



MATLAB 7.1

Ohjelmointiharjoitus

© Matti Lähteenmäki
2005
www.tamk.fi/~mlahteen/

MATLAB[®]
The Language of Technical Computing



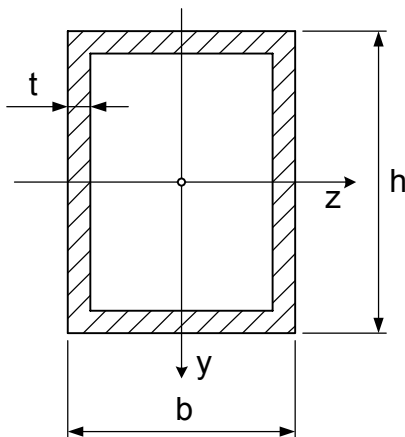
Copyright 1984-2005, The MathWorks, Inc.

SISÄLLYSLUETTELO

1	Ohjelman kirjoittaminen editori/debuggerilla	3
2	Ohjelman ajaminen komentoikkunassa	4
3	Ohjausrakenteiden if-elseif-else ja return käyttö	6
4	Ohjausrakenteiden while ja break käyttö	7
5	Tulostus kuvaikkunaan	9
6	Ohjausrakenteiden for ja continue käyttö sekä tiedosto I/O	11
7	Ohjausrakenteen switch-case-otherwise käyttö	13
8	Alifunktion käyttö	15

1 Ohjelman kirjoittaminen editori/debuggerilla

Tarkastellaan ensimmäisenä sovelluksena kuvan mukaista putkipalkin poikkileikkausta. Laaditaan ohjelma, joka laskee poikkileikkauksen pinta-alan (A), pääneliömomentit (I_z, I_y) ja taivutusvastukset pääakseleiden suhteen (W_z, W_y) ja tulostaa lasketut arvot komentoikkunaan. Lähtötietoina annetaan putken mitat b, h ja t , putkipalkin nurkkapyöristyksiä ei siis oteta huomioon.



$$A = bh - (b - 2t)(h - 2t)$$

$$I_z = \frac{bh^3}{12} - \frac{(b - 2t)(h - 2t)^3}{12} \quad W_z = \frac{I_z}{h/2}$$

$$I_y = \frac{hb^3}{12} - \frac{(h - 2t)(b - 2t)^3}{12} \quad W_y = \frac{I_y}{b/2}$$

$$b > 0 \quad h > 0 \quad t > 0 \quad t \leq \frac{b}{2} \quad t \leq \frac{h}{2}$$

Ohjelmat kannattaa kirjoittaa MATLABin editori/debuggerilla, joten käynnistä se esimerkiksi työpöydän File valikosta valitsemalla New > M-file. Kirjoita poikkipintasuureet laskeva ohjelma editorilla seuraavan kuvan mukaisesti ja tallenna se MATLABin työhakemistoosi nimellä `rhs.m`. Huomaa, että funktiolla `rhs` on kolme syöttötietoina annettavaa argumenttia b, h ja t ja se palauttaa poikkipintasuureiden arvot sijoitettuna argumentteihin A, I_z, I_y, W_z ja W_y .

```

D:\matlab_work\rhs.m
File Edit View Text Debug Breakpoints Web Window Help
[Icons] Stack: Base
1 function [A,Iz,Iy,Wz,Wy]=rhs(b,h,t)
2 % RHS laskee suorakulmaisen putkipoikkileikkauksen
3 % pinta-alan, neliömomentit ja taivutusvastukset.
4 %
5 % Muuttujat:
6 % b=leveys h=korkeus t=seinämän paksuus
7 % A=poikkileikkauksen ala
8 % Iz=neliömomentti z-akselin suhteen
9 % Iy=neliömomentti y-akselin suhteen
10 % Wz=taivutusvastus z-akselin suhteen
11 % Wy=taivutusvastus y-akselin suhteen
12 %
13 % Pinta-alan laskenta.
14 A=b*h-(b-2*t)*(h-2*t);
15 %
16 % Neliömomentti z-akselin suhteen.
17 Iz=b*h^3/12-(b-2*t)*(h-2*t)^3/12;
18 %
19 % Taivutusvastus z-akselin suhteen.
20 Wz=Iz/(h/2);
21 %
22 % Neliömomentti y-akselin suhteen.
23 Iy=h*b^3/12-(h-2*t)*(b-2*t)^3/12;
24 %
25 % Taivutusvastus y-akselin suhteen.
26 Wy=Iy/(b/2);
Ready

```

2 Ohjelman ajaminen komentoikkunassa

Kun funktio `rhs.m` on sijoitettu MATLABin hakupolkuun, se voidaan suorittaa komentoikkunasta käyttämällä komentona funktion `m`-tiedostonimeä `rhs`. Funktiota kutsuttaessa sille on kerrottava syöttötietoargumenttien arvot funktion nimen jälkeen tulevissa kaarisulkeissa pilkuilla erotettuina. Lisäksi on funktion kutsussa määriteltävä ne muuttujat, joihin funktion palauttamien arvojen sijoitetaan. Tämä tapahtuu tavanomaisella sijoituskäskyllä antamalla muuttujien nimet hakasulkeissa pilkuilla erotettuina. Syöttötietoina voivat luonnollisesti olla aikaisemmin komentoikkunassa määritellyt muuttujat, kuten seuraavassa esimerkissä.

```
>> format compact
>> b=60; h=100; t=5;
>> [A, Iz, Iy, Wz, Wy]=rhs(b, h, t)
A =
    1500
Iz =
  1962500
Iy =
   862500
Wz =
   39250
Wy =
   28750
```

