The Mathematica® Journal

A Visualization Tool for the Vibration of Euler-Bernoulli and Timoshenko Beams

Prithvi Akella , Evan G. Hemingway, Oliver M. O'Reilly

This article details the background material on the classic Euler-Bernoulli and Timoshenko beam theories. A tool designed for visualizing the deformation of beams in static equilibrium and dynamic vibration under an applied loading is also demonstrated. The treatment of these topics is intended to be accessible to upper-division undergraduate engineering students.

■ The Euler-Bernoulli Beam

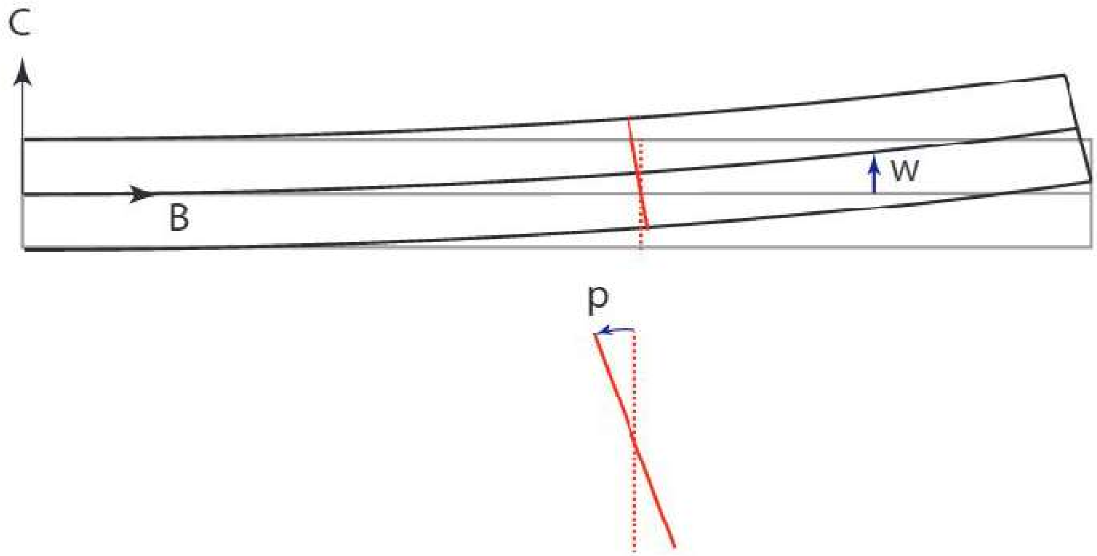

Figure 1. A deformed Euler-Bernoulli beam and its reference configuration in which the centerline is straight and parallel to the x-axis. The cross-sections of the beam remain plane and orthogonal to the centerline after deformation. The upward displacement of a material point on the centerline at a location x in the reference configuration is denoted by w(x,t) and the counterclockwise rotation of a cross-section at a location x in the reference configuration is denoted by the angle θ(x,t).

The Euler-Bernoulli beam model is the simplest linear beam theory and is discussed in most elementary textbooks on vibration (see, e.g. [6]). It works particularly well for slender beams or beams with a high shear modulus, where deformations due to bending are more significant than those due to shear. The centerline of the beam is defined to be the line of geometric centers along the length and it is parametrized by the coordinate x in the reference configuration. We assume the centerline to be inextensible while the cross-sections remain plane and normal to the centerline. To apply the theory, we require knowledge of the following beam properties: ρ, the volume mass density, A, the cross-sectional area, ℓ, the length, E, the modulus of elasticity, and I, the area moment of inertia. Referring to Figure 1, if w and u are the respective vertical and horizontal displacements of a point on the centerline relative to a straight reference configuration, then geometric considerations show that

$$\frac{\partial w}{\partial x} = \sin(\theta), \; 1 + \frac{\partial u}{\partial x} = \cos(\theta), \tag{1}$$

where the coordinate x parametrizes the initially straight centerline. Assuming small deflections, these relations simplify to

$$\frac{\partial w}{\partial x} \simeq \theta, \ \frac{\partial u}{\partial x} \simeq 0. \tag{2}$$

The signed curvature κ of the centerline is related to the angle θ as:

$$\kappa = \frac{\partial \theta}{\partial x}. \tag{3}$$

Thus, for small deflections of the beam, κ can be approximated by $\frac{\partial^2 w}{\partial x^2}$.

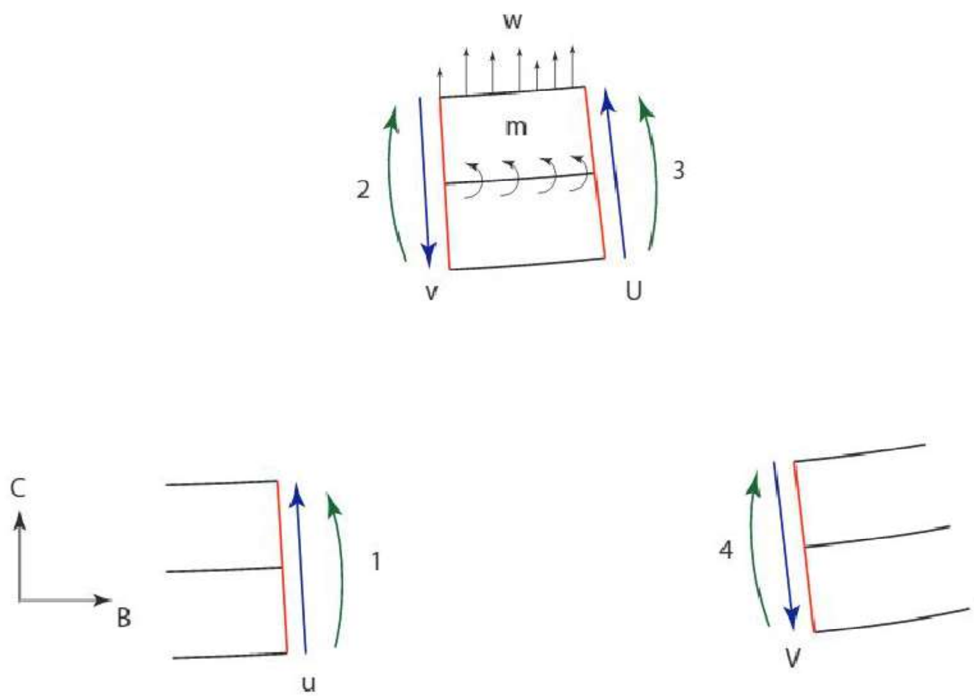

Figure 2. TRY TO MAKE NON-FUZZY A differential element of an Euler-Bernoulli beam at position x = x^* and time t = t^*. The bending moment and shear force at x^*+dx are approximated using Taylor series expansions: M $(x^* + \mathrm{dx}, t^*) = M(x^*, t^*) + \frac{\partial M}{\partial x}(x^*, t^*)\,\mathrm{dx}$ and $\mathrm{V}(x^* + \mathrm{dx}, t^*) = \mathrm{V}(x^*, t^*) + \frac{\partial V}{\partial x}(x^*, t^*)\,\mathrm{dx}$.

Referring to Figure 2, the forces acting on a cross-section of the beam include a shear force V and an axial force, which are the resultants of shear and normal tractions integrated over the cross-sectional area. The axial force ensures that the centerline remains inextensible. The curvature of the beam is due to a bending moment, M, which is the resultant torque in the z-direction due to normal tractions about the centerline. We assume the classic constitutive relation

$$M = \mathrm{EI}\,\frac{\partial^2 w}{\partial x^2}, \tag{4}$$

where we have substituted $\frac{\partial^2 w}{\partial x^2}$ for the centerline curvature and EI is known as the flexural rigidity.

In Euler-Bernoulli beam theory, the inertia of the rotating differential elements and the deformation due to shear is assumed to be negligible, so that cross-sections remain normal to the centerline. Consequently, the balance of angular momentum for an element of the beam yields the relation

$$0 = \frac{\partial M}{\partial x} + m + V. \tag{5}$$

Here, m = m(x,t) is an applied moment per unit length about the centerline of the beam in the -y-direction. The balance of linear momentum in the z-direction yields

$$\rho \mathrm{A} \frac{\partial^2 w}{\partial t^2} = \frac{\partial V}{\partial x} + f. \tag{6}$$

Here, ρA is the linear mass density of the beam, and f = f(x,t) is an applied force per unit length in the z-direction. Together, m and f make up the applied loading. Assuming no applied forces in the horizontal direction, small deformations, and inextensibility, the balance of linear momentum in the horizontal direction will show that the axial force in the beam is constant.

Combining equations (4), (5), and (6) and assuming that ρA and EI are independent of x (i.e. the beam is homogenous and prismatic) yields a constant coefficient fourth order linear PDE which is the equation of motion for the Euler-Bernoulli beam:

$$f_a - \mathrm{EI} \frac{\partial^4 w}{\partial x^4} = \rho \mathrm{A} \frac{\partial^2 w}{\partial t^2}, \tag{7}$$

where f_a = f - $\frac{\partial m}{\partial x}$ is defined as the total applied loading. To solve this partial differential equation, we require the initial conditions

$$w(x, t_0), \quad \frac{\partial w}{\partial t}(x, t_0), \tag{8}$$

at time t = t_0 along with the four independent boundary conditions featuring

$$w(x^*, t), \; \frac{\partial w}{\partial x}(x^*, t), \; \frac{\partial^2 w}{\partial x^2}(x^*, t), \; \frac{\partial^3 w}{\partial x^3}(x^*, t), \tag{9}$$

at locations x = x^*. For instance, $x^*$is typically chosen to be 0 or ℓ. To obtain solutions for the static case, we first set $\frac{\partial^2 w}{\partial t^2}$ = 0 in equation (7). Then, one performs four simple integrations while applying four independent boundary conditions to obtain the solution. The resulting displacement w = w(x) is the solution up to a rigid translation and constant velocity. The following function defines the deformed centerline and generates the visualization in Figure 3.

```
StaticEulercenterline [Emod_ , Iarea_ , TotalLength_ ,
    Lsupport_ , Rsupport_ , t_ , b_] := Module [{},
    Clear [position , w, q, eqn1 , eqn2 , eqn3 , eqn4 , eqn5 ,
     solution ];
    q = t + b;
    eqn1 = w''''[x] == q / (Emod Iarea);
    eqn2 = Switch [Lsupport , Fix, w[0] == 0, S, w[0] == 0,
      Free, w''[0] == 0];
    eqn3 = Switch [Lsupport , Fix, w'[0] == 0, S, w''[0] == 0,
      Free, w'''[0] == 0];
    eqn4 = Switch [Rsupport , Fix, w[TotalLength ] == 0,
      S, w[TotalLength ] == 0, Free, w''[TotalLength ] == 0];
    eqn5 = Switch [Rsupport , Fix, w'[TotalLength ] == 0,
      S, w''[TotalLength ] == 0, Free,
      w'''[TotalLength ] == 0];
    solution = DSolve [{eqn1 , eqn2 , eqn3 , eqn4 , eqn5 },
      w[x], {x, 0, TotalLength }];
    position [x_] := Evaluate [solution [[1]] [[1]] [[2]]];
    Return [position ]];

StaticEulerSections [position_ , TLength_] := Module [{},
    Clear [newposition , slope , shear , moment , radius ,
     theta , crosssection , graphing ];
    newposition [x_] := position [x];
    slope [x_] := newposition '[x];
    shear [x_] := position '''[x];
    moment [x_] := position ''[x];
    radius [x_] := (1 / newposition ''[x]) / 5;
    theta [x_] := ArcTan [newposition '[x]];
    crosssection [var1_ , var2_ , var3_ , constant_ ,
      TotalLength_ , newposition_ , theta_] :=
     If [var1 ≥ var2 && var2 ≥ var3,
      Graphics [
       Line [
         {{var2 + Sin [constant theta [var2]] TotalLength / 10,
           constant newposition [var2] -
            Cos [constant theta [var2]] TotalLength / 10},
          {var2 - Sin [constant theta [var2]] TotalLength / 10,
           constant newposition [var2] +
            Cos [constant theta [var2]] TotalLength / 10}}]],
      {}];
    graphing [var1_ , n_ , place_ , constant_ , TotalLength_ ,
      newposition_ , theta_] :=
```

```
 Module[{sep, sections}, sections = 0 Range[n + 1];
  sep = TotalLength / n;
  For[i = 1, i <= n + 1, i++,
   sections[[i]] = crosssection[var1, (i - 1) sep,
     place, constant, TotalLength, newposition, theta]];
  Return[sections]];
Return[Manipulate[Show[
   Plot[d newposition[x], {x, l, u + 0.01},
    PlotRange → {{0, TLength}, {-c, c}},
    AxesLabel → {"x", "w[x]"}],
   If[line, Graphics[
     Line[{{u + Sin[b theta[u]] TLength / 10,
        b newposition[u] - Cos[b theta[u]] TLength / 10},
       {u - Sin[b theta[u]] TLength / 10,
        b newposition[u] + Cos[b theta[u]] TLength / 10}}]],
    Plot[0, {x, 0, 1}]],
   If[line, graphing[u, number, l, b, TLength,
     newposition, theta], Plot[0, {x, 0, 1}]]],
  {u, 0.01, TLength - 0.01},
  Item["u-sets RHS x-axis graph limit",
   Alignment → Left],
  {l, 0.01, TLength - 0.01},
  Item["l sets the LHS x-axis graph limit",
   Alignment → Left],
  {b, 10 000},
  Item[
   "b magnifies the graph and all associated
     material (cross-sections, angles, etc)",
   Alignment → Left],
  {c, TLength},
  Item["c sets the pos/neg values on the y-axis",
   Alignment → Left],
  {number, 5},
  Item[
   "number sets the number of cross-sections the
     program will animate", Alignment → Left],
  {line, {True, False}},
  Item["Line switches the cross-sections on/off",
   Alignment → Left]]]];
```

