Example: Static analysis of 2D frame structure

Input Data

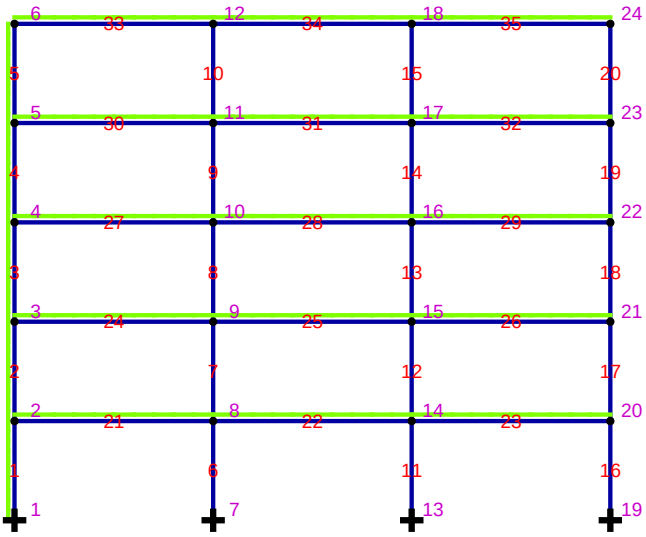
Example: Static analysis of 2D frame structure

```
<< AceFEM`;
Nr = 3; r = 1000.; Ne = 5; e = 500.;
Em = 3000; ν = 0; h1 = 60.; h2 = 40; b = 20.;
SMTInputData[];
SMTAddDomain["column", "SET2L1DFHYCBW3U1BVenant", {Em, ν, h1 b, 0, b h1^3 / 12}];
SMTAddDomain["beam", "SET2L1DFHYCBW3U1BVenant", {Em, ν, h2 b, 0, b h2^3 / 12}];
Do[SMTMesh["column", "L1", {Ne}, {{x, 0}, {x, Ne e}}];, {x, 0, Nr r, r}];
Do[SMTMesh["beam", "L1", {Nr}, {{0, y}, {Nr r, y}}];, {y, e, Ne e, e}];
SMTAnalysis[];
BeamNodeData["Y" == 0 &, "u", 0];
BeamNodeData["Y" == 0 &, "v", 0];
BeamNodeData["Y" == 0 &, "ϕ", 0];
BeamElementData["X" == 0 &, "py", -0.1];
Do[BeamElementData["Y" == y &, "py", -0.02], {y, e, Ne * e, e}];
```

Example: Static analysis of 2D frame structure

```
BeamShowMesh[]
```

Example: Static analysis of 2D frame structure



Nonlinear analysis

Example: Static analysis of 2D frame structure

```
SMTNextStep[1, 0.1];
While[
 While[step = SMTConvergence[10^-8, 10, {"Adaptive", 8, 0.001, 1, 1.}],
  SMTNewtonIteration[];];
 If[step[[4]] === "MinBound", SMTStatusReport[" Error: Δλ < Δλmin"]; Abort[];];
 If[Not[step[[1]]], SMTShowMesh["DeformedMesh" → True, "Show" -> "Window"]];
 step[[3]]
 , If[step[[1]], SMTStepBack[];];
 SMTNextStep[1, step[[2]]]
]
```

Results

Example: Static analysis of 2D frame structure

```
BeamNodeData["X" == 0 &, "u"]
```

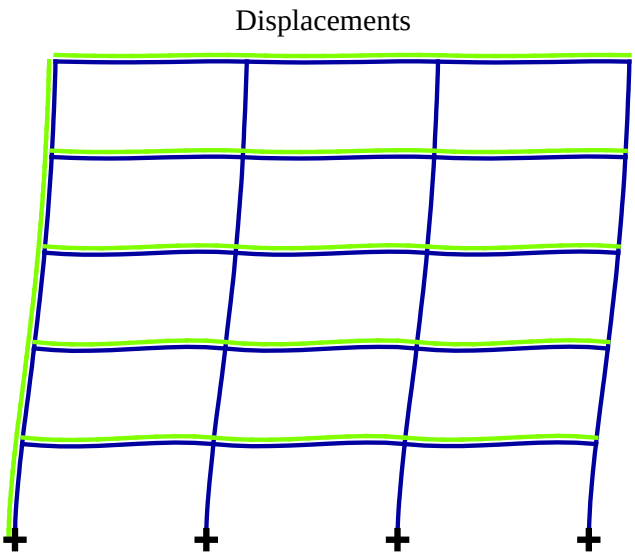
Example: Static analysis of 2D frame structure

```
{0., 1.87602, 4.98315, 7.71073, 9.58098, 10.6018}
```

Example: Static analysis of 2D frame structure

```
BeamShowMesh["u", "Scale" → 20]
```

