Priedai

C priedas. Plokštės optimizavimo programinis kodas

4.1 poskyrio, pirmojo skaitinio pavyzdžio programinis kodas parašytas MATLAB programavimo terpėje

```
tic
clc; clear; format short; close all
```

Pradinis masyvas. Duomenys, kurie nesikeis viso algoritmo metu

```
E = 210e6;          % Tamprumo modulis kPa
sigmay = 210e3;     % Takumo itempis  kPa
v = 1/3;            % Puasono koeficientas
BE = 6;             % Baigtiniu elementu skaicius
t = 0.1;           % Plokstes storis m
```

```
R = 0.9;            % Plokstes spindulys m
b = R/(2*BE);       % m \ puse BE ilgio

fi = [ 1 -0.5 ; -0.5 1]; % Mizeso salygu matricele
t = ones(BE,1)*t;
load('A.mat'); % Pusiausvyros matrica A is failo.

[m,n] = size(A);
[~,nr] = rref(A); % gausas-zordanas ir parodo kurie stulp neprklausomi
'nr'
At = A';
[~,stul] = size(nr); zero = zeros(1,n);

% apkrova nuo Momento m =36.25 ir q = 100 :
F1 = [ 0;2.35619;4.71239;0;0;9.42478;11.781;0;0;16.4934;18.8496;0;0;...
    23.5619;25.9181;0;0;30.6305;32.9867;0;0;37.6991;40.0553;204.989];

% apkrova nuo Momento ir q = -95: (pries apkrova minusas??)
F2 = [0;-2.23838;-4.47677;0;0;-8.95354;-11.1919;0;0;-15.6687;-17.9071;0;0;
...
    -22.3838;-24.6222;0;0;-29.099;-31.3374;0;0;-35.8142;-38.0526;204.989];

j = 2 ; % Apkrovos virsuniu skaicius


for dd = 1:BE
ro(dd,1) = b+(dd-1)*2*b;
K = E*(t(dd)^3)/(12*(1-v^2));
dk = D_matrica(K,v,ro(dd),b);
D(6*dd-5:6*dd, 6*dd-5:6*dd) = dk;
end  % Pasiduodamumo matricos sudarymas

L = 4*pi*b*ro; % Elementu plotu vektorius (apvaliu ziedu plotai)

alfa = D\A'/(A/D*A');          % Irazu influentine matrica
beta = inv(A/D*A');            % Poslinkiu influentine matrica
Se1 = alfa*F1;                     % Irazos
Se2 = alfa*F2;                     % Irazos
ue1 = beta*F1;                   % Tampraus skaiciavimo poslinkiai

M_teta1 = zeros(m,6*BE);
M_teta2 = zeros(n-m,6*BE);
```

```
% Sudeliojame irazas pagal mazgus Mro ir Mfi
p = size(Se1);
MSe = zeros(3*BE,2);
for i=1:2:p
    MSe((i+1)/2,1)= Se1(i);
    MSe((i+1)/2,2)= Se1(i+1);
end

% Nustatom pradini taska pirmam optimizacijos uzdaviniui (tamprumo ribos
M0)
for i=1:3*BE
    vSep = [ Se1(2*i-1); Se1(2*i) ];
    VM0p(i,:) = (vSep'*fi*vSep)^(1/2);
end
M0p = max(VM0p);          % Visi M0
numeriai = 3*[1:BE];      % atrenkam po viena is elemento

