

# A CALCPAD PROGRAM

## FOR ANALYSIS OF PLANE FRAMES WITH VARIABLE CROSS-SECTIONS



(using the finite element method)

by

Eng. Nedelcho Ganchovsky

# Table of contents

I. Introduction.....	1
II. Calcpad source code.....	2
III. Output.....	12
IV. Comparison with SAP 2000.....	21
V. Conclusions.....	25

## I. Introduction

This Calcpad program performs analysis of plane frames with variable cross-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

```
1  #include svg_drawing.cpd
2  "Analysis of plane frames with variable cross-sections
3  '<h4>Joint coordinates - xJ; yJ</h4>
4  #hide
5  #deg
6  dz = 10^-12
7  Precision = 10^-8
8  x_J = [0; 0; 8; 16; 16]*m
9  y_J = [0; 8; 10; 8; 0]*m
10 #show
11 x_J', 'y_J
12 n_J = len(x_J)
13 '<h4>Elements - [J1; J2]</h4>
14 #hide
15 e_J = [1; 2|3; 2|3; 4|5; 4]
16 #show
17 transp(e_J)
18 n_E = n_rows(e_J)
19 'Element endpoint coordinates
20 x_1(e) = x_J.e_J.(e; 1)', 'y_1(e) = y_J.e_J.(e; 1)
21 x_2(e) = x_J.e_J.(e; 2)', 'y_2(e) = y_J.e_J.(e; 2)
22 'Element length - 'l(e) = sqrt((x_2(e) - x_1(e))^2 + (y_2(e) - y_1(e))^2)
23 'Element directions
24 c(e) = (x_2(e) - x_1(e))/l(e)', 's(e) = (y_2(e) - y_1(e))/l(e)
25 'Transformation matrix
26 'Diagonal 3x3 block - 't(e) = [c(e); s(e); 0|-s(e); c(e); 0|0; 0; 1]
27 'Generation of the full transformation matrix
28 T(e) = add(t(e); add(t(e); matrix(6; 6); 1; 1); 4; 4)
29 '<h4>Supports - [Joint; cx; cy; cr]</h4>
30 #hide
31 c = [1; 10^20kN/m; 10^20kN/m; 0kNm|5; 10^20kN/m; 10^20kN/m; 10^20kNm]
32 #show
33 c
34 n_c = n_rows(c)
35 '<h4>Loads - [Element, qx, qy]</h4>
36 #hide
37 q = [1; 10kN/m; 0kN/m|2; 0kN/m; -20kN/m|3; 0kN/m; -10kN/m]
38 n_q = n_rows(q)
39 q_x = vector(n_E)*kN/m
40 q_y = vector(n_E)*kN/m
41 $Repeat{q_x.(q.(i; 1)) = q_x.(q.(i; 1)) + q.(i; 2) @ i = 1 : n_q}
42 $Repeat{q_y.(q.(i; 1)) = q_y.(q.(i; 1)) + q.(i; 3) @ i = 1 : n_q}
43 #show
44 q
45 'Load values on elements
46 q_x
47 q_y
```

```

48 '<h4>Scheme of the structure</h4>
49 #hide
50 w = max(x_J)
51 h = max(y_J)
52 W = 240
53 H = h*W/w
54 k = W/w
55 #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:5pt; width:'W + 150'pt; height:'H + 200*H/W'pt">
56 #def thin_style$ = style = "stroke:green; stroke-width:1; fill:none"
57 #def thick_style$ = style = "stroke:green; stroke-width:2; fill:none"
58 k_q = m/kN
59 #show
60 #val
61 svg$
62 #for i = 1 : n_E
63     #hide
64     x1 = x_1(i)*k
65     y1 = (h - y_1(i))*k
66     x2 = x_2(i)*k
67     y2 = (h - y_2(i))*k
68     q_xi = q_x.i
69     q_yi = q_y.i
70     α = atan2(c(i); s(i))
71     #if α ≥ 135
72         α = α - 180
73     #end if
74     #if α < -45
75         α = α + 180
76     #else if α < 0
77         α = 360 + α
78     #end if
79     #if q_xi ≠ 0kN/m
80         #hide
81         x3 = x2 - q_xi*k_q, 'y3 = y2
82         x4 = x1 - q_xi*k_q, 'y4 = y1
83         x = (x3 + x4)/2 - 5*sign(q_xi)
84         y = (y3 + y4)/2
85         #show
86         '<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;" />
87         text$(x;y;α;qx='abs(q_xi)')
88     #end if
89     #if q_yi ≠ 0kN/m
90         #hide
91         x3 = x2, 'y3 = y2 + q_yi*k_q
92         x4 = x1, 'y4 = y1 + q_yi*k_q
93         x = (x3 + x4)/2

```

```

94         y = (y3 + y4)/2 + 5*sign(q_yi)
95         #show
96         '<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;" />
97         text$(x;y;α;qy='abs(q_yi)')
98     #end if
99     #show
100     line$(x1; y1; x2; y2; main_style$)
101 #loop
102 '<g id="frame">
103 #for i = 1 : n_E
104     #hide
105     x1 = x_1(i)*k
106     y1 = (h - y_1(i))*k
107     x2 = x_2(i)*k
108     y2 = (h - y_2(i))*k
109     #show
110     line$(x1; y1; x2; y2; main_style$)
111 #loop
112 #for i = 1 : n_c
113     #hide
114     j = c.(i; 1)
115     x1 = x_J.j*k
116     y1 = (h - y_J.j)*k
117     δ = w/30*k*sign(x1 - w/2*k)
118     x2 = x1 - δ
119     y2 = y1 - abs(δ)
120     x3 = x1 + δ
121     y3 = y1 + abs(δ)
122     #show
123     #if c.(i; 2) ≠ 0kN/m
124         #if c.(i; 3) ≠ 0kN/m
125             #if c.(i; 4) ≠ 0kNm
126                 line$(x1; y1; x1; y3; thin_style$)
127                 line$(x2; y3; x3; y3; thick_style$)
128             #else
129                 line$(x2; y3; x3; y3; thick_style$)
130                 line$(x2; y3; x1; y1; thin_style$)
131                 line$(x3; y3; x1; y1; thin_style$)
132             #end if
133         #else
134             #if c.(i; 4) ≠ 0kNm
135                 line$(x1; y1; x2; y1; thin_style$)
136                 line$(x2; y2; x2; y3; thick_style$)
137                 line$(x2 - δ/2; y2; x2 - δ/2; y3; thick_style$)
138             #else
139                 line$(x2; y2; x1; y1; thin_style$)
140                 line$(x2; y3; x1; y1; thin_style$)
141                 line$(x2; y2; x2; y3; thin_style$)

```

```

142         line$(x2 -  $\delta/2$ ; y2; x2 -  $\delta/2$ ; y3; thick_style$)
143     #end if
144 #end if
145 #else
146     #if c.(i; 3)  $\neq$  0kN/m
147         #if c.(i; 4)  $\neq$  0kNm
148             line$(x1; y1; x1; y3; thin_style$)
149             line$(x2; y3; x3; y3; thick_style$)
150             line$(x2; y3 + abs( $\delta$ )/2; x3; y3 + abs( $\delta$ )/2; thick_style$)
151         #else
152             line$(x2; y3; x3; y3; thin_style$)
153             line$(x2; y3; x1; y1; thin_style$)
154             line$(x3; y3; x1; y1; thin_style$)
155             line$(x2; y3 + abs( $\delta$ )/2; x3; y3 + abs( $\delta$ )/2; thick_style$)
156         #end if
157     #else
158         line$(x2; y2; x3; y3; thick_style$)
159     #end if
160 #end if
161 #loop
162 '</g>'
163 #for i = 1 : n_E
164     #hide
165     x = (x_1(i) + x_2(i))*k/2
166     y = (h - (y_1(i) + y_2(i))/2)*k
167     #show
168     text$(x + 0.8m*sign(w/2 - x)*k; y + 0.6m*k; e'i')
169 #loop
170 #for i = 1 : n_J
171     point$(x_J.i*k; (h - y_J.i)*k; point_style$)
172     text$((x_J.i - 0.7m*sign(w/2 - x_J.i))*k; (h - y_J.i - 0.4m)*k;
173     J'i')
174 #loop
175 dimv$((w + 2m)*k; (h - y_J.4)*k; h*k; 'y_J.4')
176 dimv$((w + 2m)*k; 0; (h - y_J.4)*k; 'h - y_J.4')
177 dimh$(0; w*k; (h + 1.5m)*k; 'w')
178 '</svg>'
179 #equ
180 '<h4>Materials</h4>'
181 'Modules of elasticity -'E = [45; 35]*GPa
182 'Poisson coefficients -'v = [0.2; 0.2]
183 'Shear modules -'G = E/(2*(1 + v))
184 'Assignments on elements -'e_M = [1; 2; 2; 1]
185 '<h4>Cross-sections</h4>'
186 #hide
187 b = vector(2); h = vector(2)
188 #show
189 'Section S1 -'b_1 = 300mm; 'h_1 = 300mm
190 'Section S2 -'b_2 = 300mm; 'h_2 = 900mm
191 'General representation