Example: Static analysis of 2D frame structure

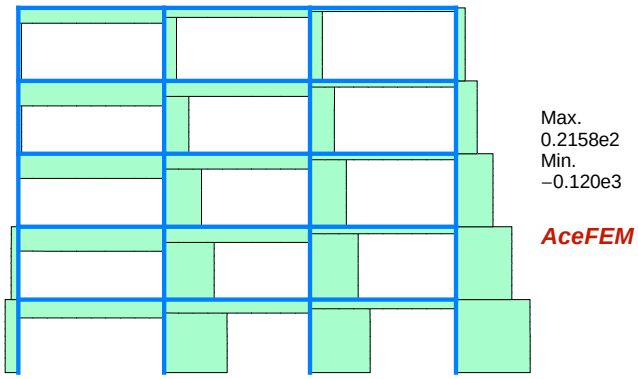


Example: Static analysis of 2D frame structure

```
BeamShowMesh["N"]
```

Example: Static analysis of 2D frame structure

Axial force N

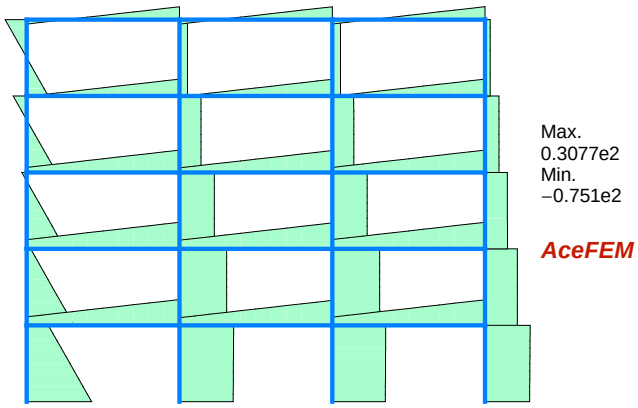


Example: Static analysis of 2D frame structure

```
BeamShowMesh["V"]
```

Example: Static analysis of 2D frame structure

Shear force V

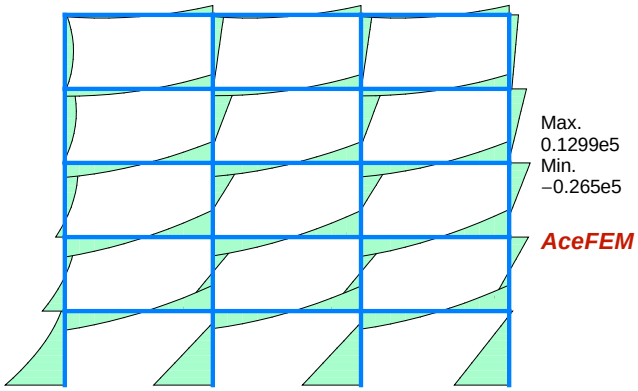


Example: Static analysis of 2D frame structure

```
BeamShowMesh["M"]
```

Example: Static analysis of 2D frame structure

Bending moment M



Example: Large displacements

Example: Large displacements

Input Data

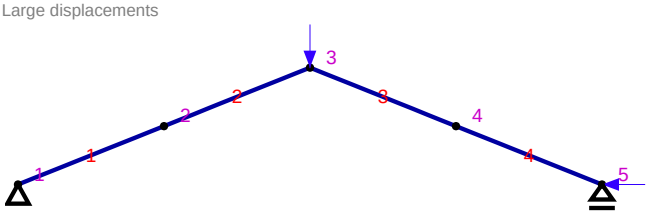
Example: Large displacements

```
<< AceFEM‘;
h = 1; b = 1; L = 100.; p = L / 5; ne = 4; Em = 21 000; ν = 0;
crit = π^2 Em b h^3 / 12 / (L)^2;
```

Example: Large displacements

```
SMTInputData[];
SMTAddDomain["roof", "SET2L1DFHYCBW3U1BVenant", {Em, ν, h b, 0, b h^3 / 12}];
SMTMesh["roof", "L1", {ne}, {{0, 0}, {L / 2, p}, {L, 0}}, "InterpolationOrder" → 1];
SMTAnalysis[];
BeamNodeData["X" == 0 &, "u", 0];
BeamNodeData["X" == 0 || "X" == L &, "v", 0];
BeamNodeData["X" == L / 2 &, "Fy", -1];
BeamNodeData["X" == L &, "Fx", -1];
BeamShowMesh[]
```

Example: Large displacements



Nonlinear analysis

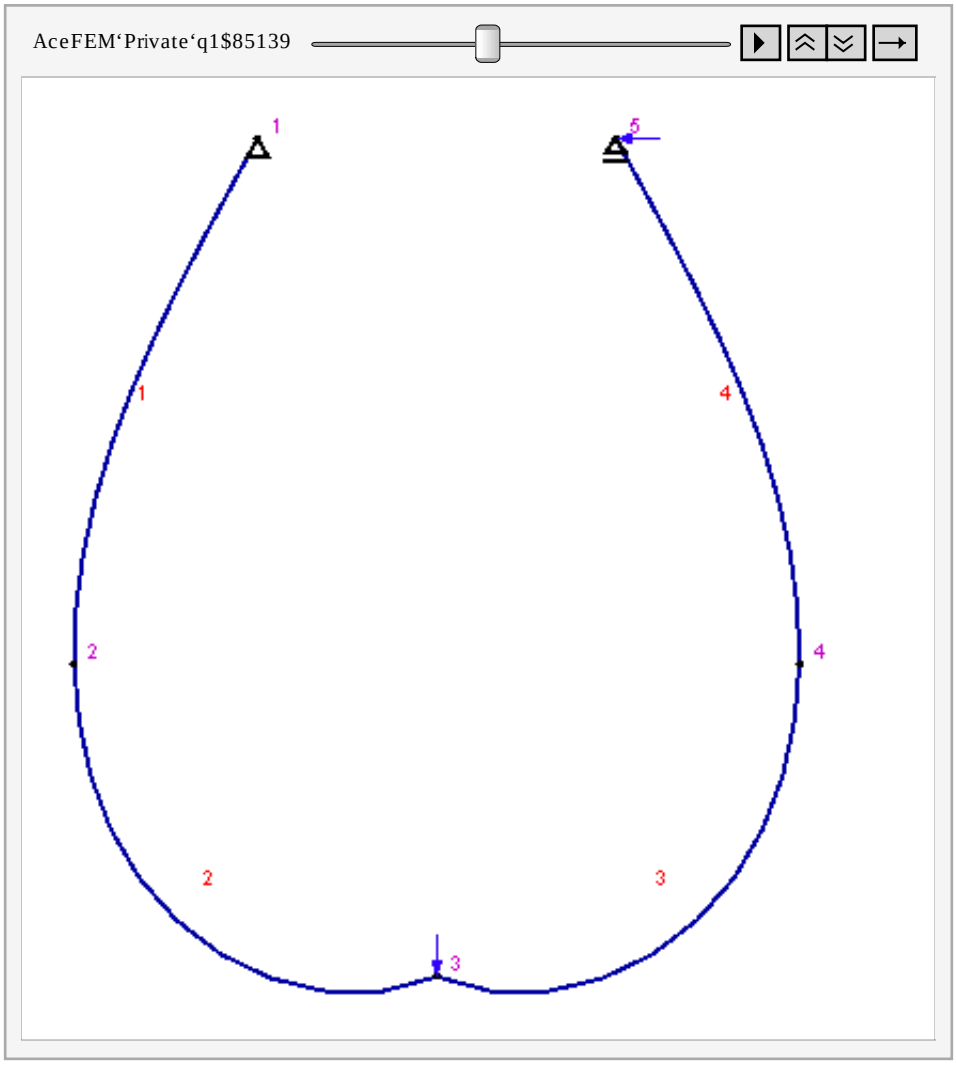
Example: Large displacements

```
SMTNextStep[1, crit / 20];
```

Example: Large displacements

```
graf = {};
Do[
 SMTNextStep[1, crit / 20];
 While[SMTConvergence[10^-8, 10], SMTNewtonIteration[];];
 AppendTo[graf, {BeamNodeData["X" == L && "Y" == 0 &, "u"][[1]] , SMTRData["Multiplier"]}];
 BeamShowMesh["Show" → "Window" | {"Animation", "Beam"}];
 , {i, 1, 100}]
```

```
SMTMakeAnimation["Beam"]
```

