Matlab megoldások

Nagy Dániel Zoltán

2021. május 29.

Tartalomjegyzék

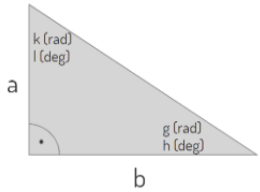
1. Gyakorlat

1.1. Házifeladatok

1.1.1. Számológép

Készíts egy **függvény**t **2 bemenet**i [a-b] és **11 kimenet**i [c-h; k-p] paraméterrel, az alábbi összefüggések alapján:

1. $c = \sqrt[a]{b}$
2. $d = a^b$
3. $e = \log_a b$
4. $f = 5.21^{((log_b \sqrt[3]{\pi})^a)}$
5. Egy derékszögű háromszög két szára a és b. Számítsd ki a befogókkal szemközti szögeket radiánban és fokban is --- ez a négy érték kerüljön a g, h, l, k változókba az alábbi ábrának megfelelő sorrendben.



6. Számítsd ki a b^a -nál kisebb prímeket egy beépített MATLAB függvénnyel, és tárold el őket az m változóba.
7. Faktoriális számolás: $n = (a * b)!$
8. Készítsd el a $10 * a * b$ érték prímtényezős felbontását (prím-faktorizáció) egy beépített MATLAB függvénnyel, az eredményt tárold el az o változóba.
9. Számold ki a b sugarú kör területét, az eredményt tárold el a p változóba. Használj beépített kulcsszót a π állandó helyén!

Solution:

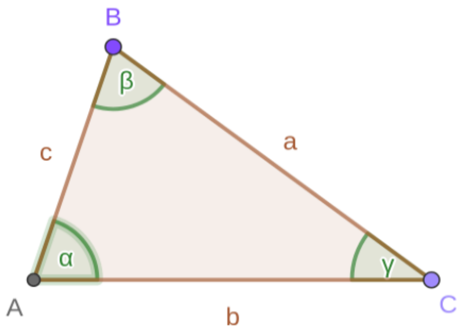
```
function [c,d,e,f,g,h,k,l,m,n,o,p]=assignment1Solution(a,b)
    c = power(b, 1/a);
    d = power(a, b);
    e = log10(b)/log10(a);
    f = power(5.21, power((log10(power(pi, 1/3))/log10(b)), a));
    g = atan(a/b);
    h = atand(a/b);
    k = atan(b/a);
    l = atand(b/a);
    m = primes(power(b,a));
    n = factorial(a*b);
    o = factor(10*a*b);
    p = power(b,2)*pi;
end
```

```
a=5; b=2.4;
[c,d,e,f,g,h,k,l,m,n,o,p]=assignment1Solution(a,b)
a=4.5; b=4;
[c,d,e,f,g,h,k,l,m,n,o,p]=assignment1Solution(a,b)
a=4.5; b=2;
[c,d,e,f,g,h,k,l,m,n,o,p]=assignment1Solution(a,b)
```

1.2. Szorgalmik

1.2.1. Háromszög

Adott egy háromszög csúcsainak összes koordinátája (xA, yA, xB, yB, xC, yC). Írj függvényt, ami kiszámíja a háromszög szögeit fokban (alpha, beta, gamma), az oldalainak hosszát (a, b, c), a kerületét (k) és a területét (t)!



Solution:

```
function [alpha, beta, gamma, a, b, c, k, t] = haromszog(x_A, y_A, x_B, y_B, x_C, y_C)
    a = power(power(x_B-x_C, 2) + power(y_B-y_C, 2), 1/2);
    b = power(power(x_C-x_A, 2) + power(y_C-y_A, 2), 1/2);
    c = power(power(x_B-x_A, 2) + power(y_B-y_A, 2), 1/2);
    alpha = acosd((power(b, 2) + power(c, 2) - power(a, 2))/(2*b*c));
    beta = acosd((power(a, 2) + power(c, 2) - power(b, 2))/(2*a*c));
    gamma = acosd((power(a, 2) + power(b, 2) - power(c, 2))/(2*a*b));
    k = a + b + c;
    t = (b*c*sind(alpha))/2;
end
```

```
[alpha, beta, gamma, a, b, c, k, t] = haromszog(0, 0, 0, 5, 3, 0)
```

1.2.2. DNS

A DNS négy különböző nukleotidtípusból felépülő lineáris polimer. Az egyes nukleotidokat rendre az A, C, T és G betűkkel jelöljük. Tegyük fel, hogy egy adott DNS láncba a különböző nukleotidok rendre a, c, t és g valószínűséggel épülnek be.

1. Mi annak a valószínűsége, hogy egy kilenc nukleotid hosszúságú DNS lánc éppen AAATCGAGT szekvenciájú?
2. Mi annak a valószínűsége, hogy egy kilenc nukleotid hosszúságú DNS lánc szekvenciája AAAAAAAAA?
3. Mi annak a valószínűsége, hogy egy kilenc nukleotid hosszú véletlen DNS szekvencia négy darab A, két darab T, két darab G és egy darab C nukleotidot tartalmaz?
4. Mi annak a valószínűsége, hogy egy > 9 nukleotid hosszúságú DNS láncban megtalálható az AAATCGAGT szekvencia?

Készíts függvényt, ami adott a, c, t, g valószínűségek esetén kiszámolja az 1-4. kérdésekre a választ, és rendre az A-D változókba teszi!

Solution:

```
n = randi([9,200]);
prob = randi(100,1,4);
prob = prob/sum(prob);
a=prob(1);
c=prob(2);
t=prob(3);
g=prob(4);
[A, B, C, D] = dna(a, c, t, g, n)
```

2. Gyakorlat

2.1. Házifeladatok

2.1.1. Vektorok; skalár és logikai indexelés

Készíts el egy függvényt az alábbiak szerint:

1. 2db bemeneti vektor (*x*: tízelemű, *y*: kételemű, legyen a nevük *a* és *b*); 4 db visszatérési érték (*c*-*f*);
2. logikai indexeléssel határozzuk meg azokat a vektorelemeket, amik az [*a*, *b*] zárt intervallumba esnek, ezek az elemek kerüljenek a *c* vektorba (tehát nem az indexvektor maga, hanem már az indexvektorral kiválasztott elemek);
3. számoljuk ki az [*a*, *b*] zárt intervallum feletti értékek átlagát, ez kerüljön a *d* kimeneti változóba;
4. egyetlen függvényhívással határozzuk meg a korábban készített *c* vektor maximumának értékét és vektoron belüli pozícióját (indexét), ez a kettő egy vektorként legyen az *e* kimeneti változó;
5. egy beépített függvénnyel határozzuk meg *x* azon elemeinek számát, amik az [*a*, *b*] zárt intervallumba esnek, ez a szám kerüljön az *f* kimeneti változóba.

Solution:

```
function [c,d,e,f] = vectorIndexing1(x,y)
    c = x(x>=y(1) & x<= y(2));
    d = mean(c);
    [e(1), e(2)] = max(c);
    f = length(c);
end
```

```
x=linspace(1,25,10)
y=sort(randi(25,2,1))
[c,d,e,f] = vectorIndexing1(x,y)
```

2.1.2. Mátrix

Készíts egy függvényt, aminek egyetlen bemenete egy `A` mátrix lesz (dimenziónként 5, 2 és 3 véletlen számmal), 5 visszatérési értékkel (`B`, `C` mátrixok, `d-f` skalárok), 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, ciklushasználat nélkül), --- `d`
- kiszámítja a második részmátrixban az **oszlopok minimumának átlagát** (beépített függvényekkel, ciklushasználat nélkül), --- `e`

Hasznos lehet: `squeeze`, kettőspont operátor, `max`, `sum`, `min`, `mean`.

Solution:

```
function [B,C,d,e]=myFunction(A)
    B = squeeze(A( :, 1, :));
    C = squeeze(A( :, 2, :));
    d = sum(max(B, [], 2));
    e = mean(min(C, [], 1));
end
```

```
A=rand(5,2,3);
[B, C, d, e] = myFunction(A)
```

2.2. Szorgalmik

2.2.1. Mátrix tulajdonságok

Adott egy négyzetes mátrix: A. Határozza meg a mátrix néhány tulajdonságát beépített Matlab függvények segítségével! A következő tulajdonságokat szeretnénk vizsgálni:

- Van-e negatív eleme?
- Hányszor fordul elő benne a maximális elem?
- Determinánsa nagyobb-e, mint elemeinek átlaga?
- Diagonális-e?
- Ortogonális-e?
- Szimmetrikus-e?
- Ferdén szimmetrikus-e?

Solution:

```
function [a,b,c,d,e,f,g] = matrixTulajdonsagok(A)
    a = length(A(A<0))>0;
    b = sum(sum(A == max(A, [], 'all')));
    c = det(A) > mean(A, 'all');
    d = sum(sum(A)) == trace(A);
    e = all(all(A') == all(inv(A)));
    f = all(all(A') == all(A));
    g = all(all(A') == all(-A));
end
```

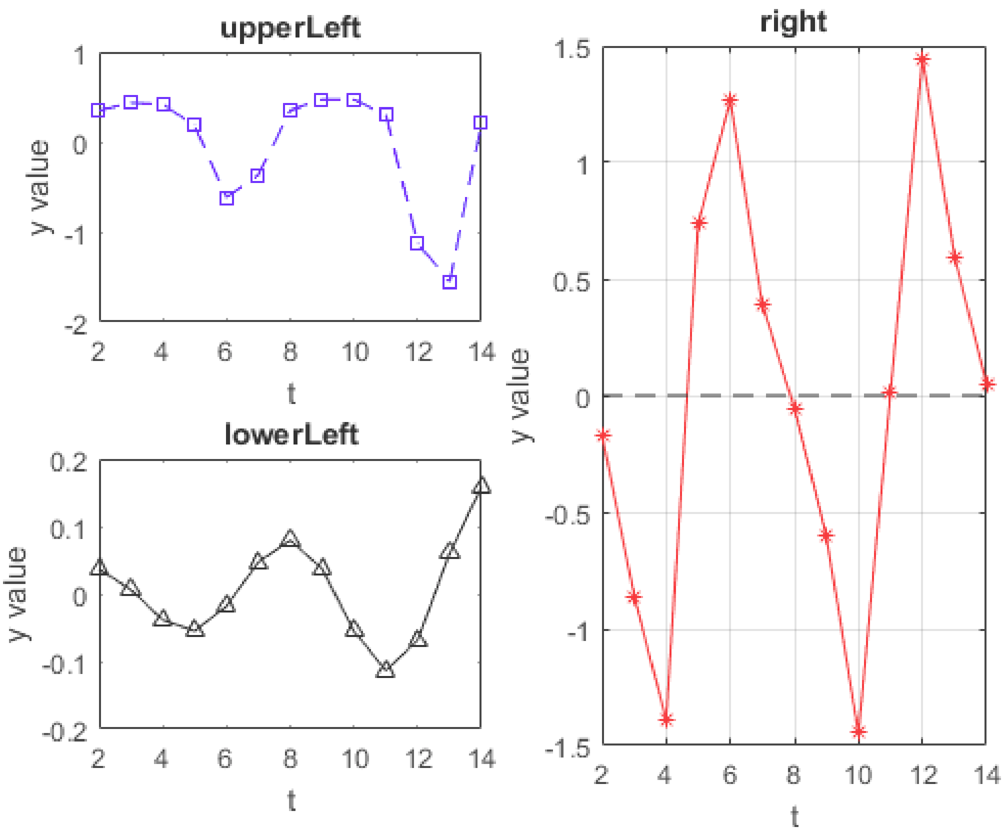
```
A=randi(5,5)-2
[a,b,c,d,e,f,g]=matrixTulajdonsagok(A)
```

3. Gyakorlat

3.1. Házifeladatok

3.1.1. 2D ábrázolás

Készíts egy **szkriptet**, amely az alábbi ábrát állítja elő:



ahol minden alábrán időfüggő (t) adatsorokat láthatunk az alábbi összefüggések szerint:

bal felső: $0.5 - 0.2e^{\ln(t)\cos(t)}$

bal alsó: $\pi^{0.1t-3}\sin(t)$

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

Kérjük ne feledkezz meg a címekről, feliratokról, a vonal/görbe stílusáról, rácsozásról, tengelyhatárokról.

Megjegyzés: a jobb oldali ábrán a 0-szinten van egy vízszintes, fekete, szaggatott vonal.