```

```

191 'Width -'b(ξ; z) = b_1 + (b_2 - b_1)*ξ
192 'Height -'h(ξ) = h_1 + (h_2 - h_1)*ξ
193 '<h4>Cross section properties</h4>'
194 'Area -'A(ξ) = $Area{b(ξ; z) @ z = 0mm : h(ξ)}|cm^2
195 'First moment of area -'S(ξ) = $Area{b(ξ; z)*z @ z = 0mm : h(ξ)}|cm^3
196 'Centroid -'z_c(ξ) = S(ξ)/A(ξ)|mm
197 'Second moment of area -'I(ξ) = $Area{b(ξ; z)*(z - z_c(ξ))^2 @ z = 0mm :
h(ξ)}|cm^4
198 'First moment of area below z -'S_1(ξ; z) = $Area{b(ξ; ζ)*(z_c(ξ) - ζ) @
ζ = 0.1mm : z}
199 'Shear area -'A_Q(ξ) = I(ξ)^2/$Area{S_1(ξ; z)^2/b(ξ; z) @ z = 0.1mm :
h(ξ) - 0.1mm}
200 '<h4>Element stiffness matrix</h4>'
201 'Elastic properties for element "e"
202 EA(e; x) = E.e_M.e*A(x/l(e))
203 EI(e; x) = E.e_M.e*I(x/l(e))
204 GA_Q(e; x) = G.e_M.e*A_Q(x/l(e))
205 'Stiffness matrix for element with variable cross-section
206 'Displacement due to F<sub>x</sub> = 1 in primary system -'u_F(e) =
$Area{1/EA(e; x) @ x = 0m : l(e)}
207 'Displacement due to F<sub>y1</sub> = 1 in primary system, with account
of shear deflections -'v_F1(e) = $Area{x^2/EI(e; x) @ x = 0m : l(e)} +
$Area{1/GA_Q(e; x) @ x = 0m : l(e)}
208 'Rotation due to F<sub>y1</sub> = 1 in primary system -'φ_F1(e) = -
$Area{x/EI(e; x) @ x = 0m : l(e)}' + 0
209 'Displacement due to M<sub>1</sub> = 1 in primary system -'v_M1(e) =
φ_F1(e)
210 'Rotation due to M<sub>1</sub> = 1 in primary system -'φ_M1(e) =
$Area{1/EI(e; x) @ x = 0m : l(e)}' + 0
211 'Determinant -'D_1(e) = φ_M1(e)*v_F1(e) - φ_F1(e)^2
212 'Displacement due to F<sub>y2</sub> = 1 in primary system -'v_F2(e) =
$Area{(l(e) - x)^2/EI(e; x) @ x = 0m : l(e)} + $Area{1/GA_Q(e; x) @ x =
0m : l(e)}
213 'Rotation due to F<sub>2</sub> = 1 in primary system -'φ_F2(e) =
$Area{(l(e) - x)/EI(e; x) @ x = 0m : l(e)}' + 0
214 'Displacement due to M<sub>2</sub> = 1 in primary system -'v_M2(e) =
φ_F2(e)
215 'Rotation due to M<sub>2</sub> = 1 in primary system -'φ_M2(e) = φ_M1(e)
216 'Determinant -'D_2(e) = φ_M2(e)*v_F2(e) - φ_F2(e)^2
217 '3x3 blocks of the stiffness matrix for element "e"
218 k_ii(e) = [D_1(e)/u_F(e)*kN*m|0; φ_M1(e)*kNm; -φ_F1(e)*kN|0; -φ_F1(e)*kN;
v_F1(e)*(kN/m)]*(kN^-2/D_1(e))
219 k_ij(e) = [-D_2(e)/u_F(e)*kN*m|0; -φ_M2(e)*kNm; φ_F2(e)*kN|0; (φ_F2(e) -
φ_M2(e)*1(e))*kN; -(v_F2(e) - φ_F2(e)*1(e))*(kN/m)]*(kN^-2/D_2(e))
220 k_ji(e) = [-D_1(e)/u_F(e)*kN*m|0; -φ_M1(e)*kNm; φ_F1(e)*kN|0; (φ_F1(e) +
φ_M1(e)*1(e))*kN; -(v_F1(e) + φ_F1(e)*1(e))*(kN/m)]*(kN^-2/D_1(e))
221 k_jj(e) = [D_2(e)/u_F(e)*kN*m|0; φ_M2(e)*kNm; -φ_F2(e)*kN|0; -φ_F2(e)*kN;
v_F2(e)*(kN/m)]*(kN^-2/D_2(e))
222 'Full element stiffness matrix
223 k_E(e) = stack(augment(k_ii(e); k_ij(e)); augment(k_ji(e); k_jj(e)))

```

```

224 'Stiffness matrices obtained in local coordinates
225 k_E(1)
226 k_E(2)
227 'Stiffness matrices obtained in global coordinates
228 transp(T(1))*k_E(1)*T(1)
229 transp(T(2))*k_E(2)*T(2)
230 '<h4>Global stiffness matrix</h4>
231 #hide
232 K = symmetric(3*n_J)
233 'Add element stiffness matrices
234 #for e = 1 : n_E
235     i = 3*e_J.(e; 1) - 2', 'j = 3*e_J.(e; 2) - 2
236     t = t(e)', 'tT = transp(t)
237     add(tT*k_ii(e)*t; K; i; i)
238     #if j > i
239         add(tT*k_ij(e)*t; K; i; j)
240     #else
241         add(tT*k_ji(e)*t; K; j; i)
242     #end if
243     add(tT*k_jj(e)*t; K; j; j)
244 #loop
245 'Add supports
246 #for i = 1 : n_c
247     j = 3*c.(i; 1) - 2
248     add(vec2diag(last(row(c; i); 3)/[kN/m; kN/m; kNm])); K; j; j)
249 #loop
250 #show
251 K
252 '<h4>Element load vector</h4>
253 'Load functions
254 'Shear - 'q_E(e; x) = -q_x.e*s(e) + q_y.e*c(e)
255 'Axial - 'n_E(e; x) = q_x.e*c(e) + q_y.e*s(e)
256 'Functions of internal forces in primary system
257 'Axial forces - 'N_0(e; x) = -$Area{n_E(e; ξ) @ ξ = 0m : x}
258 'Shear forces - 'Q_0(e; x) = $Area{q_E(e; ξ) @ ξ = 0m : x}
259 'Bending moments - 'M_0(e; x) = $Area{q_E(e; ξ)*(x - ξ) @ ξ = 0m : x}
260 'Reactions at element ends
261 'Displacements along "x" due to axial loads - 'u_n(e) = $Area{N_0(e; x)/
EA(e; x) @ x = 0m : l(e)}
262 'Displacements along "y" due to lateral loads - 'v_q(e) = $Area{M_0(e;
x)*x/EI(e; x) @ x = 0m : l(e)} + $Area{Q_0(e; x)/GA_Q(e; x) @ x = 0m :
l(e)}
263 'Rotations due to lateral loads - 'φ_q(e) = -$Area{M_0(e; x)/EI(e; x) @ x
= 0m : l(e)}
264 'Element endpoint loads in local coordinate system
265 'For joint "1":
266 F_x1(e) = -u_n(e)/u_F(e)
267 F_y1(e) = (φ_M1(e)*v_q(e) - φ_F1(e)*φ_q(e))/D_1(e)
268 M_1(e) = (v_F1(e)*φ_q(e) - φ_F1(e)*v_q(e))/D_1(e)
269 'For joint "2":

```



```

270 F_x2(e) = -F_x1(e) - N_0(e; 1(e))
271 F_y2(e) = -F_y1(e) + Q_0(e; 1(e))
272 M_2(e) = -M_1(e) + F_y1(e)*l(e) - M_0(e; 1(e))
273 'Element endpoint loads in global coordinate system
274 'For joint "1":
275 F'_x1(e) = F_x1(e)*c(e) - F_y1(e)*s(e)
276 F'_y1(e) = F_x1(e)*s(e) + F_y1(e)*c(e)
277 'For joint "2":
278 F'_x2(e) = F_x2(e)*c(e) - F_y2(e)*s(e)
279 F'_y2(e) = F_x2(e)*s(e) + F_y2(e)*c(e)
280 'Element load vector
281 F_E(e) = [F'_x1(e); F'_y1(e); M_1(e)*m^-1; F'_x2(e); F'_y2(e); M_2(e)*m^-1]*kN^-1
282 #novar
283 #for e = 1 : n_E
284     F_E(e)
285 #loop
286 #varsub
287 '<h4>Global load vector</h4>
288 #hide
289 F = vector(3*n_J)
290 #for i = 1 : n_q
291     e = q.(i; 1)
292     #for jj = 1 : 2
293         j = 3*e_J.(e; jj) - 3
294         F.(j + 1) = F.(j + 1) + take(3*jj - 2; F_E(e))
295         F.(j + 2) = F.(j + 2) + take(3*jj - 1; F_E(e))
296         F.(j + 3) = F.(j + 3) + take(3*jj; F_E(e))
297     #loop
298 #loop
299 #show
300 F
301 '<h4>Results</h4>
302 '<p><strong>Solution of the system of equations</strong></p>
303 Z = clsolve(K; F)
304 '<p><strong>Joint displacements</strong></p>
305 z_J(j) = slice(Z; 3*j - 2; 3*j)
306 z(j) = round(z_J(j)/δz)*δz*1000*[mm; mm; 1]
307 #novar
308 #for j = 1 : n_J
309     z(j)
310 #loop
311 #varsub
312 '<p><strong>Support reactions</strong></p>
313 r(i) = row(c; i)', 'j(i) = take(1; r(i))
314 R(i) = -z_J(j(i))*[m; m; 1]*last(r(i); 3)
315 #novar
316 #for i = 1 : n_c
317     #val
318     '<p>Joint <b>J'j(i)' -