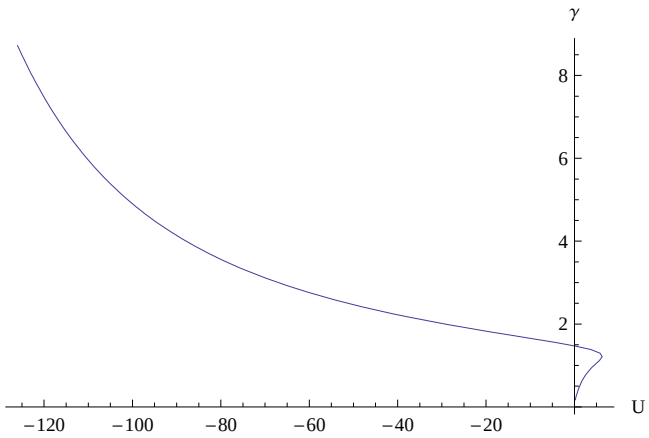


Results

Example: Large displacements

```
ListLinePlot[graf, AxesLabel → {"U", "γ"}]
```

Example: Large displacements



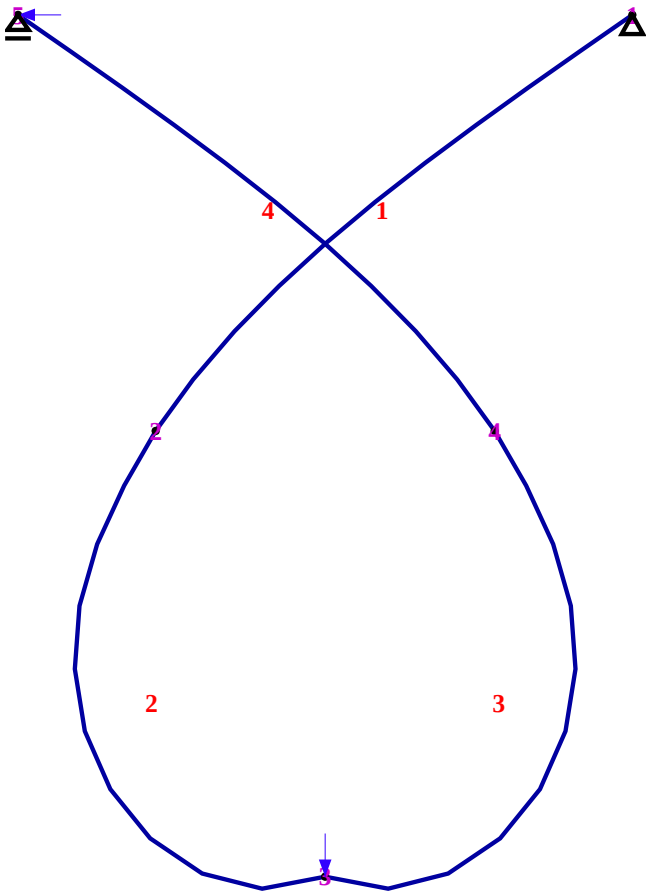
Example: Large displacements

```
BeamShowMesh["All"]
```

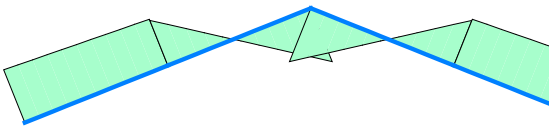
Example: Large displacements

Results of static analysis for γ=8.72226

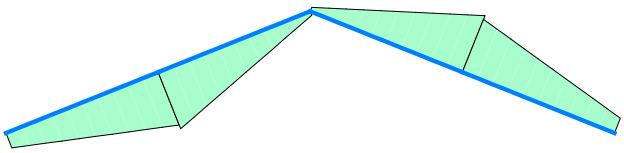
Max[|u|]=0.1259e3 Max[|ϕ|]=0.2917e1



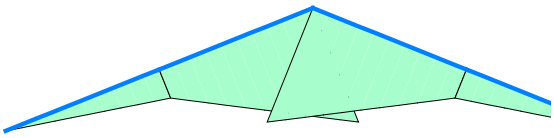
N: Min= −0.9648e1 Max= 0.95594e1



V: Min= −0.5171e1 Max= 0.51714e1



M: Min= 0 Max= 0.25874e3



Example: Bifurcation analysis

Input Data

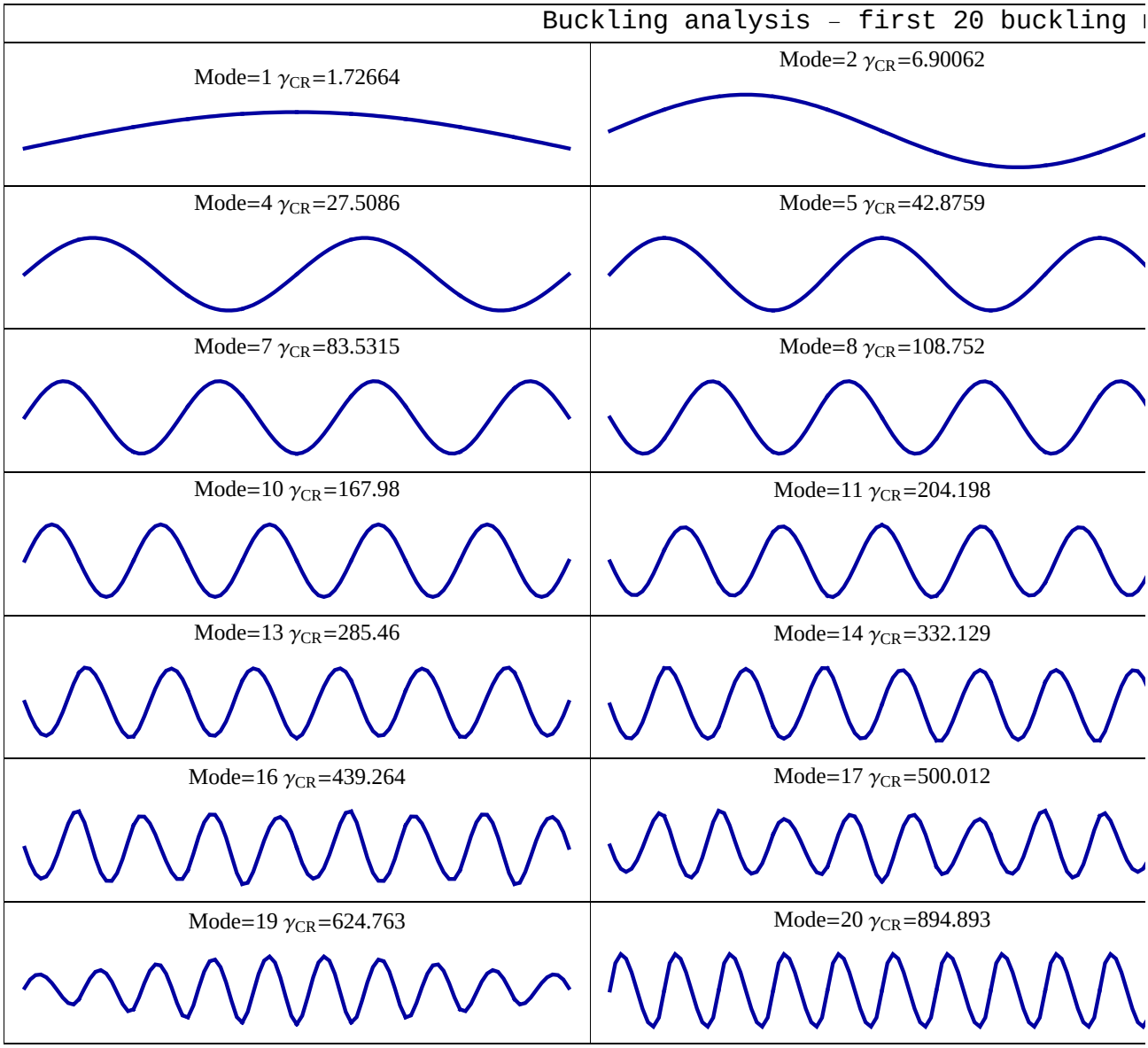

```
<< AceFEM`;
h = 1; b = 1; L = 100; ne = 10; Em = 21 000; ν = 0.;
SMTInputData["CDriver"];
SMTAddDomain["beam", "SET2L1DFHYCBW3U1BVenant", {Em, ν, h b, 0, b h^3 / 12}];
SMTMesh["beam", "L1", {ne}, {{0, 0}, {L, 0}}];
SMTAnalysis[];
BeamNodeData["X" == 0 &, "u", 0];
BeamNodeData["X" == 0 || "X" == L &, "v", 0];
BeamNodeData["X" == L &, "Fx", -1];
```