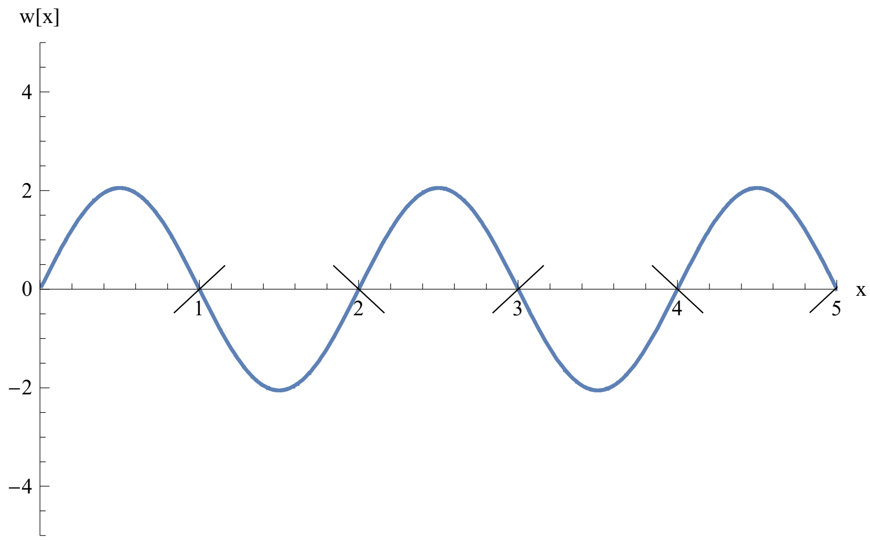

Figure 3. Example of a static deformation of an Euler-Bernoulli beam under a sinusoidal applied traction load. Given pin joints as support structures, the four independent boundary conditions require that V = 0 and M = 0 at both ends of the beam. Here, we used the following data:

Emod = 100 Pa Iarea: 5 m^4 TLength: 5 m t = Sin[π x] b = 0

Free Vibrations

The more interesting case concerns the dynamic vibrations of the beam. To proceed, we first set f = 0 and m = 0 and solve the free vibration problem. The governing equation (7) is written in standard form as

$$c^2 \frac{\partial^4 w}{\partial x^4} + \frac{\partial^2 w}{\partial t^2} = 0, \tag{10}$$

where

$$c = \frac{\text{EI}}{\rho \text{A}}. \tag{11}$$

Next, we assume that the mode shape profile is independent of time so that the desired solution is separable:

$$w(x, t) = W(x)\, Q(t). \tag{12}$$

Substituting the assumed form of w(x,t) into the partial differential equation (10) produces the equation

$$\frac{Q}{\frac{d^2 Q}{\mathrm{dt}^2}} = \frac{-W}{c^2 \frac{d^4 W}{\mathrm{dx}^4}} = \frac{-1}{\omega^2}, \tag{13}$$

where ω is an undetermined constant. A solution for Q(t) is easily obtained as

$$Q(t) = Y \sin(\omega \mathrm{t}) + Z \cos(\omega \mathrm{t}), \tag{14}$$

where Y and Z are real constants to be determined from initial conditions. The mode shape W(x) is obtained as the eigenfunction with eigenfrequency ω in the differential eigenvalue problem:

$$c^2 \frac{d^4 W}{\mathrm{dx}^4} - \omega^2 W = 0. \tag{15}$$

By assuming a solution of the form W = Be$^{\mathrm{rx}}$, we obtain the dispersion relation

$$c^2 r^4 - \omega^2 = 0, \tag{16}$$

which yields:

$$\begin{aligned} r_1 &= \sqrt{\frac{\omega}{c}}\,,\; r_2 = -r_1, \\ r_3 &= i\sqrt{\frac{\omega}{c}}\,,\; r_4 = -r_3. \end{aligned} \tag{17}$$

The quantity $\sqrt{\frac{\omega}{c}}$ is known as the wavenumber. The general solution may be expressed as a linear combination of exponential functions:

$$W(x) = \sum_{i=1}^{4} B_i e^{r_i x}. \tag{18}$$

Currently, B_i and r_i are complex. Since we require ω to be real and Q has been shown to be real, we rewrite W(x) in the form

$$W(x) = A_1 \cos(\beta \mathrm{x}) + A_2 \sin(\beta \mathrm{x}) + A_3 \cosh(\beta \mathrm{x}) + A_4 \sinh(\beta \mathrm{x}), \tag{19}$$

where each A_i is a real constant and we have defined $\beta = \sqrt{\frac{\omega}{c}}$ for notational convenience.

Three of the A_i are found by imposing four boundary conditions on W(x). Simple boundary conditions include: no allowable displacement (w = 0) , no allowable cross-section rotation $\left(\frac{\partial w}{\partial x} = 0\right)$, no applied couple (EI $\frac{\partial^2 w}{\partial x^2}$ = 0), and no applied shear force (EI $\frac{\partial^3 w}{\partial x^3}$ = 0). For example, the boundary conditions on the displacement for a fixed (clamped or cantilevered)-free beam are

$$\begin{aligned} &w(x = 0, t) = 0,\; \frac{\partial w}{\partial x}(x = 0, t) = 0, \\ &\frac{\partial^2 w}{\partial x^2}(x = \ell, t) = 0,\; \frac{\partial^3 w}{\partial x^3}(x = \ell, t) = 0, \end{aligned} \tag{20}$$

which imply the following boundary conditions on the mode shape:

$$W(x=0) \ = \ 0, \ \frac{d\,W}{d\,x}(x=0)=0, \ \frac{d^2\,W}{d\,x^2}(x=\ell)=0, \ \frac{d^3\,W}{d\,x^3}(x=\ell)=0. \tag{21}$$

It is convenient to formulate a matrix equation using the four boundary conditions:

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{22}$$

For a non-trivial mode shape the determinant of the coefficient matrix M must be zero. This yields the frequency equation for ω:

$$\det(M) = 0. \tag{23}$$

Any 4-tuple $[A_1, A_2, A_3, A_4]^T$ in the 1-dimensional null space of M provides three of the four constants A_i. Continuing the aforementioned example, we find for the fixed-free beam

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -c(\beta\ell) & -s(\beta\ell) & \mathrm{ch}(\beta\ell) & \mathrm{sh}(\beta\ell) \\ s(\beta\ell) & -c(\beta\ell) & \mathrm{sh}(\beta\ell) & \mathrm{ch}(\beta\ell) \end{pmatrix}, \tag{24}$$

where we have used the abbreviated notations: c(-) = cos(-), s(-) = sin(-), ch=cosh(-), and sh(-) = sinh(-). With the aid of some trigonometric identities, the frequency equation is found to be

$$\cos(\beta\ell)\cosh(\beta\ell) \ = \ -1, \tag{25}$$

which is a transcendental equation for the natural frequency ω. Equation (25) yields infinitely many natural frequencies where $\omega_i < \omega_{i+1}$ and $\omega_1$is known as the fundamental frequency. It follows that there are infinitely many corresponding mode shapes W_i(x). In our example, a 4-tuple up to a scalar A_i in the null space of M corresponding to ω_i is found to be:

$$A_i\left[\frac{s(\beta_i\,\ell)+\mathrm{sh}(\beta_i\,\ell)}{-c(\beta_i\,\ell)-\mathrm{ch}(\beta_i\,\ell)}, 1, \frac{s(\beta_i\,\ell)+\mathrm{sh}(\beta_i\,\ell)}{c(\beta_i\,\ell)\ +\ \mathrm{ch}(\beta_i\,\ell)}, -1\right]^T. \tag{26}$$

Since A_i is arbitrary, we may set A_i=1 for every natural frequency. Then, the general solution for the fixed-free beam has the form

$$w(x,t) \ = \ \sum_{i=1}^{\infty} W_i(x)\,Q_i(t), \tag{27}$$

where

$$W_i(x) = \frac{s(\beta_i\,\ell) + \mathrm{sh}(\beta_i\,\ell)}{-c(\beta_i\,\ell) - \mathrm{ch}(\beta_i\,\ell)}\,c(\beta_i\,x) + s(\beta_i\,x) + \frac{s(\beta_i\,\ell) + \mathrm{sh}(\beta_i\,\ell)}{c(\beta_i\,\ell) + \mathrm{ch}(\beta_i\,\ell)}\,\mathrm{ch}(\beta_i\,x) - \mathrm{sh}(\beta_i\,x)\,, \tag{28}$$
$$Q_i(t) = Y_i\,s(\omega_i\,t) + Z_i\,c(\omega_i\,t).$$

It remains to determine the coefficients Y_i and Z_i from the initial conditions.

Non-Dimensionalization Procedure

To non-dimensionalize the equation of motion for the free vibration, we begin by introducing the following non-dimensional variables:

$$\hat{w} = \frac{w}{\ell},\ \hat{x} = \frac{x}{\ell},\ \hat{t} = \frac{\mathrm{ct}}{\ell^2}, \tag{29}$$

where $\mathrm{c} = \sqrt{\frac{\mathrm{EI}}{\rho\mathrm{A}}}$ is a constant.

The dimensionless form of the partial differential equation (10) is

$$\hat{w}'''' + \ddot{\hat{w}} = 0, \tag{30}$$

where o' denotes $\frac{\partial o}{\partial \hat{x}}$ and $\dot{o}$ denotes $\frac{\partial o}{\partial \hat{t}}$.

Furthermore, the governing equation may also be non-dimensionalized as

$$\hat{f}_a - \hat{w}'''' = \ddot{\hat{w}}. \tag{31}$$

Here,

$$\hat{f}_a = \frac{f_a\,\ell^3}{\mathrm{EI}}. \tag{32}$$

With the aforementioned definitions for the superposed dot and prime.

$$\hat{W} = \frac{W}{\ell},\ \hat{\beta} = \beta\ell,\ \hat{\omega} = \frac{\omega\ell^2}{c},\ \hat{M} = \frac{\mathrm{M}\ell}{\mathrm{EI}} = \hat{w}'',\ \hat{V} = -\hat{M}'. \tag{33}$$

The highlights of the non-dimensional treatment present themselves as follows; first for the fixed-free boundary conditions:

$$\hat{w}\left(\hat{x} = 0, \hat{t}\right) = 0,\ \hat{w}'\left(\hat{x} = 0, \hat{t}\right) = 0,\ \hat{w}''\left(\hat{x} = 1, \hat{t}\right) = 0,\ \hat{w}'''\left(\hat{x} = 1, \hat{t}\right) = 0. \tag{34}$$

Secondly, boundary conditions on the now non-dimensional mode shapes are:

$$\hat{W}(\hat{x} = 0) = 0,\ \hat{W}'(\hat{x} = 0) = 0,\ \hat{W}''(\hat{x} = 1) = 0,\ \hat{W}'''(\hat{x} = 1) = 0. \tag{35}$$

Given that we non-dimensionalized W as $\hat{W} = \frac{W}{\ell}$, we also require that A_i are replaced with $\hat{A}_i = \frac{A_i}{\ell}$. Lastly, the only change from the frequency equation for the fixed-free case discussed earlier is:

$$\cos(\hat{\beta})\cosh(\hat{\beta}) = -1. \tag{36}$$

Implementation

In the code, we solve for the first ten $\hat{\omega}_i$ that satisfy the frequency equation (36). Mode shapes for five simple cases may be found in Rao [6, Chapter 8]. The classical solutions are used in the code and are listed here for completeness: for a free-free beam,

$$\hat{W}_i(\hat{x}) = F_{1\,i}(\hat{x}) - \frac{F_{2\,i}(1)}{F_{4\,i}(1)} F_{3\,i}(\hat{x}), \tag{37}$$

a fixed-fixed beam,

$$\hat{W}_i(\hat{x}) = F_{2\,i}(\hat{x}) - \frac{F_{2\,i}(1)}{F_{4\,i}(1)} F_{4\,i}(\hat{x}), \tag{38}$$

a fixed-free beam,

$$\hat{W}_i(\hat{x}) = F_{2\,i}(\hat{x}) - \frac{F_{4\,i}(1)}{F_{1\,i}(1)} F_{4\,i}(\hat{x}), \tag{39}$$

a fixed-pinned beam,

$$\hat{W}_i(\hat{x}) = F_{2\,i}(\hat{x}) - \frac{F_{2\,i}(1)}{F_{4\,i}(1)} F_{4\,i}(\hat{x}), \tag{40}$$

and a pinned-pinned beam,

$$\hat{W}_i(x) = \sin\left(\hat{\beta}_i\,\hat{x}\right). \tag{41}$$

The functions $F_{1\,i\,.2\,i\,.3\,i\,.4\,i}$ are

$$\begin{aligned} F_{1\,i}(\hat{x}) &= \cosh\left(\hat{\beta}_i\,\hat{x}\right) + \cos\left(\hat{\beta}_i\,\hat{x}\right), \\ F_{2\,i}(\hat{x}) &= \cosh\left(\hat{\beta}_i\,\hat{x}\right) - \cos\left(\hat{\beta}_i\,\hat{x}\right), \\ F_{3\,i}(\hat{x}) &= \sinh\left(\hat{\beta}_i\,\hat{x}\right) + \sin\left(\hat{\beta}_i\,\hat{x}\right), \\ F_{4\,i}(\hat{x}) &= \sinh\left(\hat{\beta}_i\,\hat{x}\right) - \sin\left(\hat{\beta}_i\,\hat{x}\right). \end{aligned} \tag{42}$$

