tartalmaz 100 darab elemet! Az így kapott vektoron értékeljük ki az alábbi összefüggést:

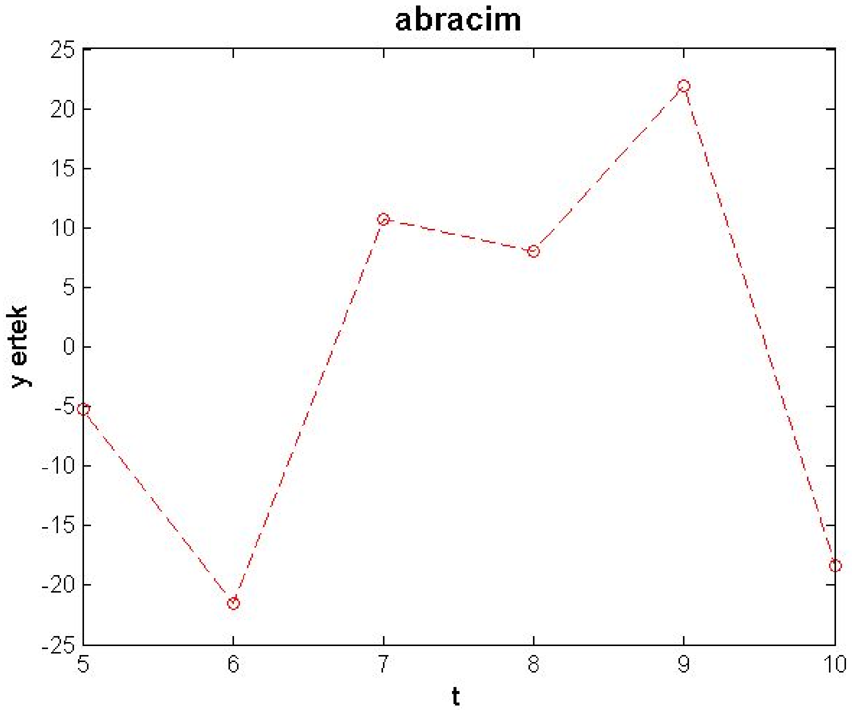
$$fv = \frac{sin(t)}{4{,}7t+3} + 0{,}1cos(t^2)$$

Határozzuk meg azt a `t` értéket, ahol `fv` értéke maximális.

(t értéke: p=0.6664 ahol fv értéke: q=0.1911, ami a 22. index alatt van)

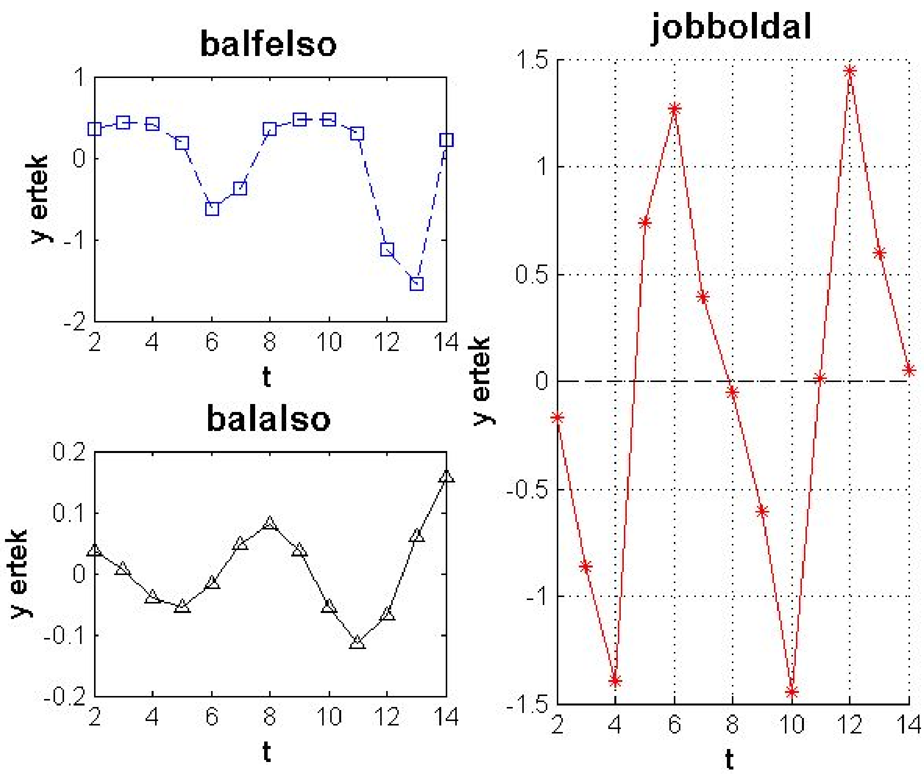
(Segítség: egyetlen beépített MATLAB függvénnyel keresd meg `fv` maximális értékét és indexét, majd az így kapott indexszel mond meg a megfelelő `t` értéket.)

2.6 Állítsd elő az alábbi ábrát:



ahol a `t` mint értelmezési tartomány az [5, 10] zárt intervallum egészeiből áll, az értékkészlet pedig `t/sin(t)` összefüggéssel írható le.

2.7 Állítsd elő az alábbi ábrát:



ahol a balfelső ábra függvénye:

$$0.5 - 0.2e^{ln(t)cos(t)}$$

a balalsó ábra függvénye:

$$\pi^{0.1t-3}sin(t)$$

a jobboldali ábra függvény pedig:

$$\frac{cos(t)}{e^{sin(t)}}$$

Megjegyzés: a subplot-on belül több cellát pl felsorolással tudsz egyszerre címezni:

```
subplot(2, 2, [2, 4]);
```

Megjegyzés: a jobboldali ábrán van rácsozás és egy szaggatott vonal az x tengelyen

További gyakorlófeladatok:

2.8 A [-0.5π, 8.5π] zárt intervallumon, 270 adatpont felett határozd meg az alábbi függvény értékeit: $x \rightarrow xcos(x)$

Logikai indexeléssel válaszd ki a 17.5 és 20.5 közötti zárt **értelmezési tartománybeli** részletet (az indexvektort és az ezzel kiválasztott ÉT-tartományt és értékkészletet mentsd ki új vektorokba). A kiválasztott értékeken belül határozd meg a **legnagyobb érték** (ÉT-beli) **helyét és értékét**.

(hely: 18.9255, érték: 18.8710)

`Find` függvény használatával válaszd ki a 20 és 25 közötti zárt **értelmezési tartománybeli** részletet (a skalár indexek vektorát, és az ezzel kiválasztott ÉT-tartományt és értékkészletet mentsd ki új vektorokba). A kiválasztott értékeken belül határozd meg a **legkisebb érték** (ÉT-beli) **helyét és értékét**.

(hely: 22.0787, érték: -21.9941)

2.9 A [0.6π, 3.6π] zárt intervallumon, 123 adatpont felett határozd meg az alábbi függvény értékeit: $x \rightarrow xsin(x)$

Logikai indexeléssel válaszd ki a -3.6 és -1 közötti zárt **értékkészlet** részletet (az indexvektort és az ezzel kiválasztott ÉT-tartományt és értékkészletet mentsd ki új vektorokba). A kiválasztott értékeken belül határozd meg a **legkisebb érték** (ÉT-beli) **helyét és értékét**.

(hely: 5.5931, érték: -3.5607)

`Find` függvény használatával válaszd ki a 4 és 9 közötti zárt **értékkészlet** részletet (a skalár indexek vektorát, és az ezzel kiválasztott ÉT-tartományt és értékkészletet mentsd ki új vektorokba). A kiválasztott értékeken belül határozd meg a **legnagyobb érték** (ÉT-beli) **helyét és értékét**.

(hely: 7.9879, érték: 7.9164)

3. Mátrixok, formázott kiiratás, vezérlőszerkezetek

3.1 Készíts egy függvényt, aminek a bemenete egy **`A`** mátrix lesz (dimenziónként 5, 2 és 3 véletlen számmal), és az alábbi lépéseket tartalmazza:

- két különálló változóba kimenti az eredeti mátrixnak a 2. dimenzió mentén található két részmátrixát (szemléletesen: az első és második oszlop közötti síknál kettévágjuk a téglatestet; ne felejtsd el az egyke dimenziók eltávolítását, hogy két darab valódi 2d-s mátrixod legyen), -- **`B és C`**
- kiszámítja az első részmátrixban a **sorok maximumának összegét** (beépített függvényekkel), -- **`d`**
- kiszámítja a második részmátrixban az **oszlopok minimumának átlagát** (beépített függvényekkel), -- **`e`**
- egy szöveges változóba kiírja ezt a két adatot némi kísérőszöveggel, egyetlen utasítással, 2-2 tizedesjegy pontosan. -- **`f`**

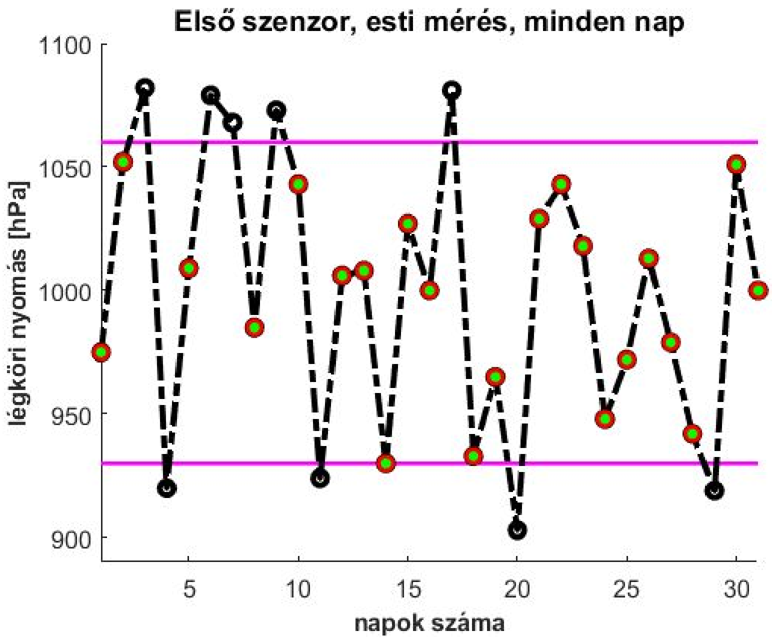
(`rng(1); rand` (kipróbáláshoz), `squeeze` és kettőspont operátor, `max, sum, min, mean, sprintf`
`"B-ben a sorok maximumának összege: 3.84, míg C-ben az oszlopok minimumának átlaga: 0.09 "`)