Nonlinear analysis

Example: Bifurcation analysis

```
SMTNextStep[1, 1];
numerical = BeamBucklingAnalysis["Modes" → 20]
```

Example: Bifurcation analysis



Example: Bifurcation analysis

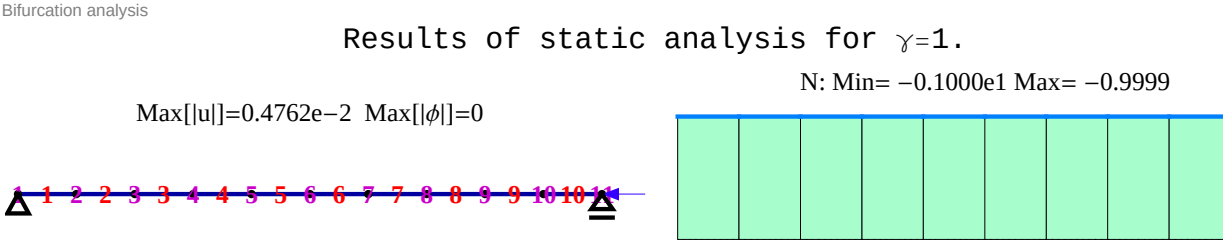
```
{1.72664, 6.90062, 15.5042, 27.5086, 42.8759, 61.5636, 83.5315, 108.752, 137.222, 167.98,
 204.198, 242.922, 285.46, 332.129, 383.292, 439.264, 500.012, 564.136, 624.763, 894.893}
```

Results

Example: Bifurcation analysis

```
BeamShowMesh["All"]
```

Example: Bifurcation analysis



Example: Bifurcation analysis

```
(* analytical solution*)
Grid[Join[{{"N", "Euler", "FEM", "Error"}}, Table[{n, euler = n^2 π^2 Em b h^3 / 12. / L^2,
    numerical[[n]], (numerical[[n]] - euler) / euler}, {n, 1, 20}]]
 ,
 Frame →
  All]
```

Example: Bifurcation analysis

N	Euler	FEM	Error
1	1.72718	1.72664	−0.000311576
2	6.90872	6.90062	−0.00117326
3	15.5446	15.5042	−0.00260077
4	27.6349	27.5086	−0.00457059
5	43.1795	42.8759	−0.00703118
6	62.1785	61.5636	−0.00988971
7	84.6319	83.5315	−0.0130019
8	110.54	108.752	−0.0161684
9	139.902	137.222	−0.0191516
10	172.718	167.98	−0.0274315
11	208.989	204.198	−0.0229262
12	248.714	242.922	−0.0232886
13	291.894	285.46	−0.0220401
14	338.527	332.129	−0.0189017
15	388.616	383.292	−0.0136992
16	442.158	439.264	−0.00654508
17	499.155	500.012	0.00171634
18	559.607	564.136	0.00809415
19	623.512	624.763	0.0020053
20	690.872	894.893	0.295309

SET2L1DFHYCBW3U1BVenant

AceGen input for Timoschenko beam element, kinematically exact continuum beam theory

```
<< AceGen‘; << AceFEM‘;
```

SET2L1DFHYCBW3U1BVenant

```
elementcodes = {"SE", "T2", "L1", "DF", "HY", "CBW3U1B", "Venant"};
SMSInitialize[StringJoin @@ elementcodes,
 "Environment" → "AceFEM", "Mode" -> "Optimal", "VectorLength" → 3500];
SMSTemplate["SMSTopology" → "L1",
  "SMSNoNodes" → 11,
  "SMSDOFGlobal" → {3, 3, 1, 1, 8, 0, 0, 0, 0, 0, 0},
  "SMSSegments" → {{1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 2}},
  "SMSDefaultIntegrationCode" → 23,
  "SMSNoTimeStorage" → 3,
  "SMSNoElementData" → 3,
  "SMSCharSwitch" → {"BeamEnvironment"},
  "SMSAdditionalNodes" → Hold[Table[(1 - i) #1 + i #2, {i, .1, .9, .1}] &],
  "SMSNodeID" → {"DFi", "DFi", "Fi -CF", "Fi -CF", "Lagr8 -F", "Post -F",
    "Post -F", "Post -F", "Post -F", "Post -F", "Post -F"},
  "SMSMMAInitialisation" →
   SMSEvaluateCellsWithTag["Environment", "CollectInput" → False],
  "SMSAdditionalGraphics" → Hold[BeamAdditionalGraphics],
  "SMSMainTitle" ->
   "Timoschenko beam element, kinematically exact continuum beam theory with
     constant thickness change, u(x) and v(x) - 4-th order polynomial, ϕ(x)
     - 3-th order polynomial, analytical integration over thickness.",
  "SMSSubTitle" -> "Hyperelastic, isothropic St. Venant material law.",
  "SMSSubSubTitle" → BeamDescription[]
 ];
```

SET2L1DFHYCBW3U1BVenant

```
[0-0] Include Tag : Environment (24 cells found, 15 evaluated)
```

SET2L1DFHYCBW3U1BVenant

```
SMSStandardModule["Tangent and residual"];

SMSDo[IpIndex, 1, SMSInteger[es$$["id", "NoIntPoints"]]];
{ξ, wGauss} ⊢ SMSReal[{es$$["IntPoints", 1, IpIndex], es$$["IntPoints", 4, IpIndex]}];
```