The general solution for any of these simple cases is found by substituting the appropriate $\hat{W}_i(\hat{x})$ into the non-dimensional version of the expansion (27), where $Q_i(\hat{x})$ always has the form (28) for every case. As in the static case,functions which identify and display the deformed centerline as in Figures 4 and 5 may be seen below.

```
VibEulerCenterline [type_, mode_] := Module[{},
   Clear[modetype, val, P, Q, R, S, FreeFreeeq,
    FixedFixedeq, FixedFreeeq, FixedPinnedeq,
    PinnedPinnedeq, position, omega, FreeFreeroots,
    FixedFixedroots, FixedFreeroots, FixedPinnedroots,
    PinnedPinnedroots];
   FreeFreeroots = {0, 4.73004, 7.85320, 10.9956,
     14.1372, 17.2788, 20.4204, 23.5619, 26.7035,
     29.8451};
   FixedFixedroots = {4.73004, 7.85320, 10.9956,
     14.1372, 17.2788, 20.4204, 23.5619, 26.7035,
     29.8451, 32.9867};
   FixedFreeroots = {1.8751, 4.6409, 7.85476, 10.9955,
     14.1372, 17.2788, 20.4204, 23.5619, 26.7035, 29.8451};
   FixedPinnedroots = {3.92660, 7.06858, 10.212,
     13.3518, 16.4934, 19.635, 22.7765, 25.9181,
     29.0597, 32.2013};
   PinnedPinnedroots = {1 * Pi, 2 * Pi, 3 * Pi, 4 * Pi,
     5 * Pi, 6 * Pi, 7 * Pi, 8 * Pi, 9 * Pi, 10 * Pi};
   modetype = Switch[type, FreeFree, FreeFreeroots,
     FixedFixed, FixedFixedroots, FixedFree,
     FixedFreeroots, FixedPinned, FixedPinnedroots,
     PinnedPinned, PinnedPinnedroots];
   val = modetype[[mode]];
   P[x_] := Cosh[val x] + Cos[val x];
   Q[x_] := Cosh[val x] - Cos[val x];
   R[x_] := Sinh[val x] + Sin[val x];
   S[x_] := Sinh[val x] - Sin[val x];
   FreeFreeeq[x_] := P[x] - (Q[1] / S[1]) R[x];
   FixedFixedeq[x_] := Q[x] - (Q[1] / S[1]) S[x];
   FixedFreeeq[x_] := Q[x] - (S[1] / P[1]) S[x];
   FixedPinnedeq[x_] := Q[x] - (Q[1] / S[1]) S[x];
   PinnedPinnedeq[x_] := Sin[val x];
   position[x_] := Switch[type, FreeFree, FreeFreeeq[x],
     FixedFixed, FixedFixedeq[x], FixedFree,
     FixedFreeeq[x], FixedPinned, FixedPinnedeq[x],
     PinnedPinned, PinnedPinnedeq[x]];
   Return[position]];

VibEulerDeformation [position_] := Module[{},
   Clear[theta, radius, slope, moment, shear, TotalLength];
   slope[x_] := position'[x];
   moment[x_] := position''[x];
```

```
shear[x_] := position'''[x];
radius[x_] := 1 / moment[x];
theta[x_] := ArcTan[slope[x]];
TotalLength = 1;
crosssection[var1_, var2_, var3_, constant_] :=
 If[var1 ≥ var2 && var2 ≥ var3,
  Graphics[
   Line[
    {{var2 + Sin[constant theta[var2]] TotalLength / 10,
      constant position[var2] -
       Cos[constant theta[var2]] TotalLength / 10},
     {var2 - Sin[constant theta[var2]] TotalLength / 10,
      constant position[var2] +
       Cos[constant theta[var2]] TotalLength / 10}}]],
  {}];
graphing[var1_, n_, place_, constant_] :=
 Module[{sep, sections}, sections = 0 Range[n + 1];
  sep = TotalLength / n;
  For[i = 1, i <= n + 1, i++,
   sections[[i]] = crosssection[var1, (i - 1) sep,
     place, constant]];
  Return[sections]];
Manipulate[Show[
  Plot[b position[x], {x, l, u + 0.01},
   PlotRange → {{-TotalLength / 5, 2 TotalLength},
     {-2 TotalLength, 2 TotalLength}},
   AxesLabel → {"x", "w[x]"}],
  If[circ,
   Graphics[Circle[{u + Sin[theta[b u]] b radius[u],
      b position[u] - Cos[theta[b u]] b radius[u]},
     Abs[b radius[u]]]], Plot[0, {x, 0, 0.01}]],
  If[line,
   Graphics[
    Line[{{u + Sin[theta[b u]] 0.1 ,
       b position[u] - Cos[theta[b u]] 0.1},
      {u - Sin[theta[b u]] 0.1,
       b position[u] + Cos[theta[b u]] 0.1}}]],
   Plot[0, {x, 0, 0.01}]],
  If[line, graphing[u, number, l, b],
   Plot[0, {x, 0, 0.01}]]],
 {u, 0.01, TotalLength - 0.01}, {circ, {True, False}},
 {line, {True, False}}, {l, 0.01, TotalLength - 0.01},
 {b, 1, 0}, {number, 5}]];
```

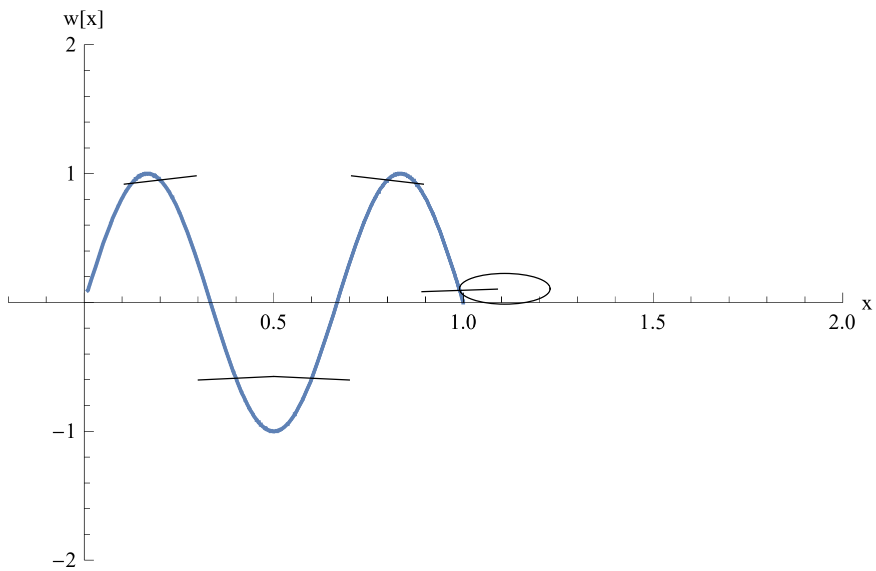

Figure 4. Example of the non-dimensionalized third mode of vibration for a Bernoulli-Euler beam with Simple end supports. As we've non-dimensionalized, the simple end conditions enforce $\hat{w}$=0 and $\hat{M}$=0 at the ends. The cross-sections and osculating circles can be seen albeit they are slightly exaggerated for visualization purposes.

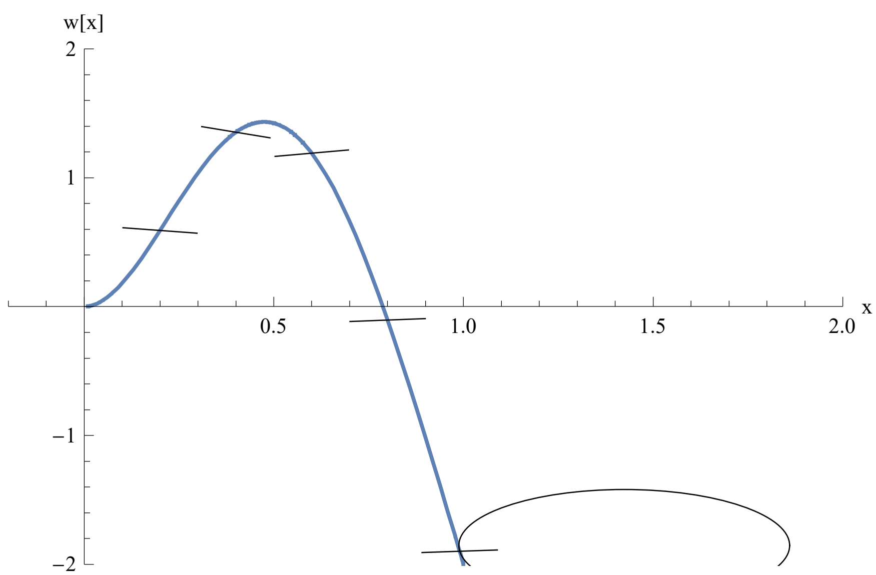

Figure 5. Example of the non-dimensionalized second mode of vibration for a Bernoulli-Euler beam with fixed and free end supports. Non-dimensionalizing enforces $\hat{w} = \hat{w}' = 0$ for the fixed end, and $\hat{V} = \hat{M} = 0$ for the free end. The cross-sections and osculating circle can be seen albeit they are slightly exaggerated for visualization purposes.

■ Timoshenko Beam Theory

In Timoshenko beam theory, the rotary inertia of a differential element of the beam is considered non-negligible and the cross-sections are allowed to rotate relative to the centerline due to shear deformation (cf. Figure 6). The Euler-Bernoulli beam theory may be obtained as a limiting case of Timoshenko beam theory when a particular dimensionless quantity in the forthcoming section is very small.

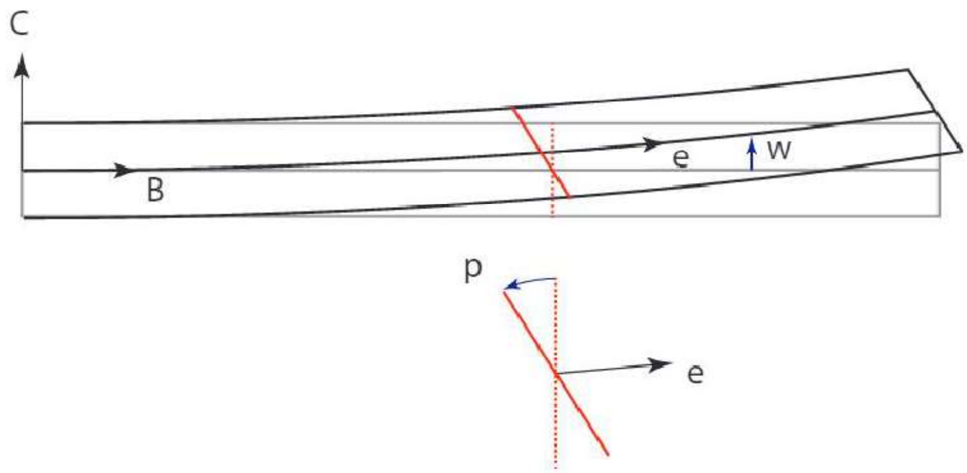

Figure 6. A deformed Timoshenko beam whose centerline is initially straight and parallel to the x-axis. The cross-sections of this beam remain plane but are not necessarily orthogonal to the centerline after deformation. The upward displacement of the beam at a location x along the center line is defined by w(x,t) and the counterclockwise positive rotation of the cross-section at a location x on the centerline is defined by the angle α(x,t).

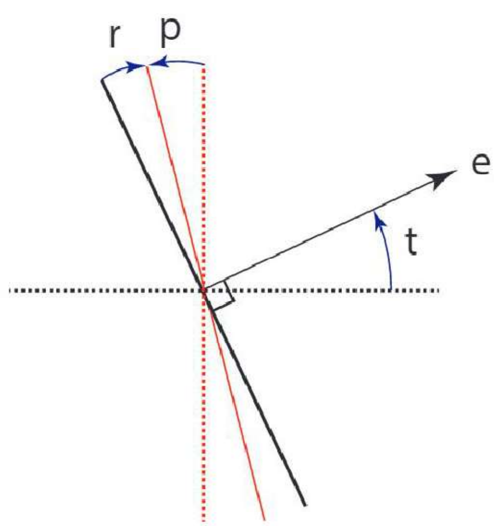

Figure 7. Positive values of the angles θ, α and β that are used to define the respective rotation of the unit tangent vector e_t to the centerline, the rotation of the cross-section, and the shear deformation of the cross-section.

To begin characterizing the shear deformation, we first define θ as the counterclockwise positive angle that the deformed tangent to the centerline, e_t, makes with the x-direction:

$$\frac{\partial w}{\partial x} = \sin(\theta)\,,\;\; 1 + \frac{\partial u}{\partial x} \; = \; \cos(\theta). \tag{43}$$

Referring to Figures 6 and 7, we define the angle that the cross-section makes with its referential counterpart by α. Finally, the shear deformation angle β is measured clockwise positive from a plane perpendicular to the cross-section. For small deflections of the beam, β may be expressed as

$$\beta \; = \; \theta \; - \; \alpha \; \approx \; \frac{\partial w}{\partial x} - \; \alpha. \tag{44}$$

For this beam theory, the complete solutions entails finding w = w(x,t) and $\alpha = \alpha$(x,t). By way of contrast, for the Euler-Bernoulli beam the complete solution entailed finding just w = w(x,t). The development of the governing equations parallels those for the Euler-Bernoulli beam. The internal forces in the beam are a shear force V, an axial force, and a bending moment M. We assume the beam is acted upon by an applied distributed force f and an applied distributed moment m. The sign conventions for these scalar kinetic quantities are as in the previous section. The shear force and bending moment are prescribed by the following constitutive relations:

$$M \; = \; \mathrm{EI}\, \frac{\partial \alpha}{\partial x}, \;\; V \; = \; \mathrm{kGA}\beta \; = \; \mathrm{kGA}\left(\frac{\partial w}{\partial x} - \alpha\right). \tag{45}$$

For the shear force, we have introduced the shear modulus G and the shear correction factor (or shear coefficient) k. The factor k was originally introduced by Timoshenko [?] ADD REFERENCE in 1921. It has been shown that k depends on the geometry of the cross-section of the beam and obtaining accurate values of k for beams of various cross-sections has received considerable attention in the literature since the 1920s (cf. [?,?,?]) ADD REFERENCE. For example, a value of k = 5/6 for beams with rectangular cross-sections is often used.