x0 = [zeros(3*BE*j+n,1); VM0p(numeriai)];

clear D
for iter = 1:15
```

optimizacijos UZDAVINYS (pertvarkyta sistema su visais apribojimais)

```
%{
matematini modelis:
    min M0

kai     [A]Sr = 0
        f(Sr+Se)<= M0^2                  netiesinis
        [Br]Sr = [Bt]*Teta_pl            netiesinis
        lambda*(f(Sr+Se)-M0^2) = 0       netiesinis
        -lambda <= 0

   nezinomieji tokia tvarka Sr, lambda, M0: (n +3*BE) = 36+18+6= 54+6=60

%}

for dd = 1:BE
ro(dd,1) = b+(dd-1)*2*b;
K = E*(t(dd)^3)/(12*(1-v^2));
dk = D_matrica(K,v,ro(dd),b);
D(6*dd-5:6*dd, 6*dd-5:6*dd) = dk;
```

```
end  % Pasiduodamumo matricos sudarymas


alfa = D\A'/(A/D*A');           % Irazu influentine matrica
beta = inv(A/D*A');             % Poslinkiu influentine matrica
Se1 = alfa*F1;                  % Tampraus skaiciavimo irazos
Se2 = alfa*F2;                  % Tampraus skaiciavimo irazos
ue2 = beta*F2;                  % Tampraus skaiciavimo poslinkiai

% Nauju matricu A1 A2 ir D1 D2 ir Teta1 Teta2 atskyrimas, geometriniu
lygciu pertvarkymui

for kint = 1:stul;
    A1(kint,:) = At(nr(kint),:);
    D1(kint,:) = D(nr(kint),:);
    M_teta1(kint,nr(kint))=1;
    zero(1,nr(kint))= nr(kint);
end      % A1 D1 Teta_p1

for q = 1:n;
    if zero(q) == 0, nr2(q)=q;
    else nr2(q)= 0;
    end
end
nr2 = nonzeros(nr2);

for kint2 = 1:(n-m);
    A2(kint2,:) = At(nr2(kint2),:);
    D2(kint2,:) = D(nr2(kint2),:);
    M_teta2(kint2,nr2(kint2))=1;
end   % A2 D2 Teta_p2

Br = ((A2/A1)*D1-D2);


Aeq = [-A, zeros(m, 3*BE*j+BE)];     % Lygybiu koeficientai: irazos ir
ribine iraza
beq = zeros(m,1);                   % Lygybiu laisvieji nariai nuliai

Aineq1 = [zeros(3*BE*j+BE,n), -eye(3*BE*j+BE)];         % -lambda <= 0 -
M0 <= 0
bineq1 = zeros(3*BE*j+BE,1);

apribojimai =
```

```
@(x,c,ceq)apribs(x,Se1,Se2,BE,A1,A2,D1,n,Br,M_teta1,M_teta2,j,ue2);
funkcija = @(x,F)fja(x,n,BE,L,j);
% =============Sprendziam optimizavimo uzdavini==================
% options = optimset('Algorithm','sqp','largescale','on','TolCon',1e-
10,...
%     'TolFun',1e-10,'TolX',1e-14,'MaxFunEvals',1e6,'MaxIter',1e6);
options = optimset('Algorithm','sqp','largescale','on','TolCon',1e-8,...
    'TolFun',1e-10,'TolX',1e-14,'MaxFunEvals',1e6,'MaxIter',1e6);

[x, fval, exitflag] = ...