Edellä funktion `rhs` kutsua ei päätetty puolipisteellä, joten MATLAB tulostaa palautetut arvot komentoikkunaan. Tulostus estyy normaaliin tapaan päättämällä komento puolipisteeseen, mutta palautettavat arvot tulevat luonnollisesti edelleen komentoikkunan muuttuja-avaruuteen, mikä näkyy seuraavasta esimerkistä.

```
>> [A, Iz, Iy, Wz, Wy]=rhs(b, h, t) ;
>> Iz
Iz =
  1962500
```

Syöttötietoargumenteille voidaan antaa arvot myös suoraan funktion kutsussa seuraavaan tapaan.

```
>> [A, Iz, Iy, Wz, Wy]=rhs(40, 60, 3)
A =
    564
Iz =
  273852
Iy =
  143132
Wz =
  9.1284e+003
Wy =
  7.1566e+003
```

Edellä käytettiin funktion `rhs` kutsuissa niin syöttötietoargumenteille kuin palautusargumenteillekin samoja nimiä, joita käytettiin vastaavaa `m`-tiedostoa kirjoitettaessa. Funktiota voidaan luonnollisesti kutsua minkä nimisiin syöttötietoargumentteihin tahansa sijoitetuilla arvoilla ja vastaavasti se voi palauttaa arvoja minkä nimisiin argumentteihin tahansa.

```
>> bee=60; hoo=100; tee=5;
>> [Ala, I_zeta, I_yy, W_zeta, W_yy]=rhs(bee, hoo, tee)
Ala =
    1500
I_zeta =
```

```

    1962500
I_YY =
    862500
W_zeta =
    39250
W_YY =
    28750

```

Funktio `rhs` antaa harhaanjohtavan tulostuksen, jos sitä kutsutaan seuraavalla tavalla.

```

>> [A,Iy,Wy]=rhs(60,100,5)
A =
    1500
Iy =
    1962500
Wy =
    862500

```

Edellä mainitun puutteen korjaamiseksi siirretään tulostustoiminnot funktiosta käsin tapahtuviksi ja lisätään syöttötietoargumentti `yks`, jonka avulla voidaan ilmaista, missä yksiköissä syöttötiedot ja tulokset ovat. Muutettu funktio tallennetaan nimellä `rhs_2`.

```

function [A,Iz,Iy,Wz,Wy]=rhs_2(b,h,t,yks)
% RHS_2 laskee suorakulmaisen putkipoikkileikkauksen
% pinta-alan, neliömomentit ja taivutusvastukset.

% Muuttujat:
% b=leveys h=korkeus t=seinämän paksuus
% yks=merkkijono, joka sisältää käytettävän yksikön
% A=poikkileikkauksen ala
% Iz=neliömomentti z-akselin suhteen
% Iy=neliömomentti y-akselin suhteen
% Wz=taivutusvastus z-akselin suhteen
% Wy=taivutusvastus y-akselin suhteen
%
% Pinta-alan laskenta.
A=b*h-(b-2*t)*(h-2*t);
fprintf('Poikkileikkauksen ala = %.5g %s^2\n',A,yks);
%
% Neliömomentti z-akselin suhteen.
Iz=b*h^3/12-(b-2*t)*(h-2*t)^3/12;
fprintf('Neliömomentti z-akselin suhteen = %.5g %s^4\n',Iz,yks);
%
% Taivutusvastus z-akselin suhteen.
Wz=Iz/(h/2);
fprintf('Taivutusvastus z-akselin suhteen = %.5g %s^3\n',Wz,yks);
%
% Neliömomentti y-akselin suhteen.
Iy=h*b^3/12-(h-2*t)*(b-2*t)^3/12;
fprintf('Neliömomentti y-akselin suhteen = %.5g %s^4\n',Iy,yks);
%
% Taivutusvastus y-akselin suhteen.
Wy=Iy/(b/2);
fprintf('Taivutusvastus y-akselin suhteen = %.5g %s^3\n',Wy,yks);

```

Kun muutettua funktiota `rhs_2` kutsutaan komentoikkunasta päättäen kutsu puolipisteeseen saadaan aikaisempaan verrattuna huomattavasti havainnollisempi tulostus.

```

>> yksik='mm';
>> [A,Iz,Iy,Wz,Wy]=rhs_2(b,h,t,yksik);
Poikkileikkauksen ala = 1500 mm^2

```

```
Neliömomentti z-akselin suhteen = 1.9625e+006 mm^4
Taivutusvastus z-akselin suhteen = 39250 mm^3
Neliömomentti y-akselin suhteen = 8.625e+005 mm^4
Taivutusvastus y-akselin suhteen = 28750 mm^3
```

Muutetussa muodossaankin `rhs_2` palauttaa edelleen palautusargumenteille arvot komentoikkunan muuttuja-avaruuteen ja mahdollisuus virheelliseen kutsuun on edelleen olemassa. Palautusargumentit voitaisiin kokonaan poistaa, mutta tällöin saataisiin pintasuureista vain tulostus komentoikkunaan, eivätkä ne olisi jatkolaskuissa suoraan käytettävissä muuttujiin sijoitettuina. Funktion `rhs_2` muuttuja-avaruus on erillään komentoikkunan muuttuja-avaruudesta ja häviää sen suorituksen päättyessä.

