ПРОЕКТ СОФТ ®
34-36 Peyo Yavorov blvd,
Sofia 1164, Bulgaria

tel +359 2 423 4455
proektsoft.bg@gmail.com
https://calcpad.eu

A CALCPAD PROGRAM

FOR ANALYSIS OF PLANE FRAMES WITH ARBITRARY CROSS-SECTIONS

(using the finite element method)

by

Eng. Nedelcho Ganchovsky

Table of Contents

I. Introduction

This Calcpad program calculates plane frames with arbitrary sections using the finite element method. The input data is entered in text format as vectors and matrices as follows:

- joint coordinates;
- joint numbers at the ends of the elements;
- material properties;
- dimensions and types of cross-sections;
- support conditions;
- load values.

As a result, diagrams of internal forces and deflections of structural elements are obtained. The schemes are automatically generated by the program, using the SVG graphical format.

II. Calcpad source code

```
#include svg_drawing.cpd
"Analysis of plane frames with arbitrary cross-sections
'<h4>Joint coordinates - xJ; yJ</h4>
#hide
#deg
δz = 10^-12
Precision = 10^-9
x_J = [0; 0; 8; 16; 16]*m
y_J = [0; 8; 10; 8; 0]*m
#show
x_J','y_J
n_J = len(x_J)
'<h4>Elements - [J1; J2]</h4>
#hide
e_J = [1; 2|2; 3|3; 4|4; 5]
#show
transp(e_J)
n_E = n_rows(e_J)
'Element endpoint coordinates
x_1(e) = x_J.e_J.(e; 1)','y_1(e) = y_J.e_J.(e; 1)
x_2(e) = x_J.e_J.(e; 2)','y_2(e) = y_J.e_J.(e; 2)
'Element length - 'l(e) = sqrt((x_2(e) - x_1(e))^2 + (y_2(e) - y_1(e))^2)
'Element direction
c(e) = (x_2(e) - x_1(e))/l(e)','s(e) = (y_2(e) - y_1(e))/l(e)
'Transformation matrix
'Diagonal 3x3 block -'t(e) = [c(e); s(e); 0|-s(e); c(e); 0|0; 0; 1]
'Generation of the full transformation matrix
T(e) = add(t(e); add(t(e); matrix(6; 6); 1; 1); 4; 4)
'<h4>Supports - [Joint; cx; cy; cr]</h4>
#hide
c = [1; 10^20kN/m; 10^20kN/m; 0kNm|5; 10^20kN/m; 10^20kN/m; 10^20kNm]
#show
c
n_c = n_rows(c)
'<h4>Loads - [Element, qx, qy]</h4>
#hide
q = [1; 10kN/m; 0kN/m|2; 0kN/m; -20kN/m|3; 0kN/m; -10kN/m]
n_q = n_rows(q)
q_x = vector(n_E)*kN/m
q_y = vector(n_E)*kN/m
$Repeat{q_x.(q.(i; 1)) = q_x.(q.(i; 1)) + q.(i; 2) @ i = 1 : n_q}
$Repeat{q_y.(q.(i; 1)) = q_y.(q.(i; 1)) + q.(i; 3) @ i = 1 : n_q}
#show
q
'Load values on elements
q_x
q_y
```

```
'<h4>Scheme of the structure</h4>
#hide
w = max(x_J)
h = max(y_J)
W = 240
H = h*W/w
k = W/w
#def svg$ = '<svg viewbox="'-3m*k' '-2m*k' '(w + 6m)*k' '(h + 4m)*k'" xmlns="http://www.w3.org/2000/svg" version="1.1" style="font-family: Georgia Pro; font-size:4pt; width:'W + 150'pt; height:'H + 200*H/W'pt">
#def thin_style$ = style = "stroke:green; stroke-width:1; fill:none"
#def thick_style$ = style = "stroke:green; stroke-width:2; fill:none"
k_q = m/kN
#show
#val
svg$
#for i = 1 : n_E
    #hide
    x1 = x_1(i)*k
    y1 = (h - y_1(i))*k
    x2 = x_2(i)*k
    y2 = (h - y_2(i))*k
    q_xi = q_x.i
    q_yi = q_y.i
    α = atan2(c(i); s(i))
    #if α ≥ 135
        α = α - 180
    #end if
    #if α < -45
        α = α + 180
    #else if α < 0
        α = 360 + α
    #end if
    #if q_xi ≠ 0kN/m
        #hide
        x3 = x2 - q_xi*k_q','y3 = y2
        x4 = x1 - q_xi*k_q','y4 = y1
        x = (x3 + x4)/2 - 5*sign(q_xi)
        y = (y3 + y4)/2
        #show
        '<polygon points="'x1','y1' 'x2','y2' 'x3','y3' 'x4','y4'" style="stroke:magenta; stroke-width:1; stroke-opacity:0.3; fill:magenta; fill-opacity:0.1;" />
        text$(x;y;α;qx='abs(q_xi)')
    #end if
    #if q_yi ≠ 0kN/m
        #hide
        x3 = x2','y3 = y2 + q_yi*k_q
        x4 = x1','y4 = y1 + q_yi*k_q
        x = (x3 + x4)/2
```

```
        y = (y3 + y4)/2 + 5*sign(q_yi)
        #show
        '<polygon points="'x1','y1' 'x2','y2' 'x3','y3' 'x4','y4'" style="stroke:dodgerblue; stroke-width:1; stroke-opacity:0.4; fill:dodgerblue; fill-opacity:0.15;" />
        text$(x;y;α;qy='abs(q_yi)')
    #end if
    #show
    line$(x1; y1; x2; y2; main_style$)
#loop
'<g id="frame">
#for i = 1 : n_E
    #hide
    x1 = x_1(i)*k
    y1 = (h - y_1(i))*k
    x2 = x_2(i)*k
    y2 = (h - y_2(i))*k
    #show
    line$(x1; y1; x2; y2; main_style$)
#loop
#for i = 1 : n_c
    #hide
    j = c.(i; 1)
    x1 = x_J.j*k
    y1 = (h - y_J.j)*k
    δ = w/30*k*sign(x1 - w/2*k)
    x2 = x1 - δ
    y2 = y1 - abs(δ)
    x3 = x1 + δ
    y3 = y1 + abs(δ)
    #show
    #if c.(i; 2) ≠ 0kN/m
        #if c.(i; 3) ≠ 0kN/m
            #if c.(i; 4) ≠ 0kNm
                line$(x1; y1; x1; y3; thin_style$)
                line$(x2; y3; x3; y3; thick_style$)
            #else
                line$(x2; y3; x3; y3; thick_style$)
                line$(x2; y3; x1; y1; thin_style$)
                line$(x3; y3; x1; y1; thin_style$)
            #end if
        #else
            #if c.(i; 4) ≠ 0kNm
                line$(x1; y1; x2; y1; thin_style$)
                line$(x2; y2; x2; y3; thick_style$)