3.2 Oldd meg az alábbi feladatot egy általad írt Matlab függvény segítségével!
Van 4 darab légnyomásmérő szenzorunk (különböző földrajzi helyeken), melyekről tudjuk, hogy hektoPascalban adják meg a légnyomásértéket. Sajnos csak a [930, 1060] hPa tartományra vannak hitelesítve, így szükséges a nyers merési adatok előzetes ellenőrzése a további feldolgozás és statisztikai elemzés előtt. Mind a négy szenzorral naponta háromszor (reggel, délben, este) mérünk, összesen egy hónapon át.
A merési eredményekéket egy 4 soros, 3 oszlopos, 31 'mélységű' háromdimenziós mátrix tarolja, a neve ez: **`'legnyomasErtekek'`**. Az értékek 900 és 1100 közötti véletlen számok lehetnek (ez alapvetően a függvény bemenete, de a kipróbáláshoz ilyen értékekkel hívd meg a függvényt).
A konkrét feladatok:

1. készítsünk a következő ábrához egy hasonlót, mely az első szenzor esti méréseinek értéket tartalmazza minden nap:

   A két rózsaszín vonal a megbízhatósági határoknál van; csak azok a merési adatpontok vannak kiszínezve, amik helyes tartományon belül vannak. Az alábbi értékek/paraméterek lettek beállítva az ábra készültekor:

   - rózsaszín: vonal színe, szélessége (2 pt);
   - az eredeti adatsor: vonal típusa és színe, vonal vastagsága (3 pt),
     marker mérete (7);
   - a helyes értékek adatsoránál (amit find-al célszerű kikeresni): marker típusa és mérete (7), marker arcának :) és szélének színe, vonal vastagsága (2 pt);
   - tengelyhatárok (x: 1 és 31 között, y: 890 és 1100 között);
   - aktuális axis betűmérete (12 pt);
   - cím szövege, és annak betűmérete (14 pt);
   - tengelyfeliratok szövege, betűmérete (12 pt), szedése (félkövér).



2. logikai indexelés segítségével nullázzuk ki a hitelesített tartományon kívüli értékeket (Ne használj ciklust!!!), innentől ezekkel dolgozzunk tovább

hitelesitettMeresiErtekek

3. az első szenzor esti merései közül, a 11. és 20. nap között (11. és 20. napot is beleértve) adjuk meg a helyes merési értékek darabszámát, azt sprintf-el be egy szöveges változóba, az alábbi módon (az adat helyességét az ábrán ellenőrizhetjük):

`“Helyes mérési értékek darabszáma (első szenzor, esti mérés, 11-20. napokra): 7”`

elsoSzenzorHelyesMereseiSzovegben

4. a déli merések közül (középső oszlopsorozat) a 2., 3. és 4. szenzorokhoz adjuk meg külön-külön az átlagértékét (csak a helyes merési értékeket felhasználva!). (Segítségképpen egy lehetséges megoldásmenet: ki kell vágnunk a nagy mátrixból a megfelelő részt, azt megszabadítani a szinguláris dimenzióitól. Ebből egy logikai indexmátrixos összefüggésben soronkénti (!) szummával kivehetjük a helyes merések darabszámát; valamint sima soronkénti (!) szummázásával megtudhatjuk szenzoronként a merések összegét. A kettő hányadosa (jól felírva) egy háromelemű vektor, amit kapni szerettünk volna.)
5. Az eredményt az alábbi módon tároljuk el egy szöveges változóba:

`“A második szenzor déli átlaga: +0991.880, a harmadiknak: +0990.550 és a negyediknek: +1005.696”`

szenzorokDeliMeresenekAtlagaSzovegben

A függvény futása során semmit ne írjon ki a konzolra.

(`rng(1)`; `rand` (kipróbáláshoz), `squeeze` és kettőspont operátor, `find`, `plot`, LineSpec, `xlim`, `ylim`, `title`, `set`, `gca`, `xlabel`, `ylabel`, logikai indexelés, `sum`, `sprintf`)

További gyakorlófeladatok:

3.3 Készíts egy szkriptet, ami az alábbi lépéseket tartalmazza:

- létrehozza a `meres = 10 + 25.*rand(5, 3)` mátrixot, amely 5 db hőmérséklet szenzor mérési eredményeit tartalmazza ºC-ban (szenzoronként 3 mérés),
- mivel a szenzorokról tudjuk, hogy 15 ºC és 30 ºC közötti tartományban működnek jól, ezért az ezen a tartományon kívül mért értékeket hibásnak tekinthetjük: a szkript nullázza ki az értéküket,
- adja meg a szkript minden egyes szenzorra, hogy a három mérésből hány volt hibás,
- adja meg azoknak a szenzoroknak a sorszámát, amelyek legalább kétszer jó értéket mérték,
- kiírja a szenzoronkénti hibaszámot és a jó szenzorok sorszámát beépített függvényekkel.

(logikai indexelés, `sum`, `find`, `fprintf`, `mat2str`)

3.4 Készíts egy szkriptet a következők szerint:

- létrehoz egy 3-dimenziós véletlen mátrixot, dimenziónként 5, 4 és 2 elemmel,
- két különálló változóba kimenti az eredeti mátrixnak a 3. dimenzió mentén található két részmátrixát (szemléletesen: mélységében kettévágjuk a téglatestet; ne felejtsd el az egyke dimenziók eltávolítását, hogy két darab valódi 2d-s mátrixod legyen),
- kiszámítja az első részmátrixban az **oszlopok minimumának összegét** (beépített függvényekkel),
- kiszámítja a második részmátrixban a **sorok maximumának szorzatát** (beépített függvényekkel),
- kiírja ezt a két adatot némi kísérőszöveggel, egyetlen utasítással, 2-2 tizedesjegy pontosan.

(`rand`, `squeeze` és kettőspont operátor, `min`, `sum`, `max`, `prod`, `fprintf`)

3.5 Készíts egy MATLAB szkriptet **`switch`** vezérlési szerkezet felhasználásával, ami az alábbi módon működik:

- bekér egy számot 1-9-ig a felhasználótól, és ha nem számot kap, vagy nincs a szám az intervallumban, akkor kiírja a következő sort, és újra vár egy számot: *"Nem jo értéket adtal meg, probald ujra!"* (szám bekérése: `input`, egy változó tartalma szám-e: `isnumeric`);
- 0-ra a következő üzenettel lép ki: *"Kilepes, tovabbi szep napot!"*
- 1-re kirajzol háromféle szöfüggvényt (`sin`, `cos`, `tan`):
    - mindet a [$-\pi$, π] zárt intervallumon, 50 adatponttal,
    - az 1-es számú ábra ablakba (direkt ábra címzés: `figure(1)`),
    - egymás alá úgy, hogy az x tengelyek igazítva legyenek,
    - legyen bekapcsolva a rács,
    - legyenek feliratozva a tengelyek, 22-es betűmérettel,
    - legyen az egész ábrának a címe *"Szogfuggvenyek"*, 24-es félkövér betűvel,
    - a felső képen az ábra legyen:
        - piros szaggatott vonallal, és
        - piros teli kör markerrel, amiknek fekete a körvonala;
    - középen:
        - zöld pontozott vonallal, és
        - gyémánt jelölőkkel, amiknek fehér a belseje;

    - alul:
        - kék, 2 pontos vastag vonallal,
        - négyzet jelölőkkel;
- páros számokra kirajzolja az $xsin(x)$ függvényt:
    - a [0.5π, 3.5π] zárt intervallumon, 149 adatponttal,
    - a 2-es számú ábra ablakba,
    - csak zöld színű, pont jelölőkkel, vonal nélkül
    - legyenek feliratozva a tengelyek, 22-es betűmérettel,
    - legyen az egész ábrának a címe *“Szogfuggveny”*, 24-es félkövér betűvel;
- páratlan számokra:
    - létrehozza az `m*m`-es `A` véletlen mátrixot, ahol `m` a bemeneti szám,
    - a mátrixnak kiszámolja az inverzét, és szép alakban kiírja a konzolra, 3 tizedesjegy pontossággal,
    - beteszi az eredeti `A` mátrix főátlóbeli elemeit egy `b` vektorba, ezt kiírja sorvektor formájában a konzolra,
    - kiszámolja az Ax=b egyenletrendszer megoldását bal osztással, ezt kiírja sorvektor formájában a konzolra
- az egyes funkciók után térjünk vissza a szám bekéréséhez, amíg a felhasználó 0-t nem ír be.

4. Lineáris egyenletrendszerek, leképezések

4.1 Készíts egy függvényt, melynek négy bemeneti paramétere (négy háromelemű vektor), és egyetlen kimeneti paramétere (egy háromelemű vektor) van. A feladat az alábbi: Anna, Béla és Cili Münchenbe utaznak a hétvégére vonattal. Amint leszállnak a RailJet-ről, elhatározzák, hogy gyümölcsöt vesznek. Be is térnek az első kisboltba, ahol:

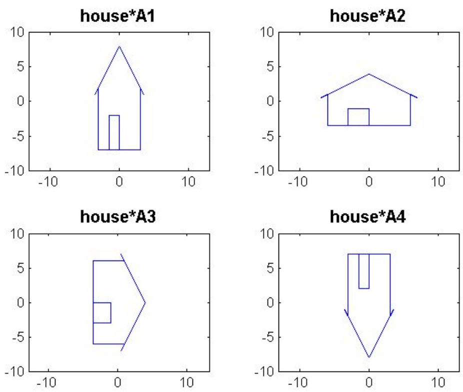
- Anna vásárol a1 almát, a2 banánt és a3 narancsot, összesen d1 EUR-ért;
- Béla b1 almát és b3 narancsot vesz d2 EUR-ért;
- Cili c2 banánt és c3 narancsot vesz d3 EUR-ért.

