Analysis of a single span Euler Bernoulli beam under different loading conditions

Initialization Code (optional)

Manipulate

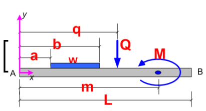
```
(*-- by Nasser M. Abbasi 12/17/2009 --*)
Manipulate[
 process[len, If[a > b, a = b, a], If[b > len, b = len, b], If[q > len, q = len, q],
  If[m > len, m = len, m], If[positiveM, M, -M], If[upW == True, -w, w], If[upQ == True, -Q, Q],
  If[StringMatchQ[youngModulusChoice, "Slider"], youngModulusSlider * 10 ^ 6,
   If[unitsForE == 2, youngModulusMenu * 6894 * 10 ^ 6, youngModulusMenu * 10 ^ 6]],
  If[StringMatchQ[momentOfInertiaChoice, "Slider"], momentOfInertia, (1 / 12) * sectionb * sectionh ^ 3],
  beamSupportType, (1 / maxDeflectionRatio) * len],

 Item[
  Column[
   {

    Grid[
     {
      { im1},
      {Control[{{len, 144, Style["L", 10]}, 1, 300, .1, ImageSize → Small, Appearance → "Labeled"}] }
     }, Frame → True, FrameStyle → Thin, Alignment → Center, Spacings → {1.33, 1}
    ],

    Grid[{
      {
       Style["distributed load w", 10],
       Button[Style["remove load", 10], w = 0, ImageSize → Small]
      },
      {
       Column[
        {
         Control[{{w, 8, Style["w", 10]}, 0, 200, 1, ImageSize → Tiny, Appearance → "Labeled"}],
         Control[
          {{a, 0.3 * len, Style["a", 10]}, 0, Dynamic[len], .01, ImageSize → Tiny, Appearance → "Labeled"}],
         Control[{{b, a + 0.2 (len - a), Style["b", 10]}, 0, Dynamic[len], .01,
           ImageSize → Tiny, Appearance → "Labeled"}]
        }
       ],
       SetterBar[ Dynamic[upW],
        {False -> Graphics[{Arrowheads[.8], Arrow[{{0, 0}, {0, -3}}]}, ImageSize → {16, 26}],
         True -> Graphics[{Arrowheads[.8], Arrow[{{0, 0}, {0, 3}}]}, ImageSize → {16, 26}]}]
      }
     }, Frame → True, FrameStyle → Thin, Spacings → {.5, Automatic}
    ],


    Grid[{
      {
       Style["applied point load Q  ", 10],
```

```
        Button[Style["remove load", 10], Q = 0, ImageSize → Small]
       },
       {
        Column[
         {
          Control[{{Q, 400, Style["Q", 10]}, 0, 1000, 1, ImageSize → Tiny, Appearance → "Labeled"}] ,
          Control[
            {{q, .7 len, Style["q", 10]}, 0.001, Dynamic[len], .001, ImageSize → Tiny, Appearance → "Labeled"}]
         }],
        SetterBar[ Dynamic[upQ],
         {False -> Graphics[{Arrowheads[.8], Arrow[{{0, 0}, {0, -3}}]}, ImageSize → {16, 26}],
          True -> Graphics[{Arrowheads[.8], Arrow[{{0, 0}, {0, 3}}]}, ImageSize → {16, 26}]}]
       }
      }, Frame → True, FrameStyle → Thin, Spacings → {.5, Automatic}
     ],


     Grid[{
       {
        Style["applied moment M", 10],
        Button[Style["remove load", 10], M = 0, ImageSize → Small]
       },
       {
        Column[
         {
          Control[{{M, 40, Style["M", 10]}, 0, 1000, 1, ImageSize → Tiny, Appearance → "Labeled"}],
          Control[
            {{m, .75 len, Style["m", 10]}, 0, Dynamic[len], .01, ImageSize → Tiny, Appearance → "Labeled"}]
         }
        ],
        SetterBar[ Dynamic[positiveM], {False -> Graphics[getMomentSymbol[.02, Black, 300, 80],
             ImageSize → 18], True -> Graphics[getMomentSymbol[.02, Black, -120, 100], ImageSize → 18]}]
       }
      }, Frame → True, FrameStyle → Thin, Spacings → {.5, Automatic}
     ],


     Grid[{
       {
        Style["maximum allowed beam deflection ratio", 10]
       },
       {
        Control[{{maxDeflectionRatio, 180, "D "}, 90, 720, 1, ImageSize → Small, Appearance → "Labeled"}]
       }
      }, Frame → True, FrameStyle → Thin, Spacings → {3.6, .3}
     ]

    } (* end column[] *)
   ], ControlPlacement → Left
  ], (* end Item[] for left side controls *)


  Item[(*now do the top controls *)
   Row[
    {

     Grid[
      {
       {Style["Select support type", 10]},
       {supportTypes},
       {Control[
          {{beamSupportType, 1, ""}, {1 -> "1", 2 -> "2", 3 → "3", 4 → "4"}, ControlType → RadioButtonBar}]}
      }, Frame → True, FrameStyle → Thin
     ],
```