                line$(x2 - δ/2; y2; x2 - δ/2; y3; thick_style$)
            #else
                line$(x2; y2; x1; y1; thin_style$)
                line$(x2; y3; x1; y1; thin_style$)
                line$(x2; y2; x2; y3; thin_style$)
```

```
                line$(x2 - δ/2; y2; x2 - δ/2; y3; thick_style$)
            #end if
        #end if
    #else
        #if c.(i; 3) ≠ 0kN/m
            #if c.(i; 4) ≠ 0kNm
                line$(x1; y1; x1; y3; thin_style$)
                line$(x2; y3; x3; y3; thick_style$)
                line$(x2; y3 + abs(δ)/2; x3; y3 + abs(δ)/2; thick_style$)
            #else
                line$(x2; y3; x3; y3; thin_style$)
                line$(x2; y3; x1; y1; thin_style$)
                line$(x3; y3; x1; y1; thin_style$)
                line$(x2; y3 + abs(δ)/2; x3; y3 + abs(δ)/2; thick_style$)
            #end if
        #else
            line$(x2; y2; x3; y3; thick_style$)
        #end if
    #end if
#loop
'</g>
#for i = 1 : n_E
    #hide
    x = (x_1(i) + x_2(i))*k/2
    y = (h - (y_1(i) + y_2(i))/2)*k
    #show
    texth$(x + 0.8m*sign(W/2 - x)*k; y + 0.6m*k; e'i')
#loop
#for i = 1 : n_J
    point$(x_J.i*k; (h - y_J.i)*k; point_style$)
    texth$((x_J.i - 0.7m*sign(w/2 - x_J.i))*k; (h - y_J.i - 0.4m)*k; J'i')
#loop
dimv$((w + 2m)*k; (h - y_J.4)*k; h*k; 'y_J.4')
dimv$((w + 2m)*k; 0; (h - y_J.4)*k; 'h - y_J.4')
dimh$(0; w*k; (h + 1.5m)*k; 'w')
'</svg>
#equ
'<h4>Materials</h4>
'Modules of elasticity -'E = [45; 35]*GPa
'Poisson coefficients -'v = [0.2; 0.2]
'Shear modules -'G = E/(2*(1 + v))
'Assignment on elements -'e_M = [1; 2; 2; 1]
'<h4>Cross-sections</h4>
#hide
b = vector(2)','h = vector(2)
#show
'Section S1 -'h.1 = 500mm'- circular - 'b_C(z) = 2*sqrt((h.1/2)^2 - (z - h.1/2)^2)
'Section S2 -'b.2 = 250mm','h.2 = 700mm'- rectangular -'b_R(z) = b.2
```

```
'General representation -'b(z; s) = take(s; b_C(z); b_R(z))
'<h4>Cross section properties</h4>
'Equations
'Area -'A(s) = $Integral{b(z; s) @ z = 0mm : h.s}
'First moment of area -'S(s) = $Integral{b(z; s)*z @ z = 0mm : h.s}|cm^3
'Centroid -'z_c(s) = S(s)/A(s)|mm
'Second moment of area -'I(s) = $Integral{b(z; s)*(z - z_c(s))^2 @ z = 0mm : h.s}
'First moment of area below z -'S_1(z; s) = $Integral{b(ζ; s)*(z_c(s) - ζ) @ ζ = 0mm : z}
'Shear area -'A_s(s) = I(s)^2/$Integral{S_1(z; s)^2/b(z; s) @ z = 0mm : h.s}
'Calculated results
'Centroids -'z_c = [z_c(1); z_c(2)]
'Areas -'A = [A(1); A(2)]
'Shear areas -'A_s = [A_s(1); A_s(2)]
'Second moments of area -'I = [I(1); I(2)]
'Assignment on elements -'e_S = [1; 2; 2; 1]
'<h4>Element stiffness matrix</h4>
'Elastic properties for element "e"
EA(e) = E.e_M.e*A.e_S.e
EI(e) = E.e_M.e*I.e_S.e
GA_s(e) = G.e_M.e*A_s.e_S.e
k_s(e) = 12*EI(e)/(GA_s(e)*l(e)^2)
α(e) = EA(e)/l(e)','β(e) = EI(e)/(l(e)^3*(1 + k_s(e)))
'Stiffness matrix coefficients for element "e"
k_11(e) = α(e)*(m/kN)','k_22(e) = 12*β(e)*(m/kN)','k_23(e) = 6*β(e)*l(e)*(1/kN)
k_33(e) = (4 + k_s(e))*β(e)*l(e)^2*(1/kNm)
k_36(e) = (2 - k_s(e))*β(e)*l(e)^2*(1/kNm)
'Assembling the 3x3 stiffness matrix blocks for element "e"
k_ii(e) = [k_11(e)|0; k_22(e); k_23(e)|0; k_23(e); k_33(e)]
k_ij(e) = [-k_11(e)|0; -k_22(e); k_23(e)|0; -k_23(e); k_36(e)]
k_ji(e) = transp(k_ij(e))
k_jj(e) = [k_11(e)|0; k_22(e); -k_23(e)|0; -k_23(e); k_33(e)]
'Full 6x6 element stiffness matrix
k_E(e) = stack(augment(k_ii(e); k_ij(e)); augment(k_ji(e); k_jj(e)))
'Stiffness matrices obtained in local coordinates
k_E(1)
k_E(2)
'Stiffness matrices obtained in global coordinates
transp(T(1))*k_E(1)*T(1)
transp(T(2))*k_E(2)*T(2)
'<h4>Global stiffness matrix</h4>
#hide
K = symmetric(3*n_J)
'Add element stiffness matrices
#for e = 1 : n_E
    i = 3*e_J.(e; 1) - 2','j = 3*e_J.(e; 2) - 2
    t = t(e)','tT = transp(t)
```

```
    add(tT*k_ii(e)*t; K; i; i)
    #if j > i
        add(tT*k_ij(e)*t; K; i; j)
    #else
        add(tT*k_ji(e)*t; K; j; i)
    #end if
    add(tT*k_jj(e)*t; K; j; j)
#loop
'Add supports
#for i = 1 : n_c
    j = 3*c.(i; 1) - 2
    add(vec2diag(last(row(c; i); 3)/[kN/m; kN/m; kNm]); K; j; j)
#loop
#show
K
'<h4>Element load vector</h4>
'Lateral load in local CS -'q_E(e) = -q_x.e*s(e) + q_y.e*c(e)
'Axial load in local CS -'n_E(e) = q_x.e*c(e) + q_y.e*s(e)
'Equivalent loads at element endpoints
F_Ex(e) = q_x.e*l(e)/2*(1/kN)','F_Ey(e) = q_y.e*l(e)/2*(1/kN)
M_E(e) = q_E(e)*l(e)^2/12*(1/kNm)
'Load vector -'F_E(e) = [F_Ex(e); F_Ey(e); M_E(e); F_Ex(e); F_Ey(e); -M_E(e)]
#novar
'Results for elements
#for e = 1 : n_E
    'Element E'e' -'F_E(e)
#loop
#varsub
'<h4>Global load vector</h4>
#hide
F = vector(3*n_J)
#for i = 1 : n_q
    e = q.(i; 1)
    #for jj = 1 : 2
        j = 3*e_J.(e; jj) - 3
        F.(j + 1) = F.(j + 1) + take(3*jj - 2; F_E(e))
        F.(j + 2) = F.(j + 2) + take(3*jj - 1; F_E(e))