Számoljuk ki, hogy mennyibe került az egyes gyümölcsök darabja, ez legyen a visszatérési vektor három értéke ([alma ára, banán ára, narancs ára]).

(a=[3, 12, 1];b=[12, 0, 2];c=[0, 2, 3];d=[2.36;5.26;2.77];gyumolcsok_ara=[0.29, 0.05, 0.89])

4.2 Készíts egy függvény egy bemeneti és egy kimeneti paraméterrel, mely egyetlen ábrát állít elő 4 alábrával az alábbi módon:

- a függvény betölti a paraméterként kapott **.mat* fájlt a `load` utasítással, mely archívum az alábbi változókat tartalmazza:
  - `kep` változó: 12 soros, 2 oszlopos mátrix, mely egy házikó 12 pontjának (x,y) koordinátáját jelenti;
  - `A1`, …, `A4` változók: 2x2-es transzformációs mátrixok;
- a függvény végezze el a betöltött `A1`, … `A4` transzformációs mátrixok által reprezentált transzformációkat külön-külön a házikó koordinátáin, és az eredményeket az egyes subplot-okba rajzolja ki, valahogy így:



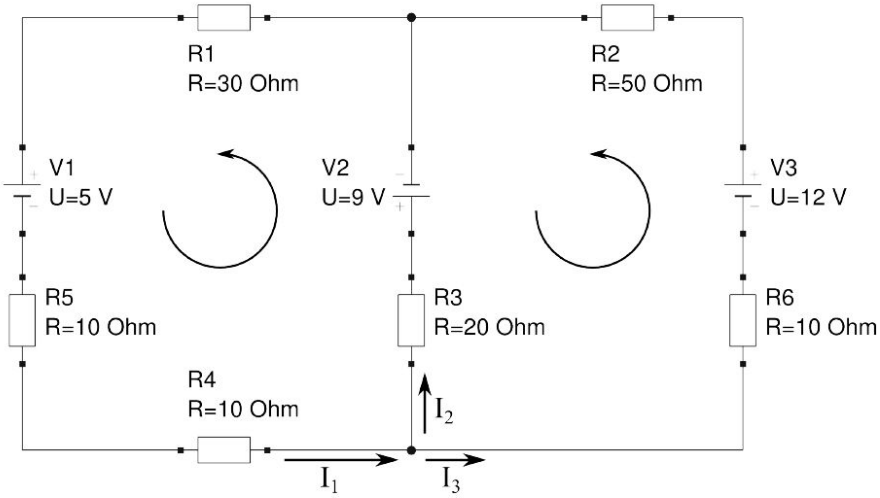
Mit jelentettek ezek a transzformációk? -- egy-egy sorral, kommentként jellemezd a forráskódban a hatásukat. A bemenetre használd a ‘house.mat’ értéket.

4.3 Készíts egy függvényt, melynek 7 bemeneti paramétere van (a - g), és három kimeneti paramétere (x - z) van. A feladat az alábbi egyenletrendszer megoldása:

a*x + b*y = e
a*y + c*x + b*z + d = f
a*z + b*y - d = g

(a = 7; b = 2; c = 1; d = 8; e = -53; f = 832; g = 428; x = -40.4551; y = 115.0930; z = 29.4020)

4.4 Készíts egy függvényt, melynek 9 bemeneti paramétere (ellenállás és feszültség értékek), és egyetlen kimeneti paramétere (egy háromelemű vektor) van. A feladat az alábbi ábrán látható kapcsolásban az áramerősségek előjeles értékének kiszámolása:



Segítség:

- **Ohm-törvény:** $R = \frac{U}{I}$
- **Kirchoff I. törvénye (csomóponti):** áramköri elágazásnál, vagy csomópontnál a csomópontba befolyó áramok összege megegyezik az onnan elfolyó áramok összegével (nincs töltésfelhalmozódás).
- **Kirchoff II. törvénye (huroktörvény):** sorosan kapcsolt áramköri elemek esetén bármely zárt áramhurokban a részfeszültségek előjelhelyes összege zérus.
- megjegyzés: az egyes csomópontokban az áramok irányának, valamint az áramhurokban a hurok irányultságának megválasztása önkényes. (Ha negatív áramokat kapunk eredményül, akkor a valós áramirány az általunk választottal ellentétes az áramkörben.)

Az alsó csomópontra felírható Kirchoff I. törvénye, míg a két áramhurokra Kirchoff II. törvénye.

$$\begin{cases} I_1 = I_2 + I_3 \\ 20I_1 + 20I_2 + 9 + 30I_1 + 5 = 0 \\ 10I_3 - 12 + 50I_3 - 9 - 20I_2 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} I_1 - I_2 - I_3 = 0 \\ 50I_1 + 20I_2 + 0I_3 = -14 \\ 0I_1 - 20I_2 + 60I_3 = 21 \end{cases}$$

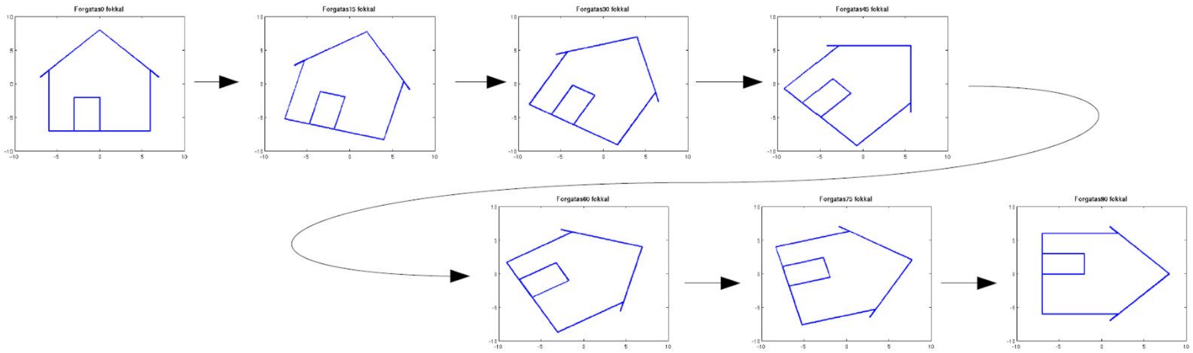
$$\Rightarrow \left[\begin{array}{ccc|c} 1 & -1 & -1 & 0 \\ 50 & 20 & 0 & -14 \\ 0 & -20 & 60 & 21 \end{array} \right]$$

(Az ábráról leolvasható adatokkal: I_1=-134.6mA, I_2=-363.5mA; I_3=228.8mA)

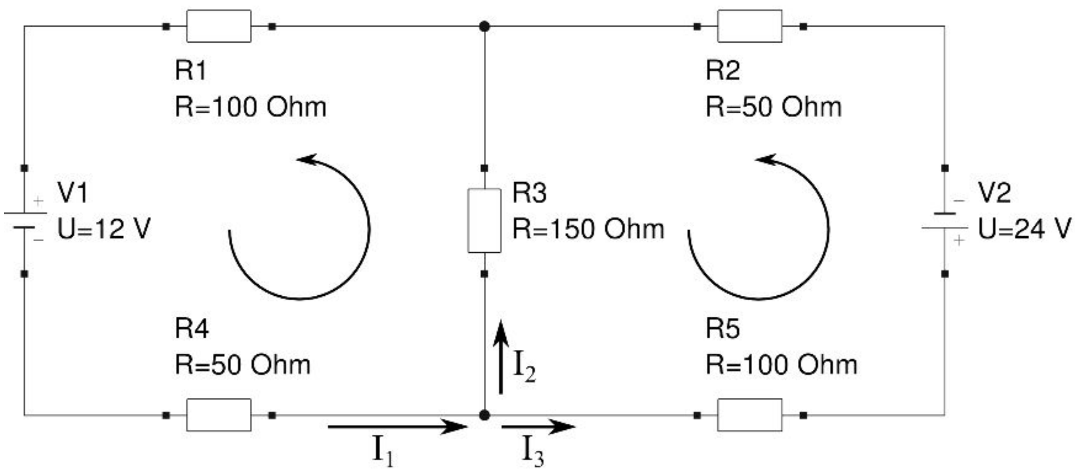
További gyakorlófeladatok:

4.5 Készíts egy függvényt bemeneti és kimeneti paraméterek nélkül, mely egyetlen ábrát állít elő, de annak tartalmát késleltetéssel frissíti az alábbi módon:

- a függvény betölti a *house.mat* fájlt a `load` utasítással, mely archivum az alábbi változókat tartalmazza:
    - `house` változó: 12 soros, 2 oszlopos mátrix, mely egy házikó 12 pontjának (x,y) koordinátáját jelenti;
    - `A1`, …, `A4` változók: 2x2-es transzformációs mátrixok;
- a függvény egy **`for`** ciklus segítségével rajzolja ki a `house` változó 15-fokokkal elforgatott változatait 90 fokig olyan módon, hogy minden egyes kirajzolás után a **`pause`** paranccsal gombnyomásig késleltetjük a program futását. (Tehát egyetlen `figure` létezik végig, és ennek tartalmát frissítjük egy cikluson belül.) A frissülő kimenet időbeni lefutása az alábbi pillanatképekhez legyen hasonló:



4.6 Készíts egy függvényt, melynek nincs bemeneti paramétere, és egyetlen kimeneti paramétere (egy háromelemű vektor) van. A feladat az alábbi ábrán látható kapcsolásban az áramerősségek előjeles értékének kiszámolása:

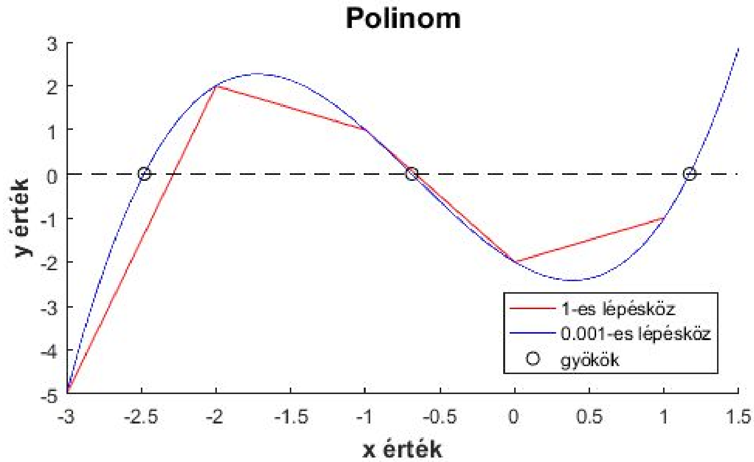


(I_1=-106.7mA, I_2=26.7mA; I_3=-133.3mA)

5. Polinomok, deriválás, integrálás

5.1 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg polinom-műveletek használatával:

- ábrázolja a `P(x) = ax³ + bx² + cx + d` polinomot a [e, f] zárt intervallumon, 1-es és 0.001-es lépésközzel;
- a görbék színe legyen: piros (1) és kék (0.001);
- ugyanezen az ábrán jelölje be a polinom gyökeit fekete körökkel; és a 0-szintet egy fekete szaggatott vonallal.

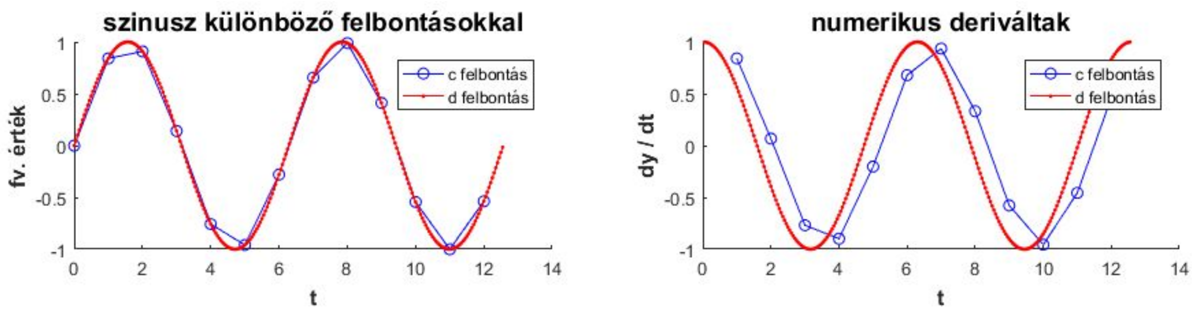


(`a=1; b=2; c=-2; d=-2; e=-3; f=1.5; függvények: polyval, roots, plot, xlim, ylim, xlabel, ylabel, title, legend`)

5.2 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- egy ábrát készít két alábrával;
- a bal alábrában ábrázol egy szinusz jelet az `[a, b]` zárt intervallumon, `c` felbontással, az egyes adatpontokat összekötött kék körökkel jelölve;
- ugyanerre az alábrára kirajzolja ugyanebben a tartományban ugyanezt a jelet, csak `d` felbontással, összekötött piros pontokkal;
- az alábrát megfelelően feliratozza-címkézi;
- kiszámítja a görbék numerikus deriváltját, és ezeket rajzolja a jobb alábrára;
- a jobb alábrát is megfelelően feliratozza-címkézi.

Valahogy így kellene kinéznie:

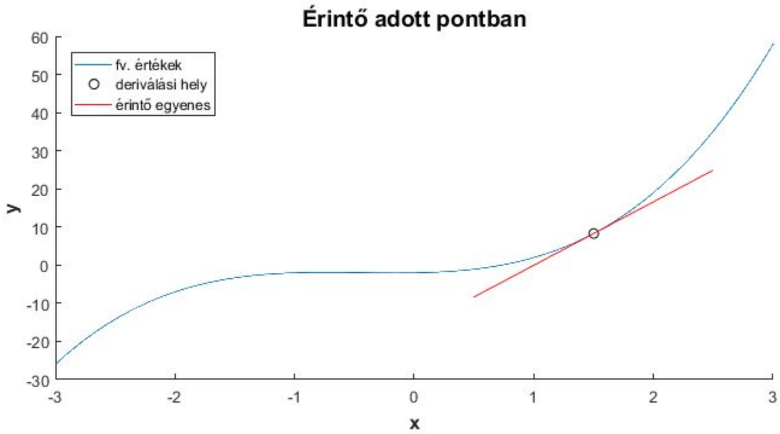


(`a=0; b=4*pi; c=1; d=0.05; függvények: sin, diff, subplot, plot, xlim, ylim, xlabel, ylabel, title, legend`)

5.3 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- legyen adott `P(x) = ax³ + bx² + cx + d` polinom;
- rajzolja ki `P(x)` értékeit 0.001 lépésközzel a `[-e, e]` zárt intervallumon (az értékeket a megfelelő beépített polinom-függvénnyel számolja ki!);
- számítsa ki a függvény deriváltját a `t0` helyen;
- a megadott `t0` helyre rajzolja be a megfelelő érintő egyenest is;
- az ábrát megfelelően feliratozza-címkézze.

Valahogy így kéne kinéznie:



(a=1.5; b=2; c=0.5; d=-2; e=3; t0=1.5; derivált=16.6250; érintő egyenes készítése: a deriválási helyen áthaladó egyenes, melynek meredeksége pont a derivált értéke, kezdő és végpontja tetszőleges az x tengelyen)

5.4 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- kiszámítja a szinusz görbe és az x tengely közötti területet a `[0, a]` zárt intervallumon, (ahol kell) `b` felbontás mellett, az alábbi módszerekkel:
  - egyszerű összeadással,
  - trapézszabály segítségével,
  - anonim függvény és a beépített `integral` függvény felhasználásával;
- a három eredményt szépen formázott módon írja be egy kimeneti stringbe, valahogy így:

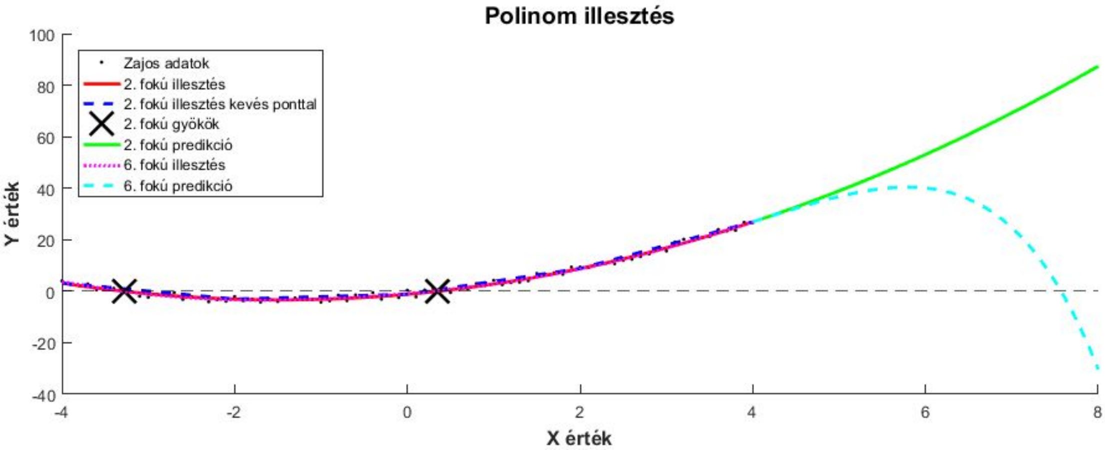
```
Integrálási eredmények:
    Összeadással: 1.547
    Trapézszabállyal: 1.504
    Függvénnyel: 1.505
```

`(a=0.7*pi; b=0.1;)`

5.5 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- Bemenet a `meresiPozicio` és `mertErtekek` melyek a polinom.mat fájlban adottak;
- illesszen az adatokra másodfokú polinomot;
- értékelje ki az illesztett polinomot
  - az eredeti (ÉT-beli) mérési pozíciók felett, majd
  - az első és utolsó mérési pozíciót is beleértve 5 ekvidisztáns pont felett is;
- számolja ki az illetsztett polinom gyökeit;

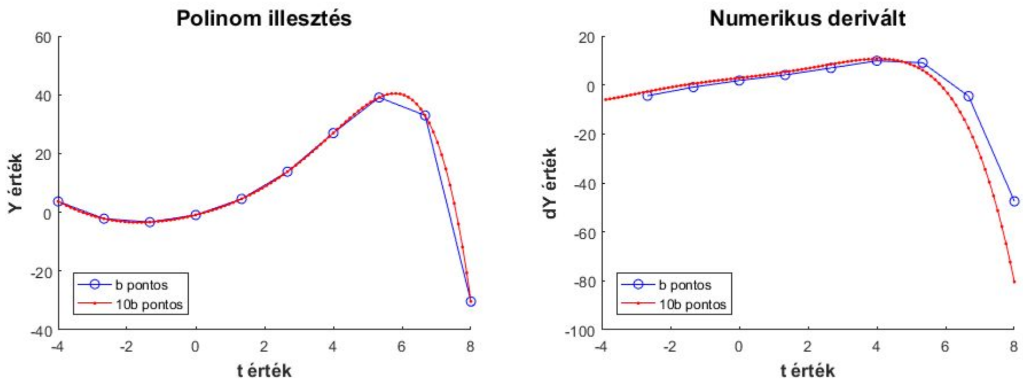
- végezzen predikciót a mérési adatokra az `[a, 2*a]` tartományon az illesztett polinom kiértékelésével (a lépésköz `b` legyen);
- illesszen egy hatodfokú polinomot is az eredeti adatokra, és ezzel is végezze el a predikciót, ugyanazon a tartományon;
- rajzolja ki az eredeti adatokat és a számolt értékeket-adatsorokat egy közös, megfelelően feliratozott ábrára.