Solution:

```
s = figure;
t = 2:14;
```

```
fun1 = 0.5 - 0.2 * exp(log(t) .* cos(t));
subplot(2,2,1);
plot(t, fun1, 'b--s');
xlim([t(1), t(end)]);
ylim([-2, 1]);
yticks([-2:1:2]);
xticks([2:2:14]);
title('upperLeft');
xlabel('t');
ylabel('y value');

fun2 = pi.^(0.1.*t-3).*sin(t);
subplot(2,2,3);
plot(t, fun2, 'k-^');
xlim([t(1), t(end)]);
yticks([-0.2:0.1:0.2]);
xticks([2:2:14]);
title('lowerLeft');
xlabel('t');
ylabel('y value');

fun3 = cos(t)./(exp(sin(t)));
fun4 = 0 * t;
subplot(2,2,[2,4]);
plot(t, fun3, 'r-*', t, fun4, 'k--');
xlim([t(1), t(end)]);
xticks([2:2:14]);
title('right');
xlabel('t');
ylabel('y value');
grid on;
```

3.1.2. Szenzorstatisztikai

Készíts egy függvényt az alábbi feladat megoldására (1 bemeneti paraméterrel: `legnyomasErtekek`, 4 visszatérési értékkel: ábra referenciája, `hitelesitettMeresiErtekek`, `elsoSzenzorHelyesMereseiSzovegben`, `szenzorokDeliMeresenekAtlagaSzovegben`):

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 mérési 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 mérési eredményeket egy 4 soros, 3 oszlopos, 31 'mélységű' háromdimenziós mátrix tárolja, a neve '`legnyomasErtekek`' (ez a függvény bemenete, az értékek 900 és 1100 közötti véletlen számok benne).

Konkrét feladatok:

1) Készítsünk a következő ábrához egy hasonlót, mely az első szenzor esti méréseinek értékét tartalmazza minden nap:

- a két rózsaszín vonal a megbízhatósági határoknál van; csak azok a mérési adatpontok vannak kiszínezve, amik helyes tartományon belül vannak,
- rózsaszín vonal: vonal színe, szélessége (2pt),
- az eredeti adatsor: vonal típusa és színe, vonal vastagsága (3pt), marker mérete (7pt),
- a helyes értékek adatsoránál (amit `find`-al célszerű kikeresni): marker típusa és mérete (7), marker közepének ('arcának') és szélének színe, vonal vastagsága (2pt),
- tengelyhatárok (x: 1 és 31 között, y: 890 és 1100 között),
- aktuális axis betűmérete (12pt),
- cím szövege és annak betűmérete (14pt), szedése (félkövér),
- tengelyfeliratok szövege, betűmérete (12pt), szedése (félkövér),
- ábrafelirat és annak elhelyezése (northeastoutside).

2) Logikai indexelés segítségével nullázzuk ki a hitelesített tartományon kívüli értékeket (ciklus használata nélkül!), az eredmény a `hitelesitettMeresiErtekek` változóba kerüljön, innentől ezzel dolgozzunk tovább.

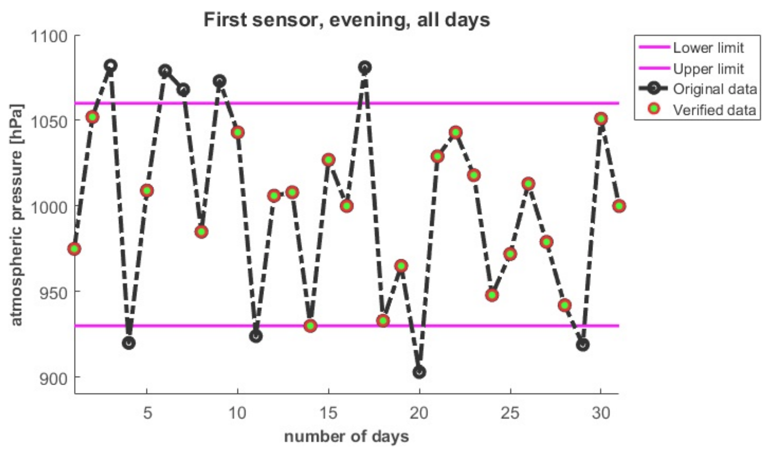
3) Az első szenzor esti mérései közül, a 11. és 20. nap között (11. és 20. napot is beleértve) adjuk meg a helyes mérési értékek darabszámát, azt `sprintf`-fel egy szöveges változóba (`elsoSzenzorHelyesMereseiSzovegben`) mentsük el (ez legyen a szöveg -- a számok persze a Te adataid szerint lesznek: "`Helyes meresi ertekek darabszama (elso szenzor, esti meres, 11-20. napokra): 7`")

4) A déli mérések közül (középső oszlopsorozat) a 2., 3. és 4. szenzorokhoz adjuk meg külön-külön az átlagértéket (csak a helyes mérési értékeket felhasználva!). (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 mérések darabszámát; valamint sima soronkénti (!) szummázással megtudhatjuk a szenzoronkénti mérések összegét. A kettő hányadosa (jól felírva) egy háromelemű vektor, amit kapni szerettünk volna.)

5) A 4)-es pont eredményét tároljuk el a `szenzorokDeliMeresenekAtlagaSzovegben` szöveges változóba az alábbi módon: "`A masodik szenzor deli atlaga: +0991.880, a harmadiknak: +0990.550 es a negyediknek: +1005.696`"

6) A függvény a futása során semmit ne írjon ki a konzolra.

Segítség: `squeeze`, kettőspont operátor, `find`, `plot`, linespec, `xlim`, `ylim`, `title`, `set`, `gca`, `xlabel`, `ylabel`, logikai indexelés, `sum`, `sprintf`.



Solution:

```
function [fig, hitelesitettMeresiErtekek, elsoSzenzorHelyesMereseiSzovegben, ...
       szenzorokDeliMeresenekAtlagaSzovegben] = myFunction(legnyomasErtekek)

fig = figure; % Ezt az abrat hasznaljuk a kirajzoltatasra.

A = squeeze(legnyomasErtekek (1, 3, :));

plot([1,31], [930,930], 'm-', 'LineWidth', 2);

hold on;
```

```
plot([1,31], [1060, 1060], 'm-', 'LineWidth', 2);

hold on;
plot(A, 'k-.o', 'LineWidth', 3, 'MarkerSize', 7);

hold on;
plot(find(A < 1060 & A > 930), A(A < 1060 & A > 930), 'o',...
    'LineWidth', 2, 'MarkerSize', 7, 'MarkerEdgeColor', 'r', 'MarkerFaceColor', 'g');

ax = gca;
ax.FontSize = 12;

xlim([1, 31]);
ylim([890, 1100]);
xticks(5:5:30);
yticks(900:50:1100);
title('First sensor, evening, all days',...
    'FontSize', 14,...
    'FontWeight', 'bold');
xlabel('number of days',...
    'FontSize', 12,...
    'FontWeight', 'bold');
ylabel('atmospheric pressure [hPa]',...
    'FontSize', 12,...
    'FontWeight', 'bold');

legend ('Lower limit', 'Upper limit', 'Original data',...
    'Verified data', 'Location', 'northeastoutside');

legnyomasErtekek(legnyomasErtekek > 1060 | legnyomasErtekek < 930) = 0;
hitelesitettMeresiErtekek = legnyomasErtekek;

db = numel(legnyomasErtekek(find(legnyomasErtekek(1, 3, 11:20) < 1060 & legnyomasErtekek(1,
    3, 11:20) > 930)));
elsoSzenzorHelyesMereseiSzovegben = sprintf('Helyes meresi ertekek darabszama (elso szenzor,
    esti meres, 11-20. napokra): %d', db);

a = squeeze(legnyomasErtekek(2, 2, [find(legnyomasErtekek(2,2, :) > 0)]));
b = squeeze(legnyomasErtekek(3, 2, [find(legnyomasErtekek(3,2, :) > 0)]));
c = squeeze(legnyomasErtekek(4, 2, [find(legnyomasErtekek(4,2, :) > 0)]));
szenzorokDeliMeresenekAtlagaSzovegben = sprintf('A masodik szenzor deli atlaga: %+09.3f, a
    harmadiknak: %+09.3f es a negyediknek: %+09.3f', sum(a)/numel(a), sum(b)/numel(b), sum(c)
    /numel(c));

end
```

```
legnyomasErtekek = 900 + round(190*rand(4, 3, 31));
[fig, hitelesitettMeresiErtekek, elsoSzenzorHelyesMereseiSzovegben,
    szenzorokDeliMeresenekAtlagaSzovegben] = myFunction(legnyomasErtekek)
```

3.1.3. Polinom

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

Bemeneti paraméterek (6db): *a, b, c, d, e, f*;

visszatérési érték (7db):

- az ábra *handler*-je,
- *P*: polinomot leíró vektor,
- *x1* és *x2*: a két értelmezési tartomány,
- *y1* és *y2*: a két polinomkiértékelés eredménye,
- *r* a polinom gyökei.

A függvény létrehozza *x1* és *x2* értelmezési tartományokat, amik az [*e*, *f*] zárt intervallumot jelentik rendre 1-es és 0.001-es lépésközzel.

A függvény kiértékeli $P(x) = ax^3 + bx^2 + cx + d$ polinomot *x1* és *x2* felett (az eredmények rendre *y1* és *y2* változókba kerüljenek).

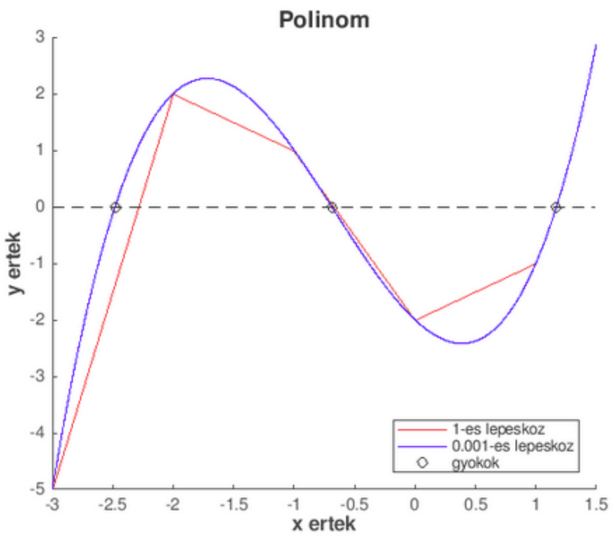
A függvény kiszámítja *P* polinom gyökeit, *r* változóban eltárolja azokat.

Ábrázolja a két polinomkiértékelés eredményét az alábbiak szerint:

- a görbék színe legyen: piros (1) és kék (0.001) (mindkét görbe folytonos vonal, nincs markerjelölés);
- ugyanezen az ábrán jelölje be a polinom gyökeit fekete körökkel; és a 0-szintet egy fekete szaggatott vonallal;
- ábracím 14-es betűmérettel;
- tengelyfeliratok 12-es betűmérettel, félkövéren szedve;
- ábrafelirat doboza '*Location*'-nel *Délkeletre* igazítva.

NB: az ellenőrzés miatt számít a kirajzolás sorrendje, ez legyen: piros vonal, kék vonal, gyökök körei, szaggatott vonal.

NB2: az ábrán ne használj ékezetes betűt (most).



Solution:

```
function [abra, P, x1, x2, y1, y2, r] = gyak5_f51_(a,b,c,d,e,f)
    P = [a, b, c, d];
    x1 = [e : 1 : f];
    x2 = [e : 0.001 : f];
    y1 = polyval(P, x1);
    y2 = polyval(P, x2);
    r = roots(P);

    abra = figure; % ne használjunk másik figure utasítást, ez után a sor után jöjjenek a
     ↪ kirajzolás részletei
    plot(x1, y1, 'r-');

    hold on;
```

```
    plot(x2, y2, 'b-');

    hold on;
    plot(r, [0, 0, 0], 'ko');
    plot([e, f], [0, 0], 'k--');
    title('Polinom', 'FontSize', 14);
    xlabel('x ertek', 'FontSize', 12, 'FontWeight', 'bold');
    ylabel('y ertek', 'FontSize', 12, 'FontWeight', 'bold');
    legend('1-es lepeskoz', '0.001-es lepeskoz', 'gyokok', 'Location', 'southeast');
end
```

```
[abra, P, x1, x2, y1, y2, r] = gyak5_f51_(1,2,-2,-2,-3,1.5)
```

3.2. Szorgalmik

3.2.1. Polinomok, kirajzolás

Adott az alábbi egyenlet:

sin(x) + sin(5*x)

x változóba hozzunk létre egy 0-tól 10-ig egyenletesen felosztott 1000-es méretű vektort.

Majd értékeljük ki x segítségével a függvényünket y-ba.

Illesszünk egy 10.-ed fokú polinomot erre, majd értékeljük ki.