    fmincon(funkcija, x0 ,Aineq1, bineq1, Aeq, beq, [], [], apribojimai,
options);
% ==============================================================
% exitflag1

M0 = x(n+3*BE*j+1:end);               % Naujas ribinis momentas
t = ((4*M0)/(sigmay)).^(1/2);  % Naujas plokstes storis
rez(iter,1) = fval;
rez(iter,2:BE+1) = M0;
rez(iter,BE+2) = exitflag;
plot(rez(:,1));
hold on
pause(0.001)
x0=x;
clear D
```

```
end

Mr = x(1:n,1);
lambda = x(n+1:n+3*BE*j,1);
Fi_lambda = repmat(eye(3*BE),1,j);
lambda_sum = Fi_lambda*lambda;       % Sumines visu virsuniu j plastines
deformacijos

for i=1:3*BE
    vSr = [ x(2*i-1); x(2*i) ];
    vSe1 = [ Se1(2*i-1); Se1(2*i) ];
    vSe2 = [ Se2(2*i-1); Se2(2*i) ];
    Fi(i,:) = (vSr+vSe1)'*fi*(vSr+vSe1) - x(n+3*BE*j+ceil(i/3))^2;
    Fi(3*BE+i,:) = (vSr+vSe2)'*fi*(vSr+vSe2) - x(n+3*BE*j+ceil(i/3))^2;
    Teta_p(2*i-1:2*i,1) =
2*(x(n+i)*fi*(vSr+vSe1)+x(n+3*BE+i)*fi*(vSr+vSe2));
end
```

```
S1 = Mr + Se1;
S2 = Mr + Se2;

Teta1=M_teta1*Teta_p;
Teta2=M_teta2*Teta_p;

ur = inv(A1)*(D1*x(1:n)+Teta1);
ceq = vertcat(x(n+1:n+3*BE*j)'*Fi,  Br*(x(1:n))+A2/A1*(M_teta1*Teta_p)-
M_teta2*Teta_p);

toc
```

```
function [dk] = D_matrica(K,niu,ro,b)
%PASID_MATRICA Summary of this function goes here
%   Detailed explanation goes here

dk = ((2*pi*b)/(15*K*(1-niu^2)))* [ (4*ro-3*b)      -niu*(4*ro-3*b)
2*(ro-b)      -2*niu*(ro-b)    -ro                niu*ro;
                                    -niu*(4*ro-3*b)   4*ro-3*b          -
2*niu*(ro-b)    2*(ro-b)        niu*ro          -ro
                                     2*(ro-b)        -2*niu*(ro-b)
16*ro          -16*niu*ro       2*(ro+b)        -2*niu*(ro+b)
                                    -2*niu*(ro-b)     2*(ro-b)          -
16*niu*ro       16*ro          -2*niu*(ro+b)      2*(ro+b)
                                    -ro               niu*ro
2*(ro+b)      -2*niu*(ro+b)     4*ro+3*b        -niu*(4*ro+3*b)
                                     niu*ro          -ro                -
2*niu*(ro+b)    2*(ro+b)       -niu*(4*ro+3*b)    4*ro+3*b       ];


end
```

```
function F = fja(x,n,BE,L,j)

F = L'*x(n+3*BE*j+1:end,1);
end
```

```
function [c,ceq] =
apribs(x,Se1,Se2,BE,A1,A2,D1,n,Br,M_teta1,M_teta2,j,ue2)

fi = [ 1 -0.5 ; -0.5 1];


Teta_p = 2*[ x(37)*fi*[(Se1(1)+x(1)); (Se1(2)+x(2))] +
x(55)*fi*[(Se2(1)+x(1)); (Se2(2)+x(2))]
             x(38)*fi*[(Se1(3)+x(3)); (Se1(4)+x(4))] +
x(56)*fi*[(Se2(3)+x(3)); (Se2(4)+x(4))]
