

DXF FÁJL KÉSZÍTÉSE MATLABBAL, GRAPHICAL USER INTERFACE (GUI)

DXF VONALLÁNC KÉSZÍTÉSE KOORDINÁTÁKBÓL

Feladat egy szöveges állományban tárolt koordinátajegyzék beolvasása, majd ebből DXF vonallánc készítése, AutoCAD-be történő beolvasása.

Az xydata.txt állományban pontok koordinátái vannak eltárolva.

```
6.7084537e+001 3.7540498e+001
6.7616798e+001 3.7562482e+001
6.7986302e+001 3.7412988e+001
```

...

Ezek a pontok egy folytonos vonal pontjai. Olvassuk be őket MATLAB-ba, majd készítsünk belőle egy alap DXF fájlt, ahol a pontok vonallánccal vannak összekötve!

A feladat első része nagyon egyszerű, mivel a fájlban azonos típusú adatok vannak azonos sorhosszakkal, így a megoldás történhet az egyszerű **load** paranccsal, majd válogassuk szét őket x és y koordinátákra!

```
> page_screen_output(0); % csak Octave-ban kell
> clc; clear all;
> load 'xydata.txt'
> x = xydata(:,1)
> y = xydata(:,2)
> plot(x,y,'r+')
```

Ezután jön a DXF fájl készítése, amihez ismernünk kell a DXF fájlok szerkezetét (lásd mellékelt DXF formátum segédlet!).

Az adott fájlban egy darab folytonos vonal pontjai találhatóak. Ha ezt AutoCAD-ben szeretnénk megjeleníteni, akkor választhatjuk a vonal, vagy a vonallánc parancsot is. A vonal esetében két pontot kell megadnunk, kezdőpontot és végpontot. Egy folytonos vonal esetében ez azt jelentené, hogy a közbülső pontokat duplán kellene megadni, egyszer, mint az előző vonal végpontja, majd mint a következő vonal kezdőpontja. Ilyenkor a vonallánc célszerűbb, mert akkor elég egyszer megadni minden pontot, csomópontként.

Az AutoCAD DXF fájlokban a sorok kettésével összetartoznak, először egy kód van, hogy mi fog következni (pl. szekció eleje, vége, vonallánc típusú rajzi elem, rétegmegadás, szín stb.), utána pedig a hozzá tartozó érték, vagy az érték kódja). A fájl első része teljesen általános, szekció eleje, rajzi elemek szekció. Meg kell adni a rajzi elem típusát is (vonallánc), hogy milyen rétegre kerüljön, utána megadhatjuk a színét (nem kötelező), végül következnek a csomópontok 66-os kóddal az elején, majd VERTEX megnevezéssel az egyes csomópontok. A fájl végén jön egy csomópontok vége jel, majd szekció vége, fájl vége.

Példa DXF vonallánra (mindig két összetartozó érték van: következő művelet kódja, majd az értéke):

```

0          Új művelet
SECTION   Szekció kezdődik
2         Szekció típusa
ENTITIES  Rajzi elemek
0         Rajzi elem típusa
POLYLINE  Vonallánc
8         Rétegmegadás
vonall    'vonall' nevű réteg
62        Szín megadása
1         1-piros (2-sárga, 3-zöld, 4-v.kék, 5-s.kék, 6-lila, 7-fekete, 8-szürke)
66        Csomópontok következnek1
1         (értéke mindig 1)
0         Új művelet
VERTEX   Új csomópont
8         Rétegmegadás2
vonall    'vonall' nevű réteg
10        X koordináta (kezdőpont)
32.65     értéke
20        Y koordináta (kezdőpont)
67.89     értéke
...
0         Új művelet
SEQEND    Csomópontok megadásának vége
0         Új művelet
ENDSEC    Szekció vége
0         Új művelet
EOF       Fájl vége

```

Először nyissuk meg írásra a fájlt, amibe menteni szeretnénk a vonalláncot, majd írjunk a fájl elejére egy megjegyzést, hogy ez egy MATLAB-bal készült vonallánc. A DXF-ben a megjegyzés kódja a 999, az AutoCAD a 999 után következő sort nem olvassa be, azt megjegyzésnek tekinti. MATLAB-ban ezt az **fprintf** használatával írhatjuk ki a fájlba!

```

> fajlnev='xypolyline.dxf'; % Fájlnév megadása
> fid=fopen(fajlnev,'w'); % Fájl megnyitása írásra
> % Megjegyzés AutoCAD dxf fájlban: 999
> fprintf(fid,'999\r\nPolyline with matlab\r\n');