Rajzoljuk ki a két függvényt az első subplotra.

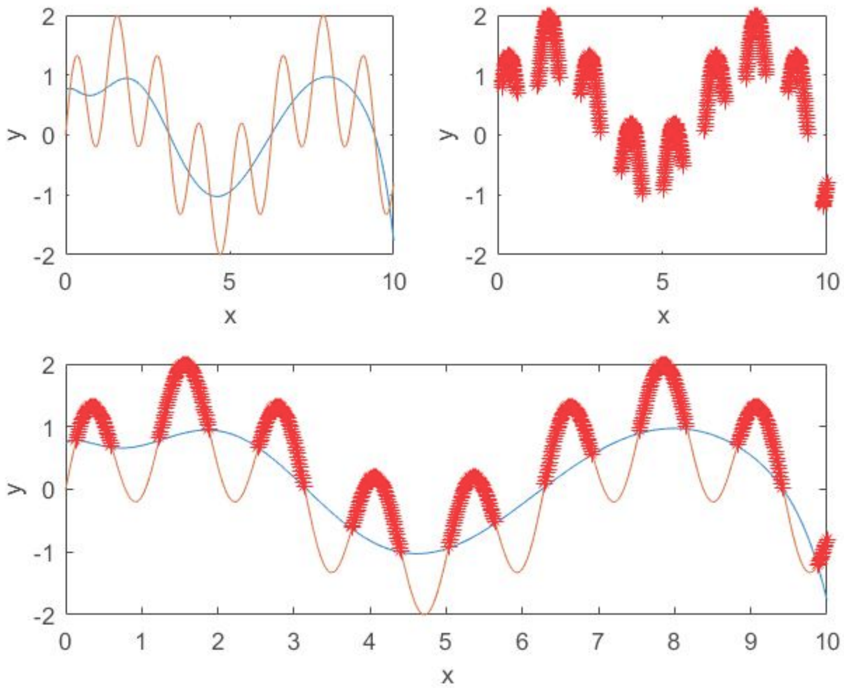
Értékeljük ki azokat a helyeket egy logikai indexvektorba, ahol függvényünk nagyobb értéket vesz fel, mint a polinom.

Rajzoljuk ki ezeket a helyeket piros csillagokkal a második subplot-ra.

Rajzoljuk ki a harmadikra és negyedikre az előző kettőt együtt.

Ne felejtsük el a tengelyfeliratot.

A függvény mindig legyen narancs, a polinom pedig kék.



Solution:

```
function [x,y,p,y2,idx,abra] = Polynom()
    abra=figure;
    x = linspace(0, 10, 1000);
    y = sin(x) + sin(5*x);
    p = polyfit(x, y, 10);
    y2 = polyval(p, x);
    idx = y > y2;

    subplot(2, 2, 1);
    plot(x, y2);
```

```
    xlabel('x');
    ylabel('y');

    hold on;
    plot(x, y);

    hold off;
    subplot(2, 2, 2);
    plot(x(find(idx)), y(idx), 'r*');
    xlabel('x');
    ylabel('y');

    subplot(2, 2, [3,4]);
    plot(x, y2, x, y, x(find(idx)), y(idx), 'r*');
    xticks(0:2:30);
    xlabel('x');
    ylabel('y');
end
```

```
[x,y,p,y2,idx]=Polynom()
```

4. Gyakorlat

4.1. Házifeladatok

4.1.1. Gyümölcsök

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]).

Solution:

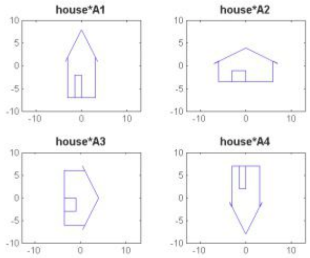
```
function gyumolcsok_ara=gyak4_f41(a,b,c,d)
    A = [a; b; c]
    B = [d]
    gyumolcsok_ara = A\B;
end
```

```
gyumolcsok_ara=gyak4_f41([3, 12, 1],[12, 0, 2],[0, 2, 3],[2.36;5.26;2.77])
```

4.1.2. Transzformáció

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 (H1, ... H4) az egyes subplot-okba rajzolja ki, valahogy így:



- Figyelj a tengelyek méretezésére!
- A címek legyenek félkövér betűvel!

Mit jelentettek ezek a transzformációk? -- egy-egy sorral, kommentként jellemezd a forráskódban a hatásukat. A bemenetre használd a 'house.mat' értéket (a Cody-ban is elérhető, itt csak a load parancsot kell megírni a függvény paraméteréhez és automatikusan ez a fájl hívódik meg).

Solution:

```
function [H1,H2,H3,H4,abra]=gyak4_f42(archivum)

% load függvény használata -> lásd help; külső tesztelésnél fontos, hogy a house.mat
% a house.mat ugyanabban a mappában legyen, mint a függvény, különben teljes elérési
% utat kell adni neki.

load('house.mat')

% H1,H2,H3,H4 rendre az A1,A2,A3,A4 transzformációk eredménye

H1 = kep*A1;
H2 = kep*A2;
H3 = kep*A3;
H4 = kep*A4;

%% Ide kerüljön az ábra kirajzoltatása
abra = figure; % ez után

subplot(2,2,1);

hold on;
plot(H1(:,1), H1(:,2));
xlim([-12, 12]);
ylim([-10, 10]);
xticks(-10:10:10);
yticks(-10:5:10);
title('house*A1',...
    'FontWeight', 'bold');

subplot(2,2,2);
plot(H2(:,1), H2(:,2));
xlim([-12, 12]);
ylim([-10, 10]);
```

```
xticks(-10:10:10);
yticks(-10:5:10);
title('house*A2',...
    'FontWeight', 'bold');

subplot(2,2,3);
plot(H3(:,1), H3(:,2));
xlim([-12, 12]);
ylim([-10, 10]);
xticks(-10:10:10);
yticks(-10:5:10);
title('house*A3',...
    'FontWeight', 'bold');

subplot(2,2,4);
plot(H4(:,1), H4(:,2));
xlim([-12, 12]);
ylim([-10, 10]);
xticks(-10:10:10);
yticks(-10:5:10);
title('house*A4',...
    'FontWeight', 'bold');

end
```

```
abra=gyak4_f42('house.mat')
```

4.1.3. Egyenletrendszer

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

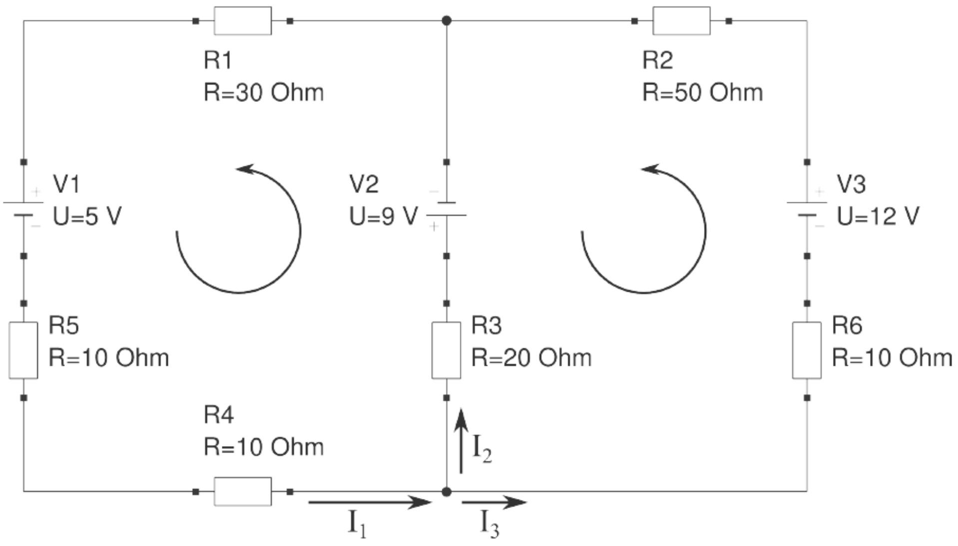
Solution:

```
function [x,y,z]=gyak4_f43(a,b,c,d,e,f,g)
    A = [a b 0 ; c a b; 0 b a]
    B = [e; f-d; g+d]
    C = A\B
    x=C(1);
    y=C(2);
    z=C(3);
end
```

```
[x,y,z]=gyak4_f43(7,2,1,8,-53,832,428)
```

4.1.4. Áramkör

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.

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

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

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

Solution:

```
function aramok=gyak4_f44_(R1,R2,R3,R4,R5,R6,V1,V2,V3)
    A = [1 -1 -1; R4+R5+R1 R3 0; 0 -R3 R6+R2]
    B = [0; -V2-V1; V3+V2]
    aramok = A\B;
end
```

```
R1=30;
```

```
R2=50;
R3=20;
R4=10;
R5=10;
R6=10;
V1=5;
V2=9;
V3=12;
aramok=gyak4_f44_(R1,R2,R3,R4,R5,R6,V1,V2,V3)
```

4.1.5. Korcsoporteloszlás

Számítsd ki egy adott $m \times m$ méretű Leslie-mátrix (populáció korcsoportjainak egyedszámváltozását leíró mátrix) sajátértékeit oszlopvektor *(d)* valamint diagonálmátrix alakban *(D)*, és keresd meg a pozitív valós sajátértékeket *(r)*, és a hozzájuk tartozó jobb oldali sajátvektorokat úgy skálázva, hogy a vektor(ok) elemeinek összege egy legyen *(R)*!

Solution:

```
function [d, D, r, R] = leslie(a, b)
    L = gallery('leslie', a, b); %létrehoz egy Leslie-mátrixot

    d = eig(L);
    D = eig(L, 'matrix');
    c=(d==real(d)&d>0);
    r = d(c);
    [a,b]=eig(L);
    a=a(:,c);
    R = a./sum(a);
end
```

```
a = [0; 2; 5; 3; 1; 1];
b = [0.3, 0.7, 0.7, 0.7, 0.6];
[d, D, r, R] = leslie(a, b)
```

4.1.6. Stochasztikus

Számítsd ki az $\mathbf{A}$ mátrix sajátértékeit diagonálmátrix formában *(D)*, jobb és bal oldali sajátvektorait *(V és U)*, a $\mathbf{V}$ mátrix $\mathbf{V}^{-1}$ inverzét *(Vi)* és a $\mathbf{V} \cdot \mathbf{D} \cdot \mathbf{V}^{-1}$ mátrixot *(B)*!

Solution:

```
function [D, V, U, Vi, B] = stochastic(A)
    V = NaN;
    D = NaN;
    U = NaN;
    [V,D,U] = eig(A);
    Vi = inv(V);
    B = V*D*Vi;
end
```

```
A = [0.2 0.3 0.1 0.1 0.3; 0.1 0.5 0.1 0.2 0.1; 0.05 0.15 0.3 0.3 0.2; 0.05 0.05 0.05 0.8
 ↪  0.05; 0.25 0.25 0.25 0.25 0.];
[D, V, U, Vi, B] = stochastic(A);
```

4.2. Szorgalmik

4.2.1. Átmeneti mátrix

Egy fizikai rendszernek van *m* különböző állapota. Azt, hogy a rendszer egy *t* időpillanatban melyik állapotban milyen valószínűséggel tartózkodik, egy $\mathbf{p}(t) = (p_1, p_2, \dots p_m)$ *valószínűségi vektor adja meg.*

A $\mathbf{p}(t)$ valószínűségi vektor időbeli változását a

$\mathbf{p}(t_{n+1}) = \mathbf{p}(t_n)\mathbf{T}$

egyenlet adja meg, ahol $\mathbf{T}$ az *átmenetivalószínűség-mátrix*, amelynek T_{ij} eleme annak a valószínűsége, hogy a rendszer a t_{n+1} időpillanatban a *j*-edik állapotban lesz, feltéve, hogy a t_n időpillanatban az *i*-edik állapotban van.

Legyen adott egy átmenetivalószínűség-mátrix és egy o állapot. Számold ki az összes *i* állapotból az *o* állapotba jutáshoz szükséges átlagos lépésszámok *(mean first passage time)* $\mathbf{k}$ vektorát! Számítsd ki az állapotok egyensúlyi valószínűségeinek $\mathbf{p_e}$ vektorát!