3 Ohjausrakenteiden `if-elseif-else` ja `return` käyttö

Funktio `rhs_2` laskee pintasuureet myös, vaikka jotkut syöttötiedoista olisivat negatiivisia ja seinämänpaksuuden ollessa suurempi kuin puolet leveydestä tai korkeudesta, mikä ei tietenkään ole suotavaa. Lisätään funktioon `rhs_2` ohjausrakenteen `if-elseif-else` avulla mahdottomien syöttötietojen tunnistus ja suorituksen keskeytys komennolla `return` mahdottomassa tapauksessa. Lisätään myös palautusarvojen alustus (nolliksi), jotta funktio palauttaa kaikissa tapauksissa jotkut arvot. Muutettu funktio tallennetaan nimellä `rhs_3`.

```
function [A,Iz,Iy,Wz,Wy]=rhs_3(b,h,t,yks)
% RHS_3 laskee suorakulmaisen putkipoikkileikkauksen
% pinta-alan, neliömomentit ja taivutusvastukset.

% Muuttujat:
% b=leveys h=korkeus t=seinämän paksuus
% yks=merkkijono, joka sisältää käytettävän yksikön
% A=poikkileikkauksen ala
% Iz=neliömomentti z-akselin suhteen
% Iy=neliömomentti y-akselin suhteen
% Wz=taivutusvastus z-akselin suhteen
% Wy=taivutusvastus y-akselin suhteen
%
% Alustetaan palautusargumentit.
A=0; Iz=0; Iy=0; Wz=0; Wy=0;
%
% Tutkitaan, onko annettu negatiivinen mitta.
if ~(b>0) | ~(h>0) | ~(t>0)
    fprintf('Mittojen tulee olla positiiviset.\n');
    return
% Tutkitaan, onko seinämän paksuus mahdollinen.
elseif (t>b/2) | (t>h/2)
    fprintf('Seinämän paksuus on liian suuri.\n');
    return
else
% Poikkileikkaus on mahdollinen, lasketaan pintasuureet.
% Pinta-alan laskenta.
A=b*h-(b-2*t)*(h-2*t);
fprintf('Poikkileikkauksen ala = %.5g %s^2\n',A,yks);
%
% Neliömomentti z-akselin suhteen.
Iz=b*h^3/12-(b-2*t)*(h-2*t)^3/12;
fprintf('Neliömomentti z-akselin suhteen = %.5g %s^4\n',Iz,yks);
%
% Taivutusvastus z-akselin suhteen.
Wz=Iz/(h/2);
```

```

fprintf('Taivutusvastus z-akselin suhteen = %.5g %s^3\n',Wz,yks);
%
% Neliömomentti y-akselin suhteen.
Iy=h*b^3/12-(h-2*t)*(b-2*t)^3/12;
fprintf('Neliömomentti y-akselin suhteen = %.5g %s^4\n',Iy,yks);
%
% Taivutusvastus y-akselin suhteen.
Wy=Iy/(b/2);
fprintf('Taivutusvastus y-akselin suhteen = %.5g %s^3\n',Wy,yks);
end

```

Seuraavassa on vielä testattu tehtyjen muutosten toimintaa.

```

>> yksik='mm';
>> [A,Iz,Iy,Wz,Wy]=rhs_3(50,100,-3,yksik);
Mittojen tulee olla positiiviset.
>> [A,Iz,Iy,Wz,Wy]=rhs_3(50,100,30,yksik);
Seinämän paksuus on liian suuri.
>> [A,Iz,Iy,Wz,Wy]=rhs_3(60,100,5,yksik);
Poikkileikkauksen ala = 1500 mm^2
Neliömomentti z-akselin suhteen = 1.9625e+006 mm^4
Taivutusvastus z-akselin suhteen = 39250 mm^3
Neliömomentti y-akselin suhteen = 8.625e+005 mm^4
Taivutusvastus y-akselin suhteen = 28750 mm^3

```

4 Ohjausrakenteiden while ja break käyttö

Toteutetaan putkipalkin poikkipintasuureet laskeva `m`-tiedosto vielä skriptinä siten, että se kysyy lähtötiedot komentoikkunassa vuorovaikutteisesti, jolloin ei siis tarvita mitään syöttötietoargumentteja. Koska skriptissä luodut muuttujat tulevat komentoikkunan muuttuja-avaruuteen, ei tarvita myöskään palautusargumentteja laskettujen pintasuureiden arvojen tuomiseksi komentoikkunaan. Laskenta voidaan ohjelmoida `while`-silmukaksi, jossa uusia poikkileikkauksia lasketaan ja tulostetaan niin kauan, kun käyttäjä haluaa. Ohjelmoidaan myös syöttötietojen testaus silmukan sisään ja mahdottomien mittojen tullessa vastaan, keskeytetään `while`-silmukka ja samalla koko skriptin suoritus käyttämällä `break`-komentoja.

```

% PUTKIPA on skripti, joka kysyy, testaa ja laskee
% putkipalkkipoikkileikkauksen pintasuureet.
%
% Lähtötietojen kysely ja testaus.
vastaus='k';
while vastaus=='k'
    b=input('Leveys [0]: ');
    if isempty(b); b=0; end
    if ~(b>0); fprintf('Mittojen tulee olla positiiviset.\n');
        break; end
    h=input('Korkeus [0]: ');
    if isempty(h); h=0; end
    if ~(h>0); fprintf('Mittojen tulee olla positiiviset.\n');
        break; end
    t=input('Paksuus [0]: ');
    if isempty(t); t=0; end
    if ~(t>0); fprintf('Mittojen tulee olla positiiviset.\n');
        break; end
    if (t>b/2)|(t>h/2); fprintf('Seinämän paksuus on liian suuri.\n');
        break; end
    yks=input('Yksikkö [mm]: ', 's');
    if isempty(yks); yks='mm'; end

```

```

% Poikkileikkaus on mahdollinen, lasketaan ja tulostetaan pintasuureet.
%
% Pinta-alan laskenta.
A=b*h-(b-2*t)*(h-2*t);
fprintf('Poikkileikkauksen ala = %.5g %s^2\n',A,yks);
%
% Neliömomentti z-akselin suhteen.
Iz=b*h^3/12-(b-2*t)*(h-2*t)^3/12;
fprintf('Neliömomentti z-akselin suhteen = %.5g %s^4\n',Iz,yks);
%
% Taivutusvastus z-akselin suhteen.
Wz=Iz/(h/2);
fprintf('Taivutusvastus z-akselin suhteen = %.5g %s^3\n',Wz,yks);
%
% Neliömomentti y-akselin suhteen.
Iy=h*b^3/12-(h-2*t)*(b-2*t)^3/12;
fprintf('Neliömomentti y-akselin suhteen = %.5g %s^4\n',Iy,yks);
%
% Taivutusvastus y-akselin suhteen.
Wy=Iy/(b/2);
fprintf('Taivutusvastus y-akselin suhteen = %.5g %s^3\n',Wy,yks);
%
% Kysytään, lasketaanko uusi poikkileikkaus.
vastaus=input('Uusi poikkileikkaus [k/e]: ','s');
if isempty(vastaus); vastaus='e'; end
end