```

```

319     #equ
320     '</b>'R(i)'\</p>
321 #loop
322 #varsub
323 '<p><strong>Element end forces</strong></p>
324 z_E(e) = [z_J(e_J.(e; 1)); z_J(e_J.(e; 2))]
325 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
326 #novar
327 #for e = 1 : n_E
328     R_E(e)
329 #loop
330 #varsub
331 '<p><strong>Element internal forces</strong></p>
332 N(e; x) = -take(1; R_E(e)) + N_0(e; x)
333 Q(e; x) = take(2; R_E(e)) + Q_0(e; x)
334 M(e; x) = -take(3; R_E(e)) + take(2; R_E(e))*x + M_0(e; x)
335 #hide
336 w = max(x_J)
337 h = max(y_J)
338 W = 240
339 H = h*W/w
340 k = W/w
341 #def red_style$ = style = "stroke:red; stroke-width:1; fill:red"
342 #deg
343 #for i = 1 : 3
344     #hide
345     R(e; x) = take(i; N(e; x); Q(e; x); M(e; x))
346     sgn = take(i; 1; 1; -1)
347     tol = 0.01*take(i; kN; kN; kNm)
348     R_max = $Sup{$Sup{R(e; x) @ x = 0m : 1(e)} @ e = 1 : n_E}
349     R_min = $Sup{abs($Inf{R(e; x) @ x = 0m : 1(e)}) @ e = 1 : n_E}
350     k_R = sgn*1m*k/max(R_min; R_max)
351     #show
352     #if i == 1
353         '<p><strong>Axial forces diagram, kN</strong></p>
354     #else if i == 2
355         '<p><strong>Shear forces diagram, kN</strong></p>
356     #else
357         '<p><strong>Bending moments diagram, kNm</strong></p>
358     #end if
359     #val
360     svg$
361     '<use href="#frame"/>
362     #for e = 1 : n_E
363         #hide
364         x1 = x_1(e)*k
365         y1 = (h - y_1(e))*k
366         x2 = x_2(e)*k
367         y2 = (h - y_2(e))*k
368         c_e = c(e)

```

```

369     s_e = s(e)
370     l_e = l(e)
371     st = l_e/10
372     xd2 = x1
373     yd2 = y1
374     #show
375     #for j = 0 : 10
376         #hide
377         xd1 = xd2
378         yd1 = yd2
379         x = j*st*k
380         v = R(e; j*st)
381         y = v*k_R
382         xd2 = x1 + x*c_e - y*s_e
383         yd2 = y1 - x*s_e - y*c_e
384         α = 90 + atan2(c_e; s_e)
385         #if α ≥ 135
386             α = α - 180
387         #end if
388         #if α < -45
389             α = α + 180
390         #else if α < 0
391             α = 360 + α
392         #end if
393         d = -15*sign(v*sgn)
394         #show
395         line$(xd1; yd1; xd2; yd2; red_style$)
396         #if (j == 0 ∨ j == 10) ∧ abs(v) > tol
397             text$(xd2 + s_e*d; yd2 + d*c_e; α; 'v')
398         #end if
399         line$(xd1; yd1; xd2; yd2; red_style$)
400     #loop
401     #hide
402     xd1 = x2
403     yd1 = y2
404     #show
405     line$(xd1; yd1; xd2; yd2; red_style$)
406 #loop
407 '</svg>
408 #loop
409 #equ
410 '<p><strong>Deformed shape</strong></p>
411 'Shape function in relative coordinates ξ = x/l (approximate)
412 ϕ_1(e; ξ) = 1 - 3*ξ^2 + 2*ξ^3
413 ϕ_2(e; ξ) = ξ*1(e)*m^-1*(1 - 2*ξ + ξ^2)
414 ϕ_3(e; ξ) = ξ^2*(3 - 2*ξ)
415 ϕ_4(e; ξ) = ξ^2*1(e)*m^-1*(-1 + ξ)
416 'Element endpoint displacements and rotations
417 z_E,loc(e) = T(e)*z_E(e)
418 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))
419 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))
420 'Displacement functions
421 u(e; ξ) = u_1(e)*(1 - ξ) + u_2(e)*ξ
422 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)
423 'Deformed shape, mm
424 #val
425 #hide
426 tol = 0.00001
427 k_R = 800
428 #show
429 svg$
430 '<use href="#frame" style="opacity:0.4;"/>
431 #for e = 1 : n_E
432     #hide
433     x1 = x_1(e)*k
434     y1 = (h - y_1(e))*k
435     x2 = x_2(e)*k
436     y2 = (h - y_2(e))*k
437     c_e = c(e)
438     s_e = s(e)
439     l_e = l(e)
440     u = u(e; 0)
441     v = v(e; 0)
442     x = u*k_R
443     y = v*k_R
444     xd2 = x1 + x*c_e - y*s_e
445     yd2 = y1 - x*s_e - y*c_e
446     #show
447     #for j = 0 : 10
448         #hide
449         xd1 = xd2
450         yd1 = yd2
451         ξ = j/10
452         u = u(e; ξ)
453         v = v(e; ξ)
454         x = ξ*l_e*k + u*k_R
455         y = v*k_R
456         xd2 = x1 + x*c_e - y*s_e
457         yd2 = y1 - x*s_e - y*c_e
458         d = -15*sign(v)
459         #show
460         line$(xd1; yd1; xd2; yd2; red_style$)
461     #loop
462 #loop
463 #for j = 1 : n_J
464     #hide
465     z_J = z_J(j)

```

```

466     u = z_J.1
467     v = z_J.2
468     x = x_J.j*k + u*k_R
469     y = (h - y_J.j)*k - v*k_R
470     dx = 15*sign(u)
471     dy = -15*sign(v)
472     #show
473     #if abs(u) > tol
474         texth$(x + dx; y; 'u*1000')
475     #end if
476     #if abs(v) > tol
477         textv$(x; y + dy; 'v*1000')
478     #end if
479 #loop
480 '</svg>'
481 #equ

```

### III. Output

#### Analysis of plane frames with variable cross-sections

##### Joint coordinates

$$\begin{aligned}
 & \quad \quad \quad J1 \quad J2 \quad J3 \quad J4 \quad J5 \\
 \vec{x}_J &= [0 \text{ m} \quad 0 \text{ m} \quad 8 \text{ m} \quad 16 \text{ m} \quad 16 \text{ m}] \\
 \vec{y}_J &= [0 \text{ m} \quad 8 \text{ m} \quad 10 \text{ m} \quad 8 \text{ m} \quad 0 \text{ m}] \\
 n_J &= \text{len}(\vec{x}_J) = 5
 \end{aligned}$$

##### Elements

$$\begin{aligned}
 e_J^T &= \begin{bmatrix} 1 & 3 & 3 & 5 \\ 2 & 2 & 4 & 4 \end{bmatrix} \\
 n_E &= \mathbf{n}_{rows}(e_J) = 4
 \end{aligned}$$

##### Element endpoint coordinates

$$x_1(e) = \vec{x}_{J.e_{J.e,1}}, y_1(e) = \vec{y}_{J.e_{J.e,1}}, x_2(e) = \vec{x}_{J.e_{J.e,2}}, y_2(e) = \vec{y}_{J.e_{J.e,2}}$$

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

$$\text{Element directions - } c(e) = \frac{x_2(e) - x_1(e)}{l(e)}, s(e) = \frac{y_2(e) - y_1(e)}{l(e)}$$