SET2L1DFHYCBW3U1BVenant, Kinematics - SET2L1DFHYCBW3U1BVenant

```
Xi ⊨ SMSReal[{nd$$[1, "X", 1], nd$$[2, "X", 1]}];
Yi ⊨ SMSReal[{nd$$[1, "X", 2], nd$$[2, "X", 2]}];

{ui, uj} ⊨ SMSReal[{nd$$[1, "at", 1], nd$$[2, "at", 1]}];
{vi, vj} ⊨ SMSReal[{nd$$[1, "at", 2], nd$$[2, "at", 2]}];
{ϕiv, ϕjv} ⊨ SMSReal[{nd$$[1, "at", 3], nd$$[2, "at", 3]}];
{ϕic, ϕjc} ⊨ SMSReal[{nd$$[3, "at", 1], nd$$[4, "at", 1]}];
{u1, v1, ϕ1, u2, v2, ϕ2, u3, v3} ⊨ SMSReal[Array[nd$$[5, "at", #] &, 8]];
alldof = {ui, vi, ϕiv, uj, vj, ϕjv, ϕic, ϕjc, u1, v1, ϕ1, u2, v2, ϕ2, u3, v3};
```

```
SMSGroupDataNames = {"E - elastic modulus", "ν - Poisson ratio",
   "A - area", "S - first area moment", "I - second area moment"};
{Em, ν, Ap, Sp, Ip} ⊨ SMSReal[Array[es$$["Data", #1] &, SMSGroupDataNames // Length]];
Ni ⊨ {(1 - ξ)/2, (1 + ξ)/2};
X ⊨ Ni.Xi; Y ⊨ Ni.Yi;
ξ ⊨ SMSFictive[];
η ⊨ SMSFictive[];
κ0 = {X, Y, ξ};
m ⊨ SMSD[κ0, ξ] / Sqrt[SMSD[κ0, ξ].SMSD[κ0, ξ]];
n ⊨ Cross[{0, 0, 1}, m];
R = κ0 + η n;

Pu ⊨ {InterpolatingPolynomial[{{-1, 1}, {1, 0}}, ξ],
   InterpolatingPolynomial[{{-1, 0}, {1, 1}}, ξ],
   InterpolatingPolynomial[{{-1, 0}, {0, 1}, {1, 0}}, ξ],
   InterpolatingPolynomial[{{-1, {0, 1}}, {0, 0}, {1, {0, 1}}}, ξ],
   InterpolatingPolynomial[{{-1, {0, 1}}, {0, {0, 0}}, {1, {0, -1}}}, ξ]};
Pϕ ⊨ {InterpolatingPolynomial[{{-1, 1}, {1, 0}}, ξ],
   InterpolatingPolynomial[{{-1, 0}, {1, 1}}, ξ],
   InterpolatingPolynomial[{{-1, {0, 1}}, {1, {0, 0}}}, ξ],
   InterpolatingPolynomial[{{-1, {0, 0}}, {1, {0, 1}}}, ξ]};

u ⊨ Pu.{ui, uj, u1, u2, u3};
v ⊨ Pu.{vi, vj, v1, v2, v3};

(*implementation of hinges*)
{cidof, cjdof} ⊨ SMSInteger[{nd$$[3, "DOF", 1], nd$$[4, "DOF", 1]}];
SMSIf[cidof ≥ 0];
    k1 ⫤ 1;
    k2 ⫤ 0;
SMSElse[];
    k1 ⫤ 0;
    k2 ⫤ 1;
SMSEndIf[k1, k2];
SMSIf[cjdof ≥ 0];
    k3 ⫤ 1;
    k4 ⫤ 0;
SMSElse[];
    k3 ⫤ 0;
    k4 ⫤ 1;
SMSEndIf[k3, k4];

ϕ ⊨ Pϕ.{ϕic k1 + ϕiv k2, ϕjc k3 + ϕjv k4, ϕ1, ϕ2};
t ⊨ Cos[ϕ] n - Sin[ϕ] m;
κ ⊨ κ0 + {u, v, 0};
Jm ⊨ SMSD[R, {ξ, η, ξ}];
Jmi ⊨ Inverse[Jm];
r = κ + η t;
```

SET2L1DFHYCBW3U1BVenant, Kinematics - SET2L1DFHYCBW3U1BVenant

```
{λ, μ} ⊨ SMSHookeToLame[Em, ν];
𝔽g = SMSSmartRestore[SMSD[r, {ξ, η, ζ}], η];
𝔽 = SMSSmartReduce[
   Simplify[{m, n, {0, 0, 1}}.(𝔽g.Jmi).Transpose[{m, n, {0, 0, 1}}]], η, Collect[#, η] &];
𝔼 = Simplify[1/2 (Transpose[𝔽].𝔽 - IdentityMatrix[3])];
𝔼[[3, 3]] = 𝔼33;
𝔼[[2, 2]] = 𝔼22;
σS = λ Tr[𝔼] IdentityMatrix[3] + 2 μ 𝔼;
pogoji = Solve[{σS[[2, 2]] == 0, σS[[3, 3]] == 0}, {𝔼22, 𝔼33}][[1]] // FullSimplify;
Πc ⊨ FullSimplify[CoefficientList[
    Expand[1/2 Sum[σS[[i, j]] 𝔼[[i, j]] , {i, 1, 3}, {j, 1, 3}] /. pogoji], η]];
Πi ⊨ Πc.Take[{Ap, Sp, Ip, 0, 0, 0}, Length[Πc]];
```

Kinematics - SET2L1DFHYCBW3U1BVenant, SET2L1DFHYCBW3U1BVenant

```
{dpx, dpy, dmz} ⊨ SMSReal[{ed$$["Data", 1], ed$$["Data", 2], ed$$["Data", 3]}];
{pxp, pyp, mzp} ⊨ SMSReal[{ed$$["hp", 1], ed$$["hp", 2], ed$$["hp", 3]}];
δλ ⊨ SMSReal[rdata$$["MultiplierIncrement"]];
{pxt, pyt, mzt} ⊨ {pxp, pyp, mzp} + δλ {dpx, dpy, dmz};
Πe ⊨ -{{u, v, 0}.m, {u, v, 0}.n, ϕ}.{pxt, pyt, mzt};
```

SET2L1DFHYCBW3U1BVenant

```
SMSExport[{pxt, pyt, mzt}, {ed$$["ht", 1], ed$$["ht", 2], ed$$["ht", 3]}];

weight ⊨ Det[Jm] wGauss;
SMSDo[i, 1, alldof // Length];
    δΠ ⊨ weight SMSD[Πi + Πe, alldof, i];
    SMSExport[δΠ, p$$[i], "AddIn" → True];
    SMSDo[j, i, alldof // Length];
        Kt ⊨ SMSD[δΠ, alldof, j];
        SMSExport[Kt, s$$[i, j], "AddIn" → True];
    SMSEndDo[];
SMSEndDo[];
```

SET2L1DFHYCBW3U1BVenant

```
SMSEndDo[];
```

SET2L1DFHYCBW3U1BVenant

```
SMSStandardModule["Postprocessing"];
```

SET2L1DFHYCBW3U1BVenant

```
T ⊰ Table[0, {i, 1, SMSNoDOFGlobal}];
SMSDo[IpIndex, 1, SMSInteger[es$$["id", "NoIntPoints"]], 1, T];
    {ξ, wGauss} ⊢ SMSReal[{es$$["IntPoints", 1, IpIndex], es$$["IntPoints", 4, IpIndex]}];

SMSEvaluateCellsWithTag["Kinematics - SET2L1DFHYCBW3U1BVenant", "CollectInput" → False];
    weight ⊨ Det[Jm] wGauss;
    T ⊰ T + weight SMSD[Πi + Πe, alldof];
SMSEndDo[T];
```

SET2L1DFHYCBW3U1BVenant

```
[15-332] Include Tag : Kinematics - SET2L1DFHYCBW3U1BVenant (3 cells found, 3 evaluated)
```

SET2L1DFHYCBW3U1BVenant

```
ii = .; pol = InterpolatingPolynomial[{{1, -1}, {11, 1}}, ii];
ξi = Table[pol /. ii -> Position[SMSSegments[[1]], i][[1, 1]], {i, 11}];
SMSDo[node, 1, 11];
    ξ = SMSFreeze[SMSPart[ξi, node]];

SMSEvaluateCellsWithTag["Kinematics - SET2L1DFHYCBW3U1BVenant", "CollectInput" → False];
```

SET2L1DFHYCBW3U1BVenant

```
[25-573] Include Tag : Kinematics - SET2L1DFHYCBW3U1BVenant (3 cells found, 3 evaluated)
```

