finite element using Ritz method for axial loaded beam

Initialization Code (optional)

Manipulate

```
Manipulate[process[totalLength, numberOfElements, area , youngModulus, traction, endLoad],
 Grid[
  {

   {
    Control[{{totalLength, 1, Column[{Style["Total", 10], Style["Length", 10]}]},
      .1, 10, .1, Appearance → "Labeled", ImageSize → Tiny }],
    Control[{{numberOfElements, 3, Column[{Style["number", 10], Style["of elements", 10]}]},
      2, 14, 1, Appearance → "Labeled", ImageSize → Tiny}],
    Control[{{area , 1, Column[{Style["A, cross", 10], Style["section", 10], Style["area", 10]}]},
      1, 10, 1, Appearance → "Labeled", ImageSize → Tiny}]
   },

   {
    Control[{{youngModulus , 1, Column[{Style["E, Young's", 10], Style["modulus", 10]}]},
      .01, 5, .01, Appearance → "Labeled", ImageSize → Tiny}],
    Control[{{traction , 1, Column[{Style["c, traction", 10]}]}, .01, 5,
      .01, Appearance → "Labeled", ImageSize → Tiny}],
    Control[{{endLoad , 0, Column[{Style["P, force at", 10], Style["end", 10]}]},
      0, 10, .01, Appearance → "Labeled", ImageSize → Tiny}]
   }
  },
  Frame → All, FrameStyle → Directive[AbsoluteThickness[.1], Gray], Spacings → {2, 0},
  ItemSize → Automatic
 ],

 {{plotWidth, 280}, ControlType → None},
 {{plotHeight, 200}, ControlType → None},
 {{plotImagePadding, 38}, ControlType → None},
 (*ContentSize→Automatic,*)
 FrameMargins → 0,
 ImageMargins → 0,
 ContinuousAction → False,
 SynchronousUpdating → True,
 ControlPlacement → Top,
 FrameMargins → 0,
 AutorunSequencing → {{2, 40}},
 ImageMargins → 0
 , Initialization :→
  (

   process[totalLength_, numberOfElements_, area_ , youngModulus_, traction_, endLoad_] :=
     Module[{numberOfNodes, N, L, E, c, A, p, y, z, B, Π, strainEnergy, externalLoad,
       pVector, gamma, d, x, eqs, b, kmat, output, sol, data, p2, p1, p3, p4, exactSolution,
       strain, strainData, stressData, actualStrain, actualStress, k, pVectorString},
      numberOfNodes = numberOfElements + 1;
```

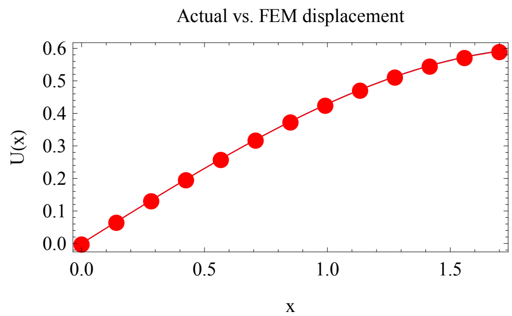
```
(N = {{(L - s)/L, s/L}}) // MatrixForm;
(B = D[N, s]) // MatrixForm;
Π = 0;
strainEnergy = Table[0, {numberOfElements}];
externalLoad = Table[0, {numberOfElements}];
pVector = Table[0, {numberOfNodes}];

gamma = Table[0, {numberOfElements}];
Do[
 d = Array[{u#} &, 2, k];
 strainEnergy[[k]] = (Simplify[(A E)/2 ∫_0^L (Transpose[d].Transpose[B].B.d) ⅆs])[[1, 1]];
 x = (k - 1) * L;
 gamma[[k]] = Simplify[c Transpose[d].∫_0^L (Transpose[N] (x + s)) ⅆs];

 If[k == numberOfElements, externalLoad[[k]] = d[[2]] * p, externalLoad[[k]] = 0];

 Π = Π + (strainEnergy[[k]] - gamma[[k]] - externalLoad[[k]])[[1, 1]];
 , {k, 1, numberOfElements}
];
Π = Simplify[Π];

(*Print["Strain energy initially=",strainEnergy];*)

d = Array[u# &, numberOfNodes, 1];
eqs = Table[0, {numberOfNodes}];
Do[
 eqs[[k]] = D[Π, d[[k]]] == 0;
 , {k, 1, numberOfNodes}
];

{b, kmat} = Normal[CoefficientArrays[eqs, d]];

pVector[[-1]] = p;
pVectorString = pVector;
pVectorString[[-1]] = "p";
output = Text[ToString[Style["(E A)/L", 14], FormatType → TraditionalForm] <>
    ToString[MatrixForm[kmat / (E A / L)], FormatType → TraditionalForm] <>
    ToString[MatrixForm[Transpose[{d}]], FormatType → TraditionalForm] <>
    " = " <> ToString[Style["c L^2 ", 14], FormatType → TraditionalForm] <>
    ToString[MatrixForm[(-b - pVector) / (c L^2)], FormatType → TraditionalForm] <>
    " + " <> ToString[MatrixForm[pVectorString], FormatType → TraditionalForm]];
pVector[[-1]] = p;
sol = LinearSolve[kmat[[2 ;; -1, 2 ;; -1]], -b[[2 ;; -1]]];
PrependTo[sol, 0];


L = totalLength / numberOfElements;
E = youngModulus;
c = traction;
A = area;
p = endLoad;
(*Print["sol=",sol];*)
data = Table[{(i - 1) * L, sol[[i]]}, {i, 1, numberOfNodes}];

p1 = ListPlot[data, Joined → True, PlotMarkers → {Automatic, Medium}, PlotStyle →
```

