



Μάθημα: Μέθοδοι Επίλυσης με Η/Υ

Τετάρτη, 9/1/2019

Διδάσκοντες: Ν.Δ. Λαγαρός (Αν. Καθηγητής), Α. Στάμος (ΕΔΙΠ), Χ. Φραγκουδάκης (ΕΔΙΠ)
Αμβ. Σαββίδης

Παραδείγματα για την 11^η παράδοση – Δομές και ανομοιογενή μητρώα

1. Υπολογισμός μήκους οδού

Να συνταχθεί script το οποίο να διαβάσει το αρχείο odos.syn που περιέχει τις συντεταγμένες x, y τους άξονα οδού και να υπολογίζει και να γράφει το συνολικό μήκος της οδού. Σε κάθε σειρά το αρχείο περιέχει όνομα σημείου (κείμενο), συντεταγμένη x (πραγματικός αριθμός) και συντεταγμένη y (πραγματικός αριθμός) χωρισμένα με ένα ή περισσότερα κενά:

K0	19655.273	23846.76
ΧΠ1	19658.740	23853.6
ΧΠ2	19659.077	23854.3
....		

Το αρχείο δίνεται στην ιστοσελίδα mycourses.ntua.gr.

Ανάλυση

α) Αν διαβαστούν οι συντεταγμένες x_i και y_i η απόσταση μεταξύ διαδοχικών σημείων δίνεται από το πυθαγόρειο θεώρημα:

$$d_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

Το άθροισμα όλων των αποστάσεων δίνει το μήκος της οδού. Η συνάρτηση `hypot(dx, dy)` υπολογίζει το πυθαγόρειο θεώρημα.

Το αρχείο θα διαβαστεί με την συνάρτηση `textscan()` η οποία επιστρέφει τις 3 στήλες του αρχείου ως στοιχεία ενός ανομοιογενούς μητρώου με 3 στοιχεία.

Κώδικας

```
clear; clc; close all;  
fr = fopen('odos.syn', 'r');  
a = textscan(fr, '%s %f %f');  
x = a{2};  
y = a{3};  
d = hypot( x(2:end)-x(1:end-1), y(2:end)-y(1:end-1) );  
disp(sum(d));  
plot(x, y);  
axis equal;
```

Αν το αρχείο έχει στην πρώτη γραμμή τίτλο:

Σημείο	X	Y
K0	19655.273	23846.76
ΧΠ1	19658.740	23853.6
ΧΠ2	19659.077	23854.3
....		

Η συνάρτηση `textscan()` μπορεί να αγνοήσει την πρώτη γραμμή:

```
clear; clc; close all;  
fr = fopen('odos2.syn', 'r');  
a = textscan(fr, '%s %f %f', 'headerlines', 1);
```

```

x = a{2};
y = a{3};
d = hypot( x(2:end)-x(1:end-1), y(2:end)-y(1:end-1) );
disp(sum(d));
plot(x, y);
axis equal;

```

Αν για κάποιο λόγο πρέπει να διαβαστεί και ο τίτλος, μπορεί να διαβαστεί με τη συνάρτηση `fgetl()` και στη συνέχεια η συνάρτηση `textscan()` θα διαβάσει από εκεί και κάτω (από τη δεύτερη σειρά και μετά):

```

clear; clc; close all;
fr = fopen('odos2.syn', 'r');
tit = fgetl(fr);
disp(tit);
a = textscan(fr, '%s %f %f');
x = a{2};
y = a{3};
d = hypot( x(2:end)-x(1:end-1), y(2:end)-y(1:end-1) );
disp(sum(d));
plot(x, y);
axis equal;

```

2. Αποθήκευση μητρώων δυσκαμψίας

Το μητρώο δυσκαμψίας $[K]$ μίας ράβδου είναι:

$$[K] = k^* \cdot \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix}$$

όπου $c=\cos(\theta)$, $s=\sin(\theta)$. Τα k^* και θ (μοίρες) δίνονται σε αρχείο `bars.txt` που δίνεται στην ιστοσελίδα `mycourses.ntua.gr`. Να φτιαχτεί script το οποίο για κάθε γραμμή του αρχείου `bars.txt`, που αντιστοιχεί σε διαφορετική ράβδο, να αποθηκεύει τα k^* και θ και το μητρώο σε δομές. Να γίνει το ίδιο με αποθήκευση σε ανομοιογενή πίνακα.