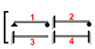
```
Grid[{
  {
   Style["choose Young's modulus (E) from ", 10],
   SetterBar[ Dynamic[youngModulusChoice], {"Menu" -> "Menu", "Slider" -> "Slider"}]
  },
  {

   Grid[{
     {
      Control[{ {youngModulusMenu, 30, ""}, {80 → Style["Osmium [80,551]", 9],
          75 → Style["Iridium [75,517]", 9], 36 → Style["Chromium [36,248]", 9],
          30 → Style["Steel [30,206]", 9], 30 → Style["Cobalt [30,206]", 9],
          29.5 → Style["Carbon steel [29.5,203]", 9], 28.5 → Style["Iron [28.5,196.5]", 9],
          24 → Style["Uranium [24,165.5]", 9], 21.3 → Style["Platinum [21.3,146.9]", 9],
          17 → Style["Copper [17,117]", 9], 16 → Style["Silicon [16,110.3]", 9],
          10.8 → Style["Gold [10.8,74]", 9], 10.5 → Style["Silver [10.5,72.3]", 9],
          10.0 → Style["Aluminum [10,69]", 9], 6.4 → Style["Magnesium [6.4,44]", 9],
          1.3 → Style["Pine wood [1.3,8.9]", 9], 0.5 → Style["fibreboard [0.5,3.4]", 9]},
         Enabled → Dynamic@StringMatchQ[youngModulusChoice, "Menu"], ControlType → PopupMenu}],

      Control[{{unitsForE, 1, "units"},
        {1 → "imperial", 2 → "metric"}, ControlType -> RadioButtonBar, Appearance → "Vertical"}]}
    }, Frame → True, FrameStyle → Directive[Dotted], Spacings → {1, Automatic}
   ],

   SpanFromLeft
  },
  {
   Control[{{youngModulusSlider, 1.6, "E"}, 1, 300 000, 1, ImageSize → Small,
     Enabled → Dynamic@StringMatchQ[youngModulusChoice, "Slider"], Appearance → "Labeled"}
   ],
   SpanFromLeft
  }
 }, Frame → True, Alignment → Left, Spacings → {1, Automatic}
],


Grid[{
  {
   Style["choose moment of inertia (I) from", 10],
   SetterBar[ Dynamic[momentOfInertiaChoice], {"crossSection" -> "section", "Slider" -> "Slider"}]
  },

  {
   Grid[
    {

     {
      Grid[
       {
        {Control[{{sectionb, 2, "b"}, .1, 100, .1, ImageSize → Small, Enabled →
            Dynamic@StringMatchQ[momentOfInertiaChoice, "crossSection"], Appearance → "Labeled"}]
        },
        {Control[{{sectionh, 8, "h"}, .1, 100, .1, ImageSize → Small, Enabled →
            Dynamic@StringMatchQ[momentOfInertiaChoice, "crossSection"], Appearance → "Labeled"}]
        }
       }, Spacings → {1, 0}, Alignment → Left
      ],
      Labeled[
       Graphics[Polygon[{ {0, 0}, {0, 5}, {2, 5}, {2, 0}}], ImageSize → 8], {"b", "h"}, {Bottom, Right}]
     }
    }, Frame → True, FrameStyle → Directive[Dotted], Spacings → {1, 0}, Alignment → Left
   ], SpanFromLeft
```



```
      },

      {
       Control[{{momentOfInertia, 57.1, "I"}, 1, 100, .1, ImageSize → Small,
         Enabled → Dynamic@StringMatchQ[momentOfInertiaChoice, "Slider"], Appearance → "Labeled"}],

       SpanFromLeft
      }
     }, Frame → True, Alignment → Left, Spacings → {1, Automatic}
    ](*end Grid[]*)
   }(*end Row[] *)
  ], ControlPlacement → Top
 ], (* end Item[] *)


 FrameMargins → 0,
 ImageMargins → 0,
 ContinuousAction → False,
 SynchronousUpdating → True,
 AutorunSequencing → {1, 2, 3},


 Initialization :→
  {

   im1 = Image[
```




```
   , ImageSize → {195, 80}];

   supportTypes = Image[
```




```
   , ImageSize → {100, 58}];
   youngModulusChoice = "Menu";
   momentOfInertiaChoice = "Slider";
   upW = False;
   upQ = False;
   positiveM = True;

   getMomentSymbol[radius_, color_, from_, to_] := Module[{data, theta},
     data = Table[{radius * Cos[theta], radius * Sin[theta]},
       {theta, from * Degree, to * Degree, If[from < to, 1 * Degree, -1 * Degree]}
      ];
     {Thickness[0.001], color, Arrowheads[Medium], Arrow[Line[data]]}];

   getBeamCurveAnnotation[y_, i_, x_] := Module[{},
     Column[{
       Text[ Row[{Style["x", Italic, 14], " = ", NumberForm[N[i], {30, 3}], " ",