        F.(j + 3) = F.(j + 3) + take(3*jj; F_E(e))
    #loop
#loop
#show
F
'<h4>Results</h4>
'<p><strong>Solution of the system of equations by Cholesky decomposition</strong></p>
Z = clsolve(K; F)
'<p><strong>Joint displacements</strong></p>
z_J(j) = slice(Z; 3*j - 2; 3*j)
z(j) = round(z_J(j)/δz)*δz*1000*[mm; mm; 1]
```

```
#novar
#for j = 1 : n_J
    z(j)
#loop
#varsub
'<p><strong>Support reactions</strong></p>
r(i) = row(c; i)','j(i) = take(1; r(i))
R(i) = -z_J(j(i))*[m; m; 1]*last(r(i); 3)
#novar
#for i = 1 : n_c
    #val
    '<p>Joint <b>J'j(i)' -
    #equ
    '</b>'R(i)'</p>
#loop
#varsub
'<p><strong>Element end forces</strong></p>
z_E(e) = [z_J(e_J.(e; 1)); z_J(e_J.(e; 2))]
R_E(e) = col(k_E(e)*T(e)*z_E(e) - T(e)*F_E(e); 1)*[1; 1; m; 1; 1; m]*kN
#novar
#for e = 1 : n_E
    R_E(e)
#loop
#varsub
'<p><strong>Element internal forces</strong></p>
N(e; x) = -take(1; R_E(e)) - n_E(e)*x
Q(e; x) = take(2; R_E(e)) + q_E(e)*x
M(e; x) = -take(3; R_E(e)) + take(2; R_E(e))*x + q_E(e)*x^2/2
#hide
w = max(x_J)
h = max(y_J)
W = 240
H = h*W/w
k = W/w
#def red_style$ = style = "stroke:red; stroke-width:1; fill:red"
#deg
#for i = 1 : 3
    #hide
    R(e; x) = take(i; N(e; x); Q(e; x); M(e; x))
    sgn = take(i; 1; 1; -1)
    tol = 0.01*take(i; kN; kN; kNm)
    R_max = $Sup{$Sup{R(e; x) @ x = 0m : l(e)} @ e = 1 : n_E}
    R_min = $Sup{abs($Inf{R(e; x) @ x = 0m : l(e)}) @ e = 1 : n_E}
    k_R = sgn*1m*k/max(R_min; R_max)
    #show
    #if i ≡ 1
        '<p><strong>Axial forces diagram, kN</strong></p>
    #else if i ≡ 2
        '<p><strong>Shear forces diagram, kN</strong></p>
    #else
```

```
        '<p><strong>Bending moments diagram, kNm</strong></p>
    #end if
    #val
    svg$
    '<use href="#frame"/>
    #for e = 1 : n_E
        #hide
        x1 = x_1(e)*k
        y1 = (h - y_1(e))*k
        x2 = x_2(e)*k
        y2 = (h - y_2(e))*k
        c_e = c(e)
        s_e = s(e)
        l_e = l(e)
        st = l_e/10
        xd2 = x1
        yd2 = y1
        #show
        #for j = 0 : 10
            #hide
            xd1 = xd2
            yd1 = yd2
            x = j*st*k
            v = R(e; j*st)
            y = v*k_R
            xd2 = x1 + x*c_e - y*s_e
            yd2 = y1 - x*s_e - y*c_e
            α = 90 + atan2(c_e; s_e)
            #if α ≥ 135
                α = α - 180
            #end if
            #if α < -45
                α = α + 180
            #else if α < 0
                α = 360 + α
            #end if
            d = -15*sign(v*sgn)
            #show
            line$(xd1; yd1; xd2; yd2; red_style$)
            #if (j ≡ 0 ∨ j ≡ 10) ∧ abs(v) > tol
                text$(xd2 + s_e*d; yd2 + d*c_e; α; 'v')
            #end if
            line$(xd1; yd1; xd2; yd2; red_style$)
        #loop
        #hide
        xd1 = x2
        yd1 = y2
        #show
        line$(xd1; yd1; xd2; yd2; red_style$)
    #loop
```

```
    '</svg>
#loop
#equ
'<p><strong>Deformed shape</strong></p>
'Shape function in relative coordinates ξ = x/l (with account to shear deflections)
Φ_1(e; ξ) = 1/(1 + k_s(e))*(1 + k_s(e) - k_s(e)*ξ - 3*ξ^2 + 2*ξ^3)
Φ_2(e; ξ) = ξ*l(e)*m^-1/(1 + k_s(e))*(1 + k_s(e)/2 - (2 + k_s(e)/2)*ξ + ξ^2)
Φ_3(e; ξ) = ξ/(1 + k_s(e))*(k_s(e) + 3*ξ - 2*ξ^2)
Φ_4(e; ξ) = ξ*l(e)*m^-1/(1 + k_s(e))*(-k_s(e)/2 - (1 - k_s(e)/2)*ξ + ξ^2)
'Element endpoint displacements and rotations
z_E,loc(e) = T(e)*z_E(e)
u_1(e) = take(1; z_E,loc(e))','v_1(e) = take(2; z_E,loc(e))','ϕ_1(e) = take(3; z_E,loc(e))
u_2(e) = take(4; z_E,loc(e))','v_2(e) = take(5; z_E,loc(e))','ϕ_2(e) = take(6; z_E,loc(e))
'Displacement functions
u(e; ξ) = u_1(e)*(1 - ξ) + u_2(e)*ξ
v(e; ξ) = v_1(e)*Φ_1(e; ξ) + ϕ_1(e)*Φ_2(e; ξ) + v_2(e)*Φ_3(e; ξ) + ϕ_2(e)*Φ_4(e; ξ)
'Deformed shape, mm
#val
#hide
tol = 0.00001
k_R = 1200
#show
svg$
'<use href="#frame" style="opacity:0.4;"/>
#for e = 1 : n_E
    #hide
    x1 = x_1(e)*k
    y1 = (h - y_1(e))*k
    x2 = x_2(e)*k
    y2 = (h - y_2(e))*k
    c_e = c(e)
    s_e = s(e)
    l_e = l(e)
    u = u(e; 0)
    v = v(e; 0)
    x = u*k_R
    y = v*k_R
    xd2 = x1 + x*c_e - y*s_e
    yd2 = y1 - x*s_e - y*c_e
    #show
    #for j = 0 : 10
        #hide
        xd1 = xd2
        yd1 = yd2
        ξ = j/10
```

```
        u = u(e; ξ)
        v = v(e; ξ)
        x = ξ*l_e*k + u*k_R
        y = v*k_R
        xd2 = x1 + x*c_e - y*s_e
        yd2 = y1 - x*s_e - y*c_e
        d = -15*sign(v)
        #show
        line$(xd1; yd1; xd2; yd2; red_style$)
    #loop
#loop
#for j = 1 : n_J
    #hide
    z_J = z_J(j)
    u = z_J.1
    v = z_J.2
    x = x_J.j*k + u*k_R
    y = (h - y_J.j)*k - v*k_R
    dx = 15*sign(u)
    dy = -15*sign(v)
    #show
    #if abs(u) > tol
        texth$(x + dx; y; 'u*1000')
    #end if
    #if abs(v) > tol
        textv$(x; y + dy; 'v*1000')
    #end if
#loop
'</svg>
#equ
```