##### Transformation matrix

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

##### Generation of the full transformation matrix

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

## Supports

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

$$n_c = n_{rows}(c) = 2$$

## Loads

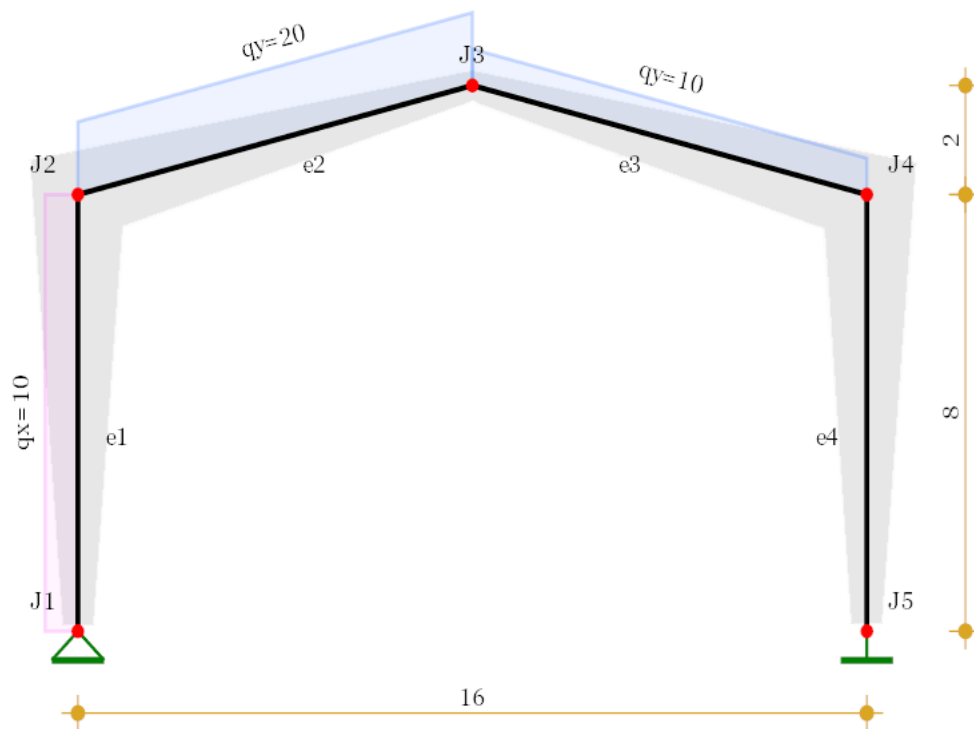
$$q = \begin{bmatrix} 1 & 10 \text{ kN/m} & 0 \text{ kN/m} \\ 2 & 0 \text{ kN/m} & -20 \text{ kN/m} \\ 3 & 0 \text{ kN/m} & -10 \text{ kN/m} \end{bmatrix}$$

Load values on elements

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

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

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

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

## Cross-sections

Section S1 -  $b_1 = 300 \text{ mm}$ ,  $h_1 = 300 \text{ mm}$

Section S2 -  $b_2 = 300 \text{ mm}$  ,  $h_2 = 900 \text{ mm}$

General representation

Width -  $b(\xi; z) = b_1 + (b_2 - b_1) \cdot \xi$

Height -  $h(\xi) = h_1 + (h_2 - h_1) \cdot \xi$

**Cross section properties**

Area - 
$$A(\xi) = \int_{0 \text{ mm}}^{h(\xi)} b(\xi; z) \, dz$$

First moment of area - 
$$S(\xi) = \int_{0 \text{ mm}}^{h(\xi)} b(\xi; z) \cdot z \, dz$$

Centroid - 
$$z_c(\xi) = \frac{S(\xi)}{A(\xi)}$$

Second moment of area - 
$$I(\xi) = \int_{0 \text{ mm}}^{h(\xi)} b(\xi; z) \cdot (z - z_c(\xi))^2 \, dz$$

First moment of area bellow z - 
$$S_1(\xi; z) = \int_{0 \text{ mm}}^z b(\xi; \zeta) \cdot (z_c(\xi) - \zeta) \, d\zeta$$

Shear area - 
$$A_Q(\xi) = \frac{I(\xi)^2}{\int_{0 \text{ mm}}^{h(\xi)} \frac{S_1(\xi; z)^2}{b(\xi; z)} \, dz}$$

**Element stiffness matrix**

Elastic properties for element "e"

$$EA(e; x) = \vec{E}_{\vec{e}_{M.e}} \cdot A\left(\frac{x}{l(e)}\right), EI(e; x) = \vec{E}_{\vec{e}_{M.e}} \cdot I\left(\frac{x}{l(e)}\right), GA_Q(e; x) = \vec{G}_{\vec{e}_{M.e}} \cdot A_Q\left(\frac{x}{l(e)}\right)$$

Stiffness matrix for element with variable cross-section

Displacement due to  $F_x = 1$  in primary system - 
$$u_F(e) = \int_{0 \text{ m}}^{l(e)} \frac{1}{EA(e; x)} \, dx$$

Displacement due to  $F_{y1} = 1$  in primary system, with account for shear deflections -

$$v_{F1}(e) = \int_{0 \text{ m}}^{l(e)} \frac{x^2}{EI(e; x)} \, dx + \int_{0 \text{ m}}^{l(e)} \frac{1}{GA_Q(e; x)} \, dx$$

Rotation due to  $F_{y1} = 1$  in primary system - 
$$\varphi_{F1}(e) = - \left( \int_{0 \text{ m}}^{l(e)} \frac{x}{EI(e; x)} \, dx \right)$$

Displacement due to  $M_1 = 1$  in primary system -  $v_{M1}(e) = \varphi_{F1}(e)$

Rotation due to  $M_1 = 1$  in primary system -  $\varphi_{M1}(e) = \int_{0m}^{l(e)} \frac{1}{EI(e; x)} dx$

Determinant -  $D_1(e) = \varphi_{M1}(e) \cdot v_{F1}(e) - \varphi_{F1}(e)^2$

Displacement due to  $F_{y2} = 1$  in primary system -  $v_{F2}(e) = \int_{0m}^{l(e)} \frac{(l(e) - x)^2}{EI(e; x)} dx + \int_{0m}^{l(e)} \frac{1}{GA_Q(e; x)} dx$

Rotation due to  $F_{y2} = 1$  in primary system -  $\varphi_{F2}(e) = \int_{0m}^{l(e)} \frac{l(e) - x}{EI(e; x)} dx$

Displacement due to  $M_2 = 1$  in primary system -  $v_{M2}(e) = \varphi_{F2}(e)$

Rotation due to  $M_2 = 1$  in primary system -  $\varphi_{M2}(e) = \varphi_{M1}(e)$

Determinant -  $D_2(e) = \varphi_{M2}(e) \cdot v_{F2}(e) - \varphi_{F2}(e)^2$

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

$$k_{ii}(e) = \left[ \frac{D_1(e)}{u_F(e)} \cdot kNm | 0; \varphi_{M1}(e) \cdot kNm; -\varphi_{F1}(e) \cdot kN | 0; -\varphi_{F1}(e) \cdot kN; v_{F1}(e) \cdot \frac{kN}{m} \right] \cdot \frac{kN^{-2}}{D_1(e)}$$

$$k_{ij}(e) = \left[ -\frac{D_2(e)}{u_F(e)} \cdot kNm | 0; -\varphi_{M2}(e) \cdot kNm; \varphi_{F2}(e) \cdot kN | 0; (\varphi_{F2}(e) - \varphi_{M2}(e) \cdot l(e)) \cdot kN; -(v_{F2}(e) - \varphi_{F2}(e) \cdot l(e)) \cdot \frac{kN}{m} \right] \cdot \frac{kN^{-2}}{D_2(e)}$$

$$k_{ji}(e) = \left[ -\frac{D_1(e)}{u_F(e)} \cdot kNm | 0; -\varphi_{M1}(e) \cdot kNm; \varphi_{F1}(e) \cdot kN | 0; (\varphi_{F1}(e) + \varphi_{M1}(e) \cdot l(e)) \cdot kN; -(v_{F1}(e) + \varphi_{F1}(e) \cdot l(e)) \cdot \frac{kN}{m} \right] \cdot \frac{kN^{-2}}{D_1(e)}$$

$$k_{jj}(e) = \left[ \frac{D_2(e)}{u_F(e)} \cdot kNm | 0; \varphi_{M2}(e) \cdot kNm; -\varphi_{F2}(e) \cdot kN | 0; -\varphi_{F2}(e) \cdot kN; v_{F2}(e) \cdot \frac{kN}{m} \right] \cdot \frac{kN^{-2}}{D_2(e)}$$

Full element stiffness matrix

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

Stiffness matrices obtained in local coordinates

$$k_E(1) = \begin{bmatrix} 921617 & 0 & 0 & -921617 & 0 & 0 \\ 0 & 4741.71 & 9483.42 & 0 & -4741.71 & 28450.3 \\ 0 & 9483.42 & 36052.8 & 0 & -9483.42 & 39814.6 \\ -921617 & 0 & 0 & 921617 & 0 & 0 \\ 0 & -4741.71 & -9483.42 & 0 & 4741.71 & -28450.3 \\ 0 & 28450.3 & 39814.6 & 0 & -28450.3 & 187788 \end{bmatrix}$$



$$k_E(2) = \begin{bmatrix} 695411 & 0 & 0 & -695411 & 0 & 0 \\ 0 & 3370.35 & 6948.16 & 0 & -3370.35 & 20844.5 \\ 0 & 6948.16 & 27216.3 & 0 & -6948.16 & 30079.7 \\ -695411 & 0 & 0 & 695411 & 0 & 0 \\ 0 & -3370.35 & -6948.16 & 0 & 3370.35 & -20844.5 \\ 0 & 20844.5 & 30079.7 & 0 & -20844.5 & 141808 \end{bmatrix}$$