(load('polinom.mat'); a=4; b=0.1;
másodfokú együtthatók, gyökök és hatodfokú együtthatók rendre:
[1.01281054033819 2.96782344341433 -1.1542631234446]
[-3.27795955385171;0.347674643484029]
[-0.00064366100830044 -0.00115075082186434 0.022587334219388
0.0167599729104976 0.818964233194576 2.93550336832322
-0.90399493037039])

5.6 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- Két adatsort kap: `meresiPozicio` és `mertErtekek` (ezek a polinom.mat fájlban adottak);
- illesszen az adatokra hatodfokú polinomot, és azt értékelje ki a `[-a, 2a]` intervallumon először `b`, majd `10b` mintaponttal;
- ábrázolja az eredményt egy ábra bal részábrájaként (`subplot`), a `b` pontosat kék vonallal és kör markerrel, a `10b` pontosat piros vonallal és pont markerrel;
- számolja ki a görbék numerikus deriváltjait is; ezeket az ábra jobb részábrájába jelenítse meg;
- legyenek tengelyfeliratok, ábracím, adatsor-magyarázat.



(bemenetiFajl='polinom.mat'; a=4; b=10;)

5.7 Készíts egy függvényt, ami az alábbi feladatokat valósítja meg:

- beolvas egy adatsort, ami a `meresiPozicio` és `mertErtekek` szerint adott;
- illesszen az adatokra hatodfokú polinomot, és azt értékelje ki azon a 10 mintapontos zárt intervallumon, melynek kezdőpontja a polinom 6. gyöke és zárópontja a polinom 1. gyökének fele;
- határozza meg a felvett görbe és az x tengely közötti előjeles területet az `integralasiModszer` változóban adott érték szerint:
  - `'osszeadas'` - egyszerű összeadással,
  - `'trapez'` - trapézszabály segítségével,
  - `'integral'` - anonim függvény és a beépített `integral` függvény felhasználásával;

(bemenetiFajl='plolinom.mat'; az intervallum: [0.2851, 3.7847]; Numerikus integrálás eredménye: Összeadással: 40.830, Trapézszabállyal: 35.429, Függvénnyel: 35.314; a bemenet vizsgálatához switch-case szerkezetet használj)

6. Differenciálegyenletek

6.1 Készíts egy függvényt 2 bemeneti paraméterrel (kezdeti értékek (`y_0 = [y1_0, y2_0]`) és időintervallum (`t=[t_0, t_max]`)), ami 1 ábrát generál, és ad vissza. A feladat az alábbi elsőrendű, kétváltozós DE megoldása:

$$\left|\begin{array}{l} \dfrac{dy_1}{dt} = \left(1 - \dfrac{y_2}{\mu_2}\right) y_1 \\ \dfrac{dy_2}{dt} = -\left(1 - \dfrac{y_1}{\mu_1}\right) y_2 \end{array}\right|$$

(Lotka-Volterra modell, amely egy adott területen a ragadozó-zsákmány egyedszám-viszonyt írja le. https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations)

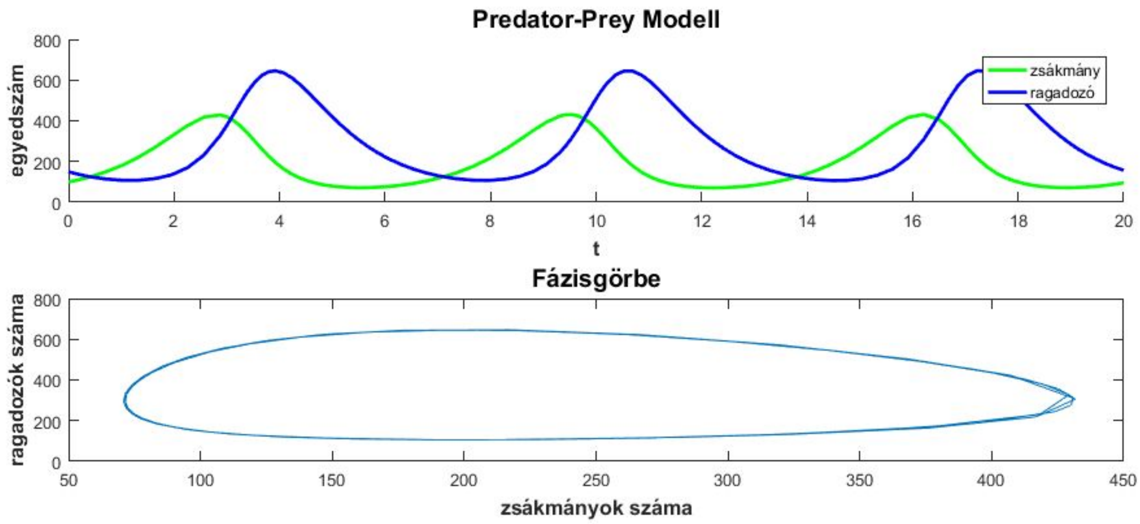
y1: zsákmány egyedszám, y2: ragadozó egyedszám;

μ1: zsákmányok környezeti eltartóképessége, **μ**2: ragadozók környezeti eltartóképessége;

Legyen most: **μ**1 = 200, **μ**2 = 300;

A differenciálegyenlet-rendszert a megoldás függvényen belül, anonim függvényként definiáld.

A függvény az alábbihoz hasonló módon rajzolja ki a felső alábrán az egyedszámok időfüggő értékét, míg az alsó alábrán az egyedszámokon felvett síkban a fázisportrét:



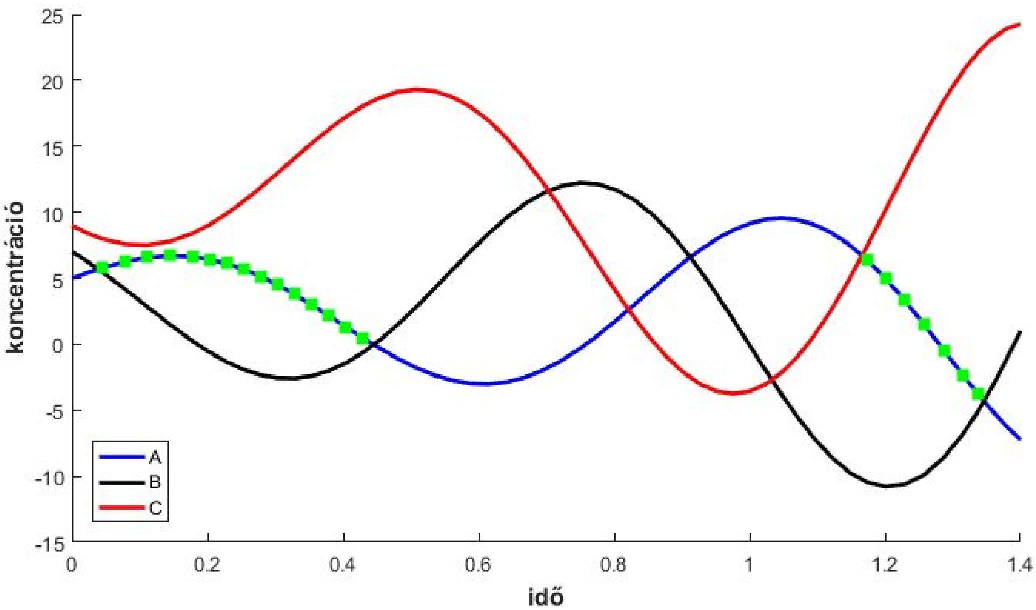
(`y_0 = [100, 150]; t = [0, 20];`)

6.2 Készíts egy függvényt, melynek 2 bemeneti paramétere (kezdeti értékek (`kezdeti=[A_0, B_0, C_0]`) és itdőintervallum (`t=[t_0, t_max]`)) és 1 visszatérési értéke van (ábra).

