



**Μάθημα: Μέθοδοι Επίλυσης με Η/Υ**

**Τετάρτη, 19/12/2018**

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

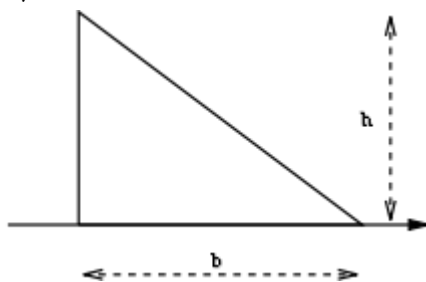
**Παραδείγματα για την 10<sup>η</sup> παράδοση - Συμβολικές πράξεις**

**1. Υπολογισμός κ.β. και ροπής αδρανείας διατομών**

Να υπολογιστεί το κέντρο βάρους και οι κεντροβαρικές δευτεροβάθμιες ροπές αδρανείας στις εξής διατομές:

- α) τριγωνικής (ορθ. Τρίγωνο)  $b \times h$
- β) ημικυκλικής διατομής ακτίνας  $R$

α)



$$S_y = \int_0^b \int_0^{h-\frac{x}{b}h} x \, dy \, dx, \quad x_c = \frac{S_y}{A}, \quad A = \frac{bh}{2}$$

$$S_x = \int_0^b \int_0^{h-\frac{x}{b}h} y \, dy \, dx, \quad y_c = \frac{S_x}{A}$$

$$I_y = \int_0^b \int_0^{h-\frac{x}{b}h} x^2 \, dy \, dx, \quad I_{yc} = I_y - x_c^2 A$$

$$I_x = \int_0^b \int_0^{h-\frac{x}{b}h} y^2 \, dy \, dx, \quad I_{xc} = I_x - y_c^2 A$$

**Κώδικας**

```
clear;
syms b h x y Sx Sy xc yc Ix Iy Ixc Iyc;
Sy = int(int(x, y, 0, h-x/b*h), x, 0, b);
Sx = int(int(y, y, 0, h-x/b*h), x, 0, b);
A = b*h/2;
xc = Sy/A
yc = Sx/A
Iy = int(int(x^2, y, 0, h-x/b*h), x, 0, b);
Ix = int(int(y^2, y, 0, h-x/b*h), x, 0, b);
Iyc = Iy - xc^2*A
Ixc = Ix - yc^2*A
```

### Αποτελέσματα

$x_c =$

$b/3$

$y_c =$

$h/3$

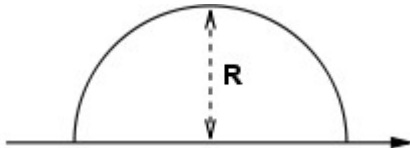
$I_{yc} =$

$(b^3 \cdot h) / 36$

$I_{xc} =$

$(b \cdot h^3) / 36$

β)



$$S_y = \int_0^\pi \int_0^R r \cos(\phi) r dr d\phi \quad , \quad x_c = \frac{S_y}{A} \quad , \quad A = \frac{\pi R^2}{2}$$

$$S_x = \int_0^\pi \int_0^R r \sin(\phi) r dr d\phi \quad , \quad y_c = \frac{S_x}{A}$$

$$I_y = \int_0^\pi \int_0^R (r \cos(\phi))^2 r dr d\phi \quad , \quad I_{yc} = I_y - x_c^2 A$$

$$S_x = \int_0^\pi \int_0^R (r \sin(\phi))^2 r dr d\phi \quad I_{xc} = I_x - y_c^2 A$$

### Κώδικας

```
clear;
```

```
syms R r phi Sx Sy xc yc Ix Iy Ixc Iyc;
```

```
Sy = int(int(r*cos(phi)*r, r, 0, R), phi, 0, pi);
```

```
Sx = int(int(r*sin(phi)*r, r, 0, R), phi, 0, pi);
```

```
A = pi*R^2/2;
```

```
xc = Sy/A
```

```
yc = Sx/A
```

```
Iy = int(int((r*cos(phi))^2*r, r, 0, R), phi, 0, pi);
```

```
Ix = int(int((r*sin(phi))^2*r, r, 0, R), phi, 0, pi);
```

```
Iyc = Iy - xc^2*A
```

```
Ixc = Ix - yc^2*A
```

### Αποτελέσματα

$x_c =$

0

$y_c =$

$(4 \cdot R) / (3 \cdot \pi)$

$I_{yc} =$

$(\pi \cdot R^4) / 8$

$I_{xc} =$

$(\pi \cdot R^4) / 8 - (8 \cdot R^4) / (9 \cdot \pi)$

## 2. Υπολογισμός απόκρισης μονώροφης οικοδομής σε σεισμό

Μικρή μονώροφη οικοδομή έχει μάζα  $m=13000$  kg, δυσκαμψία  $k=10^7$  N/m και απόσβεση  $c=80000$  N/(m s). Η οικοδομή υπόκειται σε σεισμική διέγερση με επιτάχυνση  $q=\ddot{u}_g$  που δίνεται από τα αρχεία (cm/s με  $\Delta t=0.02$  sec) **earthquake1.dat** και **earthquake2.dat** στην ιστοσελίδα mycourses.