          Style["y(x)", Italic, 14], " = ", NumberForm[N[y /. x → i], {30, 3}]}]]}]];

   getMaxAbsoluteDeflection[y_, L_, var_] := Module[{data},
     data = Table[{Abs[N[y /. var → i]], i}, {i, 0, L, L / 100.}];
     (*sorts by default is small to large*)
     data = Sort[data];
     {data[[-1, 2]], data[[-1, 1]]}];
                 (*---- main Manipulate process ---*)
   process[L_, a_, b_, q_, m_, M_, w_, Q_, ee_, ii_, beamSupportType_, maxDeflection_] :=
    Module[{shear, moment, R1, R2, y, sol, headings, yMax, yMaxLocation, pDeflection, pMoment,
      pShear, pAnnotated, commonEpilog, ydata, x, d, c1, title, shearSaved, M2, M1, slope, c2,
      verticalEquilibrium, momentEquilibrium, b1, b2, pDeflectionOpts, clockWiseStartingAngle = 320,
      clockWiseEndingAngle = 30, antiClockWiseStartingAngle = 30, antiClockWiseEndingAngle = 320, data,
      maxDeflectionEstimate, yLeftEnd, yRightEnd, slopeLeftEnd, slopeRightEnd, u = UnitStep, yAtMax},

     (*---            Notations             ----*)
     (* R1: left end reaction                  *)
```

```
(* R2: right end reaction                *)
(* M1: left end moment reaction          *)
(* M2: right end moment reaction         *)
(* M:  is the external applied moment    *)
(* w:  is the external distributed load  *)
(* Q:  is the external point load        *)

(*--- set up the main equations --*)
shear = R1 - w (x - a) u[x - a] + w (x - b) u[x - b] - Q u[x - q];
moment = M1 + R1 x - w/2 (x - a)^2 u[x - a] + w/2 (x - b)^2 u[x - b] - Q (x - q) u[x - q] - M u[x - m];
slope = M1 x + R1 x^2/2 - w/6 (x - a)^3 u[x - a] + w/6 (x - b)^3 u[x - b] - Q/2 (x - q)^2 u[x - q] - M (x - m) u[x - m] + c1;
y = 1/(ee ii) (M1 x^2/2 + R1 x^3/6 - w/24 (x - a)^4 u[x - a] +
    w/24 (x - b)^4 u[x - b] - Q/6 (x - q)^3 u[x - q] - M/2 (x - m)^2 u[x - m] + c1 x + c2);
verticalEquilibrium = R1 + R2 - Q - w (b - a);
momentEquilibrium = M1 + w (b - a) ((a + b)/2) + Q q - M - M2 - R2 L;

(*--- based on loading type, setup boundary conditions and solve ---*)
yLeftEnd = y /. x → 0;
yRightEnd = y /. x → L;
slopeLeftEnd = slope /. x → 0;
slopeRightEnd = slope /. x → L;

Which[

 beamSupportType == 1, {(*simple supported at both ends*)
  M1 = 0;
  M2 = 0;
  sol = First@Solve[Simplify[{yLeftEnd == 0, yRightEnd == 0}], {c1, c2}];
  c1 = c1 /. sol;
  c2 = c2 /. sol;
  sol =
   ToRules@Quiet[Reduce[{verticalEquilibrium == 0, momentEquilibrium == 0}, {R1, R2}], RowReduce::luc];
  R1 = R1 /. sol;
  R2 = R2 /. sol;},

 beamSupportType == 2, {(*fixed one side, simple supported at other*)
  M2 = 0;
  sol = First@Solve[{yLeftEnd == 0, slopeLeftEnd == 0}, {c1, c2}];
  c1 = c1 /. sol;
  c2 = c2 /. sol;
  sol = ToRules@Quiet[Reduce[Simplify[{yRightEnd == 0, verticalEquilibrium == 0, momentEquilibrium == 0}],
      {R1, R2, M1}], RowReduce::luc];
  R1 = R1 /. sol;
  R2 = R2 /. sol;
  M1 = M1 /. sol;},

 beamSupportType == 3, {(*fixed one side, free at other*)
  M2 = 0; R2 = 0;
  sol = First@Solve[{yLeftEnd == 0, slopeLeftEnd == 0}, {c1, c2}];
  c1 = c1 /. sol;
  c2 = c2 /. sol;
  sol = ToRules@Quiet[
     Reduce[Simplify[{verticalEquilibrium == 0, momentEquilibrium == 0}], {R1, M1}], RowReduce::luc];
  R1 = R1 /. sol;
  M1 = M1 /. sol;},

 beamSupportType == 4, {(*fixed at both sides*)
```



```
  sol = First@Solve[{yLeftEnd == 0, slopeLeftEnd == 0}, {c1, c2}];
  c1 = c1 /. sol;
  c2 = c2 /. sol;
  sol = ToRules@Quiet[Reduce[Simplify[{yRightEnd == 0, slopeRightEnd == 0,
        verticalEquilibrium == 0, momentEquilibrium == 0}], {R1, R2, M1, M2}], RowReduce::luc];
  R1 = R1 /. sol;
  R2 = R2 /. sol;
  M1 = M1 /. sol;
  M2 = M2 /. sol;}
];

(*-- special cases all result in no defelection  --*)