Tippek és rávezető kérdések:

- az *i* állapotokból egy adott *o* állapotba jutáshoz szükséges k_i átlagos lépésszámok a

  $k_1 = T_{11}k_1 + T_{12}k_2 + \dots T_{1m}k_m + 1$

  $k_2 = T_{21}k_1 + T_{22}k_2 + \dots T_{2m}k_m + 1$

  $\vdots$

  $k_m = T_{m1}k_1 + T_{m2}k_2 + \dots + T_{mm}k_m + 1$

  egyenletrendszer segítségével számíthatók. Használjuk ki, hogy k_o értelemszerűen nulla!
- Egy adott mátrix sajátvektorai azok a vektorok, amelyek a mátrixszal való szorzás hatására skalárszorosukra (az adott sajátvektorhoz tartozó sajátértékszeresükre) változnak. Hogyan változik egy rendszer valószínűségi vektora egyensúlyban? Mennyi a valószínűségi vektor elemeinek az összege?

Solution:

```
szom = [0, 1, 0, 1, 1, 0, 0, 0, 0, 1;
1, 0, 1, 1, 1, 0, 1, 0, 0, 1;
0, 1, 0, 1, 1, 1, 0, 0, 0, 0;
1, 1, 1, 0, 0, 1, 0, 0, 1, 0;
1, 1, 1, 0, 0, 0, 0, 0, 0, 0;
0, 0, 1, 1, 0, 0, 0, 1, 1, 0;
0, 1, 0, 0, 0, 0, 0, 1, 1, 1;
0, 0, 0, 0, 0, 1, 1, 0, 0, 0;
0, 0, 0, 1, 0, 1, 1, 0, 0, 1;
1, 1, 0, 0, 0, 0, 1, 0, 1, 0]
ener = [-0.1 -0.5 -0.5 -3 -2 -2 0 -1 -1.5 -1];
T = min(ones(length(ener)), exp(-ener) .* exp(ener')) / length(ener) .* szom;
T(logical(eye(length(ener)))) = 1 - (sum(T, 2) - T(logical(eye(length(ener)))))
[mi, n] = min(ener);
[k, pe] = mfpt(T, n)
```

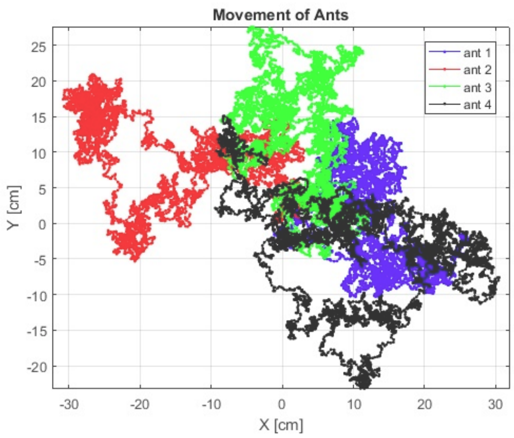
5. Gyakorlat

5.1. Házifeladatok

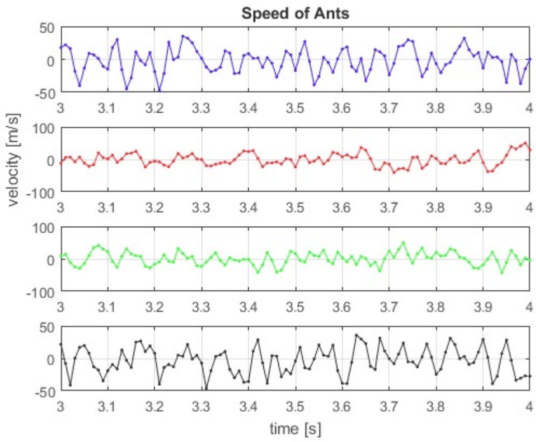
5.1.1. Ants - 2D plot, numerical derivation

A biológusok négy hangya útját követték egy kamerával, amely egy percig 100 Hz-en rögzíti a képeket (6000 rekord). A mérés eredménye egy 4x2x6000 mátrixban van tárolva, ahol az első dimenzió a hangyaszivár azonosító számát (1-4), a második dimenziót jelöli a tengely (1 az X és 2 a Y-nak felel meg), és a a harmadik dimenzió az időt jelöli. A hangyák mozgásának vizsgálatához oldja meg az alábbi feladatokat:

1. Az eredeti mátrix alapján különböző színekkel rajzold ki a hangyák ösvényeit (Ne felejtsd el eltávolítani a singleton dimenziókat a rajzoláshoz). Az ábrának a következőnek kell lennie (a tengely egyenlő távolságra van).



2. Számítsd ki a hangyák sebességét az utazás minden pontján numerikus deriválással. Minden lépés $\sqrt{x^2+y^2}$ cm és $\frac{1}{100}$s alatt, így a deriválás eredménye $\frac{m}{s}$ lesz. Az eredmény egy 4x5999-es mátrix.
3. Számítsd ki az összes hangyára az átlagos, maximális és minimális sebességet, és tárold őket egy változóban, ahol az első oszlop az átlag, a második a maximum és a harmadik a minimum értékeket tartalmazza.
4. Rajzold ki a sebesség görbéket minden hangya számára 3 és 4 másodperc között (mindet tartalmazza). Az ábra a következőképpen néz ki:



Note: The matrix is given with random values, so the figure is changing with each run. For the time vector for the second figure, simply create a vector with values from 1 to 6000 and divide it by 100.

Solution:

```
function [fig1,antSpeed,antStatistics,fig2]=myFunction(antRecords)
%% 1) Draw the path of the ants, based on the original matrix, with different colors (Do not
↪  forget to remove singleton dimensions for plotting).
fig1=figure;

hold on
Ax=squeeze(antRecords(1,1,:));
```

```
Ay=squeeze(antRecords(1,2,:));
plot(Ax, Ay, '-b.')

Bx=squeeze(antRecords(2,1,:));
By=squeeze(antRecords(2,2,:));
plot(Bx, By, '-r.')

Cx=squeeze(antRecords(3,1,:));
Cy=squeeze(antRecords(3,2,:));
plot(Cx, Cy, '-g.')

Dx=squeeze(antRecords(4,1,:));
Dy=squeeze(antRecords(4,2,:));
plot(Dx, Dy, '-k.')

title('Movement of Ants', 'FontWeight', 'bold')
xlabel('X [cm]')
ylabel('Y [cm]')
legend('ant 1', 'ant 2', 'ant 3', 'ant 4', 'Location', 'northeast')
grid on

%% 2) Calculate the speed of the ants at every point of their travel using numerical
 ↪  derivation.
t=1:6000;
t=t./100;
x=squeeze(antRecords(:,1,:));
y=squeeze(antRecords(:,2,:));
s=sqrt(x.*x+y.*y);
ds=diff(s,1,2);
dt=diff(t);
antSpeed=ds./dt;
%% 3) Calculate the average, the maximum, and the minimum speed of all ants, and store them
 ↪  in a variable, where the first column is the average the second is the maximum and the
 ↪  third is the minimum.
a=mean(antSpeed,2);
b=max(antSpeed,[],2);
c=min(antSpeed,[],2);
antStatistics=[a b c]
%% 4) Draw the speed curves for each ant.
fig2 = figure;

subplot(4,1,1)
plot(3:0.01:4, antSpeed(1,300:400), '-b.')
title('Speed of Ants', 'FontWeight', 'bold')
grid on

subplot(4,1,2)
plot(3:0.01:4, antSpeed(2,300:400), '-r.')
ylim([-100, 100])
yticks([-100, 0, 100])
ylabel('velocity [m/s]')
grid on

subplot(4,1,3)
plot(3:0.01:4, antSpeed(3,300:400), '-g.')
yticks([-100, 0, 100])
```

```
ylim([-100, 100])
grid on

subplot(4,1,4)
plot(3:0.01:4, antSpeed(4,300:400), '-k.')
xlabel('time [s]')
grid on

end
```

```
A = rand(4,2,6000)-.5;
antRecords = cumtrapz(A,3);
[fig1,antSpeed,antStatistics,fig2]=myFunction(antRecords);
```

5.1.2. Equations, polynomials, numerical integration, text display

Kérem, oldja meg az alábbi egyenletrendszert egy függvényben. Ahol a bemeneti változók: a, b, c, d.

ax + bz	= 79
cy + dz + 9	= 18
ax + dy +cz - 1	= 496

Tegye a kimenetet egy sorvektorba, amelyet pCoefficients-nek hívnak. (x, y, z)

Most vegye az x, y, z értékeket egy másodrendű polinom együtthatójaként, és értékelje ki a [-5; 5] zárt intervallumban, 1000 egyenlő távolságra elosztott pontottal. Ezt a vektort a pDomain-ba, az erre illesztett polinomot a pValues-ba mentse.

Számítsuk ki a kiszámítottszámított polinom értékek által megadott görbe alatti területet, és tegyük be a pArea változóba.

Számítsd ki a pValues maximális értékét, és azt pDomain értéket ahol ez a maximális érték található, és tedd a pMax változóba.

Írja le a számítás eredményeit egy szövegben (pText), például (sortöréseket tartalmaz):

```
The coefficients of the polynomial are: [383.500, 61.000, -43.500].

The maximum is 9849.000 at 5.000.

The area is 3.152e+04.
```

Solution:

```
function [pCoefficients,pDomain,pValues,pArea,pMax,pText] = myFunction(a,b,c,d)

A = [a,0,b; 0,c,d; a,d,c];
B = [79; 9; 497];
P = A\B;
pCoefficients = P;

x = linspace(-5, 5, 1000);
pDomain = x;
pValues = polyval(P, x);
pArea = trapz(x, pValues);

[max_value, index] = max(pValues);
pMax = [max_value, x(index)];

pText = sprintf('The coefficients of the polynomial are: [%.3f, %.3f, %.3f].\n The maximum is
 ↪  %.3f at %.3f.\n The area is %.3e', pCoefficients, pMax, pArea);

end
```

```
coeffs=randi(100,1,4);
[pCoefficients,pDomain,pValues,pArea,pMax,pText] =
 ↪  myFunction(coeffs(1),coeffs(2),coeffs(3),coeffs(4));
disp(pText)
```

5.2. Szorgalmik

5.2.1. Interpolációs polinom

Adott öt adatpont (x, y). Határozzuk meg a második, harmadik és negyedik pontbeli deriváltat interpolációs polinom segítségével! Minden egymást követő három pontra illesszünk legfeljebb másodfokú polinomot, és a középső pontra számítsuk ki a deriváltat analitikusan (a_2, a_3, a_4), és a pont $(x - 10^{-6}, x + 10^{-6})$ környezetét megvizsgálva numerikusan (n_2, n_3, n_4)!

Solution:

```
x = [1., 1.2, 1.7, 2.6, 2.8];
y = [0.2280, 0.5100, 0.4800, -0.2040, -0.1140];
[n2, n3, n4, a2, a3, a4] = interpolacios(x, y);
```

5.2.2. Pillangók

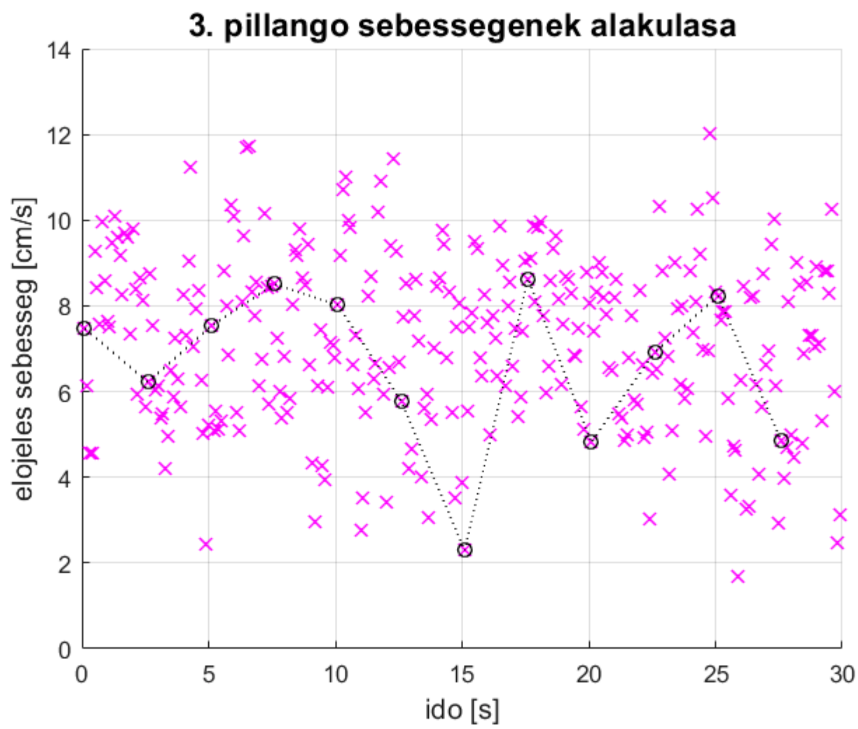
Biológus barátaink ismételt kutatásaik során pillangók mozgását rögzítették. A mérés során 5 pillangó x- és y-koordinátáját követték nyomon fél percen keresztül, a mérési pontokat 0.1 másodpercenként rögzítve.