```

Seuraavassa on esitetty skriptin putkipa testausta muutamilla lähtötiedoilla.

```

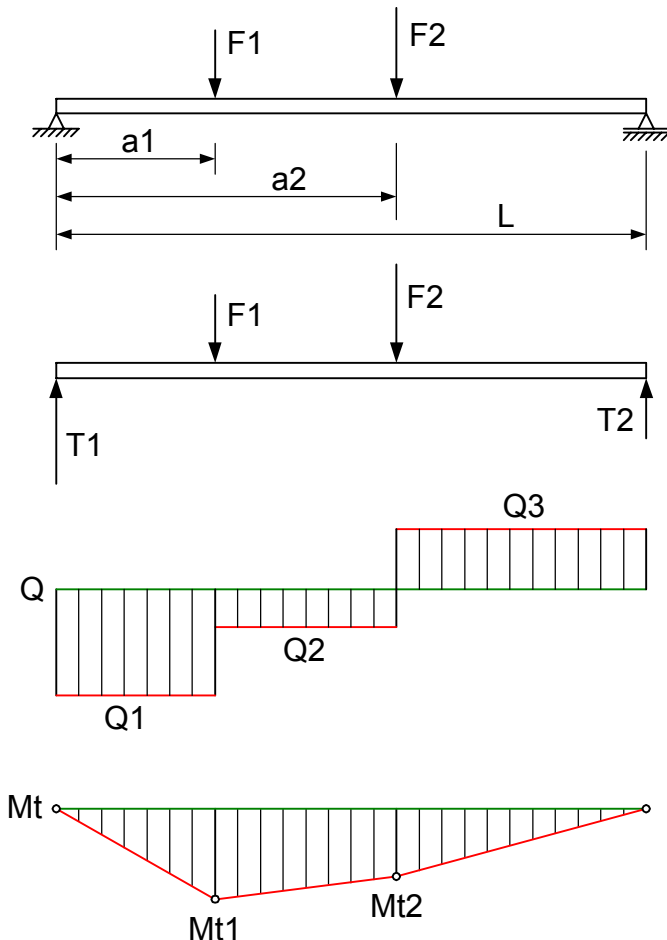
>> putkipa
Leveys [0]: 60
Korkeus [0]: 100
Paksuus [0]: 5
Yksikkö [mm]: mm
Poikkileikkauksen ala = 1500 mm^2
Neliömomentti z-akselin suhteen = 1.9625e+006 mm^4
Taivutusvastus z-akselin suhteen = 39250 mm^3
Neliömomentti y-akselin suhteen = 8.625e+005 mm^4
Taivutusvastus y-akselin suhteen = 28750 mm^3
Uusi poikkileikkaus [k/e]: k
Leveys [0]: 40
Korkeus [0]: 80
Paksuus [0]: -5
Mittojen tulee olla positiiviset.
>> putkipa
Leveys [0]: 40
Korkeus [0]: 120
Paksuus [0]: 21
Seinämän paksuus on liian suuri.
>> putkipa
Leveys [0]: 60e-3
Korkeus [0]: 100e-3
Paksuus [0]: 5e-3
Yksikkö [mm]: m
Poikkileikkauksen ala = 0.0015 m^2
Neliömomentti z-akselin suhteen = 1.9625e-006 m^4
Taivutusvastus z-akselin suhteen = 3.925e-005 m^3
Neliömomentti y-akselin suhteen = 8.625e-007 m^4
Taivutusvastus y-akselin suhteen = 2.875e-005 m^3
Uusi poikkileikkaus [k/e]: e
>>

```


5 Tulostus kuvaikkunaan

Tässä esimerkissä tarkastellaan yksinkertaista kaksitukista palkkia, jonka kuormituksena on kaksi pistevoimaa. Laaditaan funktio, joka laskee palkin tukireaktiot T_1 ja T_2 sekä piirtää kuvaikkunoihin palkin leikkausvoima- ja taivutusmomenttikuvan. Syöttötietoina funktiolle annetaan mitat a_1 , a_2 ja L

sekä voimien suuruudet F_1 ja F_2 . Kuvassa on $a_1 < a_2$, mutta laaditaan ohjelma niin, että voi olla myös $a_1 = a_2$ tai $a_2 < a_1$. Voimien suuruudet F_1 ja F_2 voivat olla myös negatiiviset.



$$T_1 = [F_1 \cdot (L - a_1) + F_2 \cdot (L - a_2)] / L$$

$$T_2 = F_1 + F_2 - T_1$$

$$Q_1 = T_1$$

$$Q_2 = Q_1 - F_1 \quad \text{kun } a_1 < a_2$$

$$Q_3 = T_1 - F_1 - F_2$$

$$Q_1 = T_1$$

$$Q_2 = Q_1 - F_1 - F_2 \quad \text{kun } a_1 = a_2$$

$$Q_3 = Q_2$$

$$M_{t1} = T_1 \cdot a_1$$

$$M_{t2} = T_2 \cdot (L - a_2)$$

Kuvaikkunoihin tulostuvien Q - ja M_t -kuvien lisäksi palautetaan komentoikkunaan vaakavektorit