III. Output

Analysis of plane frames with arbitrary cross-sections

Joint coordinates

$$\begin{array}{rccccc} & \text{J1} & \text{J2} & \text{J3} & \text{J4} & \text{J5} \\ \vec{x}_J = & [0\ \text{m} & 0\ \text{m} & 8\ \text{m} & 16\ \text{m} & 16\ \text{m}] \end{array}$$

$$\vec{y}_J = [0\ \text{m} \quad 8\ \text{m} \quad 10\ \text{m} \quad 8\ \text{m} \quad 0\ \text{m}]$$

Number of joints - $n_J = \mathbf{len}\left(\vec{x}_J\right) = 5$

Elements

$$\begin{array}{cc} \text{J1} & \text{J2} \end{array}$$

$$e_J = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix}$$

Number of elements - $n_E = \mathbf{n}_{rows}\left(e_J\right) = 4$

Supports

$$\begin{array}{cccc} \text{J} & c_x & c_y & c_r \end{array}$$

$$c = \begin{bmatrix} 1 & 10^{20}\ \text{kN/m} & 10^{20}\ \text{kN/m} & 0\ \text{kNm} \\ 5 & 10^{20}\ \text{kN/m} & 10^{20}\ \text{kN/m} & 10^{20}\ \text{kNm} \end{bmatrix}$$

Number of supports - $n_c = \text{n}_{rows}(c) = 2$

Loads

$$\begin{array}{ccc} e & q_x & q_y \end{array}$$

$$q = \begin{bmatrix} 1 & 10\ \text{kN/m} & 0\ \text{kN/m} \\ 2 & 0\ \text{kN/m} & -20\ \text{kN/m} \\ 3 & 0\ \text{kN/m} & -10\ \text{kN/m} \end{bmatrix}$$

Load values on elements

$$\begin{array}{rcccc} & \text{E1} & \text{E2} & \text{E3} & \text{E4} \\ \vec{q}_x = & [10\ \text{kN/m} & 0\ \text{kN/m} & 0\ \text{kN/m} & 0\ \text{kN/m}] \end{array}$$

$$\vec{q}_y = [0\ \text{kN/m} \quad -20\ \text{kN/m} \quad -10\ \text{kN/m} \quad 0\ \text{kN/m}]$$

Element endpoint coordinates

$x_1(e) = \vec{x}_{J.e_{J.e1}}$, $y_1(e) = \vec{y}_{J.e_{J.e1}}$,

$x_2(e) = \vec{x}_{J.e_{J.e2}}$, $y_2(e) = \vec{y}_{J.e_{J.e2}}$

Element length - $l(e) = \sqrt{\left(x_2(e) - x_1(e)\right)^2 + \left(y_2(e) - y_1(e)\right)^2}$

Element direction - $c(e) = \dfrac{x_2(e) - x_1(e)}{l(e)}$, $s(e) = \dfrac{y_2(e) - y_1(e)}{l(e)}$

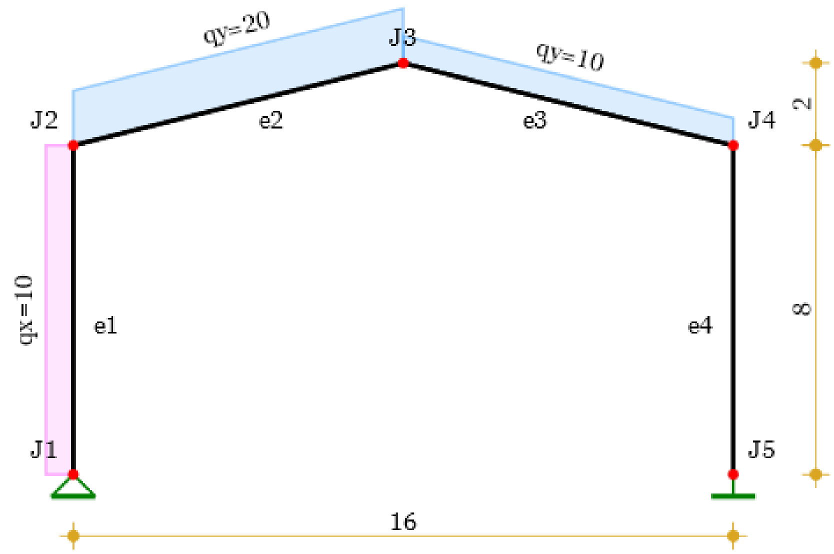
Transformation matrix

Diagonal 3x3 block - $t(e) = [c(e);\ s(e);\ 0 \mid -s(e);\ c(e);\ 0 \mid 0;\ 0;\ 1]$

Generation of the full transformation matrix

$T(e) = \mathbf{add}\left(t(e);\ \mathbf{add}\left(t(e);\ \mathbf{matrix}(6;\ 6);\ 1;\ 1\right);\ 4;\ 4\right)$

Scheme of the structure



Materials

Modules of elasticity - $\vec{E} = [45\ \text{GPa} \quad 35\ \text{GPa}]$

Poisson coefficients - $\vec{\nu} = [0.2 \quad 0.2]$

Shear modules - $\vec{G} = \dfrac{\vec{E}}{2 \cdot (1+\vec{\nu})} = [18.75\ \text{GPa} \quad 14.58\ \text{GPa}]$

Assignment on elements - $\vec{e}_M = [1 \quad 2 \quad 2 \quad 1]$

Cross-sections

Section S1 - $\vec{h}_1 = 500\ \text{mm}$ - circular - $b_C(z) = 2 \cdot \sqrt{\left(\dfrac{\vec{h}_1}{2}\right)^2 - \left(z - \dfrac{\vec{h}_1}{2}\right)^2}$

Section S2 - $\vec{b}_2 = 250\ \text{mm}$, $\vec{h}_2 = 700\ \text{mm}$ - rectangular - $b_R(z) = \vec{b}_2$

General representation - $b(z;\ s) = \textbf{take}(s;\ b_C(z);\ b_R(z))$

Cross section properties

Equations

Area - $A(s) = \displaystyle\int_{0\ \text{mm}}^{\vec{h}_s} b(z;\ s)\ \text{d}z$

First moment of area - $S(s) = \displaystyle\int_{0\ \text{mm}}^{\vec{h}_s} b(z;\ s) \cdot z\ \text{d}z$

Centroid - $z_c(s) = \dfrac{S(s)}{A(s)}$

Second moment of area - $I(s) = \displaystyle\int_{0\ \text{mm}}^{\vec{h}_s} b(z;\ s) \cdot (z - z_c(s))^2\ \text{d}z$

First moment of area below z - $S_1(z\,;\,s)=\int_{0\,\mathrm{mm}}^{z} b(\zeta\,;\,s)\cdot\left(z_c(s)-\zeta\right)\,\mathrm{d}\,\zeta$

Shear area - $A_s(s)=\dfrac{I(s)^2}{\displaystyle\int_{0\,\mathrm{mm}}^{\vec{h}_s}\frac{S_1(z\,;\,s)^2}{b(z\,;\,s)}\,\mathrm{d}z}$

Calculated results

Centroids - $\vec{z}_c=[z_c(1);\ z_c(2)]=[250\ \mathrm{mm}\quad 350\ \mathrm{mm}]$

Areas - $\vec{A}=[A(1);\ A(2)]=[196350\ \mathrm{mm}^2\quad 175000\ \mathrm{mm}^2]$

Shear areas - $\vec{A}_s=[A_s(1);\ A_s(2)]=[176715\ \mathrm{mm}^2\quad 145833\ \mathrm{mm}^2]$