Λύση με δομές

```

clear; clc; close all;
a = dlmread('bars.txt');
for i=1:size(a,1)
    b(i).ks = a(i, 1);
    b(i).theta = a(i, 2);
    c = cosd(b(i).theta);
    s = sind(b(i).theta);
    M = b(i).ks .* [c.^2 c.*s
                   c.*s s.^2];
    b(i).K = [ M -M
              -M M];
end
celldisp(struct2cell(b));

```

Λύση με ανομοιογενή πίνακα

```

clear; clc; close all;
a = dlmread('bars.txt');
for i=1:size(a,1)
    ks = a(i, 1);
    theta = a(i, 2);

```

```

c = cosd(theta);
s = sind(theta);
M = ks .* [c.^2 c.*s
            c.*s s.^2];
b{i, 1} = [ M -M; -M M];
b{i, 2} = ks;
b{i, 3} = theta;
end
celldisp(b);

```

3. Αναίρεση ανάγνωσης

Να συνταχθούν συναρτήσεις fopen1(), fclose(1), fgetl1(), fungetl1(), feof1() οι οποίες να ανοίγουν/κλείνουν/διαβάζουν ένα αρχείο με την πρόσθετη δυνατότητα ότι μπορεί να αναιρείται η ανάγνωση μίας γραμμής.

Λύση

Μία λύση είναι όταν διαβάζεται μία γραμμή από το αρχείο μέσω της συνάρτησης fgetl1(), αυτή να αποθηκεύεται σε μία μεταβλητή. Αν ο χρήστης αναιρέσει την ανάγνωση μέσω της συνάρτησης fungetl1(), τότε η επόμενη κλήση της συνάρτησης fgetl1() θα επιστρέφει την αποθηκευμένη γραμμή, αντί να διαβάσει την επόμενη γραμμή από το αρχείο. Συνεπώς πρέπει να υπάρχει επιπλέον ένας κωδικός σε μία άλλη μεταβλητή, ο οποίος θα καθορίζει αν η συνάρτηση fgetl1() θα επιστρέφει την αποθηκευμένη γραμμή, ή αν θα διαβάζει και θα επιστρέφει μία νέα γραμμή από το αρχείο.

Αυτές οι μεταβλητές καθώς επίσης και το ανοιγμένο αρχείο (fid) θα ομαδοποιηθούν σε μία δομή fid1. Επειδή η δομή fid1 αλλάζει μέσα σε κάθε συνάρτηση, θα επιστρέφεται ως όρισμα εξόδου από την κάθε συνάρτηση.

Κώδικας

```

function [ fid1 ] = fopen1(fn)
%Open a file with unread capabilities
fid1.fid = fopen(fn, 'r');
fid1.isunread = false;
fid1.prevline = '';
end

function [ fid2, dline ] = fgetl1(fid1)
%Read a line of the file, or return the previous line.
fid2 = fid1;
if fid2.isunread
    dline = fid2.prevline;
    fid2.isunread = false;
else
    dline = fgetl(fid2.fid);
    fid2.prevline = dline;
end
end

function [ fid2 ] = fungetl1(fid1)
%Unread previously read line
fid2 = fid1;
if fid2.isunread
    error('Only 1 line can be unread');
else
    fid2.isunread = true;
end
end

function [r] = feof1( fid1 )
%Return true if end of file.
r = feof(fid1.fid);

```

```

end

function [ ] = fclose1( fid1 )
%Close the file.
fclose(fid1.fid);
end

```

Παράδειγμα χρήσης

Το παρακάτω script γράφει σε ένα αρχείο 4 γραμμές με περιεχόμενα 'aaa', 'bbb', 'ccc', 'ddd' χρησιμοποιώντας τις έτοιμες συναρτήσεις του Matlab, και στη συνέχεια τις διαβάζει μέσω των νέων συναρτήσεων. Η δεύτερη γραμμή διαβάζεται 2 φορές μέσω της αναίρεσης ανάγνωσης.

```

clear; clc; close all;
fid = fopen('temp.txt', 'w');
fprintf(fid, '%s\n', 'aaa');
fprintf(fid, '%s\n', 'bbb');
fprintf(fid, '%s\n', 'ccc');
fprintf(fid, '%s\n', 'ddd');
fclose(fid);

fid = fopen1('temp.txt');
[fid, dline] = fgetl1(fid);
disp(dline);
[fid, dline] = fgetl1(fid);
disp(dline);
fid = fungetl1(fid);
[fid, dline] = fgetl1(fid);
disp(dline);
[fid, dline] = fgetl1(fid);
disp(dline);
fclose1(fid);
%Close the file.