```

Be kell iktatnunk sorvége jeleket is a fájlba **\r\n**, itt vegyük figyelembe, hogy a sorvége karaktert különböző operációs rendszerek máshogy jelölik. A sor vége jel Windows esetében: **\r\n**, Mac (OS 9-) esetében **\r**, Unix/Linux esetében: **\n**. Az adott programtól

¹ A 66-os kód jelöli, hogy csomópontok következnek, ez nem minden CAD programnál kötelező elem, de a tanszéki AutoCAD MAP 2008 változatnál igen.

² Lásd mint előbb. Nem minden CAD programnál kötelező minden csomópontnál megadni a réteg nevét is, de az AutoCAD MAP 2008 esetében igen.

is függ, hogy mit tud értelmezni sor végének. Sok olyan program van, ami mind a `\n` mind a `\r\n` jelet elfogadja. Ilyen például a Wordpad, de ha Notepad-et használunk, akkor az csak a `\r\n` jelet tudja értelmezni. Ugyanígy van olyan CAD program, ami elfogadja mindkét jelölést, de van, ami csak a `\r\n`-t Windows alatt. Miután mindig két összetartozó érték van két különböző sorban, egyszerűsíthetjük a dolgot, ha készítünk egy anonim függvényt, ami a két érték után berak egy-egy sorvége jelet, hogy jobban áttekinthető legyen a programunk a későbbiekben.

```
> % Függvény összetartozó értékpárok két sorba írásához
> dxf = @(a,b) [a '\r\n' b '\r\n']
> dxf('10', '20.124') % tesztelés: 10\r\n20.124\r\n
```

Ezután jöhetnek a tényleges rajzi elemek.

```
> fprintf(fid, [dxf('0', 'SECTION'), dxf('2', 'ENTITIES')]);
> fprintf(fid, [dxf('0', 'POLYLINE'), dxf('8', 'vonal')]);
> fprintf(fid, [dxf('62', '1'), dxf('66', '1')]);
> % A csomópontok adatai egyenként ciklusban
> for i=1:length(x)
>   fprintf(fid, [dxf('0', 'VERTEX'), dxf('8', 'vonal')]);
>   xs = num2str(x(i)); ys = num2str(y(i));
>   fprintf(fid, [dxf('10', xs), dxf('20', ys)]);
> end
> % lezárások
> fprintf(fid, [dxf('0', 'SEQEND'), dxf('0', 'ENDSEC'), dxf('0', 'EOF')]);
> fclose(fid); % Fájll lezárása
> type('xy polyline.dxf') % Nézzük meg mi van a fájlban
```

Ezután próbáljuk megnyitni valamilyen CAD-es programmal a fájlunkat (pl. AutoCAD, AutoCAD Map, BricsCAD). Ha minden jól megy és nem írtunk el semmit, akkor végre megnézhetjük, hogy mi is volt a fájlban! Méghozzá piros színnel, mivel a szín beállításánál 1-es kódot (piros) adtunk meg. Mentsük is el az állományunkat valamilyen más néven, de szintén ASCII DXF fájlként a programból! Nyissuk meg egy szövegszerkesztővel a két fájlt (pl. notepad++). Mit tapasztalunk? Az általunk létrehozott fájl kb. 300 sorból áll, a CAD-es szoftverből elmentett változat pedig több ezerből, pedig semmit nem változtattunk a rajzon!

 AZ ÁLTALUNK KÉSZÍTETT ALAP PROGRAM EGYBEN

```

> page_screen_output(0); % csak Octave-ban kell
> clc; clear all;
>
> load 'xydata.txt'
> x = xydata(:,1)
> y = xydata(:,2)
> plot(x,y,'r+')
>
> fajlnev='xypolyline.dxf'; % Fájlnév megadása
> fid=fopen(fajlnev,'w'); % Fájl megnyitása írásra
> % Megjegyzes AutoCAD dxf fájlban: 999
> fprintf(fid,'999\r\nPolyline with matlab\r\n');
>
> % Függvény összetartozó értékpárok két sorba írásához
> dxf = @(a,b) [a '\r\n' b '\r\n']
> dxf('10','20.124') % tesztelés: 10\r\n20.124\r\n
>
> fprintf(fid, [dxf('0','SECTION'), dxf('2','ENTITIES')]);
> fprintf(fid, [dxf('0','POLYLINE'), dxf('8','vonal')]);
> fprintf(fid, [dxf('62','1'), dxf('66','1')]);
> % A csomópontok adatai egyenként ciklusban
> for i=1:length(x)
>   fprintf(fid, [dxf('0','VERTEX'), dxf('8','vonal')]);
>   xs = num2str(x(i)); ys = num2str(y(i));
>   fprintf(fid, [dxf('10', xs), dxf('20', ys)]);
> end
> % lezárások
> fprintf(fid, [dxf('0','SEQEND'), dxf('0','ENDSEC'), dxf('0','EOF')]);
> fclose(fid); % Fájl lezárása
  