Να υπολογιστεί η επιτάχυνση της οικοδομής και να γίνει γράφημα όπου θα φαίνεται η σεισμική επιτάχυνση  $\ddot{u}_g$  και η επιτάχυνση της οικοδομής  $\ddot{u}$  σε σχέση με το χρόνο  $t$ .

Ισχύει:

$$m\ddot{u} + c\dot{u} + ku = -mq$$

όπου  $u(t)$  είναι η μετατόπιση της οικοδομής. Επίσης, θεωρώντας γραμμική μεταβολή της επιτάχυνσης (του εδάφους) για το χρόνο  $\Delta t$  μεταξύ δύο διαδοχικών τιμών  $q_{j-1}$  και  $q_j$  από το αρχείο, η επιτάχυνση μπορεί να γραφεί:

$$q = \ddot{u}_g = a + bt, \quad a = q_{j-1}, \quad b = \frac{q_j - q_{j-1}}{\Delta t}$$

Έτσι η εξίσωση κίνησης γίνεται:

$$m\ddot{u} + c\dot{u} + ku = -m(a + bt)$$

με αρχική μετατόπιση  $u(0)=u_0$  και αρχική ταχύτητα  $\dot{u}(0)=v_0$ .

### Κώδικας

```
clear;
syms a b uo vo u(t) q(t);
k=10e6;
c=80e3;
m=13000;
q(t) = a+b*t;
du = diff(u, t);
ddu = diff(u, t, 2);
f(t)=dsolve(k*u+c*du+m*ddu==-m*q, u(0)==uo, du(0)==vo)
df = diff(f, t);
f(0) %Επαλήθευση
df(0) %Επαλήθευση
fw = fopen('q2', 'w');
fprintf(fw, '%s\n', char(f));
fprintf(fw, '%s\n', char(df));
fclose(fw);

as = dlmread('earthquake1.dat');
as = as ./ 100;
n = length(as);
dt = 0.02;

ukt = zeros(n, 1);
vkt = zeros(n, 1);
akt = zeros(n, 1);
ukt(1) = 0;
vkt(1) = 0;
akt(1) = -(k*ukt(1)+c*vkt(1)+m*as(1))/m;
j = 1;
fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
n = 50;
for j=2:n
    i=j-1;
    bn = (as(j)-as(i))/dt;
```

```

    an = as(i);
    fn = subs(f, [a, b, uo, vo], [an, bn, ukt(i), vkt(i)]);
    dfn = subs(df, [a, b, uo, vo], [an, bn, ukt(i), vkt(i)]);
    ukt(j) = fn(dt);
    vkt(j) = dfn(dt);
    akt(j) = -(k*ukt(j)+c*vkt(j)+m*as(j))/m;
    fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
end
t = dt:dt:dt*n;
%plot(t(1:n), ukt(1:n), t(1:n), vkt(1:n), t(1:n), akt(1:n));
plot(t(1:n), as(1:n), t(1:n), akt(1:n));
legend('seismos', 'ktirio');