```

Το script δίνει τα αποτελέσματα:

```

aaa
bbb
bbb
ccc

```

4. Ανάγνωση τοπογραφικών μετρήσεων

Σε αρχείο κειμένου υπάρχουν μετρήσεις τοπογραφικών στάσεων. Από κάθε στάση έχουν μετρηθεί αποστάσεις και οριζόντιες γωνίες προς ένα ή περισσότερα σημεία, ως φαίνεται παρακάτω:

Σ2		
Σ1	83.905	2.77945
Σ3	68.774	228.04833
Σ3	68.775	28.04775
Σ1	83.929	202.77937
Σ2		
Σ1	83.939	99.54290
Σ3	68.755	324.81135
Σ3	68.747	124.81021
Σ1	83.922	299.54238
Σ3		
Σ2	68.780	398.28055
Σ4	66.344	142.90860
Σ4	66.340	342.91023
Σ2	68.743	198.28167

Κάθε στάση μπορεί να έχει μετρηθεί μία φορά (πχ στάση Σ3) ή πολλές φορές (πχ στάση Σ2).

1. Να συνταχθεί συνάρτηση σε Matlab η οποία να διαβάζει όλες τις στάσεις και τα σημεία κάθε στάσης, να ομαδοποιεί όλες τις φορές που έχει μετρηθεί μία στάση σε ανομοιογενές μητρώο ή

μητρώο από δομές, και τέλος να επιστρέφει ένα ανομοιογενές που μητρώο που περιέχει όλες τις ομάδες.

2. Να ελέγξετε τη συνάρτηση με script που διαβάσει το αρχείο od01.ana που βρίσκεται στην ιστοσελίδα mycourses.

Λύση

Οι μετρήσεις μίας στάσης διαχωρίζονται από τις μετρήσεις της επόμενης στάσης από μία σειρά που περιέχει μόνο το όνομα της στάσης (δεν περιέχει απόσταση ούτε γωνία). Συνεπώς συντάσσουμε συνάρτηση που διαβάζει το όνομα της στάσης, και στη συνέχεια σημεία (όνομα, απόσταση και γωνία) μέχρι να βρει μία σειρά που περιέχει όνομα σημείου αλλά όχι απόσταση ούτε γωνία (ή μέχρι το τέλος του αρχείου). Αυτό το σημείο είναι η ονομασία της επόμενης στάσης και για αυτό **ανααιρούμε** την ανάγνωση αυτού το σημείου (όπως στην προηγούμενη άσκηση). Η συνάρτηση επιστρέφει μία δομή με το όνομα της στάσης και τα σημεία της στάσης:

```
function [ fid, sta] = readstation(fid)
%Read one station and its points.
%   ost is the name of the station.
%   osm are the names of the points.
%   d are the distances.
%   a are the horizontal angles.
[fid, dline] = fgetl1(fid);
sta.ost = char(dline);    % Erase trailing blanks
n = 0;
while true
    if feof1(fid)
        break
    end
    [fid, dline] = fgetl1(fid);
    dline = char(dline);    % Erase trailing blanks
    k = find(dline==' ');
    if isempty(k)           % There is no distance, nor angle
        fid = fungetl1(fid);
        break;
    end
    n = n + 1;
    sta.osm{n} = dline(1:k(1)-1);
    temp = sscanf(dline(k(1):end), '%f');
    if length(temp) ~= 2
        error(['Error reading point ' osm{n}]);
    end
    sta.d(n) = temp(1);
    sta.a(n) = temp(2);
end
end
```

Στη συνέχεια συντάσσουμε μία συνάρτηση που διαβάζει όλες τις στάσεις, τις χωρίζει σε ομάδες με το ίδιο όνομα (σε μητρώο από δομές), και επιστρέφει ένα ανομοιογενές μητρώο με όλες τις ομάδες:

```
function [ group ] = readgroups(fid)
%Read all stations and return groups of same station
n = 0;
while ~feof1(fid)
    n = n + 1;
    [fid, stas(n)] = readstation(fid);
end
k = 0;
i = 1;
for j=2:n
    if ~strcmp(stas(j).ost, stas(i).ost)
        k = k + 1;
        group{k} = stas(i:j-1);
        i = j;
    end
end
group{k} = stas(i:n);
```

```

        end
    end
    k = k + 1;
    group{k} = stas(i:end);
end

```

Έλεγχος

Το παρακάτω script διαβάζει το αρχείο od01.ana και τυπώνει όλες τις ομάδες:

```

clear; clc; close all;
fid = fopen('od01.ana');
group = readgroups(fid);
for k=1:length(group)
    fprintf('group %d\n', k);
    fprintf('-----\n');
    stas = group{k};
    for i=1:length(stas)
        stas(i)
    end
end
end