Amit a számunkra nyújtanak (ez töltődik be a butterflyRecords.mat fájlból --- ezek a függvényed bemenetei):

- 2x5x300-as méretű `butterflyRecords` tömb, itt az egyes dimenziók: 1: x- és y-koordináta; 2: az 5 különböző pillangó adatai; 3: idődimenzió;
- 1x300-as méterű `t` sorvektor, ami az egyes mérési időpontok idejét tartalmazza.

Amit tőlünk kérnek: vizsgáljuk meg alaposabban a **3. pillangó** viselkedését, és az alábbi számításokat végezzük el

- minden mérési időpontra számoljuk ki a pillangó origótól vett euklideszi távolságát, ez kerüljön a **2. visszatérési érték**be (`bf3_tav_origotol`);
- számoljuk ki a pillangó sebességét, ez kerüljön a **3. visszatérési érték**be (`bf3_sebesseg`);
- a kiszámolt időfüggő sebességértékeket rajzoljuk ki az alul látható ábrához hasonlóan, specifikus kérések:
- --- az ábra referenciája a függvényünk **1. visszatérési érték**e;
- --- legyen az ábrának "rácsos" háttere;
- --- rajzoljuk ki az összes sebesség-értéket rózsaszínű, x-jelölésekkel, összekötetlenül;
- --- alulmintavételezve a sebesség-értékeket, csak minden 25.-et rajzoljuk ki fekete körökkel jelölve, pontozott vonallal összekötve (tehát: 1., 26., 51., stb.);
- --- legyenek tengelyfeliratok, 12-es betűmérettel;
- --- legyen ábrafelirat 14-es betűmérettel, félkövér szedéssel;
- a kiszámolt sebességértékből (!) számoljuk ki a pillangó által megtett összes utat trapézmódszerrel, ez kerüljön a **4. visszatérési érték**be (`bf3_megtett_ut`);
- egy szöveges változóba (**5. visszatérési érték**) is tároljuk le e legutóbbi eredményt, a szövegben a beszúrt számérték fixpontos alakú legyen pontosan 4 tizedesjeggyel, a szöveg végén legyen sortörés is: `"A 3. pillango altal megtett ut nagysaga: xxx.xxxx centimeter."` (itt szándékosan hiányzik a szám a példából).

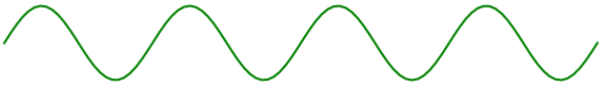


Solution:

```
load('butterflyRecords.mat');
[fig1, bf3_tav_origotol, bf3_sebesseg, bf3_megtett_ut, bf3_szov] =...
    hetfo_1214_f3(butterflyRecords, t);
```

5.2.3. Függöny

Legalább milyen széles (*fsz*) függönyre van szükségünk ahhoz, hogy teljesen eltakarjon egy *asz* szélességű ablakot, figyelembe véve, hogy a függöny négy hullámhosszúságú szinuszfüggvénynek megfelelő alakot vesz föl (lásd az ábrát)?



Tipp: egy függvénnyel (tehát nem paraméteresen) megadott görbe ívhosszát az

$$s = \int_a^b \sqrt{1 + \left(\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right)^2}$$

képlettel számíthatjuk.

Solution:

```
asz = 1;
fsz = fuggony(asz);
```

6. Gyakorlat

6.1. Házifeladatok

6.1.1. Ragadozó-zsákmány

Az alabbi feladatban a ragadozo-zsakmany modellt (Lotka-Volterra modell, amely egy adott teruleten a ragadozo-zsakmany egyedszam-viszonyt irja le. https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations) vizsgaljuk meg, mely ketvaltozos differencialegyenletkent irhato le.

Keszits egy fuggvenyt mely 1 abrat general, 2 bemeneti parameterrel:

1. kezdeti ertekek `y_0=[y1_0, y2_0]`;
2. idointervallum `t=[t_0, t_max]`;

visszateresi ertekei

1. az abra hivatkozasa;
2. a megoldasfuggveny mint pontsorozat ertelmezesi-tartomanya (T) es
3. ertekkeszlete (Y) (ez utobbi ketto az ode45 visszateresi erteke is, ugye).

A feladat az alabbi elsorendu, ketvaltozos DE megoldasa:

$$\frac{dy_1}{dt} = \left(1 - \frac{y_2}{\mu_2}\right) y_1$$

$$\frac{dy_2}{dt} = -\left(1 - \frac{y_1}{\mu_1}\right) y_2$$

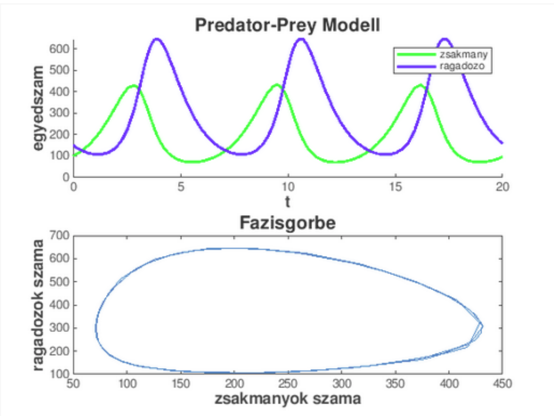
y_1: zsakmany egyedszam, y_2: ragadozo egyedszam;

μ_1: zsakmany kornyezeti eltartokepessege, μ_2: ragadozo kornyezeti eltartokepessege

Legyen most $\mu_1 = 200$ es $\mu_2 = 300$;

a differencialegyenlet-rendszert a kododban anonim fuggvenykent definiald.

A megoldasod az alabbihoz hasonlo modon rajzolja ki a felso alabran az egyedszamok idofuggo erteket, mig az also alabran az egyedszamokon felvett sikban a fazisportret:



(Nagyjaboli elvarasok: felul folytonos zold es kek vonal 2-es vastagsaggal, abrafelirat 14-es betumerettel, tengelyfeliratok felkoveren szedve 12-es meretben, felso alabrahoz abramagyarazat (legend).)

Solution:

```
function [fig, T, Y] = LotkaVolterra(y_0, t)
% elsorendu, ketvaltozos differencialegynelet megoldasa,
% Lotka-Volterra ragadozo-zsakmany modell
    fig = figure;

    m1=200;
    m2=300;
    F = @(t,y) [(1-(y(2)/m2))*y(1); -(1-(y(1)/m1))*y(2)];
    [T,Y]= ode45(F,t,y_0);
```

```
    subplot(2,1,1);
    hold on;
    plot(T,Y(:,1),'g-','LineWidth',2);
    plot(T,Y(:,2),'b-','LineWidth',2);
    title("Predator-Pray Modell", "FontSize", 14, "FontWeight", "bold");
    ylabel("t", "FontSize", 12,"FontWeight","bold");
    xlabel("egyedszam", "FontSize", 12,"FontWeight","bold");
    legend({'zsakmany', 'ragadozo'}, 'Location', 'northeast');

    subplot(2,1,2);
    plot(Y(:,1),Y(:,2),'b-','LineWidth',2);
    title("Fazisgorbe", "FontSize", 14, "FontWeight", "bold");
    xlabel("zsakmanyok szama", "FontSize", 12,"FontWeight","bold");
    ylabel("ragadozok szama", "FontSize", 12,"FontWeight","bold");

end
```

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

6.1.2. Kemiai reakcio 3 anyaggal

Az alabbi feladatban egy kemiai reakciot vizsgalunk, mely soran harom anyag koncentraciovaltozasat kiserjuk figyelemmel, a folyamatot egy haromvaltozos elsorendu differencialegyenlet-rendszer irja le.

Keszits egy fuggvenyt, mely 1 abrat general, 2 bemeneti parametere van:

1. kezdeti ertekek `y_0=[A_0, B_0, C_0];`
2. idointervallum `t=[t_0, t_max]`);

visszateresi ertekei:

1. az abra hivatkozasa;
2. a megoldasfuggveny mint pontsorozat ertelmezesi-tartomanya (`T`) es
3. ertekkeszlete (`Y` --- haromoszlopos matrix, az egyes anyagok koncentracioertekei) (ez utobbi kettoÃÂ (`T` es `Y`) az ode45 visszateresi erteke is, ugye);
4. logikai indexvektor, a lentebb reszletezett logikai indexeles eredmenyvektora.

A feladat az alabbi elsoÂrendu, haromvaltozos DE megoldasa:

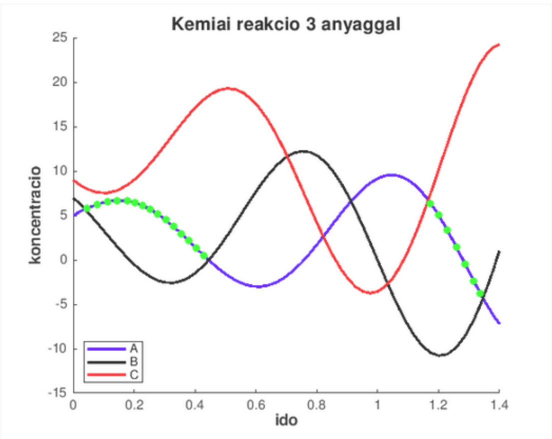
$$\frac{dA}{dt} = 1.2A + 4.1B - 1.7C$$

$$\frac{dB}{dt} = -8A - 1.4B + 2.1C$$

$$\frac{dC}{dt} = 2.1A - 7.2B + 1.3C$$

Oldd meg ezt a differencialegyenlet-rendszert a bemeneten kapott idointervallumon, a szinten bemenetkent kapott kezdeti ertekek mellett. A differencialgyenlet-rendszert a megoldasodon belul anonim fuggvenykent definiald.

Hatarozd meg azokat az indextartomanyokat **logikai indexeles** segitsegevel, amikor az elsoÂ anyag koncentracioja nagyobb, mint a masodike, de kisebb, mint a harmadike. Abrazold a megoldasfuggvenyeket (tengelyfeliratok, betumeret, vonalsziÂnek es tiÂpusok, abrafelirat), valamint jelold meg az A-anyag feltetelnek eleget tevo ertekeit kulon zold pontokkal, osszekotes nelkul, az alabbi abrahoz hasonloan:



(Abra megjegyzesek: anyagok vonalszinei:kek, fekete, piros, mindharom 2-es vonalvastagsag; a zold pontoknal a markermeret 20-as, cim 14-es betumeret, tengelyfeliratok 12-esek, felkover szedessel, van abramagyarazat (legend) az elso harom vonalhoz.)

Solution:

```
function [fig, T, Y, logind] = kemiai_3anyag(y_0, t)
% Kemiai egyenlet, 3 anyaggal
    fig=figure;

    y_d = @(t,y) [1.2*y(1)+4.1*y(2)-1.7*y(3);
                  -8*y(1)-1.4*y(2)+2.1*y(3);
                  2.1*y(1)-7.2*y(2)+1.3*y(3)]

    [T,Y] = ode45(y_d, t, y_0)
    logind = Y(:,1)>Y(:,2) & Y(:,1)<Y(:,3)

    hold on;
    plot(T,Y(:,1),'-b','LineWidth',2)
    plot(T,Y(:,2),'-k','LineWidth',2)
```

```
    plot(T,Y(:,3),'-r','LineWidth',2)
    title('Kemiai reakcio 3 anyaggal','FontWeight','bold','FontSize',14)
    xlabel('ido','FontWeight','bold','FontSize',12)
    ylabel('koncentracio','FontWeight','bold','FontSize',12)

    Y1 = Y(:,1)
    plot(T(logind),Y1(logind),'.g','MarkerSize',20)
    legend({'A','B','C'},'Location','southwest')
    yticks(-15:5:25)
    ylim([-15 25])
end
```

```
t = [0 1.4];
y_0 = [5, 7, 9];
[~, ~, ~] = kemiai_3anyag(y_0, t);
```

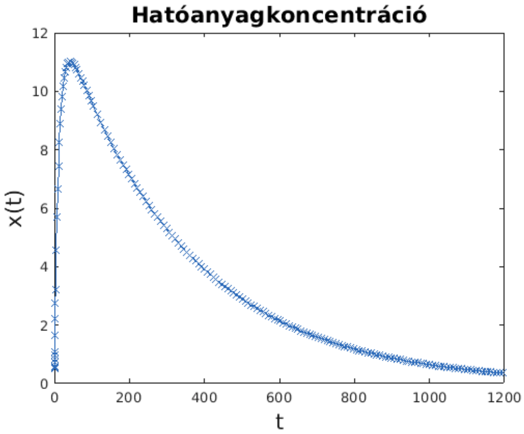
6.2. Szorgalmik

6.2.1. Kiürülési idő

Kiürül-e három órán belül (`kiurulE`), és ha igen, pontosan mennyi idő alatt (`kiurulesiIdo`) az a hatóanyag a szervezetből, aminek szervezeten belüli koncentrációváltozását a

$$x' = \Phi(t) - \lambda x$$

differenciálegyenlet írja le, ahol $\Phi(t)$ az adagolt mennyiség az idő függvényben (`phi`), λ a kiürülési állandó (`lambda`), a hatóanyag kezdeti koncentrációja pedig x_0 (`x0`) ? (Kiürülésnek tekintjük, ha a hatóanyagkoncentráció a kimutathatósági határt jelentő $0,001$ alá csökken.) Ábrázold a hatóanyagkoncentráció időfüggését az adagolás megkezdésétől számított első húsz percre, a lenti ábrának megfelelően. (Az idő mértékegységeként mindenhol másodpercet használjunk!)