If[ ( (a == L || b == 0 || (b - a) == 0) && (w ≠ 0 && Q == 0 && M == 0)) ||
   (w == 0 && Q == 0 && M == 0) || (w == 0 && Abs[Q] > 0 && (q == 0 || (q == L && beamSupportType ≠ 3))) ,
 {
  y = 0;
  yMax = 0;
  yMaxLocation = L / 2;
  maxDeflectionEstimate = .5;
 },
 {
  {yMaxLocation, yMax} = getMaxAbsoluteDeflection[y, L, x];
  maxDeflectionEstimate = maxDeflection + .3 * maxDeflection;
 }
];

yAtMax = y /. x -> yMaxLocation;
commonEpilog := {
  Which[

   beamSupportType == 1,
   {
    Polygon[{{0.05 * L, -maxDeflectionEstimate / 4}, {0, 0}, {-0.05 * L, -maxDeflectionEstimate / 4}}],
    {PointSize[.05], Black, Point[{L, -.122 maxDeflectionEstimate}]}
   },

   beamSupportType == 2,
   { (*left and right support*)
    {Thickness[.04], Black, Line[{  {0, maxDeflectionEstimate / 8}, {0, -maxDeflectionEstimate / 8}}]},
    {PointSize[.05], Black, Point[{L, -.122 maxDeflectionEstimate}]},

    (* left end moment *)
    (*notice sign. Positive means here clockwise which is negative*)
    If[M1 > 0,
     {
      Opacity[1],
      Black,
      Inset[ Graphics[getMomentSymbol[0.22 * L, Black, 280, 75]],
       {-0.08 * L, 0}, {0, 0}, 0.125 * L],
      Text[Style[NumberForm[N[M1], {30, 2}], 12],
       {-0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
     },
     {
      Opacity[1], Black, Inset[ Graphics[getMomentSymbol[0.22 * L, Black, 80, 270]],
       {-0.08 * L, 0}, {0, 0}, 0.125 * L],
      Text[Style[NumberForm[N[-M1], {30, 2}], 12],
       {-0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
     }
    ]
   },

   beamSupportType == 3,
   {
    {Thickness[.04],
```

```
      Black,
      Line[{  {0, maxDeflectionEstimate / 8}, {0, -maxDeflectionEstimate / 8}}]
     },
     If[M1 > 0,
      {
       Opacity[1],
       Black,
       Inset[ Graphics[getMomentSymbol[0.15 * L, Black, 300, 80]], {-0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[M1], {30, 2}], 12],
        {-0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
      },
      {
       Opacity[1], Black,
       Inset[ Graphics[getMomentSymbol[0.15 * L, Black, 80, 270]], {-0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[-M1], {30, 2}], 12], {-0.08 * L, 0.65 maxDeflectionEstimate},
        {0, 1}, {1, 0}]
      }
     ]
    },

    beamSupportType == 4,
    {
     {
      Thickness[.04],
      Black,
      Line[{  {0, maxDeflectionEstimate / 8}, {0, -maxDeflectionEstimate / 8}}]
     },
     {
      Thickness[.04],
      Black,
      Line[{  {L, maxDeflectionEstimate / 8}, {L, -maxDeflectionEstimate / 8}}]
     },
     If[M1 > 0,
      {
       Black,
       Inset[ Graphics[getMomentSymbol[0.15 * L, Black, 300, 80]], {-0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[M1], {30, 2}], 12],
        {-0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
      },
      {
       Black,
       Inset[ Graphics[getMomentSymbol[0.15 * L, Black, 80, 270]], {-0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[-M1], {30, 2}], 12],
        {-0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
      }
     ],
     If[M2 > 0,
      { (* this is RHS moment, positive means anticlock wise*)
       Black,
       Inset[ Graphics[getMomentSymbol[0.125 * L, Black, -120, 100]], {L + 0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[M2], {30, 2}], 12],
        {L + 0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
      },
      {
       Black,
       Inset[ Graphics[getMomentSymbol[0.125 * L, Black, 120, -90]], {L + 0.08 * L, 0}, {0, 0}, 0.125 * L],
       Text[Style[NumberForm[N[-M2], {30, 2}], 12],
        {L + 0.08 * L, 0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}]
      }
     ]
    }
   ],

   (*stright line*)
```