             x(39)*fi*[(Se1(5)+x(5)); (Se1(6)+x(6))] +
x(57)*fi*[(Se2(5)+x(5)); (Se2(6)+x(6))]
             x(40)*fi*[(Se1(7)+x(7)); (Se1(8)+x(8))] +
x(58)*fi*[(Se2(7)+x(7)); (Se2(8)+x(8))]
             x(41)*fi*[(Se1(9)+x(9)); (Se1(10)+x(10))]  +
x(59)*fi*[(Se2(9)+x(9)); (Se2(10)+x(10))]
             x(42)*fi*[(Se1(11)+x(11)); (Se1(12)+x(12))] +
x(60)*fi*[(Se2(11)+x(11)); (Se2(12)+x(12))]
             x(43)*fi*[(Se1(13)+x(13)); (Se1(14)+x(14))] +
x(61)*fi*[(Se2(13)+x(13)); (Se2(14)+x(14))]
             x(44)*fi*[(Se1(15)+x(15)); (Se1(16)+x(16))] +
x(62)*fi*[(Se2(15)+x(15)); (Se2(16)+x(16))]
             x(45)*fi*[(Se1(17)+x(17)); (Se1(18)+x(18))] +
x(63)*fi*[(Se2(17)+x(17)); (Se2(18)+x(18))]
             x(46)*fi*[(Se1(19)+x(19)); (Se1(20)+x(20))] +
x(64)*fi*[(Se2(19)+x(19)); (Se2(20)+x(20))]
             x(47)*fi*[(Se1(21)+x(21)); (Se1(22)+x(22))] +
x(65)*fi*[(Se2(21)+x(21)); (Se2(22)+x(22))]
             x(48)*fi*[(Se1(23)+x(23)); (Se1(24)+x(24))] +
x(66)*fi*[(Se2(23)+x(23)); (Se2(24)+x(24))]
             x(49)*fi*[(Se1(25)+x(25)); (Se1(26)+x(26))] +
x(67)*fi*[(Se2(25)+x(25)); (Se2(26)+x(26))]
             x(50)*fi*[(Se1(27)+x(27)); (Se1(28)+x(28))] +
```

```
x(68)*fi*[(Se2(27)+x(27)); (Se2(28)+x(28))]
           x(51)*fi*[(Se1(29)+x(29)); (Se1(30)+x(30))] +
x(69)*fi*[(Se2(29)+x(29)); (Se2(30)+x(30))]
           x(52)*fi*[(Se1(31)+x(31)); (Se1(32)+x(32))] +
x(70)*fi*[(Se2(31)+x(31)); (Se2(32)+x(32))]
           x(53)*fi*[(Se1(33)+x(33)); (Se1(34)+x(34))] +
x(71)*fi*[(Se2(33)+x(33)); (Se2(34)+x(34))]
           x(54)*fi*[(Se1(35)+x(35)); (Se1(36)+x(36))] +
x(72)*fi*[(Se2(35)+x(35)); (Se2(36)+x(36))]];


Fi = [ (Se1(1)+x(1))^2-(Se1(1)+x(1))*(Se1(2)+x(2))+(Se1(2)+x(2))^2-x(73)^2
       (Se1(3)+x(3))^2-(Se1(3)+x(3))*(Se1(4)+x(4))+(Se1(4)+x(4))^2-x(73)^2
       (Se1(5)+x(5))^2-(Se1(5)+x(5))*(Se1(6)+x(6))+(Se1(6)+x(6))^2-x(73)^2
       (Se1(7)+x(7))^2-(Se1(7)+x(7))*(Se1(8)+x(8))+(Se1(8)+x(8))^2-x(74)^2
       (Se1(9)+x(9))^2-(Se1(9)+x(9))*(Se1(10)+x(10))+(Se1(10)+x(10))^2-
x(74)^2
       (Se1(11)+x(11))^2-
(Se1(11)+x(11))*(Se1(12)+x(12))+(Se1(12)+x(12))^2-x(74)^2
       (Se1(13)+x(13))^2-
(Se1(13)+x(13))*(Se1(14)+x(14))+(Se1(14)+x(14))^2-x(75)^2
       (Se1(15)+x(15))^2-
(Se1(15)+x(15))*(Se1(16)+x(16))+(Se1(16)+x(16))^2-x(75)^2
       (Se1(17)+x(17))^2-
(Se1(17)+x(17))*(Se1(18)+x(18))+(Se1(18)+x(18))^2-x(75)^2
       (Se1(19)+x(19))^2-
(Se1(19)+x(19))*(Se1(20)+x(20))+(Se1(20)+x(20))^2-x(76)^2