Egy kémiai reakció során három anyagot vegyítünk (A, B, C), melyek koncentrációváltozását az alábbi differenciálegyenlet rendszer írja le:

$$\frac{dA}{dt} = 1.2A + 4.1B - 1.7C$$
$$\frac{dB}{dt} = -8A - 1.4B + 2.1C$$
$$\frac{dC}{dt} = 2.1A - 7.2B + 1.3C$$

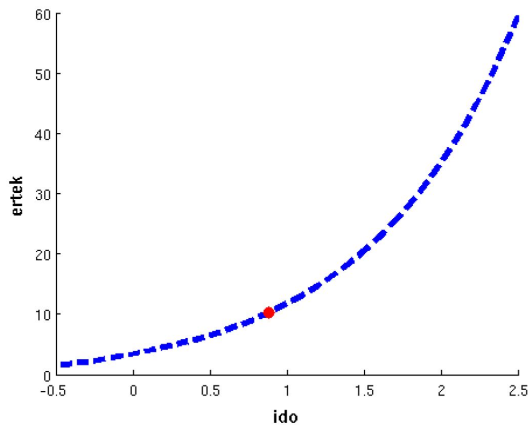
Oldd meg ezt a differenciálegyenlet rendszert a bemeneten kapott időintervallumon, a szintén bemenetként kapott kezdeti értékek mellett. A differenciálegyenlet-rendszert a megoldás függvényen belül, anonim függvényként definiáld.
Határozd meg azokat az indextartományokat *logikai indexelés* segítségével, amikor az első anyag koncentrációja nagyobb, mint a másodiké, de kisebb, mint a harmadiké. Ábrázold a megoldásfüggvényeket (tengelyfeliratok, betűméret, vonalszínek és típusok, legend), valamint jelöld meg az A-anyag feltételnek eleget tevő értékeit külön, zöld kocka markerekkel, összekötés nélkül:



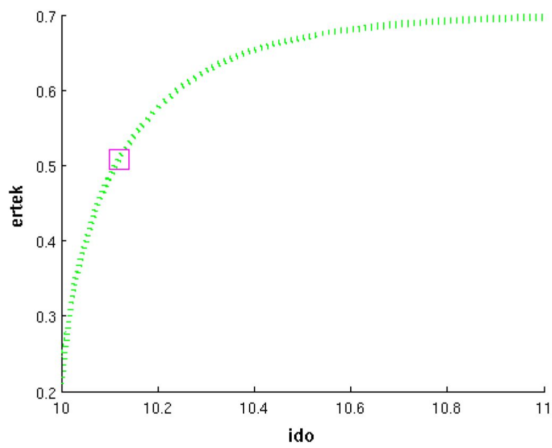
`(t= [0, 1.4]; kezdeti=[5, 7, 9];)`

További gyakorló feladatok:

6.3 Készíts egy függvényt, melynek 0 bemeneti paramétere és 1 visszatérési értéke van; 1 ábrát generál, és a konzolra nem ír ki semmit. A feladat az alábbi differenciálegyenlet megoldása a [-0.5, 2.5] időintervallumon, y0=1.5 kezdeti érték mellett: $x' = 1.03x + 1.3$
Állapítsd meg a `find` függvény használatával, hogy a megoldásfüggvény értéke mikor lépi át először a 10-es értéket, ez legyen a függvényed visszatérési értéke. *(nagyjából 0.9 körül)*
Ábrázold a megoldásfüggvényt (tengelyfeliratok, betűméret, vonalszínek és típusok), és jelöld a megtalált adatpontot, valahogy így:



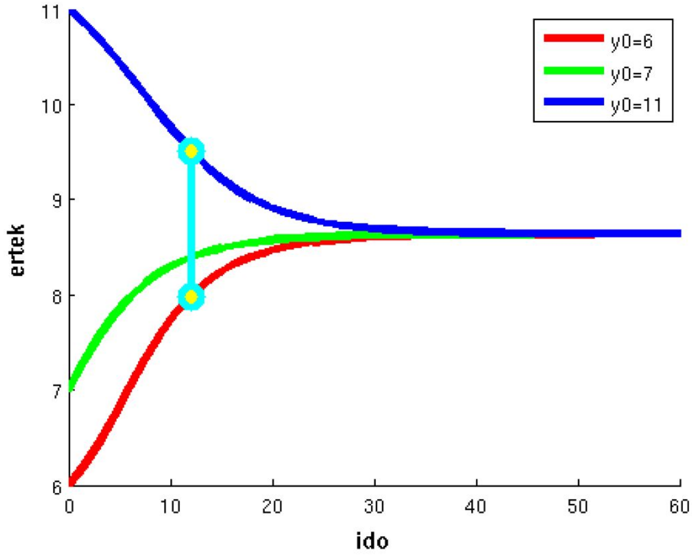
6.4 Készíts egy függvényt, melynek 0 bemeneti paramétere és 1 visszatérési értéke van; 1 ábrát generál, és a konzolra nem ír ki semmit. A feladat az alábbi differenciálegyenlet megoldása a [10, 11] időintervallumon, y0=0.21 kezdeti érték mellett: $x' = \frac{2.1}{x} - 3$
Állapítsd meg a `find` függvény használatával, hogy a megoldásfüggvény értéke mikor lépi át először a 0.5 értéket, ez legyen a függvényed visszatérési értéke. *(nagyjából 10.1 körül)*
Ábrázold a megoldásfüggvényt (tengelyfeliratok, betűméret, vonalszínek és típusok), és jelöld meg a megtalált adatpontot, valahogy így:



6.5 Készíts egy függvényt, melynek 0 bemeneti paramétere és 1 visszatérési értéke van; 1 ábrát generál, és a konzolra nem ír ki semmit. A feladat az alábbi differenciálegyenlet megoldása a [0, 60] időintervallumon, y0=6, y0=7 és y0=11 kezdeti értékek mellett:
$x' = \frac{sin(x)+cos(x)}{x}$
Állapítsd meg a `find` függvény használatával, hogy a harmadik és az első megoldásfüggvények értékének különbsége mikor lesz először kisebb 1.7-nél, ez legyen a függvényed visszatérési értéke

(egy elég kerek szám, a nagy integrálási lépések miatt...)

Ábrázold a megoldásfüggvényeket (tengelyfeliratok, betűméret, vonalszínek és típusok, legend), és kösd össze a megtalált adatpontokat, valahogy így:



6.5 Készíts egy függvényt, melynek 0 bemeneti paramétere és 0 visszatérési értéke van; 3 ábrát generál, és a konzolra nem ír ki semmit.
Vizsgáljuk meg a Van der Pol oszcillátor viselkedését különböző paraméterek mellett. Az egyenlet alalkja: $\ddot{x} + \mu(x^2 - 1)\dot{x} + x = 0$

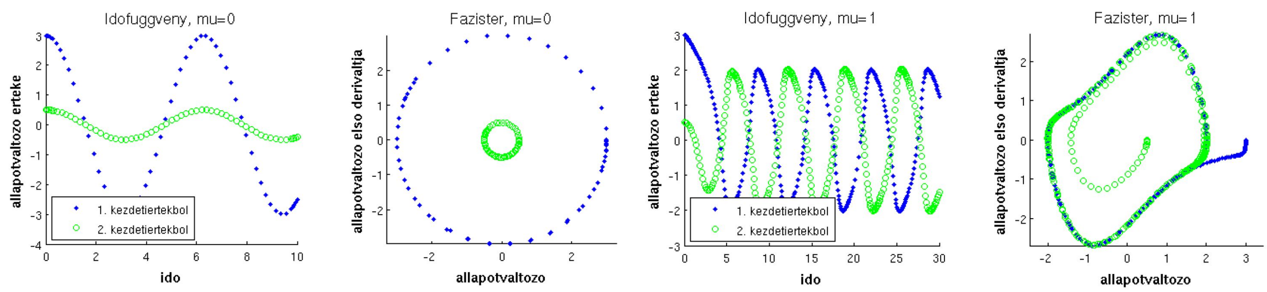
átírva elsőrendű rendszerre az alábbi eljárást követve: $y_1 = x \text{ es } y_2 = \dot{x}$
Mellyel:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} y_2 \\ \mu(1 - y_1^2)y_2 - y_1 \end{pmatrix}$$

Vizsgáljuk meg a rendszer viselkedését különböző μ paraméterek esetén. Minden esetben több kezdetiértékkel is próbálkozzunk, így szemléletesebb lesz a fázisképen a periodikus pálya vonzó hatása:

- ‘A’ eset: μ = 0, időintervallum: [0, 10], kezdeti értékek: [3; 0] és [0.5; 0]
- ‘B’ eset μ = 1, időintervallum: [0, 30], kezdeti értékek: [3; 0] és [0.5; 0]
- ‘C’ eset μ = 5, időintervallum: [0, 50], kezdeti értékek: [3; 0] és [0.5, 0]

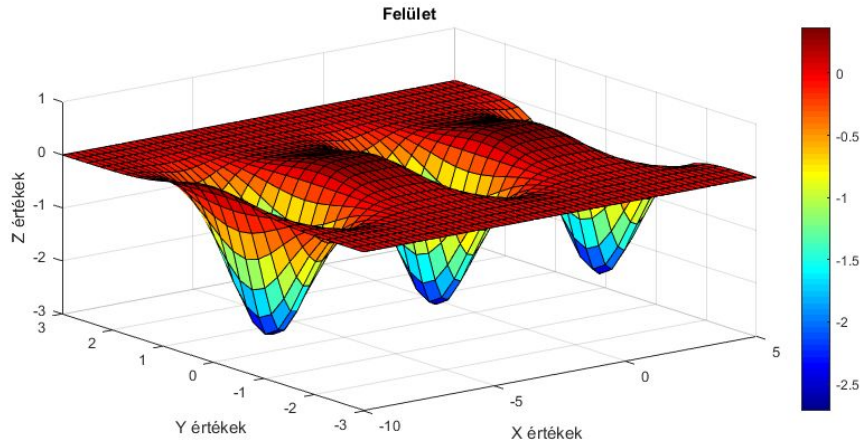
Az alábbihoz hasonló ábrákat generáljon a függvény:



7. 3D ábrázolás

7.1 Ábrázold az alábbi képlettel megadott felületet az `[x_min,x_max]×[y_min,y_max]` intervallumon `0.25`-ös felbontással. Az ábrát feliratozd megfelelően (cím és tengelyfeliratok)

$$z = sin(x) \cdot e^{-sin(x)-y^2}$$

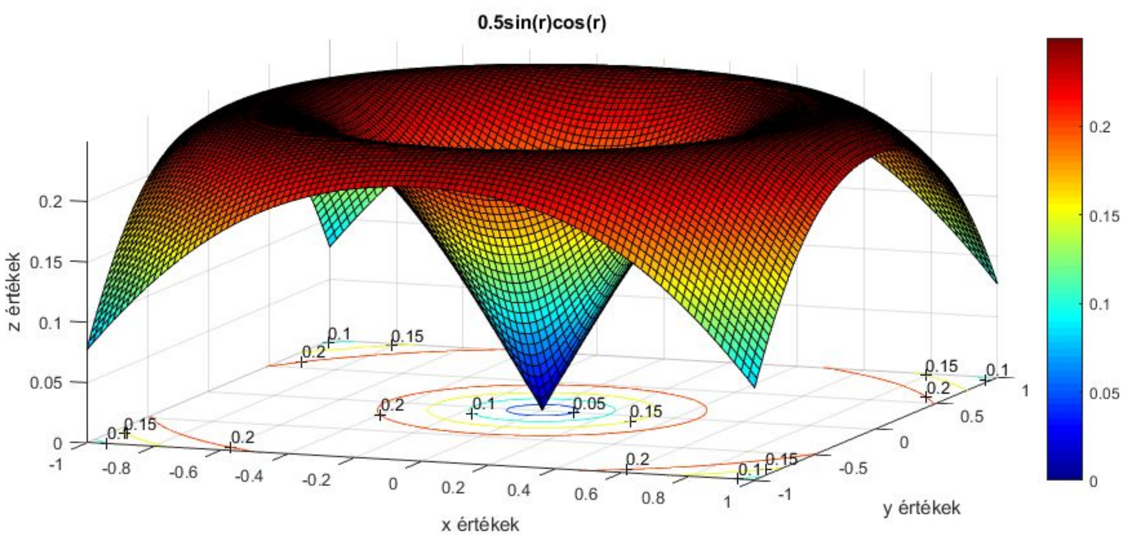


`(x_min=-10; x_max= 5;y_min=-3;y_max=3)`

7.2 Ábrázold az alábbi képlettel megadott felületet a szintvonalaival együtt az `[x_min,x_max]×[y_min,y_max]` tartományon, `0.02`-es lépésközzel, körkörösen minden irányban! (tehát mintha az XY síkban ábrázolt függvényt megforgatnád a Z tengely körül) A szintvonalakra írd rá az adott vonal értékét is! Az ábrát feliratozd megfelelően (cím, tengelyfeliratok) és a nézőpontot állítsd `[Az;El]` értékekre!

$$z = 0.5 \cdot sin(r) \cdot cos(r)$$

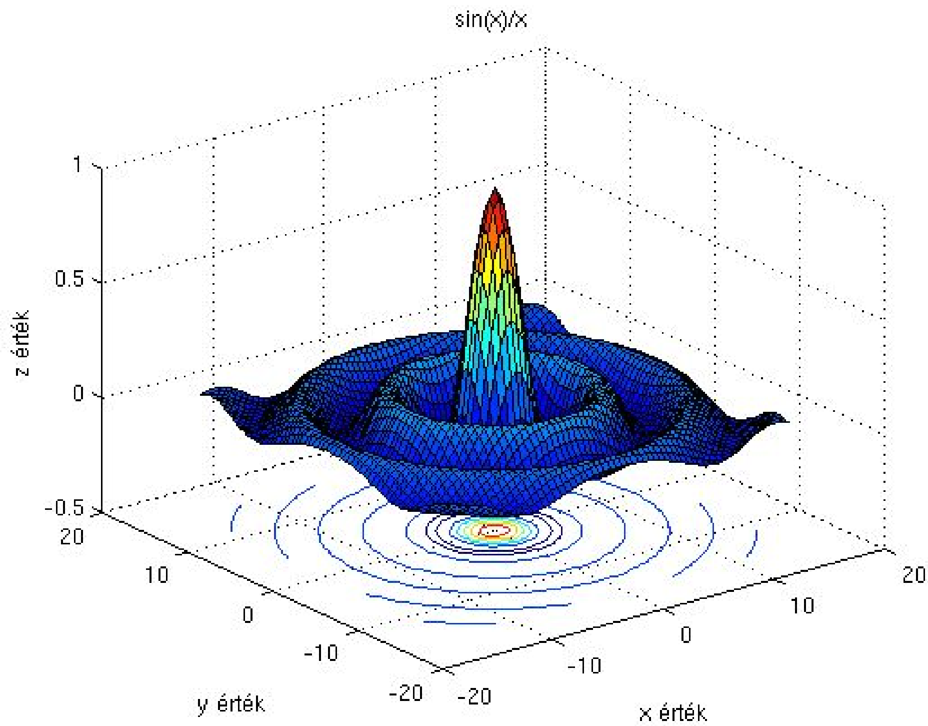
Az eredmény az alábbi ábrához hasonló legyen:



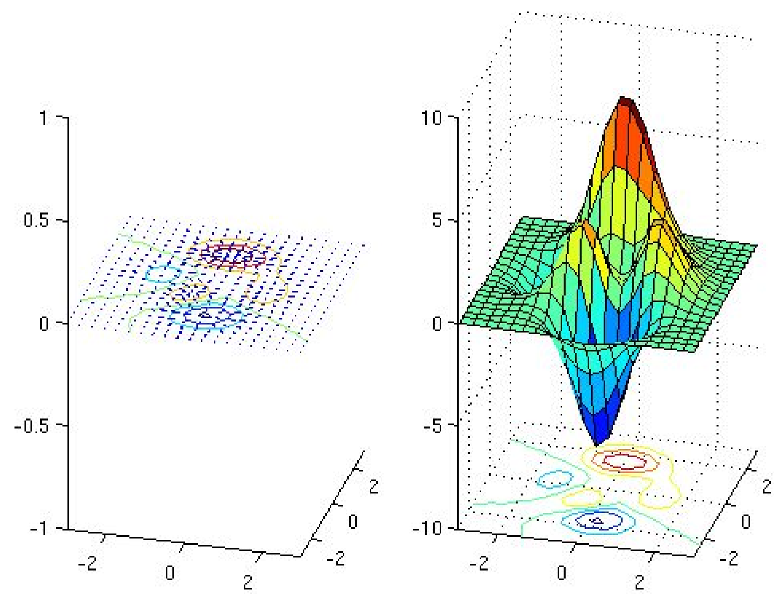
`(x_min=-1; x_max= 1;y_min=-1;y_max=1;Az= 20;El = 20;)`

További gyakorlófeladatok:

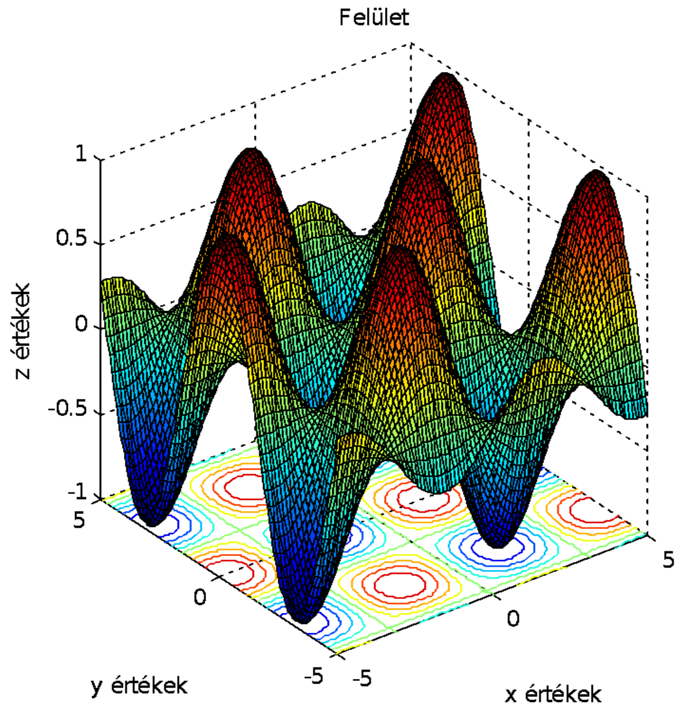
7.3 Ábrázoljuk a sin(x)/x függvényt a szintvonalaival együtt a [-15, 15] × [-15, 15] intervallumon 0.5-ös felbontással, körkörösen minden irányban!



7.4 Számítsuk ki a **peaks(20)** parancs által megadott felület gradiens mezőjét! Ábrázoljuk a felületet és a gradiens mezőt egymás mellett elhelyezkedő subplotokon, mindegyiken kirajzolva a szintvonalakat is!

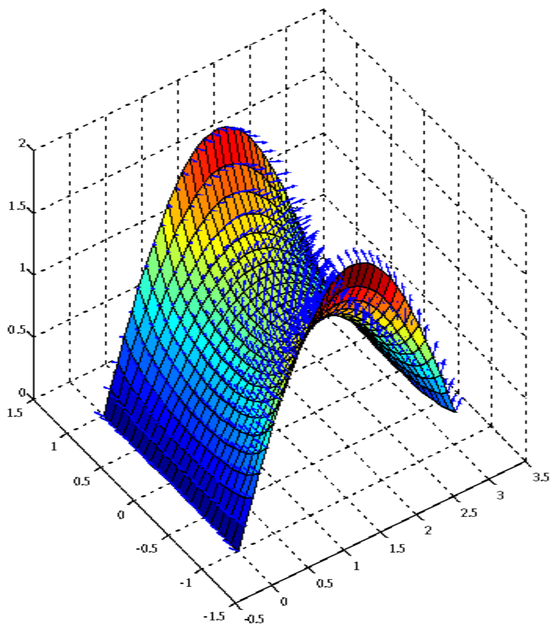


7.5 Ábrázold a z = sin(x)cos(y) képlettel megadott felületet a szintvonalaival együtt a [-5; 5]x[-5; 5] tartományon, 0.1-es lépésközzel! Az ábrát feliratozd megfelelően (cím, tengelyfeliratok)!