$T = [T_1 \ T_2]$, $Q = [Q_1 \ Q_2 \ Q_3]$ ja $M_t = [M_{t1} \ M_{t2}]$. Ohjelman listaus on seuraavassa

```
function [T,Q,Mt]=palkki(a1,a2,L,F1,F2)
% PALKKI laskee yksinkertaisen kaksitukisen palkin tukireaktiot
% ja piirtää sen leikkausvoima- ja taivutusmomenttikuvan.
% Kuormituksena on kaksi pistevoimaa.
%
%Muuttujat:
% a1,a2= pistevoiman etäisyys palkin vasemmasta päästä
% L=palkin pituus
% F1,F2=pistevoimien suuruudet
% T=vektori, joka sisältää tukireaktioiden arvot
% Q=vektori, joka sisältää leikkausvoimien arvot
% Mt=vektori, joka sisältää taivutusmomenttien arvot
%
% Alustetaan palautusargumentit.
T=[0 0]; Q=[0 0 0]; Mt=[0 0];
%
% Tutkitaan, missä järjestyksessä voimat ovat funktion kutsussa.
% Vaihdetaan muuttujien nimet, mikäli a2<a1.
```

```

if a2<a1
    apu=a1; a1=a2; a2=apu;
    apu=F1; F1=F2; F2=apu;
end
%
% Tukireaktioiden laskenta.
T(1)=(F1*(L-a1)+F2*(L-a2))/L;
T(2)=F1+F2-T(1);
%
% Leikkausvoimien laskenta ja piirtäminen.
Q(1)=T(1);
if a1==a2
    Q(2)=T(1)-F1-F2;
else
    Q(2)=T(1)-F1;
end
Q(3)=T(1)-F1-F2;
x=[0 a1 a1 a2 a2 L]; y=[Q(1) Q(1) Q(2) Q(2) Q(3) Q(3)];
figure(1);
h1=plot(x,y,'r-',[0 L],[0 0],'g-');
set(h1,'LineWidth',2);
title('Leikkausvoimakuva');
xlabel('x');
ylabel('Q');
ylim([1.2*min(y) 1.2*max(y)]);
grid on;
%
% Taivutusmomentin laskenta ja piirtäminen.
Mt(1)=T(1)*a1; Mt(2)=T(2)*(L-a2);
xt=[0 a1 a2 L]; yt=[0 Mt(1) Mt(2) 0];
figure(2);
h2=plot(xt,yt,'r-',[0 L],[0 0],'g-');
set(h2,'LineWidth',2);
title('Taivutusmomenttikuva');
xlabel('x');
ylabel('Mt');
ylim([1.2*min(yt) 1.2*max(yt)]);
grid on;

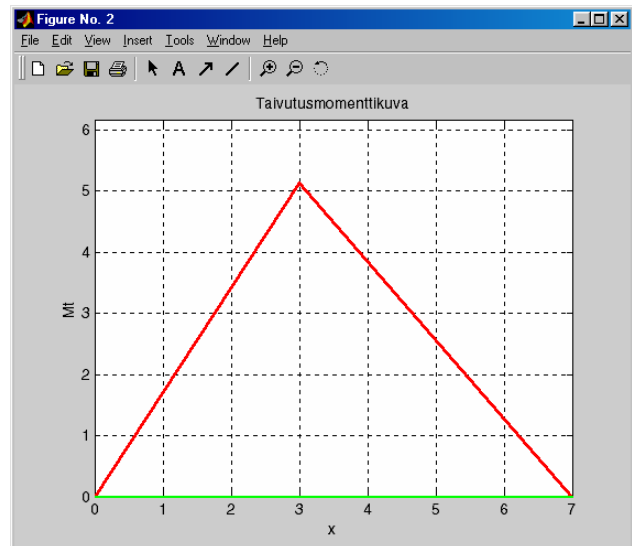
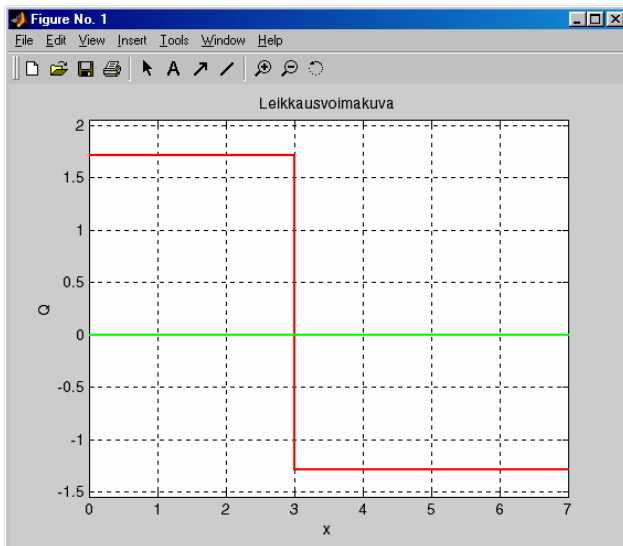
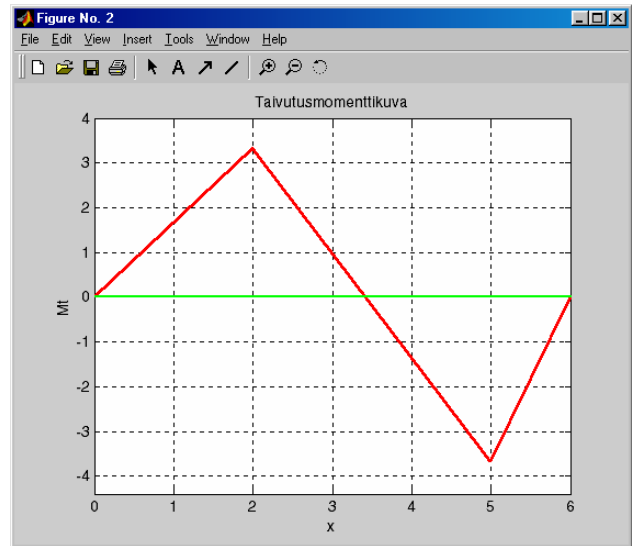
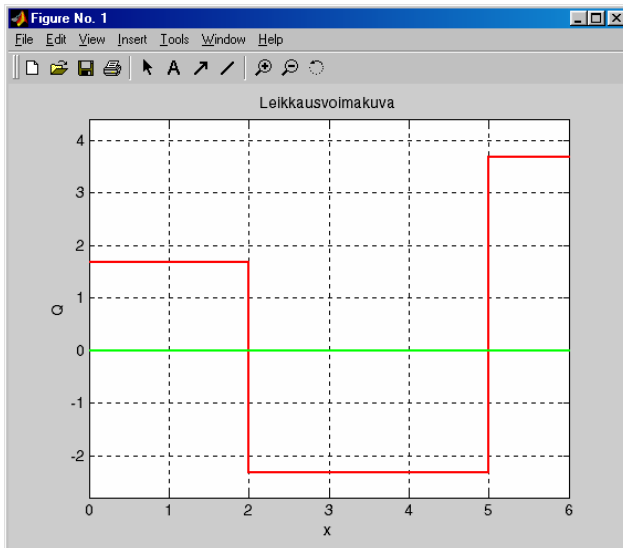
```