       (Se1(21)+x(21))^2-
(Se1(21)+x(21))*(Se1(22)+x(22))+(Se1(22)+x(22))^2-x(76)^2
       (Se1(23)+x(23))^2-
(Se1(23)+x(23))*(Se1(24)+x(24))+(Se1(24)+x(24))^2-x(76)^2
       (Se1(25)+x(25))^2-
(Se1(25)+x(25))*(Se1(26)+x(26))+(Se1(26)+x(26))^2-x(77)^2
       (Se1(27)+x(27))^2-
(Se1(27)+x(27))*(Se1(28)+x(28))+(Se1(28)+x(28))^2-x(77)^2
       (Se1(29)+x(29))^2-
(Se1(29)+x(29))*(Se1(30)+x(30))+(Se1(30)+x(30))^2-x(77)^2
       (Se1(31)+x(31))^2-
(Se1(31)+x(31))*(Se1(32)+x(32))+(Se1(32)+x(32))^2-x(78)^2
       (Se1(33)+x(33))^2-
(Se1(33)+x(33))*(Se1(34)+x(34))+(Se1(34)+x(34))^2-x(78)^2
       (Se1(35)+x(35))^2-
(Se1(35)+x(35))*(Se1(36)+x(36))+(Se1(36)+x(36))^2-x(78)^2
```

```
        (Se2(1)+x(1))^2-(Se2(1)+x(1))*(Se2(2)+x(2))+(Se2(2)+x(2))^2-x(73)^2
        (Se2(3)+x(3))^2-(Se2(3)+x(3))*(Se2(4)+x(4))+(Se2(4)+x(4))^2-x(73)^2
        (Se2(5)+x(5))^2-(Se2(5)+x(5))*(Se2(6)+x(6))+(Se2(6)+x(6))^2-x(73)^2
        (Se2(7)+x(7))^2-(Se2(7)+x(7))*(Se2(8)+x(8))+(Se2(8)+x(8))^2-x(74)^2
        (Se2(9)+x(9))^2-(Se2(9)+x(9))*(Se2(10)+x(10))+(Se2(10)+x(10))^2-
x(74)^2
        (Se2(11)+x(11))^2-
(Se2(11)+x(11))*(Se2(12)+x(12))+(Se2(12)+x(12))^2-x(74)^2
        (Se2(13)+x(13))^2-
(Se2(13)+x(13))*(Se2(14)+x(14))+(Se2(14)+x(14))^2-x(75)^2
        (Se2(15)+x(15))^2-
(Se2(15)+x(15))*(Se2(16)+x(16))+(Se2(16)+x(16))^2-x(75)^2
        (Se2(17)+x(17))^2-
(Se2(17)+x(17))*(Se2(18)+x(18))+(Se2(18)+x(18))^2-x(75)^2
        (Se2(19)+x(19))^2-
(Se2(19)+x(19))*(Se2(20)+x(20))+(Se2(20)+x(20))^2-x(76)^2
        (Se2(21)+x(21))^2-
(Se2(21)+x(21))*(Se2(22)+x(22))+(Se2(22)+x(22))^2-x(76)^2
        (Se2(23)+x(23))^2-
(Se2(23)+x(23))*(Se2(24)+x(24))+(Se2(24)+x(24))^2-x(76)^2
        (Se2(25)+x(25))^2-
(Se2(25)+x(25))*(Se2(26)+x(26))+(Se2(26)+x(26))^2-x(77)^2
        (Se2(27)+x(27))^2-
(Se2(27)+x(27))*(Se2(28)+x(28))+(Se2(28)+x(28))^2-x(77)^2
        (Se2(29)+x(29))^2-
(Se2(29)+x(29))*(Se2(30)+x(30))+(Se2(30)+x(30))^2-x(77)^2
        (Se2(31)+x(31))^2-
(Se2(31)+x(31))*(Se2(32)+x(32))+(Se2(32)+x(32))^2-x(78)^2
        (Se2(33)+x(33))^2-
(Se2(33)+x(33))*(Se2(34)+x(34))+(Se2(34)+x(34))^2-x(78)^2
        (Se2(35)+x(35))^2-
(Se2(35)+x(35))*(Se2(36)+x(36))+(Se2(36)+x(36))^2-x(78)^2  ];


ur = A1\(D1*x(1:n)+M_teta1*Teta_p);
% c = vertcat(Fi, A1\(D1*x(1:12)+M_teta1*Teta_p)-ur_lim);

c = vertcat(Fi, -ur(1)-0.02-ue2(1));

ceq = vertcat(x(n+1:n+3*BE*j)'*Fi,  Br*(x(1:n))+A2/A1*(M_teta1*Teta_p)-
M_teta2*Teta_p);

end
```