Stiffness matrices obtained in global coordinates

$$\text{transp}(T(1)) \cdot k_E(1) \cdot T(1) = \begin{bmatrix} 4741.71 & 0 & -9483.42 & -4741.71 & 0 & -28450.3 \\ 0 & 921617 & 0 & 0 & -921617 & 0 \\ -9483.42 & 0 & 36052.8 & 9483.42 & 0 & 39814.6 \\ -4741.71 & 0 & 9483.42 & 4741.71 & 0 & 28450.3 \\ 0 & -921617 & 0 & 0 & 921617 & 0 \\ -28450.3 & 0 & 39814.6 & 28450.3 & 0 & 187788 \end{bmatrix}$$

$$\text{transp}(T(2)) \cdot k_E(2) \cdot T(2) = \begin{bmatrix} 654703 & 162833 & 1685.18 & -654703 & -162833 & 5055.53 \\ 162833 & 44078.6 & -6740.7 & -162833 & -44078.6 & -20222.1 \\ 1685.18 & -6740.7 & 27216.3 & -1685.18 & 6740.7 & 30079.7 \\ -654703 & -162833 & -1685.18 & 654703 & 162833 & -5055.53 \\ -162833 & -44078.6 & 6740.7 & 162833 & 44078.6 & 20222.1 \\ 5055.53 & -20222.1 & 30079.7 & -5055.53 & 20222.1 & 141808 \end{bmatrix}$$

**Global stiffness matrix**

$$K = \begin{bmatrix} 10^{20} & 0 & -9483.42 & -4741.71 & 0 & -28450.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10^{20} & 0 & 0 & -921617 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -9483.42 & 0 & 36052.8 & 9483.42 & 0 & 39814.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4741.71 & 0 & 9483.42 & 659445 & 162833 & 23394.7 & -654703 & -162833 & -1685.18 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -921617 & 0 & 162833 & 965696 & 20222.1 & -162833 & -44078.6 & 6740.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ -28450.3 & 0 & 39814.6 & 23394.7 & 20222.1 & 329596 & 5055.53 & -20222.1 & 30079.7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -654703 & -162833 & 5055.53 & 1309406 & 0 & 3370.35 & -654703 & 162833 & 5055.53 & 0 & 0 & 0 \\ 0 & 0 & 0 & -162833 & -44078.6 & -20222.1 & 0 & 88157.3 & 0 & 162833 & -44078.6 & 20222.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1685.18 & 6740.7 & 30079.7 & 3370.35 & 0 & 54432.6 & -1685.18 & -6740.7 & 30079.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -654703 & 162833 & -1685.18 & 659445 & -162833 & 23394.7 & -4741.71 & 0 & 9483.42 \\ 0 & 0 & 0 & 0 & 0 & 0 & 162833 & -44078.6 & -6740.7 & -162833 & 965696 & -20222.1 & 0 & -921617 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5055.53 & 20222.1 & 30079.7 & 23394.7 & -20222.1 & 329596 & -28450.3 & 0 & 39814.6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4741.71 & 0 & -28450.3 & 10^{20} & 0 & -9483.42 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -921617 & 0 & 0 & 10^{20} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9483.42 & 0 & 39814.6 & -9483.42 & 0 & 10^{20} \end{bmatrix}$$

**Element load vector**

Load functions

$$\text{Lateral} - q_E(e; x) = -\vec{q}_{x,e} \cdot s(e) + \vec{q}_{y,e} \cdot c(e)$$

$$\text{Axial} - n_E(e; x) = \vec{q}_{x,e} \cdot c(e) + \vec{q}_{y,e} \cdot s(e)$$

Functions of internal forces in primary system

$$\text{Axial forces} - N_0(e; x) = - \int_{0m}^x n_E(e; \xi) d\xi$$

$$\text{Shear forces} - Q_0(e; x) = \int_{0m}^x q_E(e; \xi) d\xi$$

$$\text{Bending moment - } M_0(e; x) = \int_{0m}^x q_E(e; \xi) \cdot (x - \xi) d\xi$$

Reactions at element ends

$$\text{Displacements along "x" due to axial loads - } u_n(e) = \int_{0m}^{l(e)} \frac{N_0(e; x)}{EA(e; x)} dx$$

$$\text{Displacements along "y" due to lateral loads - } v_q(e) = \int_{0m}^{l(e)} \frac{M_0(e; x) \cdot x}{EI(e; x)} dx + \int_{0m}^{l(e)} \frac{Q_0(e; x)}{GA_Q(e; x)} dx$$

$$\text{Rotations due to lateral loads - } \varphi_q(e) = - \int_{0m}^{l(e)} \frac{M_0(e; x)}{EI(e; x)} dx$$

Element endpoint loads in local coordinate system

For joint "1":

$$F_{x1}(e) = - \frac{u_n(e)}{u_F(e)}$$

$$F_{y1}(e) = \frac{\varphi_{M1}(e) \cdot v_q(e) - \varphi_{F1}(e) \cdot \varphi_q(e)}{D_1(e)}$$

$$M_1(e) = \frac{v_{F1}(e) \cdot \varphi_q(e) - \varphi_{F1}(e) \cdot v_q(e)}{D_1(e)}$$

For joint "2":

$$F_{x2}(e) = -F_{x1}(e) - N_0(e; l(e))$$

$$F_{y2}(e) = -F_{y1}(e) + Q_0(e; l(e))$$

$$M_2(e) = -M_1(e) + F_{y1}(e) \cdot l(e) - M_0(e; l(e))$$

Element endpoint loads in global coordinate system

For joint "1":

$$F'_{x1}(e) = F_{x1}(e) \cdot c(e) - F_{y1}(e) \cdot s(e)$$

$$F'_{y1}(e) = F_{x1}(e) \cdot s(e) + F_{y1}(e) \cdot c(e)$$

For joint "2":

$$F'_{x2}(e) = F_{x2}(e) \cdot c(e) - F_{y2}(e) \cdot s(e)$$

$$F'_{y2}(e) = F_{x2}(e) \cdot s(e) + F_{y2}(e) \cdot c(e)$$

Element load vector

$$F_E(e) = [F'_{x1}(e); F'_{y1}(e); M_1(e) \cdot m^{-1}; F'_{x2}(e); F'_{y2}(e); M_2(e) \cdot m^{-1}] \cdot kN^{-1}$$

$$F_E(1) = [31.43 \quad 0 \quad -25.11 \quad 48.57 \quad 0 \quad 93.68]$$

$$F_E(2) = [-0.675 \quad -64.96 \quad 51.75 \quad 0.675 \quad -99.97 \quad -193.14]$$

$$F_E(3) = [0.338 \quad -32.48 \quad -25.88 \quad -0.338 \quad -49.98 \quad 96.57]$$

$$F_E(4) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

**Global load vector**

$$\vec{F} = \begin{bmatrix} 31.43 & 0 & -25.11 & 49.25 & -99.97 & -99.46 & -0.338 \\ -97.44 & 25.88 & -0.338 & -49.98 & 96.57 & 0 & 0 & 0 \end{bmatrix}$$

## Results

### Solution of the system of equations

$$\vec{Z} = \text{clsolve}(K; \vec{F})$$

$$\vec{Z} = \begin{bmatrix} 1.06 \times 10^{-19} & -1.34 \times 10^{-18} & -0.00122 & 0.0112 & -0.000145 \\ -0.0022 & 0.0146 & -0.0139 & 0.00199 & 0.0179 \\ -0.000124 & -0.000536 & 6.94 \times 10^{-19} & -1.14 \times 10^{-18} & -1.48 \times 10^{-18} \end{bmatrix}$$

### Joint displacements

$$z_J(j) = \text{slice}(\vec{Z}; 3 \cdot j - 2; 3 \cdot j)$$

$$z(j) = \text{round}\left(\frac{z_J(j)}{\delta z}\right) \cdot \delta z \cdot 1000 \cdot [\text{mm}; \text{mm}; 1]$$

$$z(1) = [0 \text{ mm} \quad 0 \text{ mm} \quad -1.22]$$

$$z(2) = [11.23 \text{ mm} \quad -0.145 \text{ mm} \quad -2.2]$$

$$z(3) = [14.55 \text{ mm} \quad -13.87 \text{ mm} \quad 1.99]$$

$$z(4) = [17.86 \text{ mm} \quad -0.124 \text{ mm} \quad -0.536]$$

$$z(5) = [0 \text{ mm} \quad 0 \text{ mm} \quad 0]$$

### Support reactions

$$r(i) = \text{row}(c; i), j(i) = \text{take}(1; r(i))$$

$$R(i) = -z_J(j(i)) \cdot [m; m; 1] \cdot \text{last}(r(i); 3)$$

$$\text{Joint J1} - R(1) = [-10.56 \text{ kN} \quad 133.56 \text{ kN} \quad 0 \text{ kNm}]$$

$$\text{Joint J5} - R(2) = [-69.44 \text{ kN} \quad 113.82 \text{ kN} \quad 148.05 \text{ kNm}]$$