Ohjelmassa on käytetty grafiikkakahvoja `h1` ja `h2`, joihin `plot` komennolla laaditut kuvat on sijoitettu. `set`-funktioilla voidaan 'tarttua' kuvaan sen kahvasta ja asetella sitten kuvan ominaisuuksia. Tässä esimerkissä on asetettu viivanpaksuus (`LineWidth`) arvoon 2. Seuraavassa ohjelmaa on vielä testattu kaksilla syöttötietoargumenteilla. Tulostuksena syntyneet rasiuskuvat ovat seuraavalla sivulla.

```

>> format compact
>> [T, Q, Mt]=palkki(2,5,6,4,-6)
T =
    1.6667    -3.6667
Q =
    1.6667    -2.3333     3.6667
Mt =
    3.3333    -3.6667
>> [T, Q, Mt]=palkki(3,3,7,-5,8)
T =
    1.7143     1.2857
Q =
    1.7143    -1.2857    -1.2857
Mt =
    5.1429     5.1429

```



Funktiota `palkki.m` voitaisiin kehittää lisäämällä siihen esimerkiksi seuraavat ominaisuudet.

a) Mahdottomien syöttötietojen `a1`, `a2` ja `L` testaus. b) Yksiköiden käyttö. c) Vektoreiden `T`, `Q` ja `Mt` havainnollisempi tulostus komentoikkunaan `fprintf`-funktiota käyttäen. d) Oleellisten `Q`- ja `Mt`-arvojen tulostuminen rasituskuviin `text`-funktiota käyttäen.

6 Ohjausrakenteiden `for` ja `continue` käyttö sekä tiedosto I/O

Palataan vielä putkipalkin poikkipintasuuresovellukseen ja laaditaan ohjelma, joka lukee tekstitiedostosta joukon mittoja järjestyksessä `b`, `h` ja `t`, laskee vastaavat poikkipintasuureet A , I_z , I_y , W_z ja W_y sekä kirjoittaa annetut lähtötiedot ja lasketut pintasuureet toiseen tekstitiedostoon. Mahdottomat mitat ohitetaan laskennassa ja tulostuksessa. Ohjelman listaus on seuraavassa.

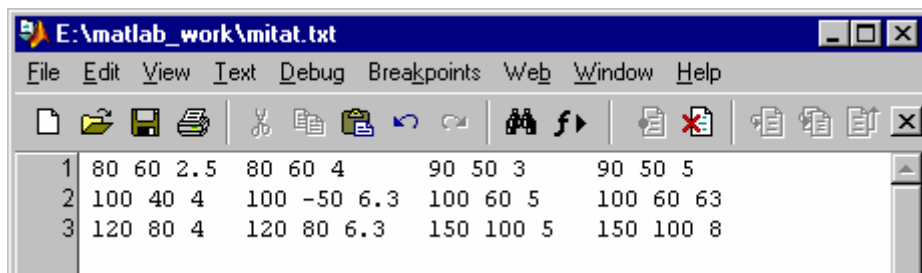
```
function [taul,virh]=pp_suureet(tied1,tied2)
% PP_SUUREET lukee tekstitiedostosta joukon putkipalkin mittoja,
% laskee niitä vastaavat pintasuureet sekä kirjoittaa lähtötiedot ja
% lasketut suureet toiseen tekstitiedostoon.
%
% Muuttujat:
% tied1 on sen tekstitiedoston nimi, jossa lähtötiedot (h,b,t) sijaitsevat.
```

```

% tied2 on sen tekstitiedoston nimi, johon tulokset kirjoitetaan.
% virh on virheellisten lähtötietojoukkojen (h,b,t) lukumäärä.
% taul on hyväksytyt lähtötiedot (h,b,t) ja vastaavat tulokset
% sisältävä taulukko.
%
% Avataan, luetaan ja suljetaan lähtötiedot sisältävä tiedosto.
tun1=fopen(tied1, 'r');
[T1, lukum]=fscanf(tun1, '%g', [3, Inf]);
fclose(tun1);
%
% Lähtötiedot ja lasketut tulokset sisältävän taulukon laadinta.
% Virheellisten lähtötietojoukkojen (h,b,t) etsiminen ja poisto taulukosta.
T=T1';
virh=0;
for k=1:size(T,1)
    h=T(k,1); b=T(k,2); t=T(k,3);
    if (~h>0) | (~b>0) | (~t>0) | (t>b/2) | (t>h/2)
        virh=virh+1;
        continue;
    end
    taul(k-virh,1)=h; % korkeus
    taul(k-virh,2)=b; % leveys
    taul(k-virh,3)=t; % paksuus
    taul(k-virh,4)=b*h-(b-2*t)*(h-2*t); % ala
    taul(k-virh,5)=b*h^3/12-(b-2*t)*(h-2*t)^3/12; % neliömomentti Iz
    taul(k-virh,6)=taul(k-virh,5)/(h/2); % taivutusvastus Wz
    taul(k-virh,7)=h*b^3/12-(h-2*t)*(b-2*t)^3/12; % neliömomentti Iy
    taul(k-virh,8)=taul(k-virh,7)/(b/2); % taivutusvastus Wy
end
%
% Avataan, kirjoitetaan ja suljetaan tulostiedosto.
tun2=fopen(tied2, 'w');
lkm1=fprintf(tun2, '      h      b      t      A      Iz      Wz
Iy      Wy\n');
lkm2=fprintf(tun2, '%4.0f    %4.0f    %4.1f    %9.3f    %13.3f    %11.3f    %13.3f
%11.3f\n', taul);
fclose(tun2);

