Příloha 6 – optimalizace elipsy

```
clear
close all
clc

global psi
global i_psi
global jk_psi
global dx
global z0
global elementy
global A0
global x0

%% ANALYTICKA ELIPSA
Na = 100000;
a = 10; % horizontal radius
b = 2; % vertical radius
x0 = 0; % x0,y0 ellipse centre coordinates
y0 = 0;
ta = linspace(0, pi/2, Na+1);
xa = x0 + a*sin(ta);
ya = y0 + b*cos(ta);
z0 = b;

points = [xa;ya];
dxy = (diff(points'))';
dl = zeros(1,Na);
for i = 1:Na
    dl(i) = norm(dxy(:,i));
end
l = [0, cumsum(dl)];
L = sum(dl);

%% NUMERICKA APROXIMACE
N = 20; % POCET ELEMENTU
l_step = L/N;
lq = 0:l_step:L;

xq = interp1(l,xa,lq,'spline');
zq = interp1(l,ya,lq,'spline');

body = [xq;zq];
d_elem = (diff(body'))';
elementy = zeros(1,N);
for i = 1:N
    elementy(i) = norm(d_elem(:,i));
end

psi = acos(diff(xq)./elementy);

dx = elementy .* cos(psi);
dz = -elementy .* sin(psi);
```

```
x_0 = [x0, x0 + cumsum(dx)];
z_0 = [z0, z0 + cumsum(dz)];
A0 = trapz(x_0,z_0);

%% NUMERICKA KRIVOST
elementy1 = [elementy(1), elementy];
dpsi = diff(psi);
dpsi = [2*psi(1),dpsi,2*(pi/2-psi(end))];
c2_avg_start = zeros(1,N);

for i = 1:N
    c2_avg_start(i) = 0.5 * ((dpsi(i)/elementy1(i)) +
(dpsi(i+1)/elementy1(i+1))); % vypocet krivosti c1 - puvodni metoda
end

c2_avg_start = [dpsi(1)/elementy1(1), c2_avg_start];
dL = cumsum(elementy);
dL = [0,dL];

%% ENERGIE
kappa = 10;
dL = cumsum(elementy);
dL = [0,dL];
U_0 = 2 * kappa * trapz(dL,c2_avg_start.^2);
U0 = U_0;

%% OPTIMALIZACE
no_imp = 0; % POCET CYKLU BEZ ZLEPSENI
xmin = zeros(1, N+1);
zmin = zeros(1, N+1);
q = 1;
k = 0.002;

while no_imp < 20000
%% NUMERICKA APROXIMACE
dx = elementy .* cos(psi);
dz = -elementy .* sin(psi);

x = [x0, x0 + cumsum(dx)];
z = [z0, z0 + cumsum(dz)];

index = 1:1:length(psi)-1;
i_psi = randsample(index, 1);
vychyleni = k * (rand - 0.5);
psi(i_psi) = psi(i_psi) + vychyleni;
index = setdiff(index, i_psi);

jk_psi = randsample(index, 2);

nula_z(zeros(1,length(jk_psi)));
fun = @nula_z;
options = optimset('Display','off','Algorithm','levenberg-marquardt');
xopt = fsolve(@nula_z, zeros(1, length(jk_psi) + 1), options);
nula_z(xopt);
```

```
dz0 = xopt(1);
dpsi = xopt(2:end);

psi2 = psi;
psi2(jk_psi) = psi2(jk_psi) + dpsi;
dx2 =  elementy .* cos(psi2);
dz2 = -elementy .* sin(psi2);
x2 = [x0, x0 + cumsum(dx2)];
z2 = [z0 + dz0, z0 + dz0 + cumsum(dz2)];
A = trapz(x2, z2);

dL = cumsum(elementy);
dL = [0,dL];
%% NUMERICKA KRIVOST
dpsi1 = diff(psi2);
dpsi1 = [2*psi2(1),dpsi1,2*(pi/2-psi2(end))];
c2_avg = zeros(1,N);

for i = 1:N
    c2_avg(i) = 0.5 * ((dpsi1(i)/elementy1(i)) +
(dpsi1(i+1)/elementy1(i+1))); % vypocet krivosti c1 - puvodni metoda
end

c2_avg = [dpsi1(1)/elementy1(1), c2_avg];

U = 2 * kappa * trapz(dL,c2_avg.^2);

%% PRUBEH VELICINY U
U_trend(q) = U; % vektor veliciny U v prubehu optimalizace
q = q+1;

if (abs(z2(end)) < 0.0000001) && (A - A0 < 0.00000001)
if (U < U0)
    psi = psi2;
    z0 = z0 + dz0;
    U0 = U(1);
    xmin = x2;
    zmin = z2;
    no_imp = 0;
else
    no_imp = no_imp + 1;
end
end

end

U = [U_0, U0, U]

%% GRAFY
figure(1)
hold on
pbaspect([11 3 1])
set(gca, 'FontSize', 30, 'fontname', 'Cambria Math', 'LineWidth', 1.5)
plot(x_0, z_0, '-ok', 'MarkerFaceColor', 'k', 'LineWidth', 2)
plot(xmin, zmin, '--ok', 'LineWidth', 2)
xlabel('{\itx} [\mum]', 'fontSize', 30, 'fontname', 'Times')
ylabel('{\itz} [\mum]', 'fontSize', 30, 'fontname', 'Times')
xlim([0 11])
ylim([0 3])
legend('pùvodní', 'optimalizovaná', 'fontSize', 30, 'fontname', 'Times',
'Location', 'best')
hold off

figure(2)
grid on
plot(U_trend, 'k')
set(gca, 'FontSize', 30, 'fontname', 'Cambria Math', 'LineWidth', 1.5)
xlabel('Poèet cyklù {\it C} [-]', 'fontSize', 30, 'fontname', 'Cambria
Math');
ylabel('Deformaèní energie {\it U} [k_{b}T]', 'fontSize', 30, 'fontname',
'Cambria Math');

%% PROMENNE FUNKCE fsolve
function res = nula_z(xopt)
    global psi
    global jk_psi
    global dx2
    global z0
    global elementy
    global x0
    global A0

    dz0 = xopt(1);
    dpsi = xopt(2:end);
    psi2 = psi;
    psi2(jk_psi) = psi2(jk_psi) + dpsi;
    dx2 =  elementy .* cos(psi2);
    dz2 = -elementy .* sin(psi2);
    x2 = [x0, x0+cumsum(dx2)];
    z2 = [z0 + dz0, z0 + dz0 + cumsum(dz2)];
    z2end = z2(end);

    A = trapz(x2, z2);

    res = [z2end, A-A0];
end
```