SET2L1DFHYCBW3U1BVenant

$$\text{md0} = \text{SMSReplaceAll}\left[\frac{\text{SMSD}[\varkappa, \xi]}{\sqrt{\text{SMSD}[\varkappa, \xi].\text{SMSD}[\varkappa, \xi]}}, \xi \to -1\right];$$

$$\text{nd0} = \text{Cross}[\{0, 0, 1\}, \text{md0}];$$

$$\text{md1} = \text{SMSReplaceAll}\left[\frac{\text{SMSD}[\varkappa, \xi]}{\sqrt{\text{SMSD}[\varkappa, \xi].\text{SMSD}[\varkappa, \xi]}}, \xi \to 1\right];$$

$$\text{nd1} = \text{Cross}[\{0, 0, 1\}, \text{md1}];$$

SET2L1DFHYCBW3U1BVenant

$$\text{L} = \sqrt{(\text{Xi}[[2]] - \text{Xi}[[1]])^2 + (\text{Yi}[[2]] - \text{Yi}[[1]])^2};$$

$$\text{xl} = (\xi + 1) / 2\,\text{L};$$

$$\text{Tx} = -\text{T}[[1]]\,\text{Pu}[[1]] + \text{T}[[4]]\,\text{Pu}[[2]];$$

$$\text{Ty} = -\text{T}[[2]]\,\text{Pu}[[1]] + \text{T}[[5]]\,\text{Pu}[[2]];$$

$$\text{Np} = -\{\text{T}[[1]], \text{T}[[2]], 0\}.\text{md0}\,\text{Pu}[[1]] + \{\text{T}[[4]], \text{T}[[5]], 0\}.\text{md1}\,\text{Pu}[[2]];$$

$$\text{Vp} = -\{\text{T}[[1]], \text{T}[[2]], 0\}.\text{nd0}\,\text{Pu}[[1]] + \{\text{T}[[4]], \text{T}[[5]], 0\}.\text{nd1}\,\text{Pu}[[2]];$$

$$\text{Mp} = -\text{T}[[3]]\,\text{Pu}[[1]] + \text{T}[[6]]\,\text{Pu}[[2]] - \text{pyt}\,\text{L} / 2\,\text{xl} + \text{pyt}\,\text{xl}^2 / 2;$$

SET2L1DFHYCBW3U1BVenant

```
Please, consider using SMSSqrt instead of Sqrt. See also: Expression Optimization
```

SET2L1DFHYCBW3U1BVenant

```
SMSNPostNames = {"N", "V", "M", "Tx", "Ty", "DeformedMeshX", "DeformedMeshY"};
SMSExport[{Np, Vp, Mp, Tx, Ty, u, v}, npost$$[node, #] &];
SMSEndDo[];
```

SET2L1DFHYCBW3U1BVenant

```
SMSWrite[];
```

SET2L1DFHYCBW3U1BVenant

```
Method : SKR 300 formulae, 4454 sub-expressions
```

SET2L1DFHYCBW3U1BVenant

```
Method : SPP 216 formulae, 2775 sub-expressions
```

SET2L1DFHYCBW3U1BVenant

```
[46] File created : SET2L1DFHYCBW3U1BVenant.c Size : 69196
```

Environment

AceFEM Environment for 2D structural analysis

Environment

Requirements:

⇒ easy determination of boundary conditions

⇒ advaced posprocessing:-smooth displacements

⇒ standard display of loads and supports

⇒ standard tasks included

Environment

```
BeamDescription[] :=
 "Additional environment commands:
\nBeamNodeData[nodes_, keyword_,
   value_] ... specify nodal boundary conditions and nodal loads
\n_ Keywords:
\n.. 'Fx' 'Fy' 'Mz' ... prescribed force ( Fx(t+1)=Fx(t)+Δγ(t) Fx )
\n.. 'Fx0' 'Fy0' 'Mz0' ... prescribed initial force ( Fx(0)=Fx0 )
\n.. 'u' 'v' 'ϕ' ... prescribed displacement
\n.. 'u0' 'v0' 'ϕ0' ... prescribed initial displacement
\nBeamElementData[element_,'LeftHinge'] ... insert right hinge
\nBeamElementData[element_,'RightHinge'] ... insert right hinge
\nBeamElementData[element_, keyword_, value_] ... specify element surface loads
\n_ Keywords:
\n.. 'px' 'py' 'mz' ... prescribed load ( px(t+1)=px(t)+Δγ(t) px )
\n.. 'px0' 'py0' 'mz0' ... prescribed initial load ( px(0)=px0 )
\nBeamShowMesh[keyword_]
\n_ Keywords:
\n.. 'N' 'M' 'V' 'u'
\nBeamLimitLoadAnalysis[] - get
   limit load (γcr) by solving det(K)=0 problem by bisection
\n_ Options:
\n..'Start'->1 ... starting load factor
\n..'MaxLoad'->10 ... stop if load factor is greater
\n..'Tolerance'->10^-3 ... tolerance
\nBeamBucklingAnalysis[] - get buckling loads (γcr) and
   buckling shapes (Φ) by solving an eigenvalue problem (K0+γcr Kσ){Φ}={0}
\n_ Options:
\n..'Modes'->6 - calculate only n lowest buckling modes
\n..'ShowModes'->True - create graphic representation of buckling modes"
```

Environment

Boundary conditions, load

Environment

SMTConstrain[
 {list of nodes}
 ,index of dof to constrain
 , dof value : 0 unconstrained
 - 1 permanent constrain
 - 2 temporarily constrain (operator split)
]

Environment

```
BeamNodeData[i_]:=BeamNodeData[True&,i]
```

Environment

```
BeamNodeData[i_,j_]:=
 SMTNodeData[
  If[Head[i]=!=Function
    ,Print["Beam environment: wrong parameter:",i];SMCAbort[];
    ,Replace[i,Function[k_]:>Function[k && "ID"=="DFi"]]
  ],
  Switch[j,"Fx","dB","Fy","dB","Mz","dB","Fx0","Bp","Fy0","Bp","Mz0","Bp",
   "u","at","v","at","ϕ","at","u0","Bp","v0","Bp","ϕ0","Bp"
   ,_,Print["Beam environment: Unknown keyword: ",j];SMCAbort[];],
  "Part"->Switch[j,"Fx",1,"Fy",2,"Mz",3,"Fx0",1,"Fy0",2,"Mz0",3,
  "u",1,"v",2,"ϕ",3,"u0",1,"v0",2,"ϕ0",3]
  ];
```

Environment

```
BeamNodeData[i_,j_,b_]:=Block[{selnode},
 If[Head[i]=!=Function
    ,Print["Beam environment: wrong parameter:",i];SMCAbort[];
 ];
 selnode=SMTFindNodes[Replace[i,Function[k_]:>Function[k && "ID"=="DFi"]]];
 SMTNodeData[
  selnode
  ,Switch[j,"Fx","dB","Fy","dB","Mz","dB","Fx0","Bp","Fy0","Bp","Mz0","Bp"
   ,"u",SMTConstrain[selnode,1,-1];"dB"
   ,"v",SMTConstrain[selnode,2,-1];"dB"
   ,"ϕ",SMTConstrain[selnode,3,-1];"dB"
   ,"u0",SMTConstrain[selnode,1,-1];"Bp"
   ,"v0",SMTConstrain[selnode,2,-1];"Bp"
   ,"ϕ0",SMTConstrain[selnode,3,-1];"Bp"
   ,_,Print["Beam environment: Unknown keyword: ",j];SMCAbort[];
   ]
  ,b
  ,"Part"->Switch[j,"Fx",1,"Fy",2,"Mz",3,"Fx0",1,"Fy0",2,"Mz0",3,
           "u",1,"v",2,"ϕ",3,"u0",1,"v0",2,"ϕ0",3]
  ];
]
```