Solution:

```
phi = @(t) exp(-0.003 * t);
lambda = 8e-2;
x0 = 0.5;
[kiurulE, kiurulesiIdo, abra]=hatoanyagKiurules(phi, lambda, x0);
```

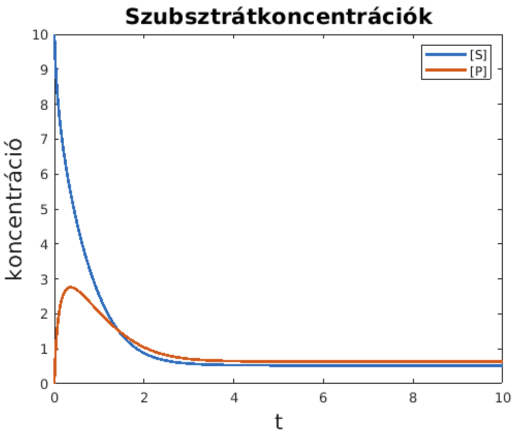
6.2.2. Enzimek

Két egymást követő biokémiai átalakulást egy-egy enzim katalizál olyan módon, hogy az első enzimreakciót a reakció terméke (P) gátolja. Az enzimreakciók mellett az első reakció szubsztrátja (szubsztrát az a vegyület, aminek az átalakulását az enzim gyorsítja) konstans mennyiségben folyamatosan termelődik, a második reakció szubsztrátja pedig az enzimreakció mellett a mennyiségével egyenesen arányos sebességgel is bomlik. A reakciókat az alábbi sebességi egyenletek írják le:

$$\frac{d[S]}{dt} = -k_{cat1}[E_1]_T \frac{1}{1+[P]/K_i} \frac{[S]}{K_{m1}+[S]} + \sigma$$

$$\frac{d[P]}{dt} = k_{cat1}[E_1]_T \frac{1}{1+[P]/K_i} \frac{[S]}{K_{m1}+[S]} - k_{cat2}[E_2]_T \frac{[P]}{K_{m2}+[P]} - \theta([P])$$

A differenciálegyenletrendszer numerikus megoldásával számítsd ki, és az alábbi ábrának megfelelően ábrázold az S és a P szubsztrátok koncentrációit (`c_s` és `c_p`) az idő függvényében a 0-10 s időintervallumra, amennyiben a k_{cat1}, k_{cat2}, K_{m1}, K_{m2}, K_i, $[E_1]_T$, $[E_2]_T$ és a σ paraméterek, valamint a $\theta([P])$ függvény adottak (`k_cat1`, `k_cat2`, `K_m1`, `K_m2`, `K_i`, `e1_T`, `e2_T`, `sigma`, `theta`, vonalvastagság 2, a cím és a tengelyfeliratok betűmérete 16, a cím szedése félkövér).



Solution:

```
[c_s, c_p, abra] = enzimReakciok(250, 35, 120, 250, 1, 3, 1, 2, @(x) 3 * x);
```

7. Gyakorlat

7.1. Házifeladatok

7.1.1. 3D feladat 1.

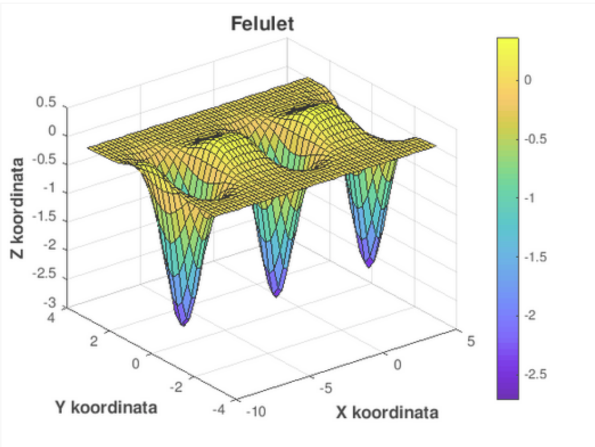
Keszits egy fuggvenyt, mely 4 bemeneti parameterrel es 1 visszateresi ertekkel rendelkezik, es abrazolja az alabbi keplettel megadott feluletet:

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

A bemeneti parameterek rendre az ertelmezesi tartomany vegpontjai: `[x_min,x_max] x [y_min,y_max]` , a felbontas mindket tengely menten 0.25 legyen.

A visszateresi ertek az abra hivatkozasa legyen.

Az abrat feliratozd megfeleloen (cim 14-es betumeret, tengelyfeliratok 12-es betumeret felkover szedessel); jelenitsd meg a z-ertekeket kodolo szinskalat is.



Solution:

```
function abra = gyak7_f71_(x_min, x_max, y_min, y_max)
    abra = figure;

    felbontas=0.25;
    [x,y]=meshgrid(x_min:felbontas:x_max,y_min:felbontas:y_max);
    z=sin(x).*exp(-sin(x)-y.^2);
    surf(x,y,z);
    title('Felulet', 'FontSize', 14)
    xlabel('X koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    ylabel('Y koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    zlabel('Z koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    colorbar;
end
```

```
xmin = -10;
xmax = 5;
ymin = -3;
ymax = 3;
gyak7_f71_(xmin, xmax, ymin, ymax);
```

7.1.2. 3D feladat 2.

Keszits egy fuggvenyt, mely 6 bemeneti parameterrel es 1 visszateresi ertekkel rendelkezik, es abrazolja az alabbi keplettel megadott feluletet, szintvonalaival egyutt:

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

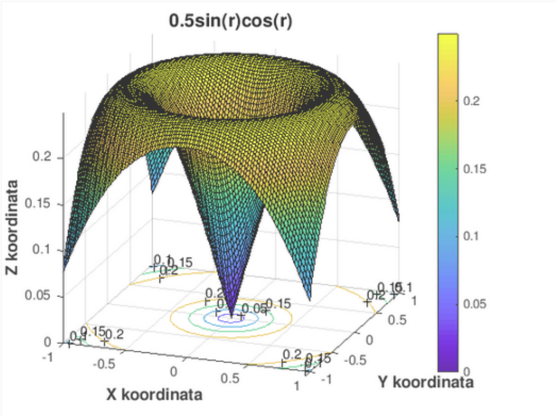
(Elkepzelhetjuk ugy is, mint egy `z=f(x)` fuggveny az `y=0` sikban, amit aztan a Z-tengely korul megforgatunk.)

A bemeneti parameterek rendre az ertelmezesi tartomany vegpontjai: `[x_min,x_max] x [y_min,y_max]` , valamint az abra nezopont-szogei (`az`, `el`). A felbontas mindket tengely menten 0.02 legyen.

A visszateresi ertek az abra hivatkozasa legyen.

Az abrat feliratozd megfeleloen (cim 14-es betumeret, tengelyfeliratok 12-es betumeret felkover szedessel); jelenitsd meg a z-ertekeket kodolo szinskalat is, valamint feliratozd a szintvonalakat, allitsd be a nezopontot.

Az eredmeny az alabbi abrahoz hasonlo legyen:



Solution:

```
function abra = gyak7_f72_(x_min, x_max, y_min, y_max, Az, El)
    abra = figure;

    felbontas=0.02;
    [x,y]=meshgrid(x_min:felbontas:x_max,y_min:felbontas:y_max);
    r=sqrt(x.^2+y.^2)
    z=0.5.*sin(r).*cos(r);
    surfc(x,y,z);
    title('0.5sin(r)cos(r)', 'FontSize', 14)
    xlabel('X koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    ylabel('Y koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    zlabel('Z koordinata', 'FontWeight', 'bold', 'FontSize', 12);
    colorbar;
    for Az= 0:Az
        view(Az,El)
        pause(felbontas)
    end
end
```

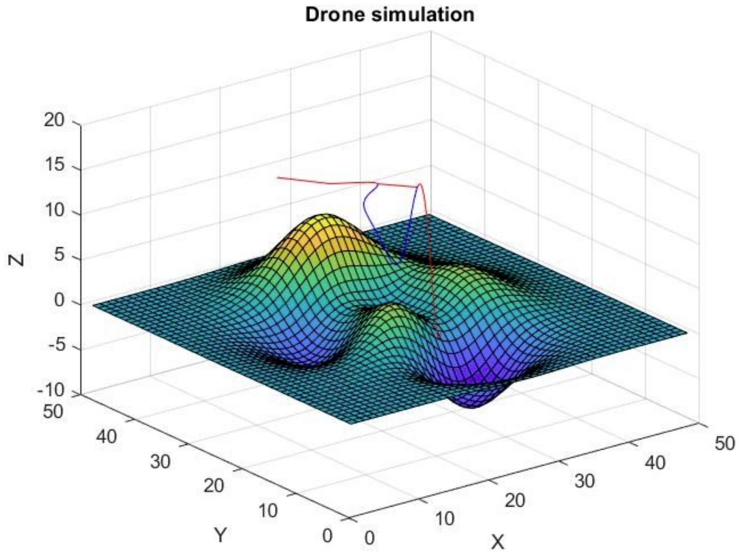
```
xmin = -1;
xmax = 1;
ymin = -1;
ymax = 1;
Az = 20;
El = 20;
gyak7_f72_(xmin, xmax, ymin, ymax, Az, El);
```

7.2. Szorgalmik

7.2.1. Drón repülés

Egy drón repülési adatait tartalmazó (drone.mat) fájlban található útvonalat kell ábrázolni:

- Töltsd be a repülési adatokat a 'drone.mat' fájlból, amiben egy 3x101-es dimenziójú mátrix található (`drone1`). Írd a sorokat rendre, az *x,y,z* változókba.
- Készíts egy ábrát és írd az *abra* változóba a handler-t!!!
- Szimulált domborzat felületét ábrázolunk, amelyhez tartozó koordinátákat a peaks paranccsal lehet elérni (egy mátrixot ad vissza a peaks parancs és ennek a felületét kell ábrázolni).
- Rajzoljuk ki a drón útvonalát 3d ábrán, kék folytonos vonallal.
- Állítsd be a tengelyhatárokat, és legyen cím és tengely feliratok is, az alábbi ábra alapján
- Az y tengelyen található 20 és 35 közötti zárt intervallumon található értékek hibásak, így azokat el kell távolítani. A helyes értékek indexeit egy *y_idx* változóba írd bele.
- Ábrázold a kiindexelt értékek alapján az útvonalat 3d ábrán, folytonos piros vonallal.