### Element end forces

$$z_E(e) = [z_J(e_{J.e,1}); z_J(e_{J.e,2})]$$

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

$$R_E(1) = [133.56 \text{ kN} \quad 10.56 \text{ kN} \quad 1.78 \times 10^{-14} \text{ kNm} \quad -133.56 \text{ kN} \quad 69.44 \text{ kN} \quad -235.56 \text{ kNm}]$$

$$R_E(2) = [59.76 \text{ kN} \quad -47.27 \text{ kN} \quad 34.36 \text{ kNm} \quad -99.76 \text{ kN} \quad -112.73 \text{ kN} \quad 235.56 \text{ kNm}]$$

$$R_E(3) = [74.98 \text{ kN} \quad -13.58 \text{ kN} \quad -34.36 \text{ kNm} \quad -94.98 \text{ kN} \quad 93.58 \text{ kN} \quad -407.5 \text{ kNm}]$$

$$R_E(4) = [113.82 \text{ kN} \quad 69.44 \text{ kN} \quad 148.05 \text{ kNm} \quad -113.82 \text{ kN} \quad -69.44 \text{ kN} \quad 407.5 \text{ kNm}]$$

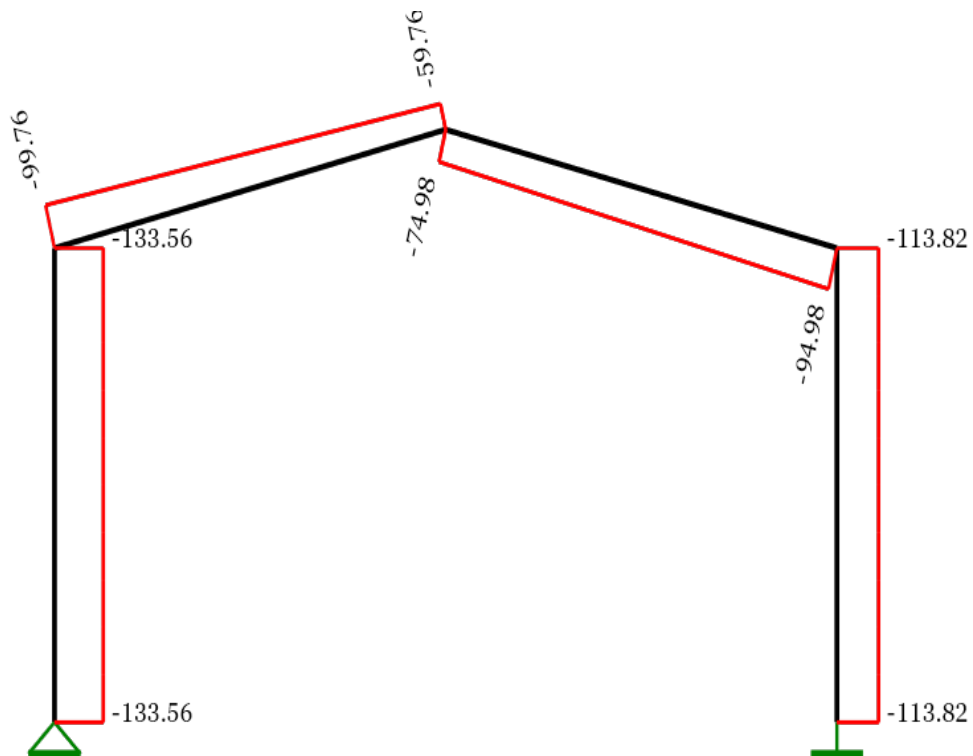
### Element internal forces

$$N(e; x) = -\text{take}(1; R_E(e)) + N_0(e; x)$$

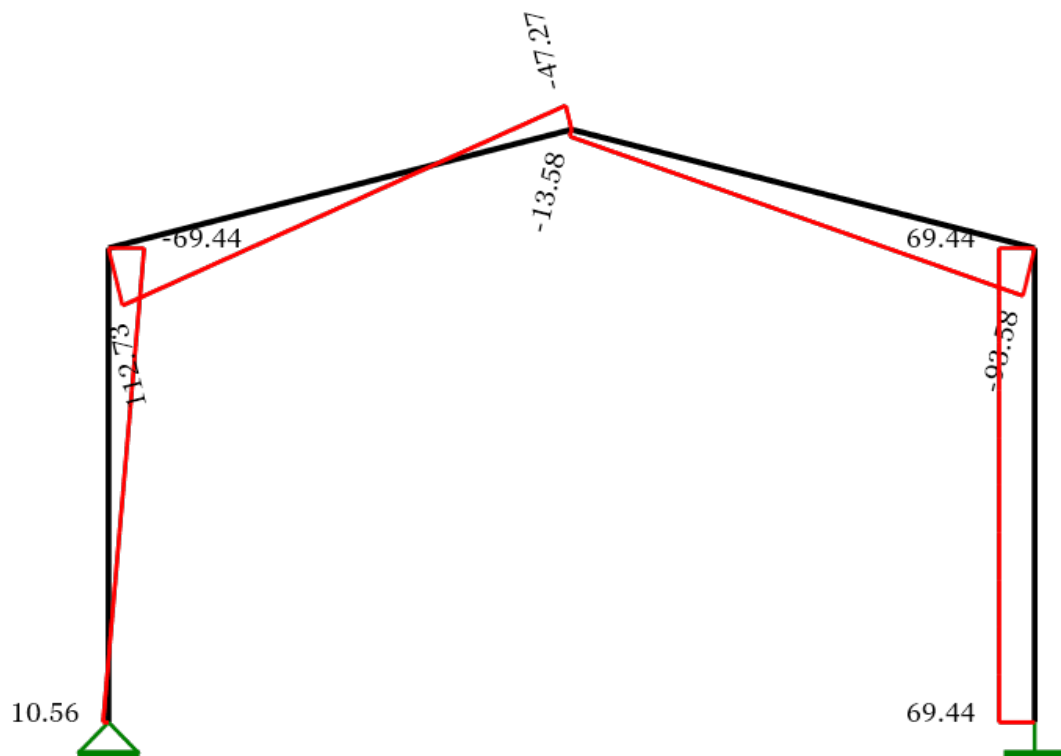
$$Q(e; x) = \text{take}(2; R_E(e)) + Q_0(e; x)$$

$$M(e; x) = -\text{take}(3; R_E(e)) + \text{take}(2; R_E(e)) \cdot x + M_0(e; x)$$

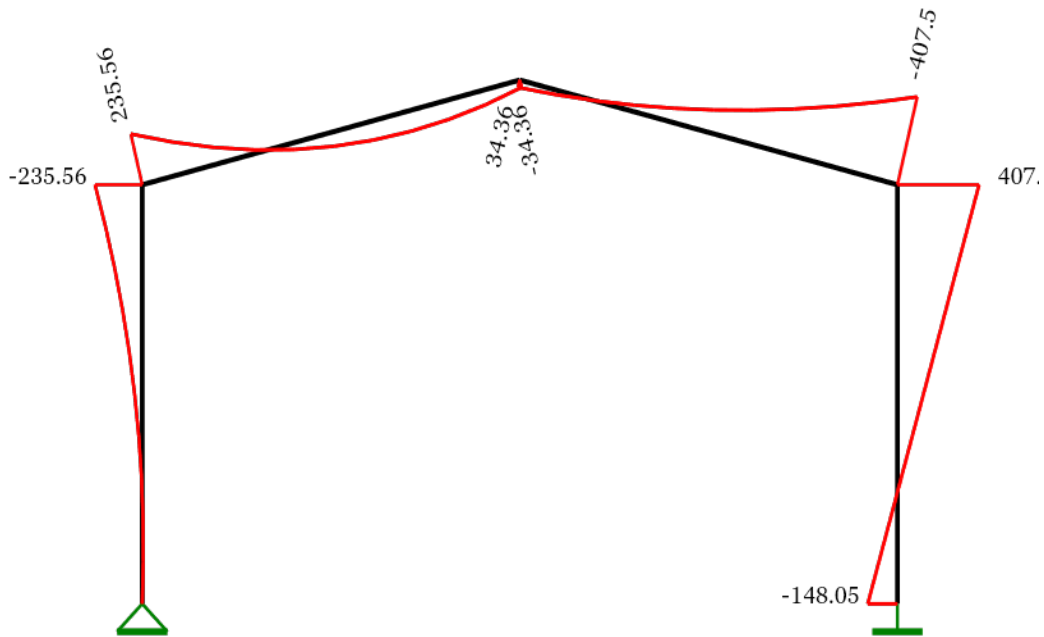
**Axial forces diagram, kN**



**Shear forces diagram, kN**



## Bending moments diagram, kNm



### Deformed shape

Shape function in relative coordinates  $\xi = x/l$  (approximate)

$$\Phi_1(e; \xi) = 1 - 3 \cdot \xi^2 + 2 \cdot \xi^3, \quad \Phi_2(e; \xi) = (1 - 2 \cdot \xi + \xi^2) \cdot \xi \cdot \frac{l(e)}{m}$$

$$\Phi_3(e; \xi) = (3 - 2 \cdot \xi) \cdot \xi^2, \quad \Phi_4(e; \xi) = (-1 + \xi) \cdot \xi^2 \cdot \frac{l(e)}{m}$$