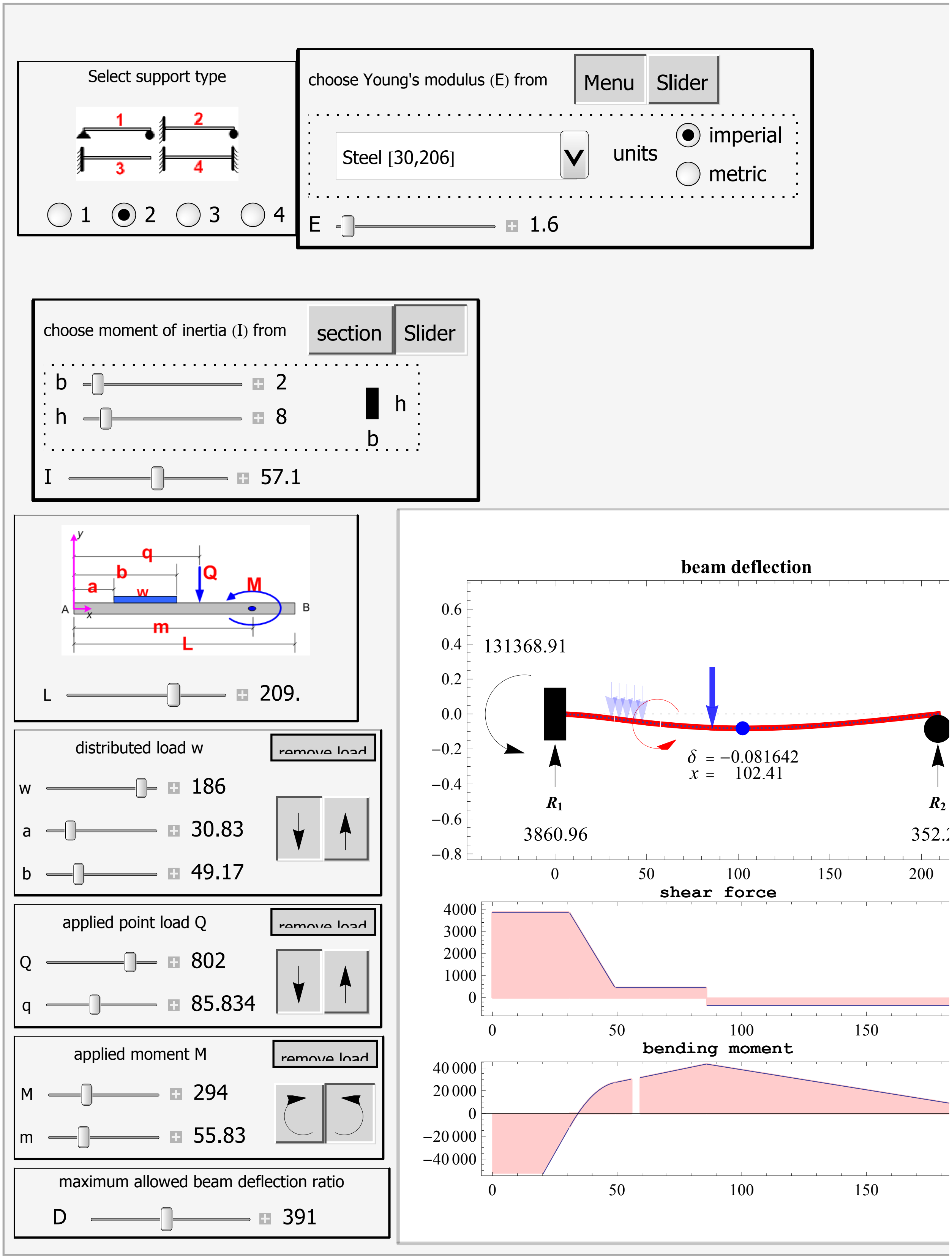


```
   {Thickness[0.0001], Dotted, Line[{{0, 0}, {L, 0}}]},

   (*Left R*)
   {
    Thickness[0.001],
    Opacity[1],
    Black,
    Arrow[{{0, -0.6 maxDeflectionEstimate}, {0, -maxDeflectionEstimate / 4}}],
    Text[Style["R1", Bold, 10], {0, -0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}],
    Text[Style[NumberForm[N[R1], {30, 2}], 12], {0, -0.9 maxDeflectionEstimate}, {0, 1}, {1, 0}]
   },

   (*right R*)
   Which[

    beamSupportType == 3, {Opacity[0]},

    beamSupportType == 1 || beamSupportType == 2 || beamSupportType == 4,
    {
     {
      Thickness[0.001],
      Black,
      Arrow[{{L, -0.6 maxDeflectionEstimate}, {L, -maxDeflectionEstimate / 4}}]},
     Text[Style["R2", Bold, 10], {L, -0.65 maxDeflectionEstimate}, {0, 1}, {1, 0}],
     Text[Style[NumberForm[N[R2], {30, 2}], 12], {L, -0.9 maxDeflectionEstimate}, {0, 1}, {1, 0}]
    }

   ]
  };

title = Style[" beam deflection ", Bold, 12];

pDeflectionOpts = {
   PlotLabel → title,
   Frame → True,
   AxesOrigin → {0, 0},
   ImagePadding → {{40, 5}, {15, 2}},
   ImageMargins → 0,
   ImageSize → 390,
   AspectRatio → .5,
   FrameTicksStyle -> Directive[10],
   Axes → {False, False},
   PlotRange -> {{-0.23 L, L + 0.23 L}, {1.1 * maxDeflectionEstimate, -1.2 * maxDeflectionEstimate}}
  };

(*-- data to allow Tooltip on the deflection curve --*)
data = Table[Tooltip[{i, If[yMax <= maxDeflection, y /. x → i, 0]},
    getBeamCurveAnnotation[y, i, x]], {i, 0, L, L / 100}];

pAnnotated := ListPlot[data, PlotStyle -> PointSize[0]];

pDeflection := Plot[y, {x, 0, L},
   PlotStyle → If[yMax <= maxDeflection, {Red, Thickness[0.01]}, {Dotted, Black, Thickness[0.001]}],
   Evaluate[pDeflectionOpts],
   Epilog → Union[{
      {

       If[yMax <= maxDeflection, Text[""],
        Text[Column[{Style["WARNING: exceeded allowed deflection ratio", Red, 10]