Second moments of area - $\vec{I}=[I(1);\ I(2)]=[3067961576\ \mathrm{mm}^4\quad 7145833333\ \mathrm{mm}^4]$

Assignment on elements - $\vec{e}_s=[1\quad 2\quad 2\quad 1]$

Element stiffness matrix

Elastic properties for element "e"

$EA(e)=\vec{E}_{\vec{e}_{M.e}}\cdot\vec{A}_{e_{S.e}}\ EI(e)=\vec{E}_{\vec{e}_{M.e}}\cdot\vec{I}_{e_{S.e}}\ GA_s(e)=\vec{G}_{\vec{e}_{M.e}}\cdot\vec{A}_{s.e_{S.e}}\ k_s(e)=\dfrac{12\cdot EI(e)}{GA_s(e)\cdot l(e)^2}$

$\alpha(e)=\dfrac{EA(e)}{l(e)}$, $\beta(e)=\dfrac{EI(e)}{l(e)^3\cdot\left(1+k_s(e)\right)}$

Stiffness matrix coefficients for element "e"

$k_{11}(e)=\alpha(e)\cdot\dfrac{\mathrm{m}}{\mathrm{kN}}$, $k_{22}(e)=12\cdot\beta(e)\cdot\dfrac{\mathrm{m}}{\mathrm{kN}}$, $k_{23}(e)=6\cdot\beta(e)\cdot l(e)\cdot\dfrac{1}{\mathrm{kN}}$

$k_{33}(e)=\left(4+k_s(e)\right)\cdot\beta(e)\cdot l(e)^2\cdot\dfrac{1}{\mathrm{kNm}}$, $k_{36}(e)=\left(2-k_s(e)\right)\cdot\beta(e)\cdot l(e)^2\cdot\dfrac{1}{\mathrm{kNm}}$

Assembling the 3x3 stiffness matrix blocks for element "e"

$k_{ii}(e)=\left[k_{11}(e)\,|\ 0;\ k_{22}(e);\ k_{23}(e)\,|\ 0;\ k_{23}(e);\ k_{33}(e)\right]$

$k_{ij}(e)=\left[-k_{11}(e)\,|\ 0;\ -k_{22}(e);\ k_{23}(e)\,|\ 0;\ -k_{23}(e);\ k_{36}(e)\right]$

$k_{ji}(e)=\mathbf{transp}\left(k_{ij}(e)\right)$

$k_{jj}(e)=\left[k_{11}(e)\,|\ 0;\ k_{22}(e);\ -k_{23}(e)\,|\ 0;\ -k_{23}(e);\ k_{33}(e)\right]$

Full 6x6 element stiffness matrix

$k_E(e)=\mathbf{stack}\left(\mathbf{augment}\left(k_{ii}(e);\ k_{ij}(e)\right);\ \mathbf{augment}\left(k_{ji}(e);\ k_{jj}(e)\right)\right)$

Stiffness matrices obtained in local coordinates

$$k_E(1)=\begin{bmatrix} 1104466 & 0 & 0 & -1104466 & 0 & 0 \\ 0 & 3210.66 & 12842.6 & 0 & -3210.66 & 12842.6 \\ 0 & 12842.6 & 68627.8 & 0 & -12842.6 & 34113.2 \\ -1104466 & 0 & 0 & 1104466 & 0 & 0 \\ 0 & -3210.66 & -12842.6 & 0 & 3210.66 & -12842.6 \\ 0 & 12842.6 & 34113.2 & 0 & -12842.6 & 68627.8 \end{bmatrix}$$

$$k_E(2)=\begin{bmatrix} 742765 & 0 & 0 & -742765 & 0 & 0 \\ 0 & 5243.46 & 21619.3 & 0 & -5243.46 & 21619.3 \\ 0 & 21619.3 & 119468 & 0 & -21619.3 & 58809.3 \\ -742765 & 0 & 0 & 742765 & 0 & 0 \\ 0 & -5243.46 & -21619.3 & 0 & 5243.46 & -21619.3 \\ 0 & 21619.3 & 58809.3 & 0 & -21619.3 & 119468 \end{bmatrix}$$

Stiffness matrices obtained in global coordinates

$$\mathbf{transp}\left(T(1)\right)\cdot k_E(1)\cdot T(1)=\begin{bmatrix} 3210.66 & 0 & -12842.6 & -3210.66 & 0 & -12842.6 \\ 0 & 1104466 & 0 & 0 & -1104466 & 0 \\ -12842.6 & 0 & 68627.8 & 12842.6 & 0 & 34113.2 \\ -3210.66 & 0 & 12842.6 & 3210.66 & 0 & 12842.6 \\ 0 & -1104466 & 0 & 0 & 1104466 & 0 \\ -12842.6 & 0 & 34113.2 & 12842.6 & 0 & 68627.8 \end{bmatrix}$$

$$\mathbf{transp}\left(T(2)\right)\cdot k_E(2)\cdot T(2)=\begin{bmatrix} 699382 & 173535 & -5243.46 & -699382 & -173535 & -5243.46 \\ 173535 & 48627.1 & 20973.8 & -173535 & -48627.1 & 20973.8 \\ -5243.46 & 20973.8 & 119468 & 5243.46 & -20973.8 & 58809.3 \\ -699382 & -173535 & 5243.46 & 699382 & 173535 & 5243.46 \\ -173535 & -48627.1 & -20973.8 & 173535 & 48627.1 & -20973.8 \\ -5243.46 & 20973.8 & 58809.3 & 5243.46 & -20973.8 & 119468 \end{bmatrix}$$

Global stiffness matrix

It is formed by adding the 3x3 blocks of the element stiffness matrices to the 3x3 blocks in the global stiffness matrix with indices corresponding to the joint numbers at the ends of the respective elements.

$$K=\begin{bmatrix} 10^{20} & 0 & -12842.6 & -3210.66 & 0 & -12842.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{20} & 0 & 0 & -1104466 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -12842.6 & 0 & 68627.8 & 12842.6 & 0 & 34113.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3210.66 & 0 & 12842.6 & 702592 & 173535 & 7599.17 & -699382 & -173535 & -5243.46 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1104466 & 0 & 173535 & 1153093 & 20973.8 & -173535 & -48627.1 & 20973.8 & 0 & 0 & 0 & 0 & 0 & 0 \\ -12842.6 & 0 & 34113.2 & 7599.17 & 20973.8 & 188096 & 5243.46 & -20973.8 & 58809.3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -699382 & -173535 & 5243.46 & 1398763 & 0 & 10486.9 & -699382 & 173535 & 5243.46 & 0 & 0 & 0 \\ 0 & 0 & 0 & -173535 & -48627.1 & -20973.8 & 0 & 97254.2 & 0 & 173535 & -48627.1 & 20973.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5243.46 & 20973.8 & 58809.3 & 10486.9 & 0 & 238937 & -5243.46 & -20973.8 & 58809.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -699382 & 173535 & -5243.46 & 702592 & -173535 & 7599.17 & -3210.66 & 0 & 12842.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 173535 & -48627.1 & -20973.8 & -173535 & 1153093 & -20973.8 & 0 & -1104466 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5243.46 & 20973.8 & 58809.3 & 7599.17 & -20973.8 & 188096 & -12842.6 & 0 & 34113.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3210.66 & 0 & -12842.6 & 10^{20} & 0 & -12842.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1104466 & 0 & 0 & 10^{20} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12842.6 & 0 & 34113.2 & -12842.6 & 0 & 10^{20} \end{bmatrix}$$