The differential element of the beam is identical to that shown for the Euler-Bernoulli beam in Figure 2. Balances of linear momentum in the z-direction and angular momentum in the y-direction yield

$$\rho\mathrm{A}\, \frac{\partial^2 w}{\partial t^2} = \frac{\partial V}{\partial x} + f, \;\; \rho\mathrm{I}\, \frac{\partial^2 \alpha}{\partial t^2} = \frac{\partial M}{\partial x} + m \; + \; V. \tag{46}$$

Substituting the constitutive relations for V and M and assuming prismatic and homogenous beams, we find a set of coupled partial differential equations for w and α:

$$\mathrm{kGA}\left(\frac{\partial^2 w}{\partial x^2} - \frac{\partial \alpha}{\partial x}\right) + f = \rho \mathrm{A} \frac{\partial^2 w}{\partial t^2},$$
$$\mathrm{EI} \frac{\partial^2 \alpha}{\partial x^2} + \mathrm{kGA}\left(\frac{\partial w}{\partial x} - \alpha\right) + m = \rho \mathrm{I} \frac{\partial^2 \alpha}{\partial t^2}. \tag{47}$$

To make the problem more tractable, we will only approach the problem from a non-dimensional perspective so that the number of parameters in the system is reduced. We begin by non-dimensionalizing the variables w, x, and t:

$$\hat{w} = \frac{w}{\ell},\ \hat{x} = \frac{x}{\ell},\ \hat{t} = t\sqrt{\frac{\mathrm{kG}}{\rho \ell^2}}. \tag{48}$$

Non-dimensional loading is conveniently defined as:

$$\hat{f} = \frac{\mathrm{f}\ell}{\mathrm{kGA}},\ \hat{m} = \frac{m}{\mathrm{kGA}}. \tag{49}$$

We also introduce the following dimensionless parameters:

$$\chi = \frac{\mathrm{EI}}{\ell^2\,\mathrm{kGA}},\ \frac{1}{s} = \frac{I}{\mathrm{A}\ell^2}. \tag{50}$$

Noticing that $\sqrt{\frac{I}{A}}$ is the radius of gyration of the cross-section, we see that s may be interpreted as the slenderness ratio for the beam. The quantity χ gives a measure of how important the effects accounted for in Timoshenko beam theory are. When χ is very small, the Euler-Bernoulli beam theory is reattained. Using these dimensionless quantities, the equations of motion (47) take on a simpler form (we have omitted the hats for notational convenience):

$$w'' - \alpha' + f = \ddot{w},$$
$$s(\chi\alpha'' + w' - \alpha + m) = \ddot{\alpha}. \tag{51}$$

The decoupled equations have the form

$$\ddot{\ddot{w}} + \mathrm{s}\chi \mathrm{w}'''' - (1 + \mathrm{s}\chi)\,\ddot{w}'' + \mathrm{s}\ddot{\mathrm{w}} = \ddot{f} - \mathrm{s}\chi f'' + \mathrm{sf} - s\,m',$$
$$\ddot{\ddot{\alpha}} + \mathrm{s}\chi\alpha'''' - (1 + \mathrm{s}\chi)\,\ddot{\alpha}'' + s\,\ddot{\alpha} = s\left(\ddot{m} - m'' + f'\right). \tag{52}$$

The solutions obtained for w = w(x,t) and $\alpha = \alpha$(x,t) using the second-order coupled partial differential equations (51) are the same as those obtained from the uncoupled fourth-order partial differential equations (52). To obtain a unique solution, a set of eight boundary conditions and four initial conditions are required. In concordance with the previous sections, functions for the obtaining the static deformed centerline and visual displays are seen below. An example of the visualization is provided in figure 8.

```
StaticTimoshenkoCenterline [Emod_ , Gmod_ , area_ , k_ ,
    Iarea_ , TotalLength_ , Lsupport_ , Rsupport_ , t_ , b_] :=
   Module [{},
    Clear [w, nonscalew , phi , nonscalephi , eqn1 , eqn2 ,
     eqn3 , eqn4 , eqn5 , eqn6 , solution , q, maxpos ];
    q[x_] := Evaluate [t + b];
    eqn1 = phi ' [x] == w ' ' [x] + q[x] / (k area Gmod);
    eqn2 = phi ' ' [x] == k area Gmod / (Emod Iarea)
        (phi [x] - w ' [x]);
    eqn3 = Switch [Lsupport , Fix , phi [0] == 0, S, phi ' [0] == 0,
      Free , phi [0] == w ' [0]];
    eqn4 = Switch [Rsupport , Fix , phi [TotalLength ] == 0,
      S, phi ' [TotalLength ] == 0, Free ,
      phi [TotalLength ] == w ' [TotalLength ]];
    eqn5 = Switch [Lsupport , Fix , w[0] == 0, S, w[0] == 0,
      Free , phi ' [0] == 0];
    eqn6 = Switch [Rsupport , Fix , w[TotalLength ] == 0, S,
      w[TotalLength ] == 0, Free , phi ' [TotalLength ] == 0];
    solution = DSolve [{eqn1 , eqn2 , eqn3 , eqn4 , eqn5 , eqn6},
      {phi [x] , w[x]}, {x, 0, TotalLength }];
    nonscalephi [x_] := Evaluate [solution [[1]] [[1]] [[2]]];
    nonscalew [x_] := Evaluate [solution [[1]] [[2]] [[2]]];
    maxpos = MaxValue [{nonscalew [x] , 0 < x < TotalLength } , x];
    w[x_] := nonscalew [x] TotalLength / maxpos ;
    phi [x_] := nonscalephi [x] TotalLength / maxpos ;
    Return [{w[x] , phi [x]}]];

StaticTimoshenkoDeformation [w_, phi_ , TotalLength_ ,
    Emod_ , Iarea_ , k_ , area_ , Gmod_] := Module [{},
    Clear [moment , shear , slope , radius , crosssection ,
     graphcircle , graphsections ];
    moment [x_] := Emod Iarea phi ' [x];
    shear [x_] := k area Gmod (w ' [x] - phi [x]);
    slope [x_] := w ' [x];
    radius [x_] := 1 / moment [x];
    crosssection [rbx_ , cbx_ , information_ , lbx_] :=
     If [rbx ≤ cbx && rbx ≥ lbx ,
      Return [line [information [[1]] , information [[2]] ,
        information [[3]] , information [[4]]]] , {}];
    graphcircle [n_ , cpos_ , constant_ , lbx_ , constant2_] :=
     Module [{sep , sections , i},
      sections = 0 Range [n + 1];
      sep = TotalLength / n;
```

```
  For[i = 1, i ≤ n + 1, i++,
   Module[{xpos, information},
    xpos = (i - 1) sep + 0.01;
    information = {{xpos, constant w[xpos]},
      constant constant2 radius[xpos],
      constant slope[xpos]};
    sections[[i]] = circlesection[xpos, cpos[[1]],
      information, lbx]]];
  Return[sections]];
graphsections[n_, cbx_, lbx_, constant_] :=
 Module[{sep, sections, i},
  sections = 0 Range[n + 1];
  sep = TotalLength / n;
  For[i = 1, i <= n + 1, i++,
   Module[{information, xpos},
    xpos = (i - 1) sep;
    information = {Tan[Pi / 2 + constant phi[xpos] + 0.01],
      xpos, constant w[xpos], TotalLength};
    sections[[i]] = crosssection[xpos, cbx,
      information, lbx]]];
  Return[sections]];
Manipulate[Show[
  Plot[b w[x], {x, l, u},
   PlotRange → {{-TotalLength / 5, TotalLength 6 / 5},
     {-TotalLength 6 / 5, TotalLength 6 / 5}},
   AxesLabel → {"x", "w[x]"}],
  If[section, line[Tan[Pi / 2 + b phi[u]], u, b w[u],
    TotalLength], Plot[0, {x, 0, 0.01}]],
  If[osc, circle[{u, b w[u]}, b magnify2 radius[u],
    b slope[u]], Plot[0, {x, 0, 0.01}]],
  If[section, graphsections[number, u, l, b], {}],
  If[osc, graphcircle[number, {u, b w[u]}, b, l,
    magnify2], {}]],
 {u, TotalLength - 0.01, 0.01}, {section, {True, False}},
 {osc, {True, False}}, {l, 0, u - 0.01},
 {magnify2, 1, 1000}, {b, 1, 0.01}, {number, 5}]];
```

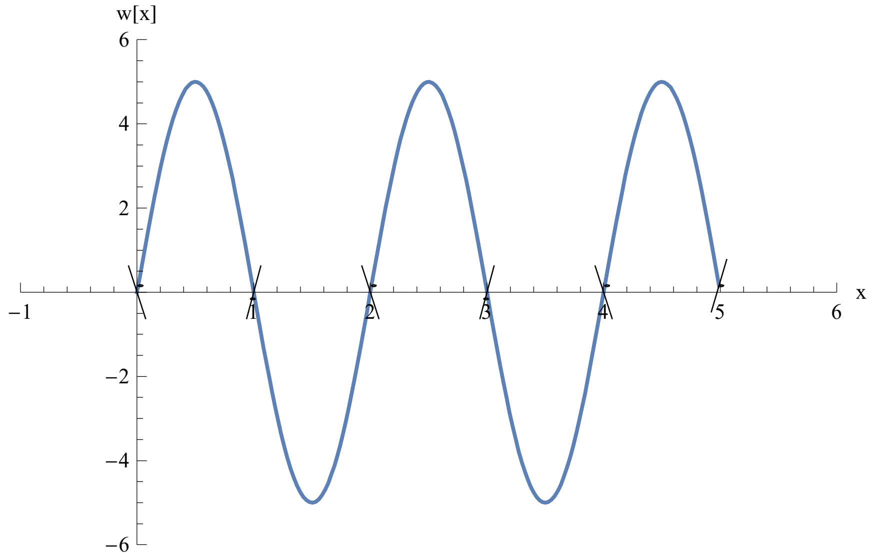

Figure 8 Example of a statically deformed pin-pinned Timoshenko beam under a sinusoidally varying traction force. Beam properties are as follows:
Emod: 1000 Pa Iarea: 5 m^4 TLength: 5 m Gmod: 100 Pa area: 4 m^2 k = 5/6
LSupport: Simple RSupport: Simple t: Sin[π x] b=0
The simple end conditions enforce w = M = 0 at both ends.

Free Vibrations

To solve for the free vibrations of the beam, we set f = 0 and m = 0 for all values of space and time. Assuming synchronous behavior of w and α, we may apply separation of variables:

$$w(x, t) = W(x)\, Q(t)\,, \;\; \alpha(x, t) = \Phi(x)\, Q(t). \tag{53}$$

Substituting assumption (53) into the equations of motion (51), we obtain

$$\begin{aligned} &W''\, Q - \Phi'\, Q = W\, \ddot{Q}, \\ &s(\chi \Phi''\, Q + W'\, Q - \Phi \mathrm{Q}) \; = \; \Phi\, \ddot{Q}\,. \end{aligned} \tag{54}$$

Let $\hat{\omega}^2$ be the (non-dimensional) constant for which $\frac{\ddot{Q}}{Q} = -\hat{\omega}^2$, where

$$\hat{\omega} = \omega \sqrt{\frac{\rho \ell^2}{\mathrm{kG}}} \tag{55}$$

and ω is a natural frequency for the system. From this point forth, we will drop the hat for convenience, though the following equations are indeed non-dimensional. Noting that, we also obtain from equation (54) a set of ordinary differential equations for the mode shapes as

$$\begin{pmatrix} 1 & 0 \\ 0 & s\chi \end{pmatrix}\begin{pmatrix} W'' \\ \Phi'' \end{pmatrix} + \begin{pmatrix} 0 & -1 \\ s & 0 \end{pmatrix}\begin{pmatrix} W' \\ \Phi' \end{pmatrix} + \begin{pmatrix} \omega^2 & 0 \\ 0 & \omega^2 - s \end{pmatrix}\begin{pmatrix} W \\ \Phi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{56}$$

The solution for Q(t) can be expressed as,

$$Q(t) \;=\; B\cos(\omega \mathrm{t} + \phi) \tag{57}$$

where B and φ are constants. If we let the beam be initially at rest, then $\varphi = 0$. We shall assume this choice of initial time in the sequel.

The coupled ordinary differential equations (56) may be decoupled as

$$s\,\chi\begin{pmatrix} W'''' \\ \Phi'''' \end{pmatrix} + (s\,\chi + 1)\,\omega^2\begin{pmatrix} W'' \\ \Phi'' \end{pmatrix} + \left(\omega^2 - s\right)\omega^2\begin{pmatrix} W \\ \Phi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{58}$$

We assume a solution of the form

$$\begin{pmatrix} W \\ \Phi \end{pmatrix} = \begin{pmatrix} A \\ B \end{pmatrix} e^{\mathrm{rx}}. \tag{59}$$

Equation (56) yields the following:

$$\begin{aligned} &W'''' + \left(\frac{1}{s\chi} + 1\right)\omega^2\, W'' + \left(\frac{\left(\frac{\omega^2}{s} - 1\right)\omega^2}{\chi}\right) W = 0, \\ &\Phi'''' + \left(\frac{1}{s\chi} + 1\right)\omega^2\, \Phi'' + \left(\frac{\left(\frac{\omega^2}{s} - 1\right)\omega^2}{\chi}\right) \Phi = 0. \end{aligned} \tag{60}$$

We define:

$$\begin{aligned} \beta_1 &= \frac{1 + s\chi}{s\chi}, \\ \beta_2 &= \frac{\omega^2 - s}{s\chi} \end{aligned} \tag{61}$$

to make our forthcoming algebraic results more compact.