          }], {0.5 L, 0.8 maxDeflectionEstimate}]
       ],
       If[Q ≠ 0,
        If[Q > 0,
         {
```

```
    Opacity[0.8],
    Thickness[0.01],
    Blue,
    If[yMax <= maxDeflection,
     Arrow[{ {q, (y /. x → q) + 0.5 * maxDeflectionEstimate}, {q, (y /. x → q)}}],
     Arrow[{ {q, +0.3 * maxDeflectionEstimate}, {q, 0}}]
    ]
   } ,
   {
    Opacity[0.8],
    Thickness[0.01],
    Blue,
    If[yMax <= maxDeflection, Arrow[{ {q, (y /. x → q) - 0.5 * maxDeflectionEstimate},
        {q, (y /. x → q)}}], Arrow[{ {q, -0.3 * maxDeflectionEstimate}, {q, 0}}]
    ]
   }
  ], Opacity[0]
 ]
},

(* moment *)
If[M ≠ 0,
 {
  If[M > 0,
   {
    Inset[ Graphics[getMomentSymbol[0.125 * L, Red, antiClockWiseStartingAngle,
        antiClockWiseEndingAngle]], {m, If[yMax <= maxDeflection, y /. x → m, 0]}, {0, 0}, 0.125 * L]
   },
   {
    Inset[ Graphics[getMomentSymbol[0.125 * L, Red, clockWiseStartingAngle,
        clockWiseEndingAngle]], {m, If[yMax <= maxDeflection, y /. x → m, 0]}, {0, 0}, 0.125 * L]
   }
  ]
 },
 {Opacity[0]}
],

(*distributed load arrows*)
Which[
 w > 0 && (b - a) > 0,
 {
  Opacity[0.2],
  Table[{ Thickness[0.001], Blue,
    If[yMax <= maxDeflection,
     Arrow[{{a + i, (y /. x → (a + i)) + 0.3 * maxDeflectionEstimate}, {a + i, y /. x → (a + i)}}],
     Arrow[{{a + i, 0.3 * maxDeflectionEstimate}, {a + i, 0}}]]
   }, {i, 0, b - a, L / 50}
  ]
 },
 w < 0 && (b - a) > 0,
 {
  Opacity[0.2],
  Table[{ Thickness[0.001], Blue,
    If[yMax <= maxDeflection,
     Arrow[{{a + i, (y /. x → (a + i)) - 0.3 * maxDeflectionEstimate}, {a + i, y /. x → (a + i)}}],
     Arrow[{{a + i, -0.3 * maxDeflectionEstimate}, {a + i, 0}}]
    ]
   }, {i, 0, b - a, L / 50}
  ]
 },
 True, Opacity[0]
],

(*point where  where deflection is *)
```

```
      {
       PointSize[.025],
       Opacity[1],
       Blue,
       Point[{yMaxLocation, If[yMax <= maxDeflection, yAtMax, 0]}]
      },