Element endpoint displacements and rotations

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

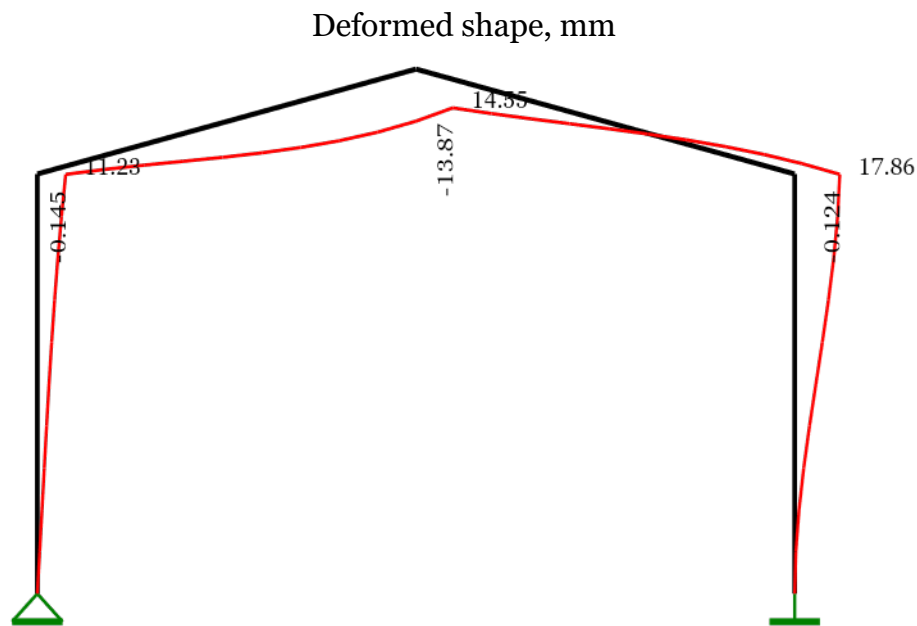
$$u_1(e) = \text{take}(1; z_{E,loc}(e)), \quad v_1(e) = \text{take}(2; z_{E,loc}(e)), \quad \varphi_1(e) = \text{take}(3; z_{E,loc}(e))$$

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

Displacement functions

$$u(e; \xi) = u_1(e) \cdot (1 - \xi) + u_2(e) \cdot \xi$$

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



## IV. Comparison with SAP 2000

To verify the obtained results, a SAP 2000 model is developed for the same structure. A non-prismatic finite element with three intermediate points is used. The input and output data from the analysis is listed below in both text and graphics.

Input data:

### STATIC LOAD CASES

STATIC CASE	CASE TYPE	SELF WT FACTOR
LOAD1	DEAD	0.0000

### JOINT DATA

JOINT	GLOBAL-X	GLOBAL-Y	GLOBAL-Z	RESTRAINTS	ANGLE-A	ANGLE-B	ANGLE-C
1	-8.00000	0.00000	0.00000	1 1 1 0 0 0	0.000	0.000	0.000
2	-8.00000	0.00000	8.00000	0 0 0 0 0 0	0.000	0.000	0.000
3	8.00000	0.00000	0.00000	1 1 1 1 1 1	0.000	0.000	0.000
4	8.00000	0.00000	8.00000	0 0 0 0 0 0	0.000	0.000	0.000
5	0.00000	0.00000	10.00000	0 0 0 0 0 0	0.000	0.000	0.000

### FRAME ELEMENT DATA

FRAME	JNT-1	JNT-2	SECTION	ANGLE	RELEASES	SEGMENTS	R1	R2	FACTOR	LENGTH
1	1	2	VAR1	0.000	000000	2	0.000	0.000	1.000	8.000
2	3	4	VAR1	0.000	000000	2	0.000	0.000	1.000	8.000
3	5	2	VAR2	0.000	000000	2	0.000	0.000	1.000	8.246
4	5	4	VAR2	0.000	000000	2	0.000	0.000	1.000	8.246

# M A T E R I A L   P R O P E R T Y   D A T A

MAT LABEL	MODULUS OF ELASTICITY	POISSON'S RATIO	THERMAL COEFF	WEIGHT PER UNIT VOL	MASS PER UNIT VOL
STEEL	199947979	0.300	1.170E-05	76.820	7.827
CONC	45000.000	0.200	9.900E-06	0.000	0.000
OTHER	24821128.4	0.200	9.900E-06	23.562	2.401
MAT1	45000.000	0.200	1.170E-05	0.000	0.000
MAT2	35000.000	0.300	1.170E-05	0.000	0.000

# F R A M E   S E C T I O N   P R O P E R T Y   D A T A

SECTION LABEL	MAT LABEL	SECTION TYPE	DEPTH	FLANGE WIDTH TOP	FLANGE THICK TOP	WEB THICK	FLANGE WIDTH BOTTOM	FLANGE THICK BOTTOM
VAR1								
FSEC10	MAT1		0.300	0.300	0.000	0.000	0.000	0.000
FSEC11	MAT1		0.600	0.300	0.000	0.000	0.000	0.000
FSEC12	MAT1		0.900	0.300	0.000	0.000	0.000	0.000
VAR2								
FSEC20	MAT2		0.300	0.300	0.000	0.000	0.000	0.000
FSEC21	MAT2		0.600	0.300	0.000	0.000	0.000	0.000
FSEC22	MAT2		0.900	0.300	0.000	0.000	0.000	0.000

# F R A M E   S E C T I O N   P R O P E R T Y   D A T A

SECTION LABEL	AREA	TORSIONAL INERTIA	MOMENTS OF INERTIA I33	OF INERTIA I22	SHEAR AREAS A2	A3
FSEC10	9.000E-02	1.141E-03	6.750E-04	6.750E-04	7.500E-02	7.500E-02
FSEC11	0.180	3.708E-03	5.400E-03	1.350E-03	0.150	0.150
FSEC12	0.270	6.401E-03	1.823E-02	2.025E-03	0.225	0.225
FSEC20	9.000E-02	1.141E-03	6.750E-04	6.750E-04	7.500E-02	7.500E-02
FSEC22	0.270	6.401E-03	1.823E-02	2.025E-03	0.225	0.225
FSEC21	0.180	3.708E-03	5.400E-03	1.350E-03	0.150	0.150

# F R A M E   S E C T I O N   P R O P E R T Y   D A T A

SECTION LABEL	S E C T I O N MODULII S33	S22	PLASTIC MODULII Z33	Z22	RADII OF GYRATION R33	R22
FSEC10	4.500E-03	4.500E-03	6.750E-03	6.750E-03	8.660E-02	8.660E-02
FSEC20	4.500E-03	4.500E-03	6.750E-03	6.750E-03	8.660E-02	8.660E-02
FSEC22	4.050E-02	1.350E-02	6.075E-02	2.025E-02	0.260	8.660E-02
FSEC11	1.800E-02	9.000E-03	2.700E-02	1.350E-02	0.173	8.660E-02
FSEC21	1.800E-02	9.000E-03	2.700E-02	1.350E-02	0.173	8.660E-02
FSEC12	4.050E-02	1.350E-02	6.075E-02	2.025E-02	0.260	8.660E-02

# F R A M E   S P A N   D I S T R I B U T E D   L O A D S   Load Case   LOAD1

FRAME	TYPE	DIRECTION	DISTANCE-A	VALUE-A	DISTANCE-B	VALUE-B
3	FORCE	GLOBAL-Z	0.0000	-20.0000	1.0000	-20.0000
4	FORCE	GLOBAL-Z	0.0000	-10.0000	1.0000	-10.0000
1	FORCE	GLOBAL-X	0.0000	10.0000	1.0000	10.0000

## Results:

### JOINT DISPLACEMENTS

JOINT	LOAD	U1	U2	U3	R1	R2	R3
1	LOAD1	0	0	0	0	1.17	0
2	LOAD1	10.80	0	-0.15	0	2.13	0
3	LOAD1	0	0	0	0	0	0
4	LOAD1	17.26	0	-0.12	0	0.51	0
5	LOAD1	14.04	0	-13.53	0	-1.87	0

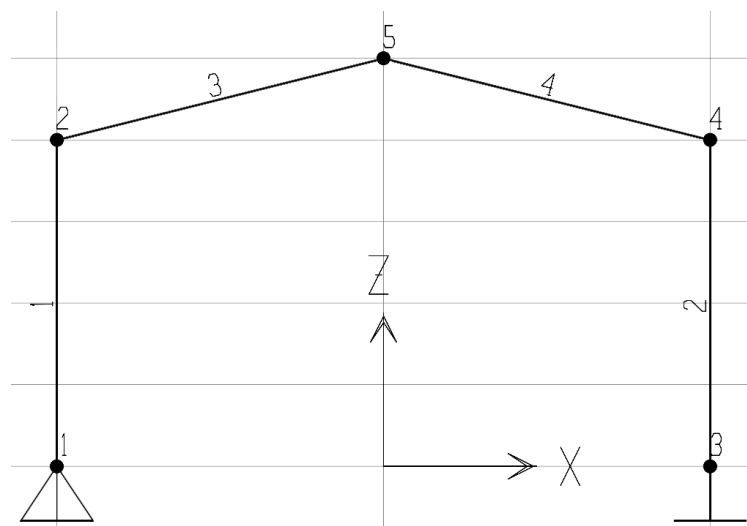
### JOINT REACTIONS

JOINT	LOAD	F1	F2	F3	M1	M2	M3
1	LOAD1	-10.6271	0.0000	133.6357	0.0000	0.0000	0.0000
3	LOAD1	-69.3729	0.0000	113.7507	0.0000	-149.2315	0.0000