Environment

```
BeamFindElements[i_]:=SMTFindElements[i,Not[FreeQ[SMTDomainData[#,"CharSwitch"],"BeamEnvi
```

Environment

```
BeamElementData[i_,"LeftHinge"]:=
   Map[(
      SMTConstrain[SMTElements[[#,3,3]],1,0];
      AppendTo[SMTGraphicsElements,{#,BeamAdditionalGraphics}];
     )&,BeamFindElements[i]
   ];
```

Environment

```
BeamElementData[i_,"RightHinge"]:=
   Map[(
      SMTConstrain[SMTElements[[#,3,4]],1,0];
      AppendTo[SMTGraphicsElements,{#,BeamAdditionalGraphics}];
     )&,BeamFindElements[i]
    ];
```

Environment

```
BeamElementData[i_,j_,k_]:=Map[(
  AppendTo[SMTGraphicsElements,{#,BeamAdditionalGraphics}];
  Switch[j
   ,"px",SMTElementData[#,"Data",k,"Part"->1]
   ,"py",SMTElementData[#,"Data",k,"Part"->2]
   ,"mz",SMTElementData[#,"Data",k,"Part"->3]
   ,"px0",SMTElementData[#,"hp",k,"Part"->1]
   ,"py0",SMTElementData[#,"hp",k,"Part"->2]
   ,"mz0",SMTElementData[#,"hp",k,"Part"->3]
   ,_,SMCError=j;SMCAbort["Wrong keyword.","BeamSurfaceLoad",""];
  ]
  )&,BeamFindElements[i]];
```

Environment

```
BeamElementData[i_,j_]:=
 SMTElementData[
   BeamFindElements[i]
  ,Switch[j
     ,"px","Data","py","Data","mz","Data","px0","hp","py0","hp","mz0","hp"
     ,_,SMCError=j;SMCAbort["Wrong keyword.","BeamElementData",""];
   ]
  ,"Part"->Switch[i,"px",1,"py",2,"mz",3,"px0",1,"py0",2,"mz0",3]
 ];

 BeamElementData[i_]:=BeamElementData[True&,i]
```

Environment

Postprocessing

Environment

SMSAdditionalGraphics[
  {element index,domain index, list of nodes},
   True if node marks are required,
   True if boundary conditions are required,
   list of node coordinates for all element nodes
]
 - returns empty list ({}) or list of graphics primitives

Environment

```
BeamAdditionalGraphics[e_,m_,b_,n_]:=
If[b
 ,{
   (* sufrace load *)
   If[Chop[Plus@@Abs[SMTElementData[e[[1]],"Data"]]]=!=0,
      { Switch[Map[Abs[#]>0&,SMTElementData[e[[1]],"Data"]],
        {True,False,False},RGBColor[1,  0.501961,  0],{False,True,False},RGBColor[0.501961,
        {False,False,True},RGBColor[1,  0,  0.501961],_,RGBColor[0.501961,  1,  0.501961]],
        AbsoluteThickness[2],
        Map[
         Line[
          {
          Offset[ 3 {-1,1} ((#[[2]]-#[[1]])/Sqrt[(#[[2]]-#[[1]]).(#[[2]]-#[[1]])])[[{2,1}
          Offset[ 3 {-1,1} ((#[[2]]-#[[1]])/Sqrt[(#[[2]]-#[[1]]).(#[[2]]-#[[1]])])[[{2,1}
         }]&,{n[[{1,3}]],n[[{3,4}]],n[[{4,5}]],n[[{5,6}]],
         n[[{6,7}]],n[[{7,8}]],n[[{8,9}]],n[[{9,10}]],n[[{10,11}]],n[[{11,2}]]}
        ]
        },{}],
    (* LeftHinge *)
    If[SMTNodeData[e[[3,3]],"DOF"][[1]]>=0,{RGBColor[1,0,0],
        AbsolutePointSize[6],Point[Offset[
            5 (n[[2]]-n[[1]])/Sqrt[(n[[2]]-n[[1]]).(n[[2]]-n[[1]])]  ,n[[1]]]]},{}]
     (* RightHinge*)
    ,If[SMTNodeData[e[[3,4]] ,"DOF"][[1]]>=0,{RGBColor[1,0,0],
        AbsolutePointSize[6],
        Point[Offset[ -5 (n[[2]]-n[[1]])/
                Sqrt[(n[[2]]-n[[1]]).(n[[2]]-n[[1]])] ,n[[2]]]]},{}]
  }
 ,{}
]
```

Environment

SMTDrawEssentialBoundary[
 {x,y,z} ..node coordinates
 ,{dof1,dof2, } True-essential False -natural
 ,NodeID
]
- returns graphics primitive

Environment

```
SMTDrawEssentialBoundary[i_,j_,"DFi"]:=(
  Switch[j
      ,{True,True,True},
          {RGBColor[0,0,0],AbsoluteThickness[3],
           Line[{Offset[{-4,0},i],Offset[{4,0},i]}],Line[{Offset[{0,-4},i],Offset[{0,4},i]
      ,{True,True,False},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{i,Offset[{-5,-9},i],Offset[{5,-9},i],i}]}
      ,{True,False,False},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{i,Offset[{7,-5},i],Offset[{7,5},i],i}],
            Line[{Offset[{11,-5},i],Offset[{11,5},i]}]}
      ,{False,True,False},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{i,Offset[{-5,-7},i],Offset[{5,-7},i],i}],
            Line[{Offset[{-5,-11},i],Offset[{5,-11},i]}]}
      ,{True,False,True},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{Offset[{-3,-7},i],Offset[{-3,7},i]}],
           Line[{Offset[{3,-7},i],Offset[{3,7},i]}]}
      ,{False,True,True},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{Offset[{-7,-3},i],Offset[{7,-3},i]}],
           Line[{Offset[{-7,3},i],Offset[{7,3},i]}]}
      ,{False,False,True},
          {RGBColor[0,0,0],AbsoluteThickness[2],
           Line[{Offset[{-7,-3},i],Offset[{7,-3},i]}],
           Line[{Offset[{-7,3},i],Offset[{7,3},i]}],
           Line[{Offset[{-3,-7},i],Offset[{-3,7},i]}],
           Line[{Offset[{3,-7},i],Offset[{3,7},i]}]}
     ,_,{}
  ])
```