To solve for the eigenfrequencies, there are three cases to consider:

Case I $\omega^2 < \mathrm{s}$,

Case II $\omega^2 = \mathrm{s}$,

Case III $\omega^2 > \mathrm{s}$.

For Cases I and II, we can assume solutions of the form

$$W(x) \;=\; \sum_{i=1}^{4} A_i\, e^{r_i x}, \;\; \Phi(x) \;=\; \sum_{i=1}^{4} B_i\, e^{r_i x}, \tag{62}$$

and solve for r_i:

$$r_1 = \omega \sqrt{\frac{-\beta_1}{2} - \frac{1}{2}\sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{\omega^2}}}\ ,\ r_2 = -r_1,$$
$$r_3 = \omega \sqrt{\frac{-\beta_1}{2} + \frac{1}{2}\sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{\omega^2}}}\ ,\ r_4 = -r_3. \tag{63}$$

For Case III, all of the r_i are purely imaginary. In contrast, for Case I, $r_{1,2}$ are purely imaginary and $r_{3,4}$ are real. Case II is the transition between the distinct spectra of the other two cases. The constants $A_{1,2,3,4}$ and $B_{1,2,3,4}$ are not independent. Indeed, substituting the expressions (62) into the first of (56), we find that

$$B_i = \frac{1}{r_i}\left(r_i{}^2 - \omega^2\right) A_i,\ (i = 1, 2, 3, 4) \tag{64}$$

When r_i are complex, then it will prove convenient to substitute trigonometric functions for exponential functions in (62).

For Case II, ω^2 = s. The general solutions to (56) for this case are

$$W(x) = C_1 \cos\left(\sqrt{\beta_1\, s}\ x\right) + C_2 \sin\left(\sqrt{\beta_1\, s}\ x\right) + C_3\, x + C_4,$$
$$\Phi(x) = D_1 \cos\left(\sqrt{\beta_1\, s}\ x\right) + D_2 \sin\left(\sqrt{\beta_1\, s}\ x\right) + D_3\, x + D_4. \tag{65}$$

Substituting the expressions (65) into the first of (56), we find that

$$D_1 = (\beta_1\, s + s)\sqrt{\frac{1}{\beta_1\, s}}\ C_2,$$
$$D_2 = -(\beta_1\, s + s)\sqrt{\frac{1}{\beta_1\, s}}\ C_1, \tag{66}$$
$$D_3 = s\, C_4,$$
$$C_3 = 0.$$

As discussed in [?] ADD REFERENCE, the presence of modes of this type depend on the boundary conditions and are only possible in beams whose material and geometric properties satisfy a very restrictive set of conditions. We leave it as an interesting exercise to compute the shear deformation $\frac{\partial w}{\partial x} - \alpha$ corresponding to the mode shape (65) with the relations (66).

Frequency Equation

It remains to compute the natural frequencies ω for the modes for Case I and III. To do this we need four independent boundary conditions. These conditions are then used to form two equations of the following form:

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \longleftrightarrow M\,x = 0. \tag{67}$$

For a non-trivial solution the determinant of the matrix M must be zero. This yields a frequency equation for ω:

$$\det(M) = 0. \tag{68}$$

As with the Euler-Bernoulli beam, the resulting null vector of M yields provides three of the four constants A_i. To uniquely determine the set of coefficients we enforce an orthonormality condition of modes that shall be explained in a coming section. However, the generation of the frequency equation is an important interim step.

Additionally, to see if a particular beam can support a Case II mode the previous analysis is modified. Again we examine the boundary conditions and formulate a matrix equation: It is convenient to formulate a matrix equation using the four boundary conditions.

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{pmatrix} \begin{pmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \longleftrightarrow C\,x = 0 \tag{69}$$

For a non-trivial solution the determinant of the matrix C must be zero. This yields an equation that must be satisfied by the material and geometric properties of the beam,

$$\det(C) = 0. \tag{70}$$

The resulting null-vectors of M provides three of the D_i in terms of a fourth which can be identified by a subsequent orthonormality condition. To better highlight how the non-dimensional values affect the solutions for the possible eigenfrequencies we developed a function to plot the contours of the 3D frequency equation as seen in Figure 9

```
ContourPlotVisualization [s_, χ_, equation_] := Module[{},
   Clear[fix];
   f = equation;
   fix = {s, χ};
   Which[
    fix[[1]] == 0,
    temp = ContourPlot[f[z, s, fix[[2]]] == 0, {z, 0, 10},
      {s, 0, 10}, PlotRange → All],
    fix[[2]] == 0,
    temp = ContourPlot[f[z, fix[[1]], g] == 0, {z, 0, 10},
      {g, 0, 10}, PlotRange → All]];
   Return[temp]
  ];
timcrosssection[pos_, angle_] := Module[{line1},
   line1 =
    Line[{{pos[[1]] - 0.2 Sin[angle],
        pos[[2]] - 0.2 Cos[angle]},
      {pos[[1]] + 0.2 Sin[angle],
        pos[[2]] + 0.2 Cos[angle]}}];
   Return[line1]];
```

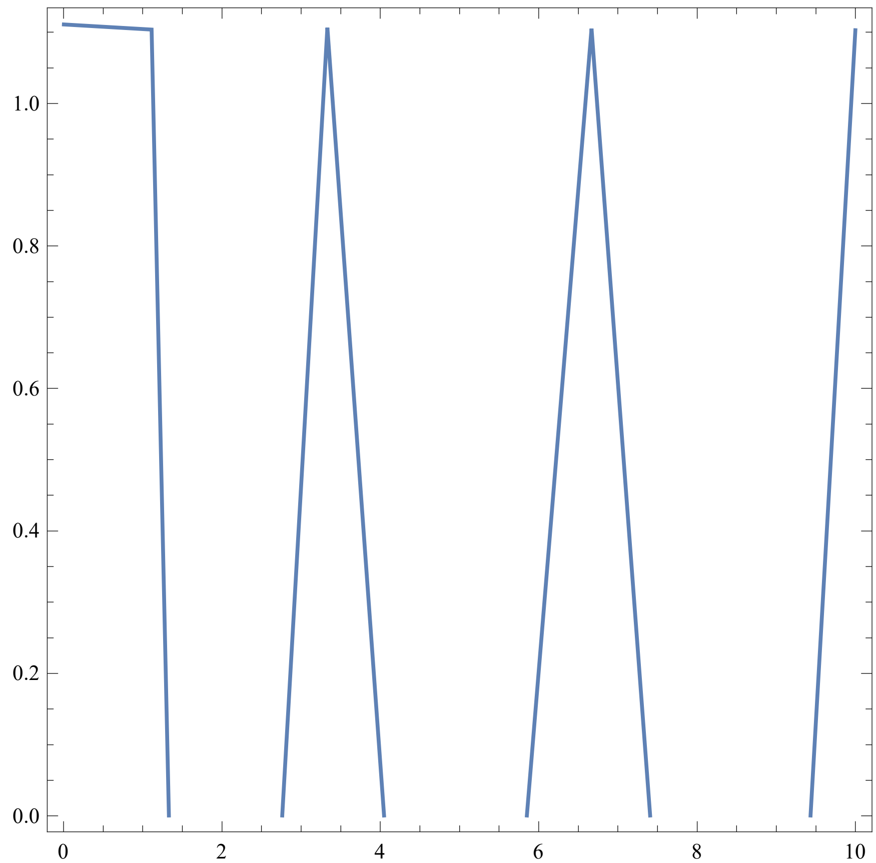

Figure 9. Example of the contour plot for the frequency equation for varying values of $z = \omega^2$ (x-axis) and χ (y-axis) for a vibrating, hinged-clamped Timoshenko beam. Note the multiple solutions for a given χ helps to highlight the multiple modes of vibration for a given beam.

Orthogonality of Modes

At this point, were we to solve for the composite shape of the vibrating beam, we would first solve the Matrix Equation highlighted at the inception of the Frequency Equation section. Doing so would yield the span of a coefficient, null-vector, i.e. $[\, A_1 \;\; A_2 \;\; A_3 \;\; A_4 \,]^T$ wherein we could backtrack to solve for the form of either Φ or W and identify the specific coefficients with equations provided by initial conditions. However, in doing so, we are not guaranteed that the over-arching deformation is a simple composite of the various mode shapes; although, were we to enforce the orthogonality of the various mode shapes, we would. Consequently, we define the following operators L_1 and L_2 as,

$$L_1 = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{s} \end{pmatrix}, \tag{71}$$

$$L_2 = \begin{pmatrix} -\frac{\partial^2}{\partial x^2} & \frac{\partial}{\partial x} \\ -\frac{\partial}{\partial x} & 1 - \chi \frac{\partial^2}{\partial x^2} \end{pmatrix}, \tag{72}$$

such that the following equation adequately represents equation 51 where f and m are set to 0:

$$\omega^2 L_1\, g \;=\; L_2\, g\,,\; g = \begin{pmatrix} w \\ \alpha \end{pmatrix}. \tag{73}$$

Furthermore, we would prefer for the operators to be self-adjoint, as that would result in distinct and orthogonal eigenpairs for each operator, vis a vis Hermitian operators. Note that L_1 as written is already self-adjoint and consequently not reproduced here; for L_2 however:

$$< L_2(g_m) \,|\, g_n > \;=\; < {L_2}^T\,(g_n) \,\big|\, g_m >, \tag{74}$$

$$\int_0^1 {g_n}^T\, L_2(g_m) \;-\; {g_m}^T\, {L_2}^T(g_n)\, \mathrm{dx} \;=0, \tag{75}$$

$$\left({\omega_m}^2 - {\omega_n}^2\right) \int_0^1 {g_n}^T\, L_1(g_m)\, \mathrm{dx} \;=\; 0. \tag{76}$$

Given that each ω_i that are solutions to the frequency equation are distinct, in order for equation 75 to be valid the corresponding integral must equal 0 for m ≠ n and some constant c for m = n. We subsequently normalize that constant c = 1 for convenience thus resulting in the Kronecker delta in the equation below,

$$\int_0^1 {g_n}^T\, L_1(g_m)\, \mathrm{dx} \;=\; \delta_{\mathrm{mn}}. \tag{77}$$

Equation 76 highlights both the orthogonality of the modes as well as the normalization condition applied to them. Thereby enforcing that any over-arching deformation is a simple composite of each mode-shape. Displaying the function here however would unnecessarily lengthen the paper; although, all functions shall be displayed in the Appendix and labeled appropriately.

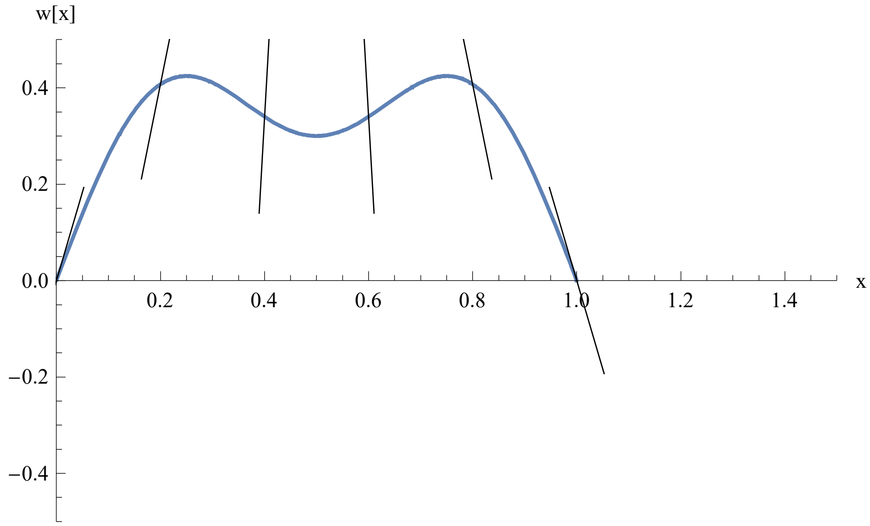

Figure 10. A vibrating Timoshenko beam with Simple end conditions. The first and third modes of vibration are shown above and the cross-sections are scaled appropriately. Note that for the non-dimensional values we used:
$s = 100 \quad \chi = 300$
Furthermore, the simple end conditions enforce that $\hat{w} = \hat{M} = 0$ at the ends of the beam.

Energy Analysis

For a given initial-value problem, it is interesting to compute how the number of modes used captures the initial energy of the deformed beam. To quantify the energy captured we need to discuss the energetics of the beam. We will discuss the Timoshenko beam as it readily reduces to the energetics of the Euler-Bernoulli beam.