```

Το script δίνει τα παρακάτω αποτελέσματα:

```

group 1
-----
ans =
    ost: 'Σ2'
    osm: {'Σ1' 'Σ3' 'Σ3' 'Σ1'}
         d: [83.9050 68.7740 68.7750 83.9290]
         a: [2.7795 228.0483 28.0478 202.7794]
ans =
    ost: 'Σ2'
    osm: {'Σ1' 'Σ3' 'Σ3' 'Σ1'}
         d: [83.9390 68.7550 68.7470 83.9220]
         a: [99.5429 324.8114 124.8102 299.5424]
ans =
    ost: 'Σ2'
    osm: {'Σ1' 'Σ3' 'Σ3' 'Σ1'}
         d: [83.9440 68.7390 68.7360 83.9390]
         a: [202.6039 27.8718 227.8713 2.6031]
ans =
    ost: 'Σ2'
    osm: {'Σ1' 'Σ3' 'Σ3' 'Σ1'}
         d: [83.9150 68.7810 68.7370 83.9480]
         a: [304.8883 130.1554 330.1543 104.8868]
group 2
-----
ans =
    ost: 'Σ3'
    osm: {'Σ2' 'Σ4' 'Σ4' 'Σ2'}
         d: [68.7800 66.3440 66.3400 68.7430]
         a: [398.2806 142.9086 342.9102 198.2817]
ans =
    ost: 'Σ3'
    osm: {'Σ2' 'Σ4' 'Σ4' 'Σ2'}
         d: [68.7370 66.3500 66.3240 68.7350]
         a: [97.6189 242.2488 42.2488 297.6192]
ans =
    ost: 'Σ3'

```

```

    osm: {'Σ2'  'Σ4'  'Σ4'  'Σ2'}
    d: [68.7840 66.3070 66.3250 68.7860]
    a: [200.1635 344.7931 144.7915 0.1635]
ans =
    ost: 'Σ3'
    osm: {'Σ2'  'Σ4'  'Σ4'  'Σ2'}
    d: [68.7380 66.3050 66.3440 68.7440]
    a: [296.7780 41.4068 241.4059 96.7778]
group 3
-----
ans =
    ost: 'Σ4'
    osm: {'Σ3'  'Σ5'  'Σ5'  'Σ3'}
    d: [66.3350 74.1480 74.1330 66.3430]
    a: [0.6856 283.2009 83.2009 200.6845]
ans =
    ost: 'Σ4'
    osm: {'Σ3'  'Σ5'  'Σ5'  'Σ3'}
    d: [66.3060 74.1220 74.1400 66.3260]
    a: [95.0632 377.5788 177.5807 295.0645]
ans =
    ost: 'Σ4'
    osm: {'Σ3'  'Σ5'  'Σ5'  'Σ3'}
    d: [66.3260 74.1740 74.1700 66.3310]
    a: [203.7977 86.3140 286.3132 3.7975]
ans =
    ost: 'Σ4'
    osm: {'Σ3'  'Σ5'  'Σ5'  'Σ3'}
    d: [66.3150 74.1230 74.1750 66.3540]
    a: [298.7238 181.2402 381.2391 98.7232]
group 4
-----
ans =
    ost: 'Σ5'
    osm: {'Σ4'  'T6'  'T6'  'Σ4'}
    d: [74.1370 41.6530 41.6530 74.1520]
    a: [1.4692 133.5320 333.5322 201.4687]
ans =
    ost: 'Σ5'
    osm: {'Σ4'  'T6'  'T6'  'Σ4'}
    d: [74.1510 41.6430 41.6490 74.1730]
    a: [102.5513 234.6137 34.6144 302.5523]
ans =
    ost: 'Σ5'
    osm: {'Σ4'  'T6'  'T6'  'Σ4'}
    d: [74.1600 41.6310 41.6520 74.1680]
    a: [198.2967 330.3595 130.3600 398.2976]
ans =
    ost: 'Σ5'
    osm: {'Σ4'  'T6'  'T6'  'Σ4'}
    d: [74.1390 41.6240 41.6770 74.1560]
    a: [304.0882 36.1518 236.1515 104.0891]
group 5
-----
ans =
    ost: 'T6'

```

```
    osm: {'Σ5'  'Σ7'  'Σ7'  'Σ5'}
      d: [41.6300 64.8730 64.8920 41.6390]
      a: [396.8237 250.2098 50.2114 196.8250]
ans =
    ost: 'T6'
    osm: {'Σ5'  'Σ7'  'Σ7'  'Σ5'}
      d: [41.6600 64.8890 64.9230 41.6360]
      a: [101.5852 354.9712 154.9714 301.5855]
ans =
    ost: 'T6'
    osm: {'Σ5'  'Σ7'  'Σ7'  'Σ5'}
      d: [41.6400 64.8800 64.8770 41.6510]
      a: [199.9810 53.3672 253.3665 399.9812]
ans =
    ost: 'T6'
    osm: {'Σ5'  'Σ7'  'Σ7'  'Σ5'}
      d: [41.6550 64.9260 64.9160 41.6590]
      a: [295.6382 149.0248 349.0244 95.6375]
```