Solution:

```
function [x,y,z,abra,y_idx] = drone()

load('drone.mat');

x = drone1(1,:);
y = drone1(2,:);
z = drone1(3,:);
y_idx = find(y < 20 | y > 35);

abra = figure;
surf(peaks(50));

hold on;
plot3(x, y, z, 'b');
plot3(x(y_idx), y(y_idx), z(y_idx), 'r');

title('Drone simulation', 'Fontsize', 14)
xlabel('X', 'Fontsize', 12);
ylabel('Y', 'Fontsize', 12);
zlabel('Z', 'Fontsize', 12);

end
```

```
[x,y,z,abra,y_idx] = drone();
```

8. Gyakorlat

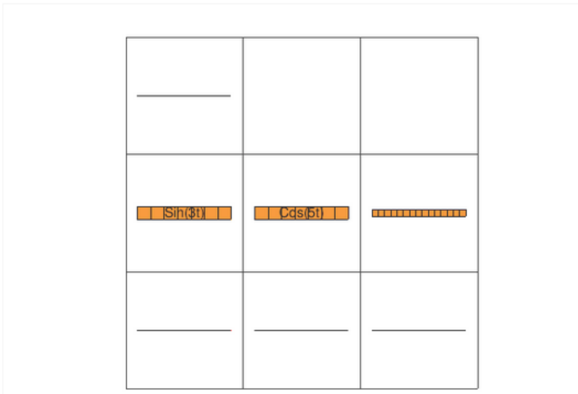
8.1. Házifeladatok

8.1.1. Cellatömb

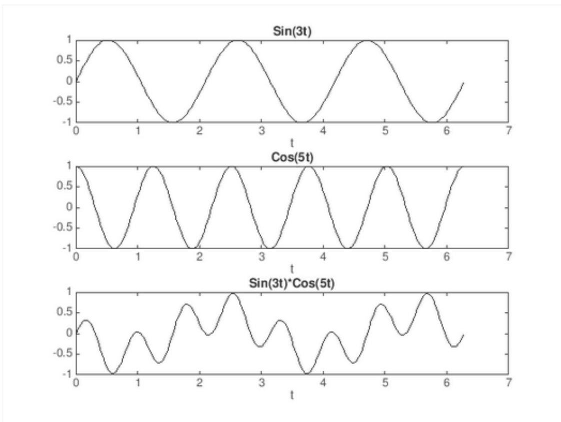
Készíts egy függvényt 1 bemeneti paraméterrel (db), és 2 visszatérési értékkel (az elkészített cellatömb és az ábra handler-je). A függvény töltsön fel egy 3x3-as cellatömböt az alábbiak szerint:

- az első eleme (1,1 index alatt) 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 cellákban;
- a harmadik sorban a stringeknek megfelelő **numerikus értékek** legyenek eltárolva a cellákban.

(Nem kell Neked kirajzolni, csak szemléltetésképpen a cellatömb ehhez hasonló legyen:



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



(Amit az ábrán ellenőrzünk: subplotok száma, ábracím egyezése a cellatömb tartalmával, for-ciklus megléte, subplotok adatsorának egyezése a cellatömb tartalmával, ábra-vonalak színe.)

Solution:

```
function [cellaTomb, abra]=gyak8_f81_(db)
    abra = figure;

    t = linspace(0, 2*pi, db);
    cellaTomb = {t,[] , []; 'Sin(3t)', 'Cos(5t)','Sin(3t)*Cos(5t)'; sin(3.*t), cos(5.*t),
    ↪  sin(3.*t).*cos(5.*t)};
    for i = 1:3
        subplot(3,1,i);
        plot(t,cellaTomb{3,i},'-k');
        title(cellaTomb{2,i});
```

```
        xlabel('t');
    end
end
```

```
db = randi([101 300]);
[cellaTomb, abra]=gyak8_f81_(db);
```

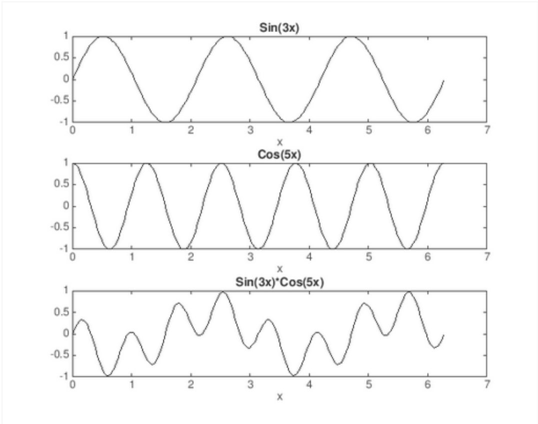
8.1.2. Struktúratömb

A 8.1 Cellatömb feladat struktúrás változata:

Készíts egy függvényt 1 bemeneti paraméterrel (db), és 2 visszatérési értékkel (az elkészített struktúra és az ábra handler-je). A függvény töltsön fel egy struktúra-tömböt az alábbiak szerint:

- készíts egy struktúrát, ami rendelkezik **x**, **y** és **nev** mezőkkel, amik rögtön tartalmazzák az első alábra adatait (**x**: 0-tól 2π-ig db darab szám, **y**: a sin(3x) kifejezés szerinti értékek, **nev**: az alábrához szükséges címke, 'Sin(3x)');
- terjeszd ki struktúra-tömbé (MATLAB Helpben keresés: Create Structure Array), hogy tartalmazza a cos(5x) és sin(3x)*cos(5x) kifejezésekhez tartalmazó **x**, **y** és **nev** adatokat is.

Ugyancsak a függvényben, ciklus használatával rajzoljuk ki a fent kiszámolt értékeket egy ábrán, 3 subplotra, fekete, folytonos vonallal! A subplotok ÉT-jét, ÉK-ját és címeit is a struktúra-tömbből használjuk közvetlenül! Az alábbihoz hasonló ábrát készítsen a függvény:



(Amit az ábrán ellenőrzünk: subplotok száma, ábracím egyezése a struktúra-tömb tartalmával, `for`-ciklus megléte, subplotok adatsorának egyezése a struktúra-tömb tartalmával, ábra-vonalak színe.)

Solution:

```
function [strukturaTomb, abra] = fgy_82_(db)
    abra = figure;

    strukturaTomb(1).x=linspace(0,2*pi, db);
    strukturaTomb(1).y=sin(3.*strukturaTomb(1).x);
    strukturaTomb(1).nev= 'Sin(3x)';
    strukturaTomb(2).x=strukturaTomb(1).x;
    strukturaTomb(3).x=strukturaTomb(1).x
    strukturaTomb(2).y=cos(5.*strukturaTomb(2).x);
    strukturaTomb(2).nev='Cos(5x)';
    strukturaTomb(3).y=sin(3.*strukturaTomb(3).x).*cos(5.*strukturaTomb(3).x);
    strukturaTomb(3).nev='Sin(3x)*Cos(5x)';
    for i=1:3
        subplot(3,1,i);
        plot(strukturaTomb(i).x, strukturaTomb(i).y,'-k');
        title(strukturaTomb(i).nev);
        xlabel('x');
    end
end
```

```
db = randi([101 300]);
[strukturaTomb, abra] = fgy_82_(db);
strukturaTomb
```

8.1.3. Diffegyenlet fájlolvasással

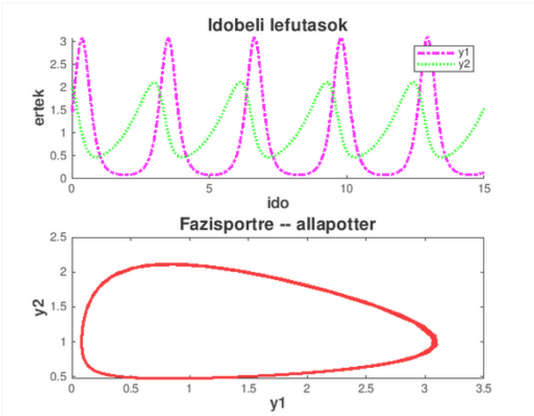
A feladat soran egy **elsorendu, ketvaltozos** differencialegyenletet kell megoldani. A fuggveny amit keszitesz rendelkezik

- 1 bemeneti parameterrel --- egy fajlnev (`bemenetiFajlNev`),
- 3 visszateresi ertekkel --- abra-referencia + a diffegyenlet megoldasfuggvenyenek ertelmezesi tartomanya (`T`) + ertekkeszlete (`Y`).

Feladatok:

- nyisd meg a parameterkent kapott szovegfajlt (ha MATLAB-ban dolgozol, a bemenetkent hasznalt `83.text` fajlt megtalalod a wiki-n, a csatolmanyok kozott);
- olvasd be (formazott szoveg olvasasa) belole a benne talalhato 4 lebegopontos szamot, ezek: 1. diffegyenlet megoldas kezdo idopontja, 2. diffegyenlet megoldas zaro idopontja, 3. y_1 kezdeti ertek, 4. y_2 kezdeti ertek;
- zard be a fajlt;
- oldd meg az alabbi diffegyenletet a beolvasott idohatarok kozott, a beolvasott kezdetiertekekkel (ha lehet, anonim fuggvenykent ird fel): $\dot{y_1} = 5ln(y_2)y_1$ $\dot{y_2} = (1 - 1.2y_1)y_2$
- abrazold a megoldasok idofuggvenyet es fazisgorbejet az alabbihoz hasonlo modon (amit ellenorzunk: vonalszinek es vonaltipusok {y1: magenta, pontozott-szaggatott (*dash-dot*), 2-es vastagsag; y2: zold pontozott (*dotted*), 2-es vastagsag; fazisportren: sima piros vonal, 2-es vastagsag}, tengelyfeliratok (12-es felkover), abracimek (14-es), abrafelirat a felso subplotnal (legend));
- a diffegyenlet megoldasat (`T` és `Y` matrixok osszefuzve: 3 oszlopos, nagyon sok soros matrix lesz) ird ki binaris modban az alabbi nevu fajlba: "`gyak_f83_kimenet.bin`", ne felejtsd el utana bezarni;
- a diffegyenlet megoldasat (`T` es `Y` matrixok) ne feledd a visszateresi ertekekben visszaadni.

Ehhez hasonlo abrat kellene kapnod:



Solution:

```
function [abra, T, Y] = f83_(bemenetiFajlNev)
    %[T, Y] = ode45(... erdemes a diffegyenletet ilyen formaban megoldani, igy egybol be
    ↪  lesznek allitva a visszateresi ertekek is

    abra = figure;

    fid=fopen(bemenetiFajlNev);
    data=fscanf(fid, '%f', 4);
    t=[data(1) data(2)];
    y=[data(3) data(4)];
    diff=@(t,y) [5*log(y(2))*y(1);
    (1-1.2*y(1))*y(2)];
    fclose(fid);

    [T, Y] = ode45(diff,t,y);
    subplot(2,1,1);
    hold on;
    plot(T,Y(:,1),'-.m','LineWidth', 2);
    plot(T,Y(:,2), 'g:','LineWidth', 2);
    title('Idobeli lefutasok', 'FontSize', 14);
    xlabel('ido', 'fontsize', 12, 'fontweight', 'bold')
    ylabel('ertek', 'fontsize', 12, 'fontweight', 'bold')
    legend('y1','y2')
```

```
    subplot(2,1,2);
    plot(Y(:,1),Y(:,2),'r-','LineWidth', 2);
    title('Fazisportre -- allapotter', 'FontSize', 14);
    xlabel('y1', 'fontsize', 12, 'fontweight', 'bold')
    ylabel('y2', 'fontsize', 12, 'fontweight', 'bold')

    filename='gyak_f83_kimenet.bin';
    fid=fopen(filename, 'w');
    fwrite(fid,T,'double');
    fwrite(fid,Y,'double');
    fclose(fid);
end
```

```
bemenetiFajlNev = '83.text'
[abra, T, Y] = f83_(bemenetiFajlNev);
```

8.1.4. 3D plot fájlolvasással

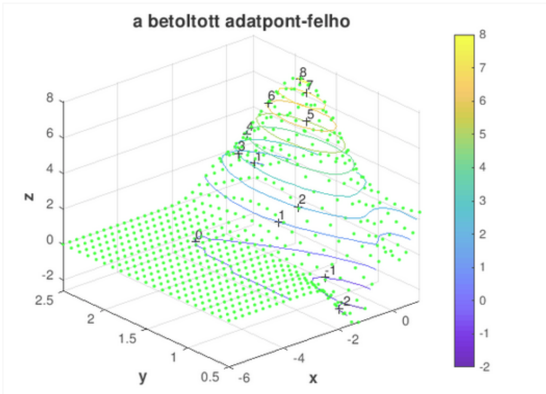
A feladat soran egy fajlbol beolvasott adatok alapjan 3D plotot kell elkesziteni, amely abranak bizonyos adatait egy strukturaban el kell tarolnod. A fuggveny amit keszitesz rendelkezik

- 1 bemeneti parameterrel --- egy fajlnev (`bemenetiFajlNev`),
- 2 visszateresi ertekkel --- abra-referencia + a kesobbiekben reszletezett struktura.