The kinetic energy of the Timoshenko beam of length ℓ is given in dimensioned form by,

$$T = \frac{1}{2}\int_0^{\ell} \rho \mathrm{A}\left(\frac{\partial w}{\partial t}\right)^2 + \rho \mathrm{I}\left(\frac{\partial \alpha}{\partial t}\right)^2 \mathrm{dx}. \tag{78}$$

The elastic potential energy of the beam is composed of the bending energy and the shearing energy,

$$U = \frac{1}{2}\int_0^{\ell} \mathrm{EI}\left(\frac{\partial \alpha}{\partial x}\right)^2 + \mathrm{kGA}\left(\frac{\partial w}{\partial x} - \alpha\right)^2 \mathrm{dx}. \tag{79}$$

The total energy E_T for the Timoshenko beam is,

$$E_T = T + U. \tag{80}$$

The dimensionless version of this energy is,

$$\frac{E_T}{\mathrm{kGA}\ell} = \frac{1}{2}\int_0^1 \left(\frac{\partial w}{\partial t}\right)^2 + \frac{1}{s}\left(\frac{\partial \alpha}{\partial t}\right)^2 + \chi\left(\frac{\partial \alpha}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x} - \alpha\right)^2 \mathrm{dx}. \tag{81}$$

The corresponding energy for a Bernoulli-Euler beam is,

$$E_{\mathrm{BE}} = \frac{1}{2}\int_0^\ell \rho\mathrm{A}\left(\frac{\partial w}{\partial t}\right)^2 + \mathrm{EI}\left(\frac{\partial^2 w}{\partial x^2}\right)^2 \mathrm{dx}. \tag{82}$$

The dimensionless version of this expression is,

$$\frac{E_{\mathrm{BE}}}{\mathrm{EI}} = \frac{1}{2}\int_0^1 \left(\frac{\partial w}{\partial t}\right)^2 + \left(\frac{\partial^2 w}{\partial x^2}\right)^2 \mathrm{dx}. \tag{83}$$

For the free vibration of a beam, the energy remains constant. Hence, by comparing the initial energy to the energy captured by the finite number of modes used to model the vibration, a measure of the accuracy of the modal approximation can be obtained.

Appendix

A: Static Bernoulli-Euler Code

For graphing the static centerline deformation of a Bernoulli-Euler beam

```
StaticEulercenterline[Emod_,Iarea_,TotalLength_,Lsupport_,Rsupp
Clear[position,w,q,eqn1,eqn2,eqn3,eqn4,eqn5,solution];
q=t+b;
eqn1= w''''[x]== q/(Emod Iarea);
eqn2= Switch[Lsupport,Fix, w[0]==0,S, w[0]==0,Free, w''[0]==0];
eqn3= Switch[Lsupport,Fix, w'[0]==0,S, w''[0]==0,Free, w'''[0]==
eqn4= Switch[Rsupport,Fix, w[TotalLength]==0,S, w[TotalLength]=
eqn5= Switch[Rsupport,Fix, w'[TotalLength]==0,S, w''[TotalLengt
solution =DSolve[{eqn1,eqn2,eqn3,eqn4,eqn5},w[x],{x,0,TotalLen
position[x_]:=Evaluate[solution[[1]][[1]][[2]]];
Return[position]]
```

For outputting the various graphs:

```
StaticEulerGraphs[position_,TotalLength_]:=Module[{},
Clear[graph1,graph2,graph3,graph4,graph5];
graph1=Plot[position[x],{x,0,TotalLength},AxesLabel→{"x","posi
graph2=Plot[position'[x],{x,0,TotalLength},AxesLabel→{"x","Slo
graph3=Plot[ArcTan[position'[x]],{x,0,TotalLength},AxesLabel→{
graph4=Plot[position''[x],{x,0,TotalLength},AxesLabel→{"x","mo
graph5=Plot[-position'''[x],{x,0,TotalLength},AxesLabel→{"x","
Return[{graph1,graph2,graph3,graph4,graph5}]]
```

For displaying the centerline with cross-sections

```
StaticEulerSections[position_,TLength_]:=Module[{},
Clear[newposition,slope,shear,moment,radius,theta,crosssection
newposition[x_]:=position[x];
slope[x_]:=newposition'[x];
shear[x_]:=position'''[x];
moment[x_]:=position''[x];
radius[x_]:=(1/newposition''[x])/5;
theta[x_]:=ArcTan[newposition'[x]];
crosssection[var1_,var2_,var3_,constant_,TotalLength_,newposit
Graphics[Line[{{var2+Sin[constant theta[var2]] TotalLength/10,
{var2-Sin[constant theta[var2]] TotalLength/10,constant newpos
graphing[var1_,n_,place_,constant_,TotalLength_,newposition_,t
sep=TotalLength/n;
For[i=1,i<=n+1,i++,sections[[i]]=crosssection[var1,(i-1) sep,p
Return[sections]];
Return[Manipulate[Show[
Plot[b newposition[x],{x,l,u+0.01},PlotRange→{{0,TLength},{-c,
If[line,Graphics[Line[{{u+ Sin[b theta[u]] TLength/10,b newpos
{u-Sin[b theta[u]] TLength/10,b newposition[u]+Cos[b theta[u]]
If[line,graphing[u,number,l,b,TLength,newposition,theta],Plot[
{u,0.01,TLength-0.01},Item["u-sets RHS x-axis graph limit",Ali
Item["l sets the LHS x-axis graph limit",Alignment→Left],{b,10
Item["b magnifies the graph and all associated material (cross
{c,TLength},Item["c sets the pos/neg values on the y-axis",Ali
{line,{True,False}},Item["Line switches the cross-sections on/
Item["number sets the number of cross-sections the program wil
```

For displaying the Varying Internal Shear and Moment Loads within the beam

```
StaticEulerShear [position_ ,TotalLength_ ]:=Module [{},
Clear [upperboundshear ,lowerboundshear ,maxshear ,upperboundmoment
shear [x_]:=position ''' [x];
moment [x_]:=position '' [x];
upperboundshear =MaxValue [{shear [x] ,0<x<TotalLength } ,x];
lowerboundshear =MinValue [{shear [x] ,0<x<TotalLength } ,x];
maxshear =If [Abs [upperboundshear ]>=Abs [lowerboundshear ] ,upperbo
upperboundmoment =MaxValue [{moment [x] ,0<x<TotalLength } ,x];
lowerboundmoment =MinValue [{moment [x] ,0<x<TotalLength } ,x];
maxmoment =If [Abs [upperboundmoment ]≥Abs [lowerboundmoment ] ,upper
Manipulate [Show [
 Plot [0,{x,0,u},PlotRange→{{0,1.5},{-1.5,1.5}},AxesLabel→{"x",'
 Graphics [Arrow [{{u,-1 shear [u]/maxshear },{u,1 shear [u]/maxshea
 Graphics [Arrow [{{u,-1moment [u]/maxmoment },{u-0.1,-1moment [u]/m
 Graphics [Circle [{u,0},Abs [moment [u]/maxmoment ],{-Pi/2,Pi/2}]]]
 {u,0.01,1},Item ["u denotes the RHS graphing limit"]]];
```

Vibrational Bernoulli-Euler Code

For identifying the vibration of the Bernoulli-Euler centerline

```
VibEulerCenterline[type_,mode_]:=Module[{},
Clear[modetype,val,P,Q,R,S,FreeFreeeq,FixedFixedeq,FixedFreeeq
position,omega,FreeFreeroots,FixedFixedroots,FixedFreeroots,Fi
FreeFreeroots={0,4.73004,7.85320,10.9956,14.1372, 17.2788,20.4
FixedFixedroots={4.73004,7.85320,10.9956,14.1372, 17.2788,20.4
FixedFreeroots = {1.8751, 4.6409, 7.85476,10.9955,14.1372,17.2
FixedPinnedroots={3.92660,7.06858,10.212,13.3518,16.4934,19.63
PinnedPinnedroots={1*Pi,2*Pi,3*Pi,4*Pi,5*Pi,6*Pi,7*Pi,8*Pi,9*P
modetype=Switch[type,FreeFree,FreeFreeroots,FixedFixed, FixedF
FixedPinned,FixedPinnedroots,PinnedPinned,PinnedPinnedroots];
val=modetype[[mode]];
P[x_]:=Cosh[val x]+Cos[val x];
Q[x_]:=Cosh[val x]-Cos[val x];
R[x_]:=Sinh[val x]+Sin[val x];
S[x_]:=Sinh[val x]-Sin[val x];
FreeFreeeq[x_]:=P[x]-(Q[1]/S[1]) R[x];
FixedFixedeq[x_]:=Q[x]-(Q[1]/S[1]) S[x];
FixedFreeeq[x_]:=Q[x]-(S[1]/P[1]) S[x];
FixedPinnedeq[x_]:=Q[x]-(Q[1]/S[1]) S[x];
PinnedPinnedeq[x_]:=Sin[val x];
position[x_]:=Switch[type,FreeFree,FreeFreeeq[x],FixedFixed,Fi
FixedPinned,FixedPinnedeq[x],PinnedPinned,PinnedPinnedeq[x]];
omega=5;
Return[position]];
```

For visualizing the vibration of the centerline

```
VibEulerVisualization[position_]:=Module[{},
Manipulate[Plot[Cos[omega t] position[x],{x,0.01,1},PlotRange→
{t,0,2 Pi/(b omega)},{b,1,1/omega}]];
```

For understanding the complete nature of the deformation of the Bernoulli-Euler centerline (replete with slope, moments, shears, etc)

```
VibEulerDeformation [position_]:=Module[{},
Clear[theta,radius,slope,moment,shear,TotalLength];
slope[x_]:=position'[x];
moment[x_]:=position''[x];
shear[x_]:=position'''[x];
radius[x_]:=1/moment[x];
theta[x_]:=ArcTan[slope[x]];
TotalLength=1;
crosssection[var1_,var2_,var3_,constant_]:=If[var1≥var2&&var2≥
Graphics[Line[{{var2+Sin[constant theta[var2]] TotalLength/10,
{var2-Sin[constant theta[var2]] TotalLength/10,constant positi
graphing[var1_,n_,place_,constant_]:=Module[{sep,sections},sec
sep=TotalLength/n;
For[i=1,i<=n+1,i++,sections[[i]]=crosssection[var1,(i-1) sep,p
Return[sections]];
Manipulate[Show[
Plot[b position[x],{x,l,u+0.01},PlotRange→{{-TotalLength/5,2To
AxesLabel→{"x","scaled position(z)"}],
If[circ,Graphics[Circle[{u+Sin[theta[b u]] b radius[u],b posit
If[line,Graphics[Line[{{u+ Sin[theta[b u]]0.1 ,b position[u]-
{u-Sin[theta[b u]] 0.1,b position[u]+Cos[theta[b u]] 0.1}}]],P
If[line,graphing[u,number,l,b],Plot[0,{x,0,0.01}]]],
{u,0.01,TotalLength-0.01},{circ,{True,False}},{line,{True,Fals
```

For identifying the Bernoulli-Euler beam’s internal shear and moment

```
StaticTimoshenkoCenterline [Emod_,Gmod_,area_,k_,Iarea_,TotalLen
Clear[w,nonscalew, phi, nonscalephi,eqn1,eqn2,eqn3,eqn4,eqn5,e
q[x_]:=Evaluate[t+b];
eqn1= phi'[x]==w''[x]+q[x]/(k area Gmod);
eqn2= phi''[x]==k area Gmod/(Emod Iarea) (phi[x]-w'[x]);
eqn3= Switch[Lsupport,Fix,phi[0]==0,S,phi'[0]==0,Free,phi[0]==w'
eqn4= Switch[Rsupport,Fix,phi[TotalLength]==0,S,phi'[TotalLengt
eqn5= Switch[Lsupport,Fix,w[0]==0,S,w[0]==0,Free,phi'[0]==0];
eqn6= Switch[Rsupport,Fix,w[TotalLength]==0,S,w[TotalLength]==0
solution=DSolve[{eqn1,eqn2,eqn3,eqn4,eqn5,eqn6},{phi[x],w[x]},
nonscalephi[x_]:=Evaluate[solution[[1]][[1]][[2]]];
nonscalew[x_]:=Evaluate[solution[[1]][[2]][[2]]];
maxpos=MaxValue[{nonscalew[x],0<x<TotalLength},x];
w[x_]:=nonscalew[x]TotalLength/maxpos;
phi[x_]:=nonscalephi[x] TotalLength/maxpos;
Return[{w[x], phi[x]}]];
```

Static Timoshenko Section

For constructing the static centerline after deformation


```
StaticTimoshenkoCenterline [Emod_,Gmod_,area_,k_,Iarea_,TotalLen
Clear[w,nonscalew, phi, nonscalephi,eqn1,eqn2,eqn3,eqn4,eqn5,e
q[x_]:=Evaluate[t+b];
eqn1= phi'[x]==w''[x]+q[x]/(k area Gmod);
eqn2= phi''[x]==k area Gmod/(Emod Iarea) (phi[x]-w'[x]);
eqn3= Switch[Lsupport,Fix,phi[0]==0,S,phi'[0]==0,Free,phi[0]==w'
eqn4= Switch[Rsupport,Fix,phi[TotalLength]==0,S,phi'[TotalLengt
eqn5= Switch[Lsupport,Fix,w[0]==0,S,w[0]==0,Free,phi'[0]==0];
eqn6= Switch[Rsupport,Fix,w[TotalLength]==0,S,w[TotalLength]==0
solution=DSolve[{eqn1,eqn2,eqn3,eqn4,eqn5,eqn6},{phi[x],w[x]},
nonscalephi[x_]:=Evaluate[solution[[1]][[1]][[2]]];
nonscalew[x_]:=Evaluate[solution[[1]][[2]][[2]]];
maxpos=MaxValue[{nonscalew[x],0<x<TotalLength},x];
w[x_]:=nonscalew[x]TotalLength/maxpos;
phi[x_]:=nonscalephi[x] TotalLength/maxpos;
Return[{w[x], phi[x]}]];
```