      (*deflection amount and location*)
      {
       Opacity[1],

       Text[ Grid[{
           {Style["δ", Italic, 12], " = ",  Style[NumberForm[N[yAtMax], {30, 6}], 11]},
          {Style["x", Italic, 12], " = ", Style[
             NumberForm[N[yMaxLocation], {30, 2}], 11]}}, Spacings → {0, 0}
         ],

         If[yMax <= maxDeflection,
          {yMaxLocation,
           If[yAtMax < 0, yAtMax - 0.15 * maxDeflectionEstimate, yAtMax + 0.1 maxDeflectionEstimate]},
          {yMaxLocation, If[yAtMax < 0, -0.15 * maxDeflectionEstimate, 0.1 maxDeflectionEstimate]}
         ],

         If[yAtMax < 0, {0, 1}, {0, -1}], {1, 0}]
      }
     },
     commonEpilog]
 ];

pMoment = Labeled[
   Plot[moment, {x, 0, L},
    Frame → True,
    ImagePadding → {{44, 1}, {15, 5}},
    ImageMargins → 0,
    ImageSize → 380,
    AspectRatio → .21,
    FrameTicksStyle -> Directive[10],
    AxesOrigin → {0, 0},
    Filling → Axis,
    FillingStyle → Lighter[Red, .8]
   ],
   Style["bending moment", Bold, 11], {{Top, Center}}, Spacings → {Automatic, -.35}
 ];

pShear = Labeled[
   Plot[shear, {x, 0, L},
    Frame → True,
    ImagePadding → {{44, 1}, {15, 5}},
    ImageMargins → 0,
    ImageSize → 380,
    AspectRatio → .21,
    FrameTicksStyle -> Directive[10],
    AxesOrigin → {0, 0},
    Filling → Axis,
    FillingStyle → Lighter[Red, .8],
    Axes → {False, False},
    PlotRange → All
   ],
   Style["shear force", Bold, 11], {{Top, Center}}, Spacings → {Automatic, -.35}
 ];

headings = Grid[{
    {Style["max deflection", 10], Style["location", 10], Style["R1", 10], Style["R2", 10]},
    {yMax, yMaxLocation, R1, R2}
```

```
    }, Frame → All, ItemSize → 7
   ];

  Grid[{
    {Item[Dynamic@Show[{pDeflection, pAnnotated}], ItemSize → Full]},
    {Item[pShear, ItemSize → Full]},
    {Item[pMoment, ItemSize → Full]}
   }, Alignment → Center, Spacings → {0, 0}, Frame → None
  ]

 ]
}
]
```