Element load vector

Lateral load in local CS - $q_E(e) = -\vec{q}_{x.e} \cdot s(e) + \vec{q}_{y.e} \cdot c(e)$

Axial load in local CS - $n_E(e) = \vec{q}_{x.e} \cdot c(e) + \vec{q}_{y.e} \cdot s(e)$

Equivalent loads at element endpoints

$$F_{Ex}(e) = \frac{\vec{q}_{x.e} \cdot l(e)}{2} \cdot \frac{1}{\text{kN}},\ F_{Ey}(e) = \frac{\vec{q}_{y.e} \cdot l(e)}{2} \cdot \frac{1}{\text{kN}},\ M_E(e) = \frac{q_E(e) \cdot l(e)^2}{12} \cdot \frac{1}{\text{kNm}}$$

Load vector - $F_E(e) = \left[F_{Ex}(e);\ F_{Ey}(e);\ M_E(e);\ F_{Ex}(e);\ F_{Ey}(e);\ \text{-}M_E(e)\right]$

Results for elements

Element **E1** - $F_E(1) = [40 \quad 0 \quad -53.33 \quad 40 \quad 0 \quad 53.33]$

Element **E2** - $F_E(2) = [0 \quad -82.46 \quad -109.95 \quad 0 \quad -82.46 \quad 109.95]$

Element **E3** - $F_E(3) = [0 \quad -41.23 \quad -54.97 \quad 0 \quad -41.23 \quad 54.97]$

Element **E4** - $F_E(4) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$

Global load vector

$\vec{F} = [40 \quad 0 \quad -53.33 \quad 40 \quad -82.46 \quad -56.62 \quad 0 \quad -123.69 \quad 54.97 \quad 0 \quad -41.23 \quad 54.97 \quad 0 \quad 0 \quad 0]$

Results

Solution of the system of equations by Cholesky decomposition

$\vec{Z} = \textbf{clsolve}(K\ ;\ \vec{F})$ = [1.88×10^{-19} -1.39×10^{-18} -0.000928 0.00809 -0.000126 -0.00274 0.0119 -0.0157 0.000699 0.0157 -9.84×10^{-5} 0.000846 6.12×10^{-19} -1.09×10^{-18} -2.3×10^{-18}]

Joint displacements

The displacements for each joint are extracted from the global vector:

$$z_J(j) = \textbf{slice}(\vec{Z}\ ;\ 3 \cdot j - 2\ ;\ 3 \cdot j),\ z(j) = \textbf{round}\left(\frac{z_J(j)}{10^{-12}}\right) \cdot 10^{-9} \cdot [\text{mm; mm; } 1]$$

Values for joints

	u	v	$\varphi \cdot 10^3$
Joint **J1** - $z(1)=$	[0 mm	0 mm	-0.928]
Joint **J2** - $z(2)=$	[8.09 mm	-0.126 mm	-2.74]
Joint **J3** - $z(3)=$	[11.88 mm	-15.67 mm	0.699]
Joint **J4** - $z(4)=$	[15.67 mm	-0.0984 mm	0.846]
Joint **J5** - $z(5)=$	[0 mm	0 mm	0]

Support reactions

They are determined by multiplying the joint displacements by the respective spring constants.

$r(i) = \mathbf{row}(c;\ i)$, $j(i) = \mathbf{take}(1;\ r(i))$, $R(i) = -z_J(j(i)) \cdot [\mathrm{m};\ \mathrm{m};\ 1] \cdot \mathbf{last}(r(i);\ 3)$

Values for supports

	F_x	F_y	M
Joint **J1** - $R(1) = [$	-18.84 kN	138.69 kN	0 kNm $]$
Joint **J5** - $R(2) = [$	-61.16 kN	108.7 kN	230.05 kNm $]$

Element end forces

Element endpoint displacements - $z_E(e) = \left[z_J(e_{J.e1});\ z_J(e_{J.e2})\right]$

Endpoint forces are determined by multiplying the element stiffness matrix by the end-point displacements in local CS and subtracting the element load vector values in local CS.

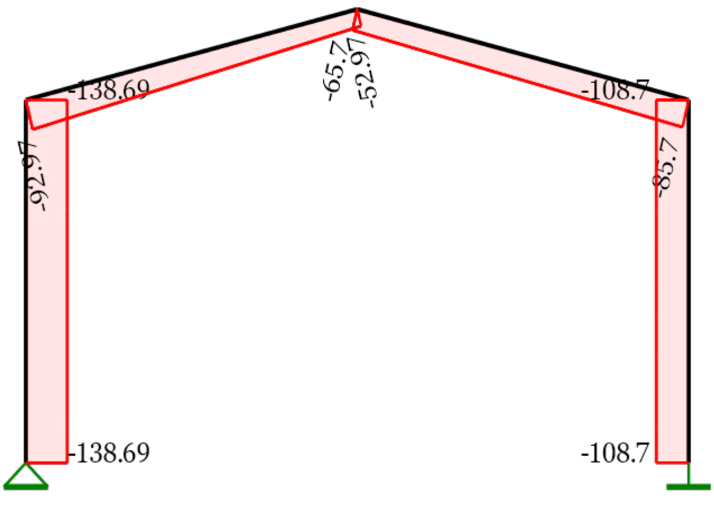
$R_E(e) = \mathrm{col}\left(k_E(e) \cdot T(e) \cdot z_E(e) - T(e) \cdot F_E(e);\ 1\right) \cdot [1;\ 1;\ m;\ 1;\ 1;\ m] \cdot kN$

End forces values for different elements

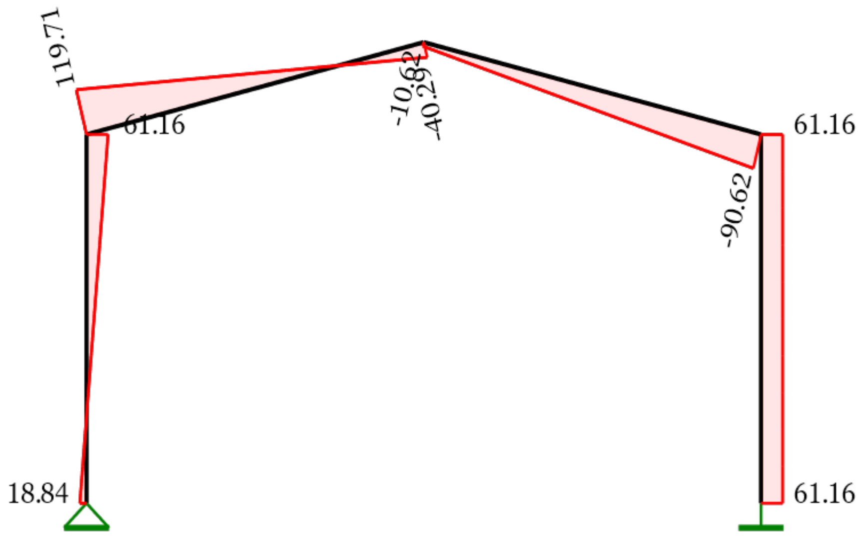
	F_{x1}	F_{y1}	M_1	F_{x2}	F_{y2}	M_2
$R_E(1) = [$	138.69 kN	18.84 kN	-1.42×10^{-14} kNm	-138.69 kN	61.16 kN	-169.29 kNm $]$
$R_E(2) = [$	92.97 kN	119.71 kN	169.29 kNm	-52.97 kN	40.29 kN	158.18 kNm $]$
$R_E(3) = [$	65.7 kN	-10.62 kN	-158.18 kNm	-85.7 kN	90.62 kN	-259.24 kNm $]$
$R_E(4) = [$	108.7 kN	61.16 kN	259.24 kNm	-108.7 kN	-61.16 kN	230.05 kNm $]$