For portraying the beam’s internal shear and moment, amongst others for a static beam.

```
StaticTimoshenkoGraphs[w_,phi_,TotalLength_,Emod_,Iarea_,k_,ar
Clear[moment,shear,slope,radius];
moment[x_]:=Emod Iarea phi'[x];
shear[x_]:= k area Gmod (w'[x]-phi[x]);
slope[x_]:=w'[x];
radius[x_]:=1/moment[x];
graph1=Plot[w[x],{x,0,TotalLength},PlotRange→All,AxesLabel→{"x
graph2=Plot[phi[x],{x,0,TotalLength},PlotRange→All,AxesLabel→{
graph3=Plot[moment[x],{x,0,TotalLength},PlotRange→All,AxesLabe
graph4=Plot[shear[x],{x,0,TotalLength},PlotRange→All,AxesLabel
graph5=Plot[slope[x],{x,0,TotalLength},PlotRange→All,AxesLabel
graph6=Plot[radius[x],{x,0,TotalLength},PlotRange→All,AxesLabe
Return[{graph1,graph2,graph3,graph4,graph5,graph6}]];
```

For portraying the deformation of the beam.

```
StaticTimoshenkoDeformation [w_,phi_,TotalLength_,Emod_,Iarea_,l
Clear[moment,shear,slope,radius,crosssection,graphcircle,graph
moment[x_]:=Emod Iarea phi'[x];
shear[x_]:= k area Gmod (w'[x]-phi[x]);
slope[x_]:=w'[x];
radius[x_]:=1/moment[x];
crosssection[rbx_,cbx_,information_,lbx_]:=If[rbx≤cbx&& rbx≥lb
information[[2]],information[[3]],information[[4]]]],{}];
graphcircle[n_,cpos_,constant_,lbx_,constant2_]:=Module[{sep,s
sections=0 Range[n+1];
sep=TotalLength/n;
For[i=1,i≤n+1,i++,
Module[{xpos,information},
xpos=(i-1) sep+0.01;
information={{xpos,constant w[xpos]},constant constant2 radius
sections[[i]]=circlesection[xpos,cpos[[1]],information,lbx]]];
Return[sections]];
graphsections[n_,cbx_,lbx_,constant_]:=Module[{sep,sections,i}
sections=0 Range[n+1];
sep=TotalLength/n;
For[i=1,i<=n+1,i++,
Module[{information,xpos},
xpos=(i-1) sep;
information={Tan[Pi/2+constant phi[xpos]+0.01],xpos,constant w
sections[[i]]=crosssection[xpos,cbx,information,lbx]]];
Return[sections]];
Manipulate[Show[
Plot[b w[x],{x,l,u},PlotRange→{{-TotalLength/5,TotalLength 6/5
If[section,line[Tan[Pi/2+b phi[u]],u,b w[u],TotalLength],Plot[
If[osc,circle[{u,b w[u]},b magnify2 radius[u],b slope[u]],Plot
If[section,graphsections[number,u,l,b],{}],
If[osc,graphcircle[number,{u,b w[u]},b,l,magnify2],{}]],
{u,TotalLength-0.01,0.01},{section,{True,False}},{osc,{True,Fa
```

For identifying the internal shear and moment of the beam (and displaying it on a graph).

```
StaticTimoshenkoShear [w_,phi_,TotalLength_,Emod_,Iarea_,k_,area
Clear[upperboundshear,lowerboundshear,maxshear,upperboundmoment
moment[x_]:=Emod Iarea phi'[x];
shear[x_]:= k area Gmod (w'[x]-phi[x]);
upperboundshear=MaxValue[{shear[x],0<x<TotalLength},x];
lowerboundshear=MinValue[{shear[x],0<x<TotalLength},x];
maxshear=If[Abs[upperboundshear]>=Abs[lowerboundshear],upperbo
upperboundmoment=MaxValue[{moment[x],0<x<TotalLength},x];
lowerboundmoment=MinValue[{moment[x],0<x<TotalLength},x];
maxmoment=If[Abs[upperboundmoment]≥Abs[lowerboundmoment],upper
Manipulate[Show[
 Plot[0,{x,0,u},PlotRange→{{-TotalLength/5,TotalLength},{-1.5,1
 Graphics[Arrow[{{u,-1 shear[u]/maxshear},{u,1 shear[u]/maxshea
 Graphics[Arrow[{{u,-1moment[u]/maxmoment},{u-0.1,-1moment[u]/m
 Graphics[Circle[{u,0},Abs[moment[u]/maxmoment],{-Pi/2,Pi/2}]]]
 {u,0.01,TotalLength-0.01}]];
```

Vibrational Timoshenko Section

For visualizing the contour plots (relative to the aforementioned non-dimensional values)

```
ContourPlotVisualization [s_,χ_,equation_]:=Module[{},
Clear[fix];
f=equation;
fix={s,χ};
Which[
 fix[[1]]==0,temp=ContourPlot[f[z,s,fix[[2]]]==0,{z,-10,10},{s,-
 fix[[2]]==0,temp=ContourPlot[f[z,fix[[1]],χ]==0,{z,-10,10},{χ,-
Return[temp]
];
timcrosssection[pos_,angle_]:=Module[{line1},
line1=Line[{{pos[[1]]-0.2 Sin[angle],pos[[2]]-0.2 Cos[angle]},
Return[line1]];
```

For identifying the complete vibration of the beam.

```
TimoshenkoVibrationVisualization [s_,χ_,temp_,equation_,modes_]
f=equation;
roots=omegafinder[s,χ,f];
placeholder=Range[Length[modes]];
```

```
For[i=1,i≤ Length[modes],i++,placeholder[[i]]={}];
repetition[constant_]:=Module[{},
z=Abs[roots[[constant]]];
```

$$\beta_1=\frac{1}{s\ \chi}+1;$$

$$\beta_2=\frac{\left(\frac{z}{s}-1\right)}{\chi};$$

$$a=\sqrt{\frac{z}{2}\left(\beta_1+\sqrt{\beta_1{}^2-\frac{4\ \beta_2}{z}}\right)};$$

$$b=\sqrt{\frac{z}{2}\left(-\beta_1+\sqrt{\beta_1{}^2-\frac{4\ \beta_2}{z}}\right)};$$

$$B_1=\frac{\left(z-a^2\right)\ A_2}{a};$$

$$B_2=\frac{\left(a^2-z\right)\ A_1}{a};$$

$$B_3=\frac{\left(z+b^2\right)\ A_4}{b};$$

$$B_4=\frac{\left(z+b^2\right)\ A_3}{b};$$

```
w[x_]:=A1Sin[a x]+A2Cos[a x]+A3Sinh[b x]+A4Cosh[b x];
phi[x_]:=B1Sin[a x]+B2Cos[a x]+B3Sinh[b x]+B4Cosh[b x];
(*Identifying the current omega^2 value that we are considering
conditions=Switch[temp,
HH,{w[0]==0,phi'[0]==0,w[1]==0},
HC ,{w[0]==0,phi'[0]==0,w[1]==0},
CC,{w[0]==0,w'[0]==0,w[1]==0},
CF,{w[0]==0,w'[0]==0,phi'[1]==0},
FF,{phi'[0]==0,w'[0]+phi[0]==0,phi'[1]==0}]; (**)
eqn1=conditions[[1]];
eqn2=conditions[[2]];
eqn3=conditions[[3]];
```

$$\text{eqn4}=\text{Integrate}\left[z\ w[x]^2-z\ \frac{\text{phi}[x]^2}{s},\{x,0,1\}\right]==1;$$

```
solution=Solve[eqn1&&eqn2&&eqn3&&eqn4,{A1,A2,A3,A4}];
set1={Re[solution[[1]][[1]][[2]]],Re[solution[[1]][[2]][[2]]],
set2={Re[solution[[2]][[1]][[2]]],Re[solution[[2]][[2]][[2]]],
test=Abs[set1];
```

```
cellindex=Position[test,Max[test]];
index=cellindex[[1]][[1]];
If[set1[[index]]==test[[index]],coeff=set1,coeff=set2];
δ=((z-a^2) coeff[[2]])/a;
ϵ=((a^2-z) coeff[[1]])/a;
ϕ=((z+b^2) coeff[[4]])/b;
η=((z+b^2) coeff[[3]])/b;
tempw[x_,t_]:= Cos[Abs[Sqrt[z]] t] (coeff[[1]] Sin[a x]+coeff[
tempphi[x_,t_]:=  Cos[Abs[Sqrt[z]] t] (δ Sin[a x]+ϵ Cos[a x]+ϕ
Return[{{tempw[x,t]},
{tempphi[x,t]}}]];placeholder[[1]]=repetition[modes[[1]]];
If[Length[modes]>1,For[i=2,i≤Length[modes],i++,placeholder[[i]
temporary1=placeholder[[Length[modes]]][[1]][[1]];
temporary2=placeholder[[Length[modes]]][[2]][[1]];
Ω[x_,t_]:=Evaluate[temporary1];
Ψ[x_,t_]:=Evaluate[temporary2];
Manipulate[Module[{},
graph1=Plot[Ω[x,t],{x,0,1},PlotRange→{{0,1.5},{-0.5,0.5}},Axes
line1=timcrosssection[{0,Ω[0,t]},mag Ψ[0,t]];
line2=timcrosssection[{0.2,Ω[0.2,t]},mag Ψ[0.2,t]];
line3=timcrosssection[{0.4,Ω[0.4,t]},mag Ψ[0.4,t]];
line4=timcrosssection[{0.6,Ω[0.6,t]},mag Ψ[0.6,t]];
line5=timcrosssection[{0.8,Ω[0.8,t]},mag Ψ[0.8,t]];
line6=timcrosssection[{1,Ω[1,t]},mag Ψ[1,t]];
Show[graph1,Graphics[{line1,line2,line3,line4,line5,line6}]]
]
,{t,0,16 Pi},{mag,1}]
];
```