Feladatok:

- nyisd meg a parameterkent kapott szovegfajlt (ha MATLAB-ban dolgozol, a bemenetkent hasznalt `84.text` fajlt megtalalod a wiki-n, a csatolmanyok kozott);
- olvasd be (formazott szoveg olvasasa) belole a benne talalhato elso 6 lebegopontos szamot, melyek megmondjak a szukseges meshgridhez az adatokat (ezek rendre *xmin, xlepes, xmax, ymin, ylepes, ymax*);
- olvasd be az osszes tobbi szamot is (szinten lebegopontosak), melyek egy felulet pontjait adjak vissza (javaslat: 21 soros, "ismeretlen" oszlopszamu matrixba olvasd be, azaz az olvasasi meretnel [21, Inf]);
- zard be a fajlt;
- a beolvasott adatokat es a majdani tengelyfeliratokat egy strukturaban tarold le, es ezt hasznald az abrazolasnal, lehetoleg a **strukturad az alabbi mezonevekkel** rendelkezzen: `meshgridpoints`, `zdata`, `cim`, `xfelirat`, `yfelirat`, `zfelirat`;
- az abran szerepeljen: 3D pontfelho (stilus: zold "pont" marker) es feliratos 3D konturvonalak szinskalaval (`contour3 + clabel + colorbar`), tengelyfeliratok 12-es meretben felkoveren, abracim 14-es meretben, a latoszoget allitsd [-41, 28] azimuth es elevation szogekre;
- az elkeszitett strukturadat mentsd el binaris matlab archivumkent az alabbi neven: "`gyak8_f84_kimenet.mat`" (valamint ugyelj, hogy a strukturad neve megegyezzen a fuggvenyed 2. visszateresi ertekevel).

Ehhez hasonlo abrat kellene kapnod:



Solution:

```
function [abra, feluletStruktura] = f84_(bemenetiFajlNev)
    abra = figure;
    % a sturkturadnal lehetoleg az alabbi mezoneveket hasznald:
    % - meshgridpoints - a bemeneti fajlbol beolvasott elso 6 szam, ami a meshgrid
    ↪ parametereit jelenti
    % - zdata - a bemeneti fajlbol beolvasott tobbi szam, ami a terbeli feluleted magassagi
    ↪ pontjait jelenti
    % - cim, xfelirat, yfelirat, zfelirat - stringek, amiket majd a 3D abrad feliratozasanal
    ↪ hasznalj
    %pl: feluletStruktura.meshgridpoints = ...

    fid=fopen(bemenetiFajlNev);
    feluletStruktura.meshgridpoints=fscanf(fid,'%f',6);
    feluletStruktura.zdata=fscanf(fid,'%f',[21 Inf]);
    feluletStruktura.cim='a betoltott adatpont-felho';
    feluletStruktura.xfelirat='x';
    feluletStruktura.yfelirat='y';
    feluletStruktura.zfelirat='z';
    fclose(fid);



    [x,y] = meshgrid(feluletStruktura.meshgridpoints(1) : feluletStruktura.meshgridpoints(2)
    ↪ : feluletStruktura.meshgridpoints(3), feluletStruktura.meshgridpoints(4) :
    ↪ feluletStruktura.meshgridpoints(5) : feluletStruktura.meshgridpoints(6));
```

```
    plot3(x,y,feluletStruktura.zdata,'g.')
    xlabel(feluletStruktura.xfelirat,'FontSize',12,'FontWeight','bold');
    ylabel(feluletStruktura.yfelirat,'FontSize',12,'FontWeight','bold');
    zlabel(feluletStruktura.zfelirat,'FontSize',12,'FontWeight','bold');
    title(feluletStruktura.cim,'FontSize', 14);
    view(-41, 28);
    hold on;
    grid on;
    [C h]=contour3(x,y,feluletStruktura.zdata);
    colorbar;
    clabel(C,h);
end
```

```
bemenetiFajlNev =  '84.text';
[abra, feluletStruktura] = f84_(bemenetiFajlNev);
feluletStruktura
```

8.2. Szorgalmik

8.2.1. Zajos adatok

Rendelkezésünkre áll egy felszínt ábrázoló grafikon (surface plot) összes adata a `filename`nevű fájlban (A változó). Korrigáld a hibás x és y adatokat, valamint simítsd a z adatokat egy 11 szélességű csúszó ablak segítségével (egydimenziós adatok simítására van függvény a matlabban), és a korrigált adatokból hozz létre egy új surface plotot (B)!

Solution:

```
filename = 'zajos_adatok.mat';
B = zajszures(filename);
```

9. Gyakorlat

9.1. Házifeladatok

9.1.1. f91 - osztályzatok

Keszits egy fuggvenyt, melynek 0 bemeneti parametere es 1 visszateresi erteke van. A visszateresi ertek egy tablazat (`tbl`) legyen, melyet az alabbi modon tolts fel adatokkal:

- az elso oszlopban az ido vektor (`t`) ertekei legyenek tarolva, melyek $[0, \pi]$ intervallumon legyenek, 0.001-es lepeskozzel;
- a masodik oszlopban az alabbi formula szerinti ertekek legyenek: $y = 5sin(3e^t)$;
- a harmadik oszlopban az iment szamolt formula ertekei alapjan a szobeli osztalyzat alljon, a kerekites szabalyainak figyelembevetelevel (megjegyzes: a szinusz fv ertekkeszlete a $[-1, 1]$ intervallumbol kerul ki, a keplete igy a $[-5, 5]$-bol, ezeket vagy hagyd meg igy es a -4.83-as osztalyzat is elegtelent jelent, vagy vedd az abszolut erteket; hatarok: y>=4.5: jeles; 4.5>y>=3.5: jo; 3.5>y>= 2.5: kozepes; 2.5>y>=1.5: elegseges; 1.5>y: elegtelen --- ekezet nelkul! ---- erdemes lehet `round` utan `for`-ciklus es `if-elseif-end` strukturaval feltolteni az elemeket);
- az oszlopok neve rendre legyen: `ido`, `eredmeny`, `ertekeles`

Solution:

```
function tbl = f91()
    t=0:0.001:pi;
    y=5*sin(3*exp(t));
    osztalyzat={'elegtelen','elegtelen','elegseges','kozepes','jo','jeles'};
    c=osztalyzat(round(abs(y))+1);
    tbl=table(t',y',c','VariableNames',{'ido','eredmeny','ertekeles'});
end
```

```
tbl = f91();
```

9.1.2. f92 - országok

Keszits egy fuggvenyt, melynek 0 bemeneti parametere es 1 visszateresi erteke (t2 --- rendezett tablazat) van.

Lepesek:

- olvasd be a country_data.xls fajlt egy tablazatba (itt cody-n es a wiki mellekletek kozott is elerheto);
- a beolvasott tablazatot rendezd sorba (sortrows) a beszelt nyelvek szama alapjan csokkeno sorrendbe, ez a rendezett tablazat keruljon t2-be;
- a fuggveny irja ki a command window-ba a **terulet**, az **eletkor median**, a **nepesseg** valamint a **beszelt nyelvek szama**nak *minimum*at, *maximum*at, *median*jat a tiz legtobb nyelvet beszelo orszag eseten (segitseg: a dolt betuvel kert jellemzoket a summary utasitas onmagaban eloallitja; a vastagon szedett adatokat indexelessel kell kinyerned a mar rendezett tablazatbol --- elso tiz sora, 4-7 oszlopai annak);
- a fuggvenyed az egesz atrendezett tablazatot mentse ki **| karakterrel** elvalasztott (Delimiter opcio) szoveges fajlba, country_data_reordered.csv neven

Solution:

```
function [t2] = f92()
    t=readtable('country_data.xls', 'ReadVariableNames', true);
    t2=sortrows(t, 'Number_of_spoken_languages', 'descend');

    summary(t2(1:10,4:7))
    writetable(t2, 'country_data_reordered.csv', 'WriteVariableNames', true, 'FileType',
     ↪  'text', 'Delimiter', '|');
end
```

```
t2 = f92();
```

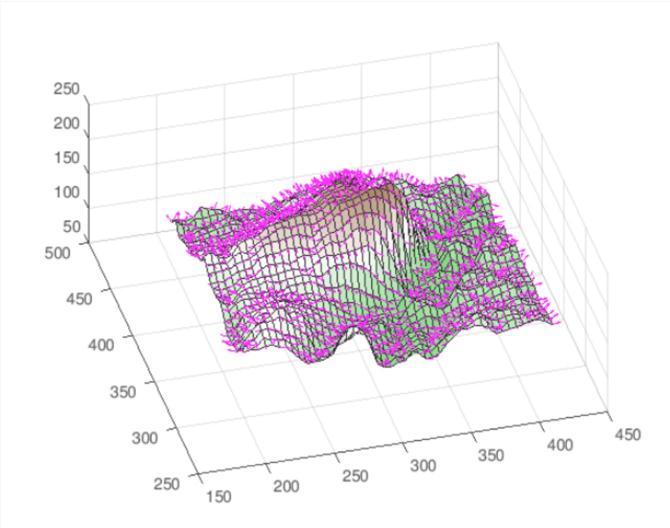
9.1.3. f93 - geodata

Egy csv-fajlban talalhato tablazat adatoszlopai alapjan feluleti abrat generalunk, az alabbiak szerint:

keszits egy 0 bemeneti parameterrel es 4 visszateresi ertekkel (`abra` handler, kiszamolt feluleti normalvektorok irany-komponensei `U`, `V`, `W`) rendelkezo fuggvenyt az alabbiak szerint:

- olvassuk be a tablazat fajlbol (`geodata.csv`) az adatokat oszlopcimekkel egyutt (`ReadVariableNames`-szel oszlopfejlecek, elvalaszto karakter a tabulator `'\t'`);
- a beolvasott adatokat tegyuk at az `X`, `Y`, `Z` matrixokba oszloponkent ugy, hogy mindegyik matrix 33x45-os meretu legyen;
- rajzoljuk ki a harom matrix alapjan a tarolt domborzat feluleti kepet (`surf`), majd szinezzuk be terkepszeru szinekkel (ehhez csinaltunk kulon colormap-et, amit betolteni a `terrain_cmap.mat` fajlbol lehet, tehat `load('terrain_cmap.mat');` es `colormap(terrain_cmap);`);
- szamoljuk ki a felulet normalvektorainak iranykomponenseit (`U`, `V`, `W` matrixok, ezeket a `surfnorm`-mal kapjuk meg), majd abrazoljuk is oket a feluleten (`quiver3`);
- allitsuk be a nezopontot -15-os azimuth, es +60-as elevation szogekre;
- mentsuk ki az abrat `print` segitsegevel 300 DPI-s felbontassal, `surf2.png` neven, png formatumban (`-dpng`).

Az alabbihoz hasonlo abrat szeretnenk kapni:



Solution:

```
function [abra, U, V, W] = f93()
    abra = figure;

    t=readtable('geodata.csv','ReadVariableNames',true);
    X=table2array(t(:,1));
    Y=table2array(t(:,2));
    Z=table2array(t(:,3));

    X=reshape(X,[33 45])
    Y=reshape(Y,[33 45])
    Z=reshape(Z,[33 45])
    surf(X,Y,Z);
    load('terrain_cmap.mat');
    colormap(terrain_cmap);
    [U V W]=surfnorm(X,Y,Z);
    hold on;
    quiver3(X,Y,Z,U,V,W);
    view([-15 60])
    print(abra,'surf2.png','-dpng','-zbuffer','-r300')
end
```

```
[abra, U, V, W] = f93();
```

9.2. Szorgalmik

9.2.1. Terhességi fehérjeszintek

Várandósság során az anya és a magzat egészségének nyomonkövetésére különböző vizsgálatokat végeznek. Egy jellemző vizsgálat különböző markerfehérjék (betegségek előrejelzésére alkalmas fehérjék) szintjének vizsgálata a vérben. Mivel ezek a fehérjeszintek fiziológiásan is változnak a terhesség során, ezért a nyers értékeket az adott terhességi korra jellemző értékekre számított mediánhoz viszonyítva adják meg. Ezt az értéket nevezik MoM (multiple of the median) értéknek.

Készíts egy olyan egy bemeneti paraméterrel (`filename`) és négy kimeneti paraméterrel (`t`, `mom_f1`, `mom_f2`, `mom_f3`) rendelkező függvényt, ami `filename` .csv fájlból beolvassa az adatokat egy táblázatba (`t`) úgy, hogy a sorok nevei a fájl első oszlopában szereplő azonosítók legyenek, az oszlopok nevei pedig a fájl első sorában szereplő nevek! A fájlban pontosvessző az elválasztókarakter.

A méréseket 30 egészséges és 14 beteg várandós anyán végezték (`csoport` oszlop). A nyers fehérjekoncentráció-adatokból (`feherje_1_koncentracio`, `feherje_2_koncentracio`, `feherje_3_koncentracio` oszlopok) számítsd ki az MoM értékeket (`mom_f1`, `mom_f2`, `mom_f3`). Ügyelj arra, hogy a medián értékeket az egészséges anyák értékei alapján számolják!

Solution:

```
[t, mom_f1, mom_f2, mom_f3] = MoM('protein_levels.csv')
```