```

 GRAPHICAL USER INTERFACE (GUI) – ADATOK BEOLVASÁSA

Most egy kicsit tegyük interaktívabbá a programunkat néhány grafikus ablakkal, amikben a felhasználó adhat meg néhány opciót, például, hogy milyen színű legyen a vonallánc, milyen fájlból olvassuk be az adatokat, milyen néven és hová mentjük az elkészült állományt, és mi legyen az AutoCAD-es réteg neve, amire a rajz kerülni fog!

A grafikus felhasználói felület adatbeviteli parancsait a program elejére írjuk be, a **load** parancs elé!

A következőkben a felhasználó válassza ki a betölteni kívánt koordináta állományt az **uigetfile** paranccsal, majd ezt töltjük is be!

```

> % GUI - koordináta fájl választás
> [koordfajl koordmappa] = uigetfile('*.txt',...
>   'Válassza ki a beolvasni kívánt koordinata fájlt!')
> koordfajl = [koordmappa koordfajl]
> xydata = load(koordfajl)
  
```

Az ezt követő **load** parancsot ezután ki is törölhetjük, vagy kommentelhetjük egy % jellel.

Korábban már használtuk a menu parancsot, ahol egy listából lehet kiválasztani elemeket, és a kiválasztott elem sorszáma lesz a visszaadott érték. Ezt célszerű

használni a színek megadásához, ott ugyanis minden színnek van megfelelő kódja (1-piros, 2-sárga, 3-zöld, 4-világoskék, 5-sötétkék, 6-lila, 7-fekete, 8-szürke). Ha a **menu** parancsban ilyen sorrendben adjuk meg őket, akkor a megfelelő kódot kapjuk vissza. Ezt utána már csak szöveggé kell alakítanunk és később használni a vonallánc kiírásához.

```
> % GUI - szín kiválasztása menüből
> % színek kódjai: 1-piros ,2-sárga, 3-zöld, 4-v.kék, 5-s.kék, 6-lila,
> % 7-fekete, 8-szürke
> szin = menu('Adja meg a vonallanc szinet!', 'piros', 'sarga', ...
>           'zold', 'vilagoskek', 'sotetkek', 'lila', 'fekete', 'szurke')
> szin = num2str(szin)
```

Utána kérdezzük meg a felhasználótól, hogy mi legyen a dxf fájl, és azon belül a réteg neve, amire a rajz kerülni fog. Ezt az **inputdlg** paranccsal tehetjük meg. Itt nem egy menüből kell kiválasztani a megfelelőt, hanem szöveges értéket/értékeket vár a program. Ha több dolgot szeretnénk megkérdezni, akkor egy kapcsos zárójelekből {'kerdes1','kerdes2'} álló cellatömbben kell megadni a kérdéseket. A kimenet mindenképpen egy cellatömb lesz, aminek az elemeit szintén kapcsos zárójelekkel kérdezhethetjük le! Ha csak egy értéket kérünk a felhasználótól, akkor a következőképpen tehetjük meg:

```
> nev = inputdlg('Fajlnev: ')
```

Ez a legegyszerűbb megadás, ilyenkor az ablak címe az alapértelmezett 'input dialog' lesz. Ha már kettő adatot kérdezzük a felhasználótól és címet is akarunk adni az ablaknak, akkor a következőképpen tehetjük meg!

```
> nevek = inputdlg({'Fajlnev: ', 'Milyen retegre keruljon a rajz? '}, ...
> 'Adja meg az adatokat!')
```