Element internal forces

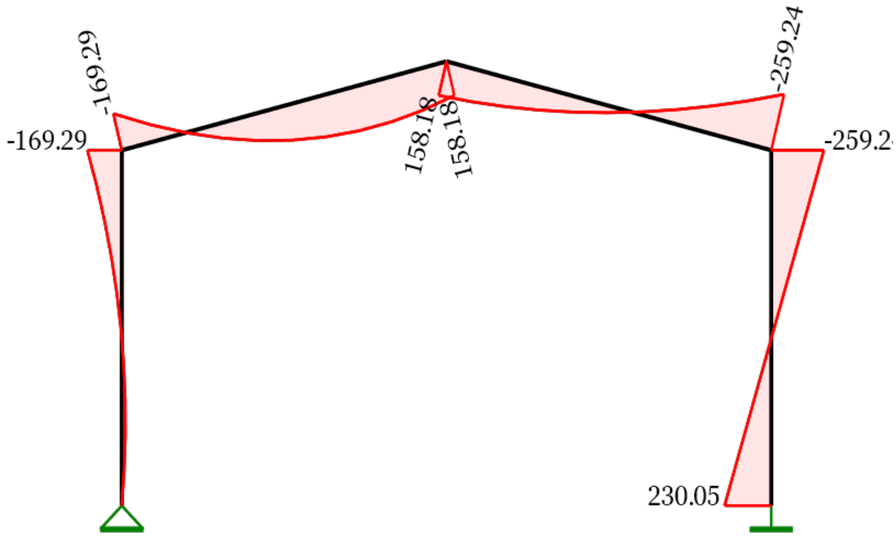
Axial forces - $N(e;\ x) = -\mathbf{take}(1;\ R_E(e)) - n_E(e) \cdot x$, kN



Shear forces- $Q(e;\ x) = \mathbf{take}(2;\ R_E(e)) + q_E(e) \cdot x$, kN



Bending moments- $M\left(e;\ x\right)=-\,\mathbf{take}\left(3;\ R_E\left(e\right)\right)+\mathbf{take}\left(2;\ R_E\left(e\right)\right)\cdot x+\dfrac{q_E\left(e\right)\cdot x^2}{2}$, kNm



Deformed shape

Shape function in relative coordinates ξ = x/l (with account to shear deflections)

$$\Phi_1\left(e;\ \xi\right)=\frac{1}{1+k_s\left(e\right)}\cdot\left(1+k_s\left(e\right)-k_s\left(e\right)\cdot\xi-3\cdot\xi^2+2\cdot\xi^3\right)$$

$$\Phi_2\left(e;\ \xi\right)=\frac{\xi\cdot l\left(e\right)\cdot m^{-1}}{1+k_s\left(e\right)}\cdot\left(1+\frac{k_s\left(e\right)}{2}-\left(2+\frac{k_s\left(e\right)}{2}\right)\cdot\xi+\xi^2\right)$$

$$\Phi_3\left(e;\ \xi\right)=\frac{\xi}{1+k_s\left(e\right)}\cdot\left(k_s\left(e\right)+3\cdot\xi-2\cdot\xi^2\right)$$

$$\Phi_4\left(e;\ \xi\right)=\frac{\xi\cdot l\left(e\right)\cdot m^{-1}}{1+k_s\left(e\right)}\cdot\left(-\frac{k_s\left(e\right)}{2}-\left(1-\frac{k_s\left(e\right)}{2}\right)\cdot\xi+\xi^2\right)$$

Element endpoint displacements and rotations

$z_{E,loc}(e) = T(e) \cdot z_E(e)$

$u_1(e) = \textbf{take}\left(1;\ z_{E,loc}(e)\right),\ v_1(e) = \textbf{take}\left(2;\ z_{E,loc}(e)\right),\ \varphi_1(e) = \textbf{take}\left(3;\ z_{E,loc}(e)\right)$

$u_2(e) = \textbf{take}\left(4;\ z_{E,loc}(e)\right),\ v_2(e) = \textbf{take}\left(5;\ z_{E,loc}(e)\right),\ \varphi_2(e) = \textbf{take}\left(6;\ z_{E,loc}(e)\right)$

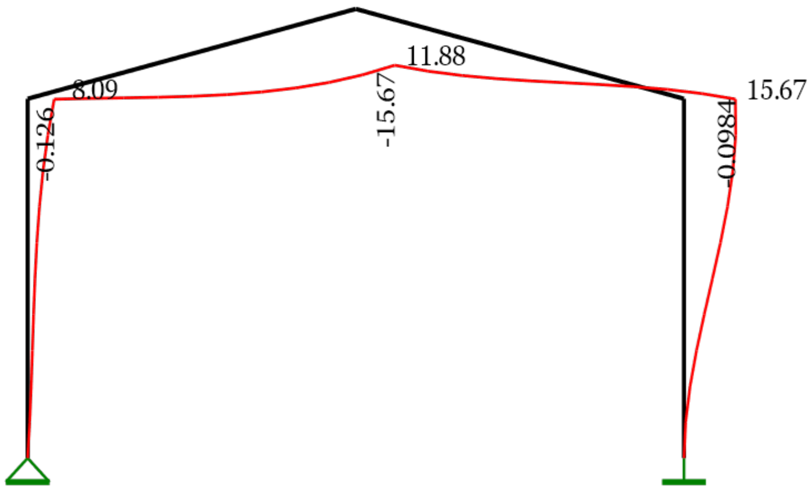
Displacement functions

$$u(e;\ \xi) = u_1(e) \cdot (1-\xi) + u_2(e) \cdot \xi + \frac{n_E(e) \cdot m}{EA(e)} \cdot \xi \cdot (1-\xi)$$

$$v(e;\ \xi) = v_1(e) \cdot \Phi_1(e;\ \xi) + \varphi_1(e) \cdot \Phi_2(e;\ \xi) + v_2(e) \cdot \Phi_3(e;\ \xi) + \varphi_2(e) \cdot \Phi_4(e;\ \xi)$$

$$+ \frac{q_E(e) \cdot l(e)^4}{24 \cdot EI(e)} \cdot \frac{\xi^2 \cdot (1-\xi)^2}{\mathrm{m}} + \frac{q_E(e) \cdot l(e)^2}{2 \cdot GA_s(e)} \cdot \frac{\xi \cdot (1-\xi)}{\mathrm{m}}$$