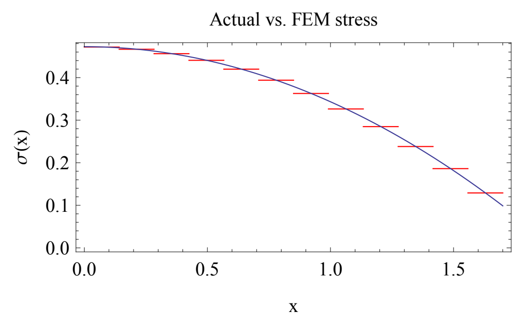
```
    {Red, PointSize[Large]}, ImageSize → {plotWidth, plotHeight}, ImagePadding → plotImagePadding,
   AspectRatio → 0.48];

(*Clear[z,y];*)
exactSolution = First@DSolve[{A E y''[z] == -c z, y[0] == 0, y'[totalLength] == P/(A E)}, y[z], z];
y = y[z] /. exactSolution;
(*Print["exact solution=",y];*)

p2 = Plot[y, {z, 0, totalLength}, ImageSize → {plotWidth, plotHeight},
   ImagePadding → plotImagePadding, AspectRatio → 0.48, PlotRange → All];

p3 = Show[{p2, p1}, Frame → True, FrameLabel → {{"U(x)", None}, {"x", "Actual vs. FEM displacement"}}];

Do[d[[i]] = {sol[[i]]}, {i, 1, numberOfNodes}];

(*Print["strain energy before=",N[strainEnergy]];*)
strain = Table[0, {numberOfElements}];

For[k = 1, k <= numberOfElements, k++,
 strain[[k]] = (B.{{sol[[k]]}, {sol[[k + 1]]}})[[1, 1]]
];

(*Print["strain =",N[strain]];*)

strainData = Table[{{L (k - 1), strain[[k]]}, {L k, strain[[k]]}}, {k, 1, numberOfElements}];

(*Print["straindata=",N[straindata]];*)

stressData = Table[{{L (k - 1), E strain[[k]]}, {L k, E strain[[k]]}}, {k, 1, numberOfElements}];

(*Print["stress data=",N[stressData]];*)

p1 = ListPlot[stressData, Joined → True, AxesOrigin → {0, 0}, PlotStyle → {Red},
   ImageSize → {plotWidth, plotHeight}, ImagePadding → plotImagePadding, AspectRatio → 0.48];

actualStrain = D[y, z];

actualStress = E actualStrain;
p2 = Plot[actualStress, {z, 0, totalLength},
   ImageSize → {plotWidth, plotHeight}, ImagePadding → plotImagePadding, AspectRatio → 0.48];

p4 = Show[{p1, p2}, Frame → True,
   PlotRange → All, FrameLabel → {{"σ(x)", None}, {"x", "Actual vs. FEM stress"}}];

Grid[{{output, SpanFromLeft}, {p3 , p4}}, Frame → None,
 Alignment → Center, Spacings → {1, 0}, ItemSize → Automatic(*{Scaled[.5],Scaled[.5]}*)]
];

 )
]
```

Total Length 1.7 | number of elements 12 | A, cross section area

E, Young's modulus 1 | c, traction 2.07 | P, force at end

$$\frac{E\,A}{L}\begin{pmatrix}1&-1&0&0&0&0&0&0&0&0&0&0&0\\-1&2&-1&0&0&0&0&0&0&0&0&0&0\\0&-1&2&-1&0&0&0&0&0&0&0&0&0\\0&0&-1&2&-1&0&0&0&0&0&0&0&0\\0&0&0&-1&2&-1&0&0&0&0&0&0&0\\0&0&0&0&-1&2&-1&0&0&0&0&0&0\\0&0&0&0&0&-1&2&-1&0&0&0&0&0\\0&0&0&0&0&0&-1&2&-1&0&0&0&0\\0&0&0&0&0&0&0&-1&2&-1&0&0&0\\0&0&0&0&0&0&0&0&-1&2&-1&0&0\\0&0&0&0&0&0&0&0&0&-1&2&-1&0\\0&0&0&0&0&0&0&0&0&0&-1&2&-1\\0&0&0&0&0&0&0&0&0&0&0&-1&1\end{pmatrix}\begin{pmatrix}u_1\\u_2\\u_3\\u_4\\u_5\\u_6\\u_7\\u_8\\u_9\\u_{10}\\u_{11}\\u_{12}\\u_{13}\end{pmatrix}=c\,L^2\begin{pmatrix}\frac{1}{6}\\1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\\frac{35}{6}\end{pmatrix}+\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\0\\p\end{pmatrix}$$





Caption

This is a demonstration written in *Mathematica* showing finite element solution by Ritz method for a simple axially loaded beam fixed at one end and free to extend on the other end

Thumbnail

Snapshots

Details (optional)

Using the Ritz method, the solution to axial deformation is found. As more elements are used, the solution is seen to converge to the exact solution.

Control Suggestions (optional)

- [ ] Slider Zoom
- [ ] Drag Locators
- [ ] Rotate and Zoom in 3D
- [ ] Automatic Animation
- [ ] Gamepad Controls
- [ ] Resize Images
- [ ] Bookmark Animation

Search Terms (optional)

finite element methods

Related Links (optional)

Authoring Information

Contributed by: Nasser M. Abbasi