```

### Υπολογισμός διαγραμμάτων χωρίς τη χρήση συμβολικών συναρτήσεων

```

clear;
as = dlmread('earthquake1.dat');
as = as ./ 100;
dt = 0.02;
n = length(as);

ukt = zeros(n, 1);
vkt = zeros(n, 1);
akt = zeros(n, 1);
ukt(1) = 0;
vkt(1) = 0;
akt(1) = -(k*ukt(1)+c*vkt(1)+m*as(1))/m;
j = 1;
fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
n = 50;
for j=2:n
    i=j-1;
    bn = (as(j)-as(i))/dt;
    an = as(i);
    [ukt(j), vkt(j)] = uv(dt, an, bn, ukt(i), vkt(i));
    akt(j) = -(k*ukt(j)+c*vkt(j)+m*as(j))/m;
    fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
end
t = dt:dt:dt*n;
%plot(t(1:n), ukt(1:n), t(1:n), vkt(1:n), t(1:n), akt(1:n));
plot(t(1:n), as(1:n), t(1:n), akt(1:n));
legend('seismos', 'ktirio');

function [ fn dfn ] = uv(t, a, b, uo, vo)
fn=(13*b)/1250000 - (13*a)/10000 - (13*b*t)/10000 + ...
exp(-(40*t)/13)*cos((20*321^(1/2)*t)/13)*((13*a)/10000 - ...
(13*b)/1250000 + uo) + (321^(1/2)*exp(-(40*t)/13)* ...
sin((20*321^(1/2)*t)/13)*(13000*a + 4121*b + ...
10000000*uo + 3250000*vo))/1605000000;

dfn=(exp(-(40*t)/13)*cos((20*321^(1/2)*t)/13)*(13000*a + ...
4121*b + 10000000*uo + 3250000*vo))/3250000 - ...
(40*exp(-(40*t)/13)*cos((20*321^(1/2)*t)/13)* ...
((13*a)/10000 - (13*b)/1250000 + uo))/13 - (13*b)/10000 - ...
(321^(1/2)*exp(-(40*t)/13)*sin((20*321^(1/2)*t)/13)* ...
(13000*a + 4121*b + 10000000*uo + 3250000*vo))/521625000 - ...
(20*321^(1/2)*exp(-(40*t)/13)*sin((20*321^(1/2)*t)/13)* ...

```

```
((13*a)/10000 - (13*b)/1250000 + uo))/13;
end
```

### Με αυτόματη παραγωγή κώδικα συνάρτησης

```
clear;
clear;
syms a b uo vo u(t) q(t);
k=10e6;
c=80e3;
m=13000;
q(t) = a+b*t;
du = diff(u, t);
ddu = diff(u, t, 2);
f(t)=dsolve(k*u+c*du+m*ddu==m*q, u(0)==uo, du(0)==vo)
df = diff(f, t);
f(0) %Επαλήθευση
df(0) %Επαλήθευση
fw = fopen('q2', 'w');
fprintf(fw, '%s\n', char(f));
fprintf(fw, '%s\n', char(df));
fclose(fw);
matlabFunction(f, df, 'File', 'f1');

as = dlmread('earthquake1.dat');
as = as ./ 100;
dt = 0.02;
n = length(as);

ukt = zeros(n, 1);
vkt = zeros(n, 1);
akt = zeros(n, 1);
ukt(1) = 0;
vkt(1) = 0;
akt(1) = -(k*ukt(1)+c*vkt(1)+m*as(1))/m;
j = 1;
fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
n = 50;
for j=2:n
    i=j-1;
    bn = (as(j)-as(i))/dt;
    an = as(i);
    [ukt(j), vkt(j)] = f1(dt, an, bn, ukt(i), vkt(i));
    akt(j) = -(k*ukt(j)+c*vkt(j)+m*as(j))/m;
    fprintf('%4d%15.10f %15.10f %15.10f\n', j, ukt(j), vkt(j), akt(j));
end
t = dt:dt:dt*n;
%plot(t(1:n), ukt(1:n), t(1:n), vkt(1:n), t(1:n), akt(1:n));
plot(t(1:n), as(1:n), t(1:n), akt(1:n));
legend('seismos', 'ktirio');

function [f,df] = f1(t,a,b,uo,vo)
%F1
% [F,DF] = F1(T,A,B,UO,VO)

% This function was generated by the Symbolic Math Toolbox version 6.0.
```

```

%      14-May-2015 23:13:01

t2 = b.*1.04e-5;
t3 = sqrt(3.21e2);
t6 = t.*(4.0e1./1.3e1);
t4 = exp(-t6);
t5 = t.*t3.*(2.0e1./1.3e1);
t7 = cos(t5);
t8 = a.*1.3e-3;
t9 = -t2+t8+uo;
t10 = a.*1.3e4;
t11 = b.*4.121e3;
t12 = uo.*1.0e7;
t13 = vo.*3.25e6;
t14 = t10+t11+t12+t13;
t15 = sin(t5);
f = a.*(-1.3e-3)+t2-b.*t.*1.3e-3+t4.*t7.*t9+t3.*t4.* ...
    t14.*t15.*6.230529595015576e-10;
if nargout > 1
    df = b.*(-1.3e-3)-t4.*t7.*t9.*(4.0e1./1.3e1)+t4.*t7.* ...
        t14.*3.076923076923077e-7-t3.*t4.*t9.*t15.*(2.0e1./1.3e1)-t3.*t4.* ...
        t14.*t15.*1.917086029235562e-9;
end

```