Deformed shape, mm



IV. Comparison with Stadyps 6.0 software

Input data

Number of joints: 5 , Number of elements: 4

Joint coordinates

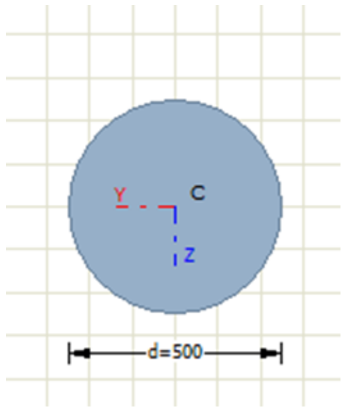
№	X	Y	№	X	Y	№	X	Y	№	X	Y	№	X	Y
1	0.00	0.00	**2**	0.00	8.00	**4**	16.00	8.00	**3**	8.00	10.00	**5**	16.00	0.00

Materials

№	E	ν	α
1	45000000	0.200	0.000012
2	35000000	0.200	0.000012

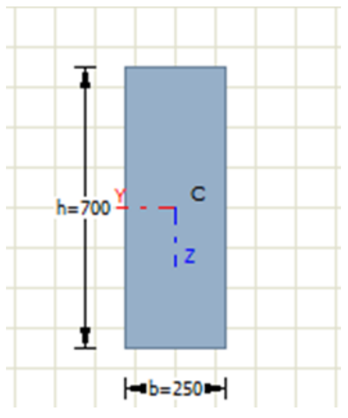
Cross-sections

Section S1 - CIRCULAR



d [mm]	C_Z [cm]	C_Y [cm]	A [cm^2]		
500.0	25.0	25.0	1963.5		
I_Y [cm^4]	I_Y [cm]	W_Y [cm^3]	I_Z [cm^4]	I_Z [cm]	W_Z [cm^3]
306796	12.5	12271.	306796	12.5	12271.
I_t [cm^4]	W_t [cm^3]	A_{qZ} [cm^2]	A_{qY} [cm^2]		
613592	24543.	1767.1	1767.1		

Section R1 - RECTANGULAR



b [mm]	h [mm]	C_Z [cm]	C_Y [cm]	A [cm^2]	
250.0	700.0	35.0	12.5	1750.0	
I_Y [cm^4]	I_Y [cm]	W_Y [cm^3]	I_Z [cm^4]	I_Z [cm]	W_Z [cm^3]
714583	20.2	20416.	91145.	7.2	7291.7
I_t [cm^4]	W_t [cm^3]	A_{qZ} [cm^2]	A_{qY} [cm^2]		
282662	11767.	1458.3	1458.3		

Elements

№	J_1	J_2	Type	Material	Section	ISZ	Winkl. Const.
1	1	2	3	1	S1	0	0.0
2	2	3	3	2	R1	0	0.0
3	3	4	3	2	R1	0	0.0
4	4	5	3	1	S1	0	0.0

Supports and springs

№	Kxm	Kym	Kz	Kxy	Kxz	Kyz	Joints
4	-1	-1	0	0	0	0	1
7	-1	-1	-1	0	0	0	5

Linearly distributed loads

№	Type	q_1	q_2	A	L	Elements
1	qx	10.00	10.00	0.00	0.00	1
2	qy	10.00	10.00	0.00	0.00	2, 2, 3

Results

Joint displacements

Joint	Ux	Uy	Rz	Joint	Ux	Uy	Rz
1	0.000	0.000	0.001	2	0.008	0.000	0.003
3	0.012	0.016	-0.001	4	0.016	0.000	-0.001
5	0.000	0.000	0.000				

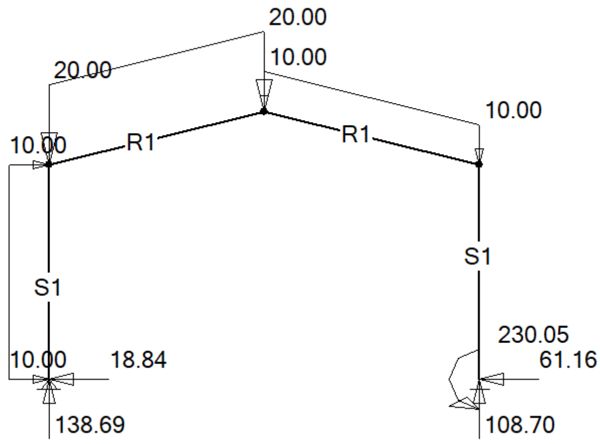
Reactions in supports

Joint	Fx	Fy	Mz	Възел	Fx	Fy	Mz
1	-18.839	-138.687	0.000	5	-61.161	-108.700	-230.046

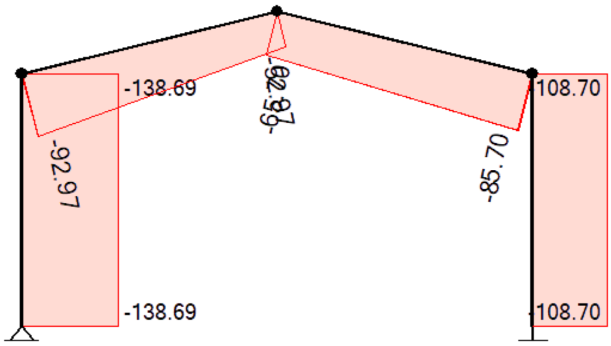
Internal forces in elements

Element	Joint	M	N	Q	Joint	M	N	Q
1	1	0.00	-138.69	18.84	2	-169.29	-138.69	-61.16
2	2	-169.29	-92.97	119.71	3	158.18	-52.97	-40.29
3	3	158.18	-65.70	-10.62	4	-259.24	-85.70	-90.62
4	4	-259.24	-108.70	61.16	5	230.05	-108.70	61.16

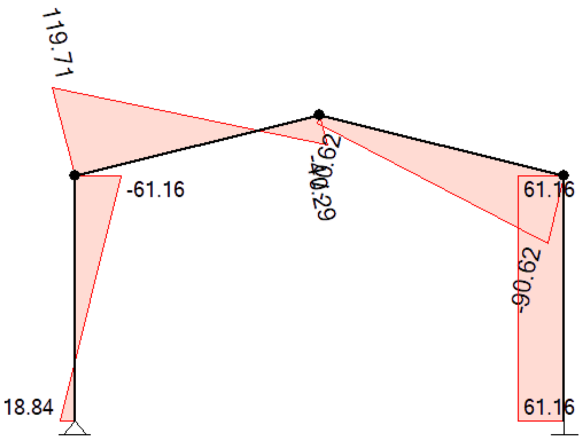
Loads and support reactions, kN



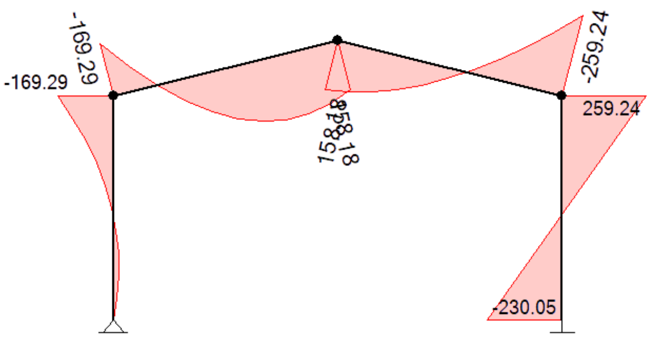
Axial forces, kN



Shear forces, kN



Bending moments, kNm



The results are almost identical.