Environment

```
BeamShowMesh["All",j___Rule]:=Module[{q1,q2,q3},

q1={Max[#[[1]]]-Min[#[[1]]],Max[#[[2]]]-Min[#[[2]]]}&[Transpose[SMTNodeData["X"]]];

q3={"Legend"->False,"Show"->False,"TextStyle"->{FontSize->12,FontFamily->"Times"},j};

q2=Join[
{SMTShowMesh["DeformedMesh"->True,"Show"->False,
 "Label"->StyleForm[
  "Max[|u|]="<>SMTNumberForm[Sqrt[SMTPost["DeformedMeshX"]^2+SMTPost["DeformedMeshY"]^2]//
   "  Max[|ϕ|]="<>SMTNumberForm[Abs[BeamNodeData["ϕ"]]//Max,6]<>"\n",FontSize->12,FontWei
   "TextStyle"->{FontSize->12,FontFamily->"Times",FontWeight->"Bold"},
   "Marks"->True,"BoundaryConditions"->True,"NodeTagOffset"->{0.05,  0.05, 0.05},Sequence@
If[Chop[SMTPost["N"]//Abs//Max]==0
  ,{}
  ,{SMTShowMesh["Field"->SMTPost["N","Smooth"->False],"Label"->{"N:  ",Automatic},Sequence
],
If[Chop[SMTPost["V"]//Abs//Max]==0
  ,{}
  ,{SMTShowMesh["Field"->SMTPost["V","Smooth"->False],"Label"->{"V:  ",Automatic},Sequence
],
If[Chop[SMTPost["M"]//Abs//Max]==0
  ,{}
  ,{SMTShowMesh["Field"->SMTPost["M","Smooth"->False],"Label"->{"M:  ",Automatic},"LineFie
]
];


Grid[Join[{{StyleForm[StringJoin[
      "Results of static analysis for γ=",ToString[SMTRData["Multiplier"]]],FontSize->14]
  SpanFromLeft}}, Partition[q2, 2, 2, 1, ""]]]

]
```

Environment

```
BeamShowMesh["N",i___Rule]:=SMTShowMesh[i,"Field"->SMTPost["N","Smooth"->False],"Legend"-
"Label"->StyleForm["Axial force N",FontSize->14]];
BeamShowMesh["M",i___Rule]:=SMTShowMesh[i,"Field"->SMTPost["M","Smooth"->False],"Legend"-
"Label"->StyleForm["Bending moment M",FontSize->14],"LineFieldScale"->-1];
BeamShowMesh["V",i___Rule]:=SMTShowMesh[i,"Field"->SMTPost["V","Smooth"->False],"Legend"-
"Label"->StyleForm["Shear force V",FontSize->14]];
BeamShowMesh["Tx",i___Rule]:=SMTShowMesh[i,"Field"->SMTPost["Tx","Smooth"->False],"Legend
"Label"->StyleForm["Force Tx",FontSize->14]];
BeamShowMesh["Ty",i___Rule]:=SMTShowMesh[i,"Field"->SMTPost["Ty","Smooth"->False],"Legend
"Label"->StyleForm["Force Ty",FontSize->14]];
BeamShowMesh["u",i___Rule]:=SMTShowMesh[i,"DeformedMesh"→True,"BoundaryConditions"→True,
"Label"->{StyleForm["Displacements",FontSize→14]}];
BeamShowMesh[i___Rule]:=SMTShowMesh[i,"DeformedMesh"→True,"BoundaryConditions"→True,"Mark
```

Environment

Limit load analysis by bisection

Environment

```
BeamLimitLoadAnalysis[g0_, gMax_, e_]:=
Module[{stepnds},
SMTNextStep[1, g0];
While[
  While[step = SMTConvergence[10^(-9), 20, {"Adaptive", 8, g0/100, gMax/2, gMax}], SMTNew
  SMTStatusReport[nds=SMTIData["DiagonalSign"]];
  step[[3]] && nds == 0
 ,If[step[[1]], SMTStepBack[];];
  SMTNextStep[1, step[[2]]]
];
SMTStepBack[];
SMTNextStep[1, SMTRData["MultiplierIncrement"]/2];
While[
 While[step = SMTConvergence[10^(-8), 20, "Analyze"], SMTNewtonIteration[];];
 SMTStatusReport[nds=SMTIData["DiagonalSign"]];
 Abs[SMTRData["MultiplierIncrement"]] > Abs[e*SMTRData["Multiplier"]],
 If[step =!= False || nds > 0, SMTStepBack[];];
 SMTNextStep[1, SMTRData["MultiplierIncrement"]/2];
];
SMTStatusReport[SMTIData["DiagonalSign"]];
Print["Critical load factor (bisection) =", SMTRData["Multiplier"]];
SMTRData["Multiplier"]
]
```

Environment

Buckling analysis

Environment

```
Options[BeamBucklingAnalysis]={"Modes"->6,"ShowModes"->True};
BeamBucklingAnalysis[allopt___Rule]:=
Module[{K0, KG,eigs, nnotzero, bload, bshape, alldofs,q1,q2,at,opt},

  opt={"Modes","ShowModes","ShowDiagrams"}/.{allopt}/.Options[BeamBucklingAnalysis];

  If[And@@Map[Not[FreeQ[#,"GeometricTangentMatrix"]]&,SMTDomainData["CharSwitch"]]
   ,SMTNewtonIteration[];
    SMTIData["GeometricTangentMatrix",1];
    K0=SMTData["TangentMatrix"];
    SMTIData["GeometricTangentMatrix",2];
    KG=SMTData["TangentMatrix"];
    SMTIData["GeometricTangentMatrix",0];
    SMTNewtonIteration[];
    SMTConvergence["Reset"];
    While[SMTConvergence[10^-8,10],SMTNewtonIteration[];];
   ,K0=SMTData["TangentMatrix"];
    While[SMTConvergence[10^-8,10],SMTNewtonIteration[];];
    KG=SMTData["TangentMatrix"]-K0;
  ];
  Off[Eigensystem::argpd];
  eigs=Re[Eigensystem[{K0,-KG},-opt[[1]]]];
  Onn[Eigensystem::argpd];
  bload=eigs[[1]]//Reverse;
  nnotzero=Min[opt[[1]], Length[bload]];
  bshape=eigs[[2]]//Reverse;
  alldofs=SMTNodeData["at"];

 q1={Max[#[[1]]]-Min[#[[1]]],Max[#[[2]]]-Min[#[[2]]]}&[Transpose[SMTNodeData["X"]]];
 q2=Table[
     at=MapIndexed[
          If[#1 < 0,Extract[alldofs, #2],bshape[[i,#1+1]]]&
       ,SMTNodeData["DOF"],{2}];
      SMTNodeData["at",at];
      SMTShowMesh["BoundaryConditions"->False,"DeformedMesh"->True,
       "Scale"->SMTRData["XYZRange"]/15/Max[Abs[{SMTPost["DeformedMeshX"],SMTPost["Defor
       "Label"->{"Mode=",i," ",SubscriptBox["ɣ","CR"]//DisplayForm,"=",
        bload[[i]]},"Marks"-> False,"Show"->False
      ]
 ,{i,1,nnotzero}];
 SMTNodeData["at",alldofs];
 Print[
   Grid[
     Join[
      {{Style[Row[{
        "Buckling analysis - first ", nnotzero, " buckling modes"}],
        FontSize -> 14], SpanFromLeft}},
      Partition[q2,3,3,1, ""]
     ]
    , Frame → All]
 ];
 bload
]
```