Select support type
1
2
3
4
choose Young's modulus (E) from
Menu
Slider
Steel [30,206]
units
imperial
metric
E 1.6
choose moment of inertia (I) from
section
Slider
b 2
h 8
h
b
I 57.1
q
b
Q
a
M
w
A
B
m
L
L 209.
distributed load w
w 186
a 30.83
b 49.17
applied point load Q
Q 802
q 85.834
applied moment M
M 294
m 55.83
maximum allowed beam deflection ratio
D 391
beam deflection
131368.91
δ = −0.081642
x = 102.41
R1
R2
3860.96
shear force
bending moment

Caption

This is an illustration of a single span Euler Bernoulli beam under 4 support conditions and with 3 different loading arrangements.

The deflection curve of the beam due to the loads, and the bending moment and the shear force diagrams are generated.

One can modify loadings, support conditions and other parameters such as Young's modulus (E) and the moment of inertia (I) and then observe the effect of these changes on the beam's deflection, moment and shear diagrams.
The beam deflection is normally found by solving the 4th order Euler Bernoulli beam differential equation using the appropriate boundary conditions. In this demonstration, the shear force diagram is used as the starting point and then integrated 3 times to obtain the deflection $y(x)$. The UnitStep[] function was used to facilitate writing the shear force and the bending moment equations.

The reactions at the supports are obtained by solving the equilibrium equations for the determinate beam cases, these are the simply supported at both ends, and the cantilever cases. For the indeterminate cases (fixed at both ends, and fixed at one end and simply supported at the other end), the slope boundary conditions were used to obtain the additional equations needed to solve for all the unknown reactions.

Thumbnail

Snapshots

Details (optional)

The support conditions for the beam are first selected. There are 4 different support conditions available to choose from. The diagram near the top left corner assists one to determine this selection. The beam length is selected using the length slider located below this diagram.
The loadings are selected by using the controls on the left side of the demonstration. A diagram is included which describes the different dimensions used to define the loading positions and the geometry. Using this diagram with the corresponding control variables, one can define different loading configurations.
3 different load types are supported: point load (Q) with units of force, distributed load (w) with units of force per unit length and a couple (M) with units of force times length.
Young's modulus (E) is selected from either the slider or by directly choosing the material from the pop-up menu. The numbers shown for the values of E in this menu selection are the actual numerical values for E for the material being selected. The first number is in imperial units (10^6 psi) and the second number is in metric units (10^9 Pa). These values are obtained from standard materials reference. When using the slider to select E, each value, internally, will be multiplied by 10^6 before being used. For example, to select a material whose E is $1,600,000$ psi, set the slider at 1.6.

The moment of inertia (I) is selected using the slider or by defining the geometry of the beam cross section. Only rectangular cross section is supported. When using the cross section option, the moment of inertia is calculated using the standard formula $I = \frac{1}{12}\,\mathrm{bh}^3$.

The amount of deflection $y(x)$ the beam will experience is limited to a fixed value controlled by the parameter (D) labeled "deflection ratio" located at the lower left part of the control area. This parameter is defined as follows: A deflection ratio of D means that the beam's maximum deflection will not be allowed to exceed $\frac{L}{D}$, where L is the beam length. For example, a value of $D = 180$ means that the maximum deflection allowed will be $\frac{L}{180}$. These ratios can be found in standard structural engineering references such as the International Building Code. By increasing the value of D, one is restricting the allowed maximum deflection of the beam to smaller values. For a typical beam used in housing construction $\frac{L}{180}$ can be used.

Once maximum beam deflection $y(x)$ (in absolute value) has reached the numerical value $\frac{L}{D}$, a warning message will be displayed inside the beam deflection plot indicating that the beam has failed. In this case, one can reduce the beam length, increase I, increase E, or reduce the current loading on the beam.
The plot range for the y-axis for the deflection curve is kept fixed at the maximum allowed deflection value in order to make it more easily to visually compare how the beam deflection change as loads and other parameters are changed.
Regarding the use of units in this illustration: The user must use numerical values for the different parameters either in metric units or imperial units. It is the user's responsibility to make sure that values being selected for the different parameters are consistent with each others. For example, when analyzing a beam of length 10 feet, and the moment of inertia is $57\ \mathrm{inches}^4$, then if the moment of inertia is entered as the number 57, then the length should be entered as 120 and not as 10.
When selecting E from the pull up menu, one chooses either metric or imperial for the units so as to be consistent with the units being used for the rest of the parameters.
In the 3 plots generated, the x-axis has length units and represents the beam length. For the deflection curve $y(x)$, the y-axis is the amount of deflection in units of length. For the bending moment diagram, the y-axis has units of force times length, and for the shear force diagram, the y-axis has units of force.

To find the amount of deflection at any location along the beam, the mouse can be moved to the location over the deflected beam curve causing this information to be be displayed on the screen. This feature was implemented using *Mathematica*'s Tooltip[] function. The small blue ball on the deflection curve indicates the location of the maximum deflection. The deflection value at that point and the distance from the left end of the beam are also displayed next to it.

Control Suggestions (optional)

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

Search Terms (optional)

Euler Bernoulli beam

shear force diagram

bending moment diagram

beam deflection curve

moment of inertia

Young's modulus

Related Links (optional)

Authoring Information

Contributed by: Nasser M. Abbasi