■ Demonstration

```
MultiRootFinder[function_, number_, increment_] :=
  Module[{rootset, root, roundedrootset, roundedroot,
    step, rootnumber, value},
   value = FindRoot[function[a], {a, 0.01}];
```

```
    root = value[[1]][[2]];
    roundedroot = Round[root 1000];
    rootset = {};
    roundedrootset = {};
    rootset = Append[rootset, root];
    roundedrootset = Append[roundedrootset, roundedroot];
    step = 1;
    rootnumber = 1;
    While[rootnumber < number, Module[{},
      value = FindRoot[function[a], {a, step increment + 1}];
      root = value[[1]][[2]];
      roundedroot = Round[root 1000] / 1000;
      If[MemberQ[roundedrootset, roundedroot] ||
        roundedroot  < 1, {},
       Module[{}, roundedrootset =
          Append[roundedrootset, roundedroot];
         rootset = Append[rootset, root]];];
      step += 1;
      rootnumber = Length[rootset]]];
    rootset = Sort[rootset];
    Return[rootset]];
omegafinder[s_, χ_, function_] := Module[{roots, temp},
    temp[beta_] := function[beta, s, χ];
    roots = MultiRootFinder[temp, 10, 0.5];
    If[roots[[1]] < 1,
     roots = Part[roots, 2 ;; Length[roots]];];
    roots = DeleteDuplicates[roots];
    Return[roots]];
StaticEulercenterline[Emod_, Iarea_, TotalLength_,
    Lsupport_, Rsupport_, t_, b_] := Module[{},
    Clear[position, w, q, eqn1, eqn2, eqn3, eqn4, eqn5,
     solution];
    q = t + b;
    eqn1 = w''''[x] == q / (Emod Iarea);
    eqn2 = Switch[Lsupport, Fix, w[0] == 0, S, w[0] == 0,
      Free, w''[0] == 0];
    eqn3 = Switch[Lsupport, Fix, w'[0] == 0, S, w''[0] == 0,
      Free, w'''[0] == 0];
    eqn4 = Switch[Rsupport, Fix, w[TotalLength] == 0,
      S, w[TotalLength] == 0, Free, w''[TotalLength] == 0];
    eqn5 = Switch[Rsupport, Fix, w'[TotalLength] == 0,
      S, w''[TotalLength] == 0, Free,
      w'''[TotalLength] == 0];
    solution = DSolve[{eqn1, eqn2, eqn3, eqn4, eqn5},
```

```
    w[x], {x, 0, TotalLength}];
  position[x_] := Evaluate[solution[[1]][[1]][[2]]];
  Return[position]];
StaticEulerGraphs[position_, TotalLength_] := Module[{},
  Clear[graph1, graph2, graph3, graph4, graph5];
  graph1 = Plot[position[x], {x, 0, TotalLength},
    AxesLabel → {"x", "w[x]"}];
  graph2 = Plot[position'[x], {x, 0, TotalLength},
    AxesLabel → {"x", "w'[x]"}];
  graph3 = Plot[ArcTan[position'[x]], {x, 0, TotalLength},
    AxesLabel → {"x", "θ"}];
  graph4 = Plot[position''[x], {x, 0, TotalLength},
    AxesLabel → {"x", "M(x)"}];
  graph5 = Plot[-position'''[x], {x, 0, TotalLength},
    AxesLabel → {"x", "V(x)"}];
  Return[{graph1, graph2, graph3, graph4, graph5}]];
StaticEulerSections[position_, TLength_] := Module[{},
  Clear[newposition, slope, shear, moment, radius,
   theta, crosssection, graphing];
  newposition[x_] := position[x];
  slope[x_] := newposition'[x];
  shear[x_] := position'''[x];
  moment[x_] := position''[x];
  radius[x_] := (1 / newposition''[x]) / 5;
  theta[x_] := ArcTan[newposition'[x]];
  crosssection[var1_, var2_, var3_, constant_,
    TotalLength_, newposition_, theta_] :=
   If[var1 ≥ var2 && var2 ≥ var3,
    Graphics[
     Line[
      {{var2 + Sin[constant theta[var2]] TotalLength / 10,
        constant newposition[var2] -
         Cos[constant theta[var2]] TotalLength / 10},
       {var2 - Sin[constant theta[var2]] TotalLength / 10,
        constant newposition[var2] +
         Cos[constant theta[var2]] TotalLength / 10}}]],
    {}];
  graphing[var1_, n_, place_, constant_, TotalLength_,
    newposition_, theta_] :=
   Module[{sep, sections}, sections = 0 Range[n + 1];
    sep = TotalLength / n;
    For[i = 1, i <= n + 1, i++,
     sections[[i]] = crosssection[var1, (i - 1) sep,
       place, constant, TotalLength, newposition, theta]];
```

```
      Return[sections]];
    Return[Manipulate[Show[
       Plot[d newposition[x], {x, l, u + 0.01},
        PlotRange → {{0, TLength}, {-c, c}},
        AxesLabel → {"x", "w[x]"}],
       If[line, Graphics[
         Line[{{u + Sin[b theta[u]] TLength / 10,
            b newposition[u] - Cos[b theta[u]] TLength / 10},
           {u - Sin[b theta[u]] TLength / 10,
            b newposition[u] + Cos[b theta[u]] TLength / 10}}]],
        Plot[0, {x, 0, 1}]],
       If[line, graphing[u, number, l, b, TLength,
         newposition, theta], Plot[0, {x, 0, 1}]]],
      {u, 0.01, TLength - 0.01},
      Item["u-sets RHS x-axis graph limit",
       Alignment → Left],
      {l, 0.01, TLength - 0.01},
      Item["l sets the LHS x-axis graph limit",
       Alignment → Left],
      {b, 10 000},
      Item[
       "b magnifies all associated material i.e.
         cross-sections", Alignment → Left],
      {d, 100}, Item["d magnifies the graph",
       Alignment → Left],
      {c, TLength},
      Item["c sets the pos/neg values on the y-axis",
       Alignment → Left],
      {number, 5},
      Item[
       "number sets the number of cross-sections the
         program will animate", Alignment → Left],
      {line, {True, False}},
      Item["Line switches the cross-sections on/off",
       Alignment → Left]]]];
StaticEulerShear[position_, TotalLength_] := Module[{},
    Clear[upperboundshear, lowerboundshear, maxshear,
     upperboundmoment, lowerboundmoment, maxmoment,
     shear, moment];
    shear[x_] := position'''[x];
    moment[x_] := position''[x];
    upperboundshear =
     MaxValue[{shear[x], 0 < x < TotalLength}, x];
    lowerboundshear = MinValue[{shear[x], 0 < x < TotalLength},
```

```
        x];
      maxshear =
       If[Abs[upperboundshear] >= Abs[lowerboundshear],
        upperboundshear, lowerboundshear];
      upperboundmoment =
       MaxValue[{moment[x], 0 < x < TotalLength}, x];
      lowerboundmoment =
       MinValue[{moment[x], 0 < x < TotalLength}, x];
      maxmoment =
       If[Abs[upperboundmoment] ≥ Abs[lowerboundmoment],
        upperboundmoment, lowerboundmoment];
      Manipulate[Show[
        Plot[0, {x, 0, u}, PlotRange → {{0, 1.5}, {-1.5, 1.5}},
         AxesLabel → {"x", "scaled magnitude"}],
        Graphics[Arrow[{{u, -1 shear[u] / maxshear},
           {u, 1 shear[u] / maxshear}}]],
        Graphics[Arrow[{{u, -1 moment[u] / maxmoment},
           {u - 0.1, -1 moment[u] / maxmoment}}]],
        Graphics[Circle[{u, 0}, Abs[moment[u] / maxmoment],
          {-Pi / 2, Pi / 2}]]],
       {u, 0.01, 1},
       Item["u denotes the RHS graphing limit"]]];
   VibEulerCenterline[type_, mode_] := Module[{},
      Clear[modetype, val, P, Q, R, S, FreeFreeeq,
       FixedFixedeq, FixedFreeeq, FixedPinnedeq,
       PinnedPinnedeq, position, omega, FreeFreeroots,
       FixedFixedroots, FixedFreeroots, FixedPinnedroots,
       PinnedPinnedroots];
      FreeFreeroots = {0, 4.73004, 7.85320, 10.9956,
        14.1372, 17.2788, 20.4204, 23.5619, 26.7035,
        29.8451};
      FixedFixedroots = {4.73004, 7.85320, 10.9956,
        14.1372, 17.2788, 20.4204, 23.5619, 26.7035,
        29.8451, 32.9867};
      FixedFreeroots = {1.8751, 4.6409, 7.85476, 10.9955,
        14.1372, 17.2788, 20.4204, 23.5619, 26.7035, 29.8451};
      FixedPinnedroots = {3.92660, 7.06858, 10.212,
        13.3518, 16.4934, 19.635, 22.7765, 25.9181,
        29.0597, 32.2013};
      PinnedPinnedroots = {1 * Pi, 2 * Pi, 3 * Pi, 4 * Pi,
        5 * Pi, 6 * Pi, 7 * Pi, 8 * Pi, 9 * Pi, 10 * Pi};
      modetype = Switch[type, FreeFree, FreeFreeroots,
        FixedFixed, FixedFixedroots, FixedFree,
        FixedFreeroots, FixedPinned, FixedPinnedroots,
```

```
      PinnedPinned, PinnedPinnedroots];
    val = modetype[[mode]];
    P[x_] := Cosh[val x] + Cos[val x];
    Q[x_] := Cosh[val x] - Cos[val x];
    R[x_] := Sinh[val x] + Sin[val x];
    S[x_] := Sinh[val x] - Sin[val x];
    FreeFreeeq[x_] := P[x] - (Q[1] / S[1]) R[x];
    FixedFixedeq[x_] := Q[x] - (Q[1] / S[1]) S[x];
    FixedFreeeq[x_] := Q[x] - (S[1] / P[1]) S[x];
    FixedPinnedeq[x_] := Q[x] - (Q[1] / S[1]) S[x];
    PinnedPinnedeq[x_] := Sin[val x];
    position[x_] := Switch[type, FreeFree, FreeFreeeq[x],
      FixedFixed, FixedFixedeq[x], FixedFree,
      FixedFreeeq[x], FixedPinned, FixedPinnedeq[x],
      PinnedPinned, PinnedPinnedeq[x]];
    omega = 5;
    Return[position]];
VibEulerVisualization[position_] := Module[{},
    Manipulate[Plot[Cos[omega t] position[x], {x, 0.01, 1},
      PlotRange → {{0, 1.5}, {-1.5, 1.5}},
      AxesLabel → {"x", "position[z]"}],
     {t, 0, 2 Pi / (b omega)}, {b, 1, 1 / omega}]];
VibEulerDeformation[position_] := Module[{},
    Clear[theta, radius, slope, moment, shear, TotalLength];
    slope[x_] := position'[x];
    moment[x_] := position''[x];
    shear[x_] := position'''[x];
    radius[x_] := 1 / moment[x];
    theta[x_] := ArcTan[slope[x]];
    TotalLength = 1;
    crosssection[var1_, var2_, var3_, constant_] :=
     If[var1 ≥ var2 && var2 ≥ var3,
      Graphics[
       Line[
        {{var2 + Sin[constant theta[var2]] TotalLength / 10,
          constant position[var2] -
           Cos[constant theta[var2]] TotalLength / 10},
         {var2 - Sin[constant theta[var2]] TotalLength / 10,
          constant position[var2] +
           Cos[constant theta[var2]] TotalLength / 10}}]],
      {}];
    graphing[var1_, n_, place_, constant_] :=
     Module[{sep, sections}, sections = 0 Range[n + 1];
      sep = TotalLength / n;
```

```
    For[i = 1, i <= n + 1, i++,
     sections[[i]] = crosssection[var1, (i - 1) sep,
       place, constant]];
    Return[sections]];
  Manipulate[Show[
    Plot[b position[x], {x, l, u + 0.01},
     PlotRange → {{-TotalLength / 5, 2 TotalLength},
       {-2 TotalLength, 2 TotalLength}},
     AxesLabel → {"x", "w[x]"}],
    If[circ,
     Graphics[Circle[{u + Sin[theta[b u]] b radius[u],
        b position[u] - Cos[theta[b u]] b radius[u]},
       Abs[b radius[u]]]], Plot[0, {x, 0, 0.01}]],
    If[line,
     Graphics[
      Line[{{u + Sin[theta[b u]] 0.1 ,
          b position[u] - Cos[theta[b u]] 0.1},
         {u - Sin[theta[b u]] 0.1,
          b position[u] + Cos[theta[b u]] 0.1}}]],
     Plot[0, {x, 0, 0.01}]],
    If[line, graphing[u, number, l, b],
     Plot[0, {x, 0, 0.01}]]],
   {u, 0.01, TotalLength - 0.01}, {circ, {True, False}},
   {line, {True, False}}, {l, 0.01, TotalLength - 0.01},
   {b, 1, 0}, {number, 5}]];
VibEulerShear[position_] := Module[{},
  Clear[upperboundshear, lowerboundshear,
   upperboundmoment, lowerboundmoment, maxshear,
   maxmoment];
  slope[x_] := position'[x];
  moment[x_] := position''[x];
  shear[x_] := position'''[x];
  radius[x_] := 1 / moment[x];
  theta[x_] := ArcTan[slope[x]];
  TotalLength = 1;
  upperboundshear =
   MaxValue[{shear[x], 0 < x < TotalLength}, x];
  lowerboundshear = MinValue[{shear[x], 0 < x < TotalLength},
    x];
  maxshear =
   If[Abs[upperboundshear] >= Abs[lowerboundshear],
    upperboundshear, lowerboundshear];
  upperboundmoment =
   MaxValue[{moment[x], 0 < x < TotalLength}, x];
```

```
    lowerboundmoment =
     MinValue[{moment[x], 0 < x < TotalLength}, x];
    maxmoment =
     If[Abs[upperboundmoment] ≥ Abs[lowerboundmoment],
      upperboundmoment, lowerboundmoment];
    Manipulate[Show[
      Plot[0, {x, 0, u},
       PlotRange → {{-TotalLength / 5, 1.5}, {-1.5, 1.5}},
       AxesLabel → {"x", "scaled magnitude"}],
      Graphics[Arrow[{{u, -Cos[omega t] shear[u] / maxshear},
         {u, Cos[omega t] shear[u] / maxshear}}]],
      Graphics[
       Arrow[{{u, Cos[omega t] moment[u] / maxmoment},
         {u - 0.1, Cos[omega t] moment[u] / maxmoment}}]],
      Graphics[Circle[{u, 0},
        Abs[Cos[omega t] moment[u] / maxmoment],
        {-Pi / 2, Pi / 2}]]],
     {u, 0.01, TotalLength - 0.01}, {t, 0, 2 Pi / omega}]];
StaticTimoshenkoCenterline[Emod_, Gmod_, area_, k_,
    Iarea_, TotalLength_, Lsupport_, Rsupport_, t_, b_] :=
   Module[{},
    Clear[w, nonscalew, phi, nonscalephi, eqn1, eqn2,
     eqn3, eqn4, eqn5, eqn6, solution, q, maxpos];
    q[x_] := Evaluate[t + b];
    eqn1 = phi'[x] == w''[x] + q[x] / (k area Gmod);
    eqn2 = phi''[x] == k area Gmod / (Emod Iarea)
        (phi[x] - w'[x]);
    eqn3 = Switch[Lsupport, Fix, phi[0] == 0, S, phi'[0] == 0,
      Free, phi[0] == w'[0]];
    eqn4 = Switch[Rsupport, Fix, phi[TotalLength] == 0,
      S, phi'[TotalLength] == 0, Free,
      phi[TotalLength] == w'[TotalLength]];
    eqn5 = Switch[Lsupport, Fix, w[0] == 0, S, w[0] == 0,
      Free, phi'[0] == 0];
    eqn6 = Switch[Rsupport, Fix, w[TotalLength] == 0, S,
      w[TotalLength] == 0, Free, phi'[TotalLength] == 0];
    solution = DSolve[{eqn1, eqn2, eqn3, eqn4, eqn5, eqn6},
      {phi[x], w[x]}, {x, 0, TotalLength}];
    nonscalephi[x_] := Evaluate[solution[[1]][[1]][[2]]];
    nonscalew[x_] := Evaluate[solution[[1]][[2]][[2]]];
    maxpos = MaxValue[{nonscalew[x], 0 < x < TotalLength}, x];
    w[x_] := nonscalew[x] TotalLength / maxpos;
    phi[x_] := nonscalephi[x] TotalLength / maxpos;
    Return[{w[x], phi[x]}]];
```

```
StaticTimoshenkoGraphs [w_, phi_, TotalLength_,
   Emod_, Iarea_, k_, area_, Gmod_] := Module[{},
   Clear[moment, shear, slope, radius];
   moment[x_] := Emod Iarea phi'[x];
   shear[x_] := k area Gmod (w'[x] - phi[x]);
   slope[x_] := w'[x];
   radius[x_] := 1 / moment[x];
   graph1 = Plot[w[x], {x, 0, TotalLength}, PlotRange → All,
     AxesLabel → {"x", "w"}];
   graph2 = Plot[phi[x], {x, 0, TotalLength},
     PlotRange → All, AxesLabel → {"x", "ϕ"}];
   graph3 = Plot[moment[x], {x, 0, TotalLength},
     PlotRange → All, AxesLabel → {"x", "moment"}];
   graph4 = Plot[shear[x], {x, 0, TotalLength},
     PlotRange → All, AxesLabel → {"x", "shear"}];
   graph5 = Plot[slope[x], {x, 0, TotalLength},
     PlotRange → All, AxesLabel → {"x", "slope"}];
   graph6 = Plot[radius[x], {x, 0, TotalLength},
     PlotRange → All, AxesLabel → {"x", "radius"}];
   Return[{graph1, graph2, graph3, graph4, graph5,
     graph6}]];
line[slope_, ptx0_, pty0_, length_] :=
  Module[{offset, ptx1, ptx2, pty1, pty2, xdev, ydev},