7.6 A quiver3 és a surfnorm parancsok Help bejegyzései alapján ábrázold az alábbi felületet és az egyes rácspontokra eső normálvektorokat a [0;3]x[-1;1] tartományon, 0.1-es lépésközzel!

$$z = \frac{\sin(x)}{\cos(y)}$$



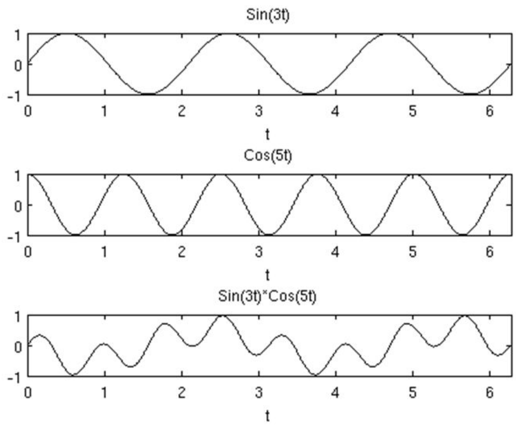
8. Cellatömbök, struktúrák és fájlműveletek

8.1 Készíts egy egy bemenetű függvényt, ami feltölt egy cellatömböt az alábbiak szerint:

- az első eleme egy sorvektor legyen (t), ami 0-tól 2π-ig tartalmaz `db` értéket (bemenet) egyenletesen elosztva
- a második sorban a 'Sin(3t)', 'Cos(5t)' és 'Sin(3t)*Cos(5t)' stringek álljanak
- a harmadik sorban a stringeknek megfelelő numerikus értékek legyenek eltárolva

A függvény kimenete az elkészített cellatömb legyen, valamint az ábra.

Ugyancsak a függvényben, ciklus használatával rajzoljuk ki a fent kiszámolt értékeket egy ábrán, 3 subplotra, fekete színnel! A subplotok címeit is a cellatömbből írjuk ki! Az alábbihoz hasonló ábrát készítsen a függvény:



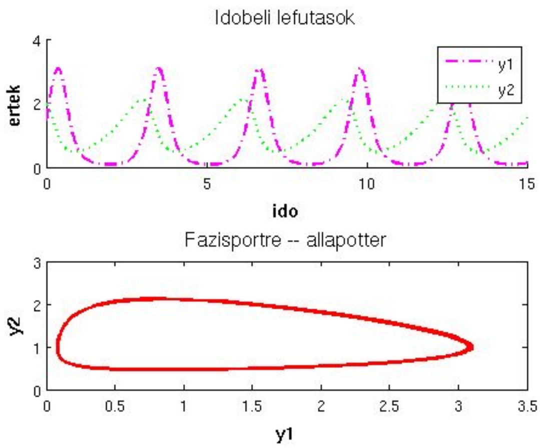
8.2 Készítsd el a 8.1-es feladatot struktúrák használatával. A struktúra mezőit nevezd `x`, `y`, és `nev`-nek.

8.3 Egy elsőrendű, kétváltozós differenciálegyenlet megoldása. A feladatot egy függvényben oldd meg, melynek bemeneti paramétere egy fájlnév (`bemenetiFajl`), valamint három visszatérési értéke van (lásd utolsó részfeladat).

- Töltsd be a `bemenetiFajl` nevű szöveges fájlt, és olvasd be a benne található négy (fixpontos) számot: az első kettő a diffegyenlet-megoldás kezdő és záróidőpontját tartalmazza, a harmadik és a negyedik pedig a két kezdetiértéket.
- Maga a diffegyenlet az alábbi alakú:

$$\dot{y_1} = 5ln(y_2)y_1$$
$$\dot{y_2} = (1 - 1.2y_1)y_2$$

- Ábrázold a megoldások időfüggvényét és fázisgörbéjét, az alábbihoz hasonló módon (vonalszínek, típusok, feliratok, betűméretek, stb.)
- Az időértékeket, és a függvény értékeket (az y mátrixot) írd ki a “`gyak8_f83_kimenet.bin`” nevű bináris fájlba, majd zárd be a fájlt.

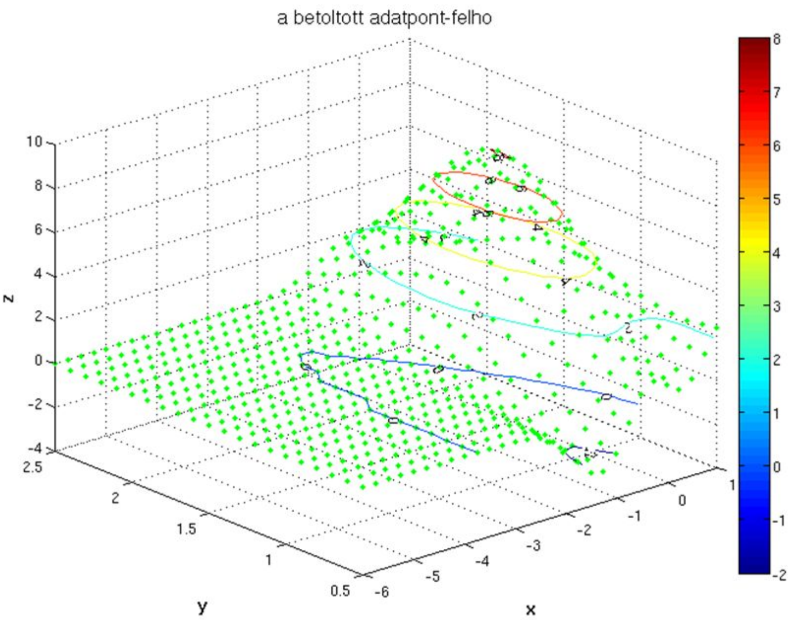


- Ez a két adat legyen a függvényed visszatérési értéke is, az ábra mellett.

(teszteléshez: `bemenetiFajl`='`83.text`';)

8.4 Készíts egy függvényt, melynek bemeneti paramétere egy fájlnév (`bemenetiFajl`), és kimeneti paramétere az alábbi ábra, valamint a megadott struktúra. A függvény feladata az alábbi 3D plot elkészítése a beolvasott fájlban adott adatok segítségével.

- A fájlban az első 6 lebegőpontos szám megmondja a szükséges meshgridhez az adatokat (sorrend: xmin, xlépés, xmax, ymin, ylépés, ymax)
- Az összes többi lebegőpontos szám egy felület pontjait adja vissza (javaslat: 21 soros mátrixba töltsük be)
- Az adatokat és a tengelyfeliratokat egy struktúrában tárold le, és ezt használd az ábrázolásnál
- Az ábrán szerepeljen: 3D pontfelhő + feliratos 3d-kontúrvonalak színskálával, tengelyfeliratok, cím megfelelő betűméretekkel, látószög: kb [-41, 28]



- Az elkészített struktúrát mentsd el a “`gyak8_f84_kimenet.mat`” fájlba, MATLAB bináris formában, valamint add vissza a kimeneten is.

(teszteléshez: `bemenetiFajl`='`84.text`';)

9. Táblázatok, fájlműveletek, képmentés

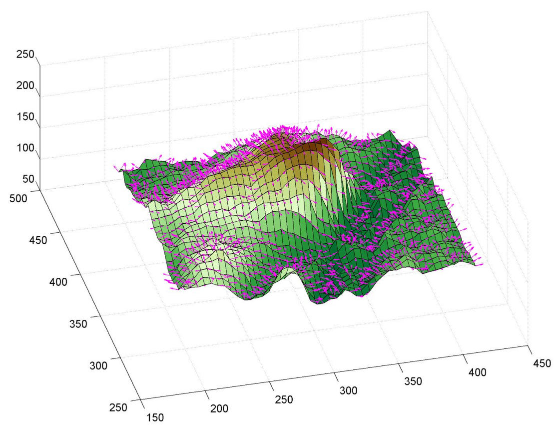
9.1 Készíts egy bemenet nélküli függvényt, ami feltölt egy táblázatot:

- az első oszlopban az idő vektor értékei legyenek tárolva (`t`), a [0,pi] intervallumon, 0,001 lépésközzel
- a második oszlopban 5*Sin(3*e^(t)) értékei legyenek
- a harmadik oszlopban a függvény értékei alapján adott szóbeli osztályzat álljon, a kerekítés szabályainak figyelembe vételével (1 - elégtelen, 2 - elégséges, 3 - közepes, 4 - jó, 5 - jeles)
- Az oszlopok neve rendre: `ido, eredmeny, ertekeles.`

9.2 Készíts egy bemenet és kimenet nélküli függvényt, ami beolvassa a `country_data.xls` fájlt egy táblázatba, majd a beolvasott adatokat a beszélt nyelvek száma alapján rendezi sorba (**`sortrows`**). A függvény írja ki a terület, az életkor medián, a népesség, valamint a nyelvek számának minimumát, maximumát, és mediánját a tíz legtöbb nyelvet beszélő ország esetén. (indexelés, **`summary`**) A függvény az egész átrendezett táblázatot | karakterrel elválasztott szöveges fájlba mentse ki, `country_data_reordered.csv` néven.

9.3 Készíts egy bemenet és kimenet nélküli függvényt, ami a kiadott `geodata.csv` fájl alapján egy ábrát generál a következők szerint:

- egy táblázatba olvassa be az adatokat a sorcímekkel együtt
- az elválasztó karakter: tabulátor
- a beolvasott adatokat tegye át az X, Y és Z mátrixokba oszloponként úgy, hogy mindegyik mátrix 33x45-ös legyen
- rajzolja ki a három mátrix alapján a tárolt domborzat felületi képét (**`surf`**), majd színezze be térképszerű színekkel (**`demcmap(`**`Z`**`)`** - ha nincs meg ez a függvény, akkor: **`load cmp; colormap(cmp)`**)
- jelenítse meg a megadott felület normálvektorait magenta színnel (**`surfnorm, quiver3)`**
- forgassa el a képet a [-15,60] szögekre
- mentse el a jobboldalihoz hasonló ábrát `surf2.png` néven



9.4 Készíts egy bemenet nélküli függvényt, ami ábrákat ment. Válassz ki szabadon az eddig elkészített házi feladatok közül legalább 5 ábrát, és mentsd ki őket LaTeX, és Word dokumentumokhoz javasolt formátumban (természetesen az ábrákat generáló kód is kerüljön be a függvénybe). Az ábrák nevei legyenek f01-f05-ig. Figyelj a betűméretre, valamint a vonalvastagságra is!