```

Tulostettavan taulukon täyttäminen on yllä ohjelmoitu for-silmukaksi ja mahdolloman lähtötietojoukon ohitus tapahtuu continue-komennolla. Funktio pp_suureet ei tutki esimerkiksi sitä, onko lähtötietojen määrä oikea (kolmella jaollinen) ja sitä olisi tietysti mahdollista kehittää monella muullakin tavalla. Funktion pp_suureet testaamiseksi kirjoitetaan MATLABin editori/debuggerilla seuraavassa kuvassa näkyvä tekstitiedosto mitat.txt, jossa on kaksi mahdotonta lähtötietojoukkoa (h,b,t). Tulokset kirjoitetaan tiedostoon outit.txt.



Kun funktio pp_suureet ajetaan MATLABin komentoikkunassa antaen syöttötietoargumenteille arvot 'mitat.txt' ja 'outit.txt', saadaan komentoikkunaan seuraava tulostus (funktion kutsua ei päätetä puolipisteeseen).

```
>> format compact
>> format short g
>> [PS, virhe]=pp_suureet('mitat.txt','outit.txt')
PS =
 80  60  2.5    675  6.2641e+005    15660  4.0016e+005    13339
 80  60  4      1056  9.4259e+005    23565  5.9635e+005    19878
 90  50  3      804   8.6425e+005    19206  3.4121e+005    13648
 90  50  5      1300  1.3308e+006    29574  5.1083e+005    20433
100  40  4      1056  1.2568e+006    25137  2.8211e+005    14106
100  60  5      1500  1.9625e+006    39250  8.625e+005     28750
120  80  4      1536  3.0904e+006    1507   1.6364e+006    40909
120  80  6.3  2361.2  4.5619e+006    76031  2.3797e+006    59492
150 100  5      2400  7.545e+006    1.006e+005  3.995e+006    79900
150 100  8      3744  1.1282e+007    1.5043e+005  5.8815e+006    1.1763e+005
virhe =
     2
```

Työhakemistoon tulee samalla oheisen kuvan mukainen tulostiedosto outit.txt.

	h	b	t	A	Iz	Wz	Iy	Wy
1	80	60	2.5	675.000	626406.250	15660.156	400156.250	13338.542
2	80	60	4.0	1056.000	942592.000	23564.800	596352.000	19878.400
3	90	50	3.0	804.000	864252.000	19205.600	341212.000	13648.480
4	90	50	5.0	1300.000	1330833.333	29574.074	510833.333	20433.333
5	100	40	4.0	1056.000	1256832.000	25136.640	282112.000	14105.600
6	100	60	5.0	1500.000	1962500.000	39250.000	862500.000	28750.000
7	120	80	4.0	1536.000	3090432.000	51507.200	1636352.000	40908.800
8	120	80	6.3	2361.240	4561886.725	76031.445	2379670.885	59491.772
9	150	100	5.0	2400.000	7545000.000	100600.000	3995000.000	79900.000
10	150	100	8.0	3744.000	11282272.000	150430.293	5881472.000	117629.440
11								
12								

7 Ohjaurakenteen switch-case-otherwise käyttö

Tarkastellaan sitten sovelluksena suoran ja tasapaksun sauvan nurjahdusvoiman laskentaa, kun sauvan puristus on keskeinen. Sauvan tuennan mukaan erotellaan neljä Eulerin tapausta viereisen kuvan mukaisesti. Sauvan nurjahdusvoiman lauseke on lujuusopin mukaan

$$P_n = \frac{\pi^2 EI}{L_n^2}$$

Sauva nurjahtaa kimmoisella alueella, jos sen hoikkuusluku on suurempi kuin kyseessä olevan materiaalin rajahoikkuusluku λ_{nr} . Hoikkuusluvun λ_n määritelmä on

$$\lambda_n = \frac{L_n}{i} \quad i = \sqrt{\frac{I}{A}}$$

N:o	I	II	III	IV
Tuenta	jäykkä - vapaa	nivel - nivel	jäykkä - nivel	jäykkä - jäykkä
L_n	2L	L	0,699L	0,5L

Nurjahdusvoiman ja hoikkuusluvun laskentaan on seuraavassa esitetty funktio `nurjahdus`, joka lisäksi tutkii onko hoikkuusluku rajahoikkuuslukua suurempi. Funktiossa `nurjahdus` Eulerin tuentatapauksen valinta on toteutettu ohjausrakenteella `switch-case-otherwise`.

```
function [Pn,lamda_n]=nurjahdus(E,A,I,L,lamda_nr,euler)
% NURJAHDUS laskee puristussauvan nurjahdusvoiman suuruuden
% ja hoikkuusluvun sekä tulostaa ne komentoikkunaan.
%
% Muuttujat:
% E=kimmomoduuli
% A=poikkileikkauksen pinta-ala
% I=poikkileikkauksen neliömomentti
% lamda_nr=rajahoikkuusluku
% euler= Eulerin tapauksen numero
% Pn=nurjahdusvoima
% lamda_n=hoikkuusluku
%
% Valitaan nurjahduspituus Eulerin tapauksen mukaiseksi.
switch euler
case 1
    Ln=2*L;
case 2
    Ln=L;
case 3
    Ln=0.699*L;
case 4
    Ln=0.5*L;
otherwise
    disp('Mahdolliset Eulerin tapausten numerot ovat 1, 2, 3 ja 4.')
    return;
end
%
% Lasketaan nurjahdusvoiman arvo.
Pn=pi^2*E*I/Ln^2;
%
% Lasketaan hoikkuusluku.
nelios=sqrt(I/A);
lamda_n=Ln/nelios;
%
% Tulostus komentoikkunaan.
if lamda_n<lamda_nr
    disp('Sauva ei nurjahda kimmoisella alueella.');
```