### FRAME ELEMENT FORCES

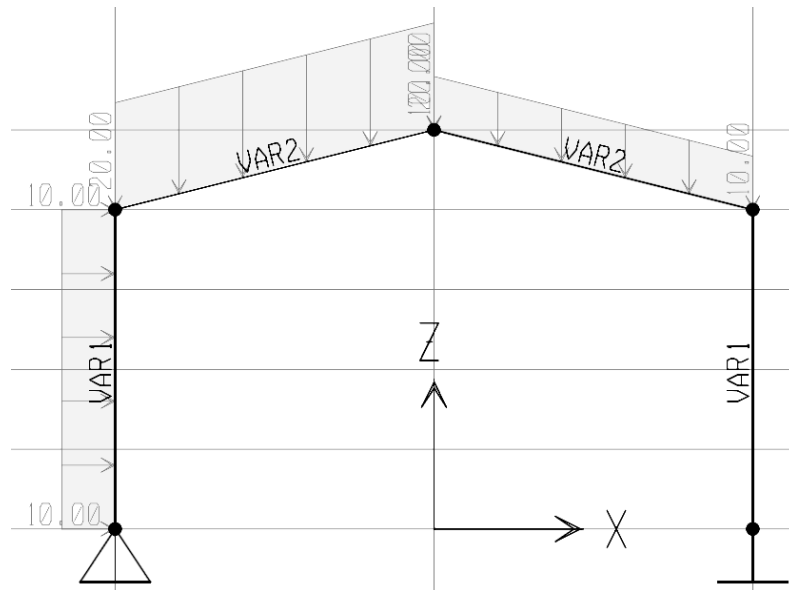
FRAME	LOC	P	V2	V3	T	M2	M3
1	0.00	-133.64	10.63	0.00	0.00	0.00	0.00
	4.00	-133.64	-29.37	0.00	0.00	0.00	37.49
	8.00	-133.64	-69.37	0.00	0.00	0.00	234.98
2	0.00	-113.75	69.37	0.00	0.00	0.00	149.23
	4.00	-113.75	69.37	0.00	0.00	0.00	-128.26
	8.00	-113.75	69.37	0.00	0.00	0.00	-405.75
3	0.00	-59.71	-47.18	0.00	0.00	0.00	35.66
	4.12	-79.71	32.82	0.00	0.00	0.00	65.26
	8.25	-99.71	112.82	0.00	0.00	0.00	-234.98
4	0.00	-74.89	13.53	0.00	0.00	0.00	35.66
	4.12	-84.89	53.53	0.00	0.00	0.00	-102.58
	8.25	-94.89	93.53	0.00	0.00	0.00	-405.75

Labels of joints and elements

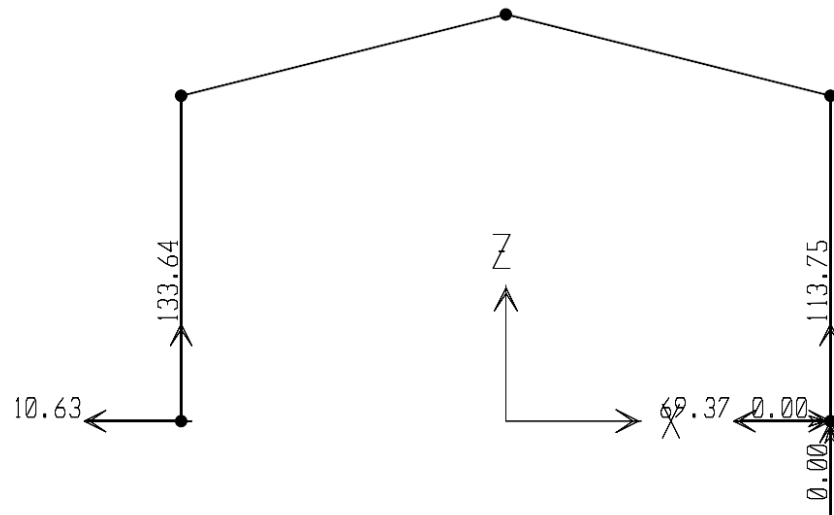




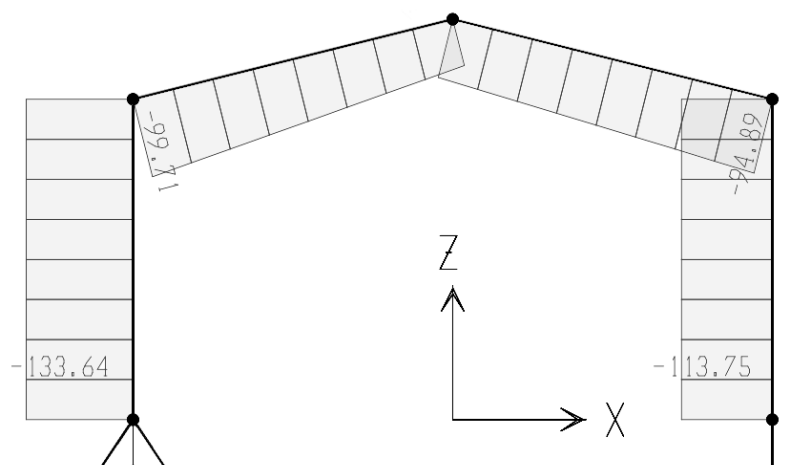
## Loads and cross sections



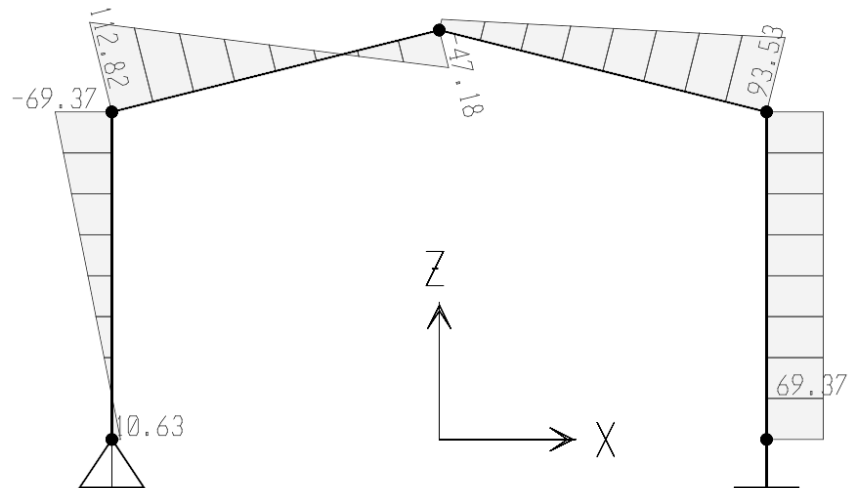
## Support reactions



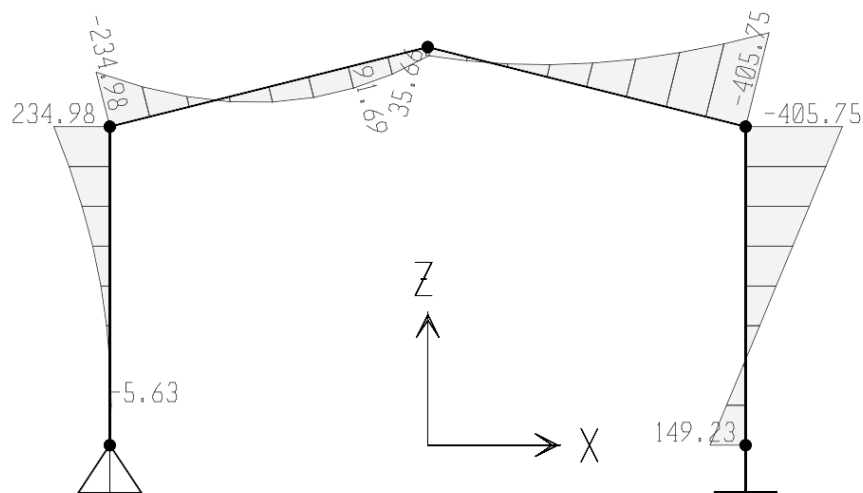
## Axial forces



### Shear forces



### Bending moments



## V. Conclusions

The results obtained by the SAP 2000 software match the Calcpad solution with an accuracy of 0.5%. This is likely due to the different way the elastic properties are approximated along the element length.