Ha azt szeretnénk, hogy legyen valamilyen alapértelmezett érték is rögtön beírva, akkor először meg kell adni, hogy hány sorba szeretnénk kiírni az alapértelmezett értékeket (legyen ez most 1) és egy cellatömbben meg kell adni a kívánt szövegeket.

```
> % GUI - fájlnev, rétegnév megadása
> nevek = inputdlg({'Fajlnev: ', 'Milyen retegre keruljon a rajz? '}, ...
>           'Adja meg az adatokat!', 1, {'proba1', 'reteg1'})
> fajlnev = [nevek{1} '.dxf']
> retteg = nevek{2}
```

Alapértelmezett esetben az új fájl az aktuális munkakönyvtárba kerül, amit a **pwd** paranccsal lehet lekérdezni. A felhasználó az **uigetdir** paranccsal választhat másik könyvtárat az aktuális könyvtár helyett, ahová menteni szeretné a DXF fájlt. A kiindulási könyvtár az aktuális munkakönyvtár legyen (pwd), és adjunk címet is az ablaknak! Ez utóbbi két adat opcionális, az uigetdir parancs önmagában is működik.

```
> % GUI - könyvtár kiválasztása, fájlnev pontosítása
> mappa = uigetdir(pwd, 'Adja meg a konyvtarat, ahova menteni szeretne!');
> fajlnev = [mappa '\' fajlnev]
```

A felhasználó által történő fájlnev megadása után ki is törölhetjük/kommentelhetjük a korábbi **fajlnev='xypolyline.dxf'**; parancsot.

Ahhoz, hogy a színt és rétegnevet is használni tudjuk, cseréljük ki az összes **dxs('8', 'vona1')** szöveget **dxs('8', retteg)**-re és a **dxs('62', '1')** szöveget **dxs('62', szin)** szövegre!

 AZ ÁLTALUNK KÉSZÍTETT VÉGLEGES PROGRAM GUI HASZNÁLATÁVAL

```

> page_screen_output(0); % csak Octave-ban kell
> clc; clear all;
>
> % GUI - koordináta fájl választás
> [koordfajl koordmappa] = uigetfile('*.txt',...
>   'Valassza ki a beolvasni kívánt koordinata fajlt!')
> koordfajl = [koordmappa koordfajl]
> xydata = load(koordfajl)
>
> % GUI - szín kiválasztása menüből
> % színek kódjai: 1-piros ,2-sárga, 3-zöld, 4-v.kék, 5-s.kék, 6-lila,
> % 7-fekete, 8-szürke
> szin = menu('Adja meg a vonallanc szinet!', 'piros', 'sarga', ...
>   'zold', 'vilagoskek', 'sotetkek', 'lila', 'fekete', 'szurke')
> szin = num2str(szin)
> % GUI - fájlnev, rétegnév megadása
> nevek = inputdlg({'Fajlnev: ', 'Milyen retegre kerüljön a rajz? '}, ...
>   'Adja meg az adatokat!', 1, {'proba1', 'reteg1'})
> fajlnev = [nevek{1} '.dxf']
> reteg = nevek{2}
>
> % GUI - könyvtár kiválasztása, fájlnev pontosítása
> mappa = uigetdir(pwd, 'Adja meg a könyvtarat, ahova menteni szeretne!');
> fajlnev = [mappa '\' fajlnev]
>
> x = xydata(:,1)
> y = xydata(:,2)
> plot(x,y, 'r+')
>
> fid=fopen(fajlnev, 'w'); % Fájl megnyitása írásra
> % Megjegyzes AutoCAD dxf fájlban: 999
> fprintf(fid, '999\r\nPolyline with matlab\r\n');
>
> % Függvény összetartozó értékpárok két sorba írásához
> dxf = @(a,b) [a '\r\n' b '\r\n']
> dxf('10', '20.124') % tesztelés: 10\r\n20.124\r\n
>
> fprintf(fid, [dxf('0', 'SECTION'), dxf('2', 'ENTITIES')]);
> fprintf(fid, [dxf('0', 'POLYLINE'), dxf('8', reteg)]);
> fprintf(fid, [dxf('62', szin), dxf('66', '1')]);
> % A csomópontok adatai egyenként ciklusban
> for i=1:length(x)
>   fprintf(fid, [dxf('0', 'VERTEX'), dxf('8', reteg)]);
>   xs = num2str(x(i)); ys = num2str(y(i));
>   fprintf(fid, [dxf('10', xs), dxf('20', ys)]);
> end
> % lezárások
> fprintf(fid, [dxf('0', 'SEQEND'), dxf('0', 'ENDSEC'), dxf('0', 'EOF')]);
> fclose(fid); % Fájl lezárása

```