Jatkossa on esitetty funktion `nurjahdus` käyttöä komentoikkunassa.

```
>> [Pn,lamda_n]=nurjahdus(200000,200,1667,400,88,4);
Sauva ei nurjahda kimmoisella alueella.
Pn = 82263.15268 lamda_n = 69.27511
>> [Pn,lamda_n]=nurjahdus(200000,200,1667,600,106,2);
Sauva nurjahtaa kimmoisella alueella.
Pn = 9140.35030 lamda_n = 207.82532
```

8 Alifunktion käyttö

Samassa m -tiedostossa voi olla myös useita funktioita. Ensimmäisenä m -tiedostossa olevaa funktiota sanotaan primääriksi funktioksi ja primäärin funktion jäljessä olevat funktiot ovat sen alifunktioita. Alifunktiot ovat ainoastaan primäärin funktion ja muiden samassa m -tiedostossa sijaitsevien alifunktioiden näkyvyysalueessa.

Tarkastellaan esimerkkinä funktiota `tasojt`, joka laskee tasojännitystilän pääjännitykset ja pääsuunnat sekä leikkausjännityksen maksimiarvon ja sen esiintymissuunnan. Lujuusopin mukaan tähän laskentaan liittyvät kaavat ovat seuraavat.

$$\sigma_k = \frac{1}{2}(\sigma_x + \sigma_y) \quad R = \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}^2} \quad \sigma_1 = \sigma_k + R \quad \sigma_2 = \sigma_k - R$$

$$\tan(2\phi) = \frac{2\tau_{xy}}{\sigma_x - \sigma_y} \quad \phi_1 = \phi \text{ tai } \phi + \pi/2 \text{ siten, että } \tau_{xy} \cdot \sin(2\phi) \geq 0$$

$$\tau_{\max} = R \quad , \text{ kun } \psi = \phi_1 - \pi/4$$

Lisäksi funktio `tasojt` piirtää annetun jännitystilän Mohrin ympyrän ja käyttää sen määrittämisessä hyväkseen alifunktiota `ympyra`, joka laskee ympyrän kehän pisteiden koordinaatteja. Funktion `tasojt` piirtämässä Mohrin ympyrässä on leikkausjännitys akseli ylöspäin lujuusopin tavanomaisesta käytännöstä poiketen.

```
function [s1,s2,fi1,fi2,tmax,psi]=tasojt(sx,sy,txy)
% TASOJT laskee tasojännitystilän pääjännitykset, pääsuunnat
% ja leikkausjännityksen maksimiarvon ja sen esiintymissuunnan
% sekä piirtää vastaavan Mohrin ympyrän kuvaikkunaan.
%
% Pääjännitysten laskenta.
sk=(sx+sy)/2;
R=sqrt(((sx-sy)/2)^2+txy^2);
s1=sk+R;
s2=sk-R;
%
%Pääsuuntien laskenta.
fii_r=0.5*atan(2*txy/(sx-sy));
if sin(2*fii_r)<0
    fii_r=fii_r+pi/2;
end
fi1=fii_r*180/pi;
fi2=fi1+90;
%
% Leikkausjännityksen maksimi xy-tasolla ja sen esiintymissuunta.
tmax=R;
psi=fi1-45;
%
%Kutsutaan alifunktiota ympyra ja piirretään Mohrin ympyra.
[mohrx,mohry]=ympyra(R,sk,0);
h=plot(mohrx,mohry,'r-');
set(h,'LineWidth',2);
```

```

ylim([-1.1*tmax 1.1*tmax]);
axis equal;
hold on;
h1=plot(s1,0,'bo',s2,0,'bo',sk,0,'ko',sx,txy,'k+',sy,-txy,'k+'...
        ,[sx sy],[txy -txy],'k-', [s1+0.2*R s2-0.2*R ],[0 0],'g--'...
        ,[0 0],[-1.2*R 1.2*R ],'g--');
set(h1,'LineWidth',2,'MarkerSize',6);
text(s1+0.05*R,0.05*R,num2str(s1));
text(s2+0.05*R,0.05*R,num2str(s2));
text(sx+0.08*R,txy,['(' num2str(sx) ',' num2str(txy) ')']);
text(sy+0.08*R,-txy,['(' num2str(sy) ',' num2str(-txy) ')']);
hold off;
%
%
function [ympx,ympy]=ympyra(r,x0,y0)
% YMPYRA on alifunktio, joka laskee ympyrän kehän
% pisteiden koordinaatteja.
%
% Muuttujat:
% r=ympyrän säde
% x0=ympyrän keskipisteen x-koordinaatti
% y0=ympyrän keskipisteen y-koordinaatti
%
delta=2*pi/500;
i=0;
for j=0:delta:2*pi
    i=i+1;
    ympx(i)=x0+r*cos(j);
    ympy(i)=y0+r*sin(j);
end

```

Oheisessa kuvassa on esimerkki funktiolla `tasojt` laaditusta Mohrin ympyrästä ja vastaavat komentoikkunan tulostukset.

```

>> [s1,s2,fi1,fi2,tmax,psi]=tasojt(-20,25,10)
s1 =
    27.122
s2 =
   -22.122
fi1 =
    78.019
fi2 =
   168.02
tmax =
    24.622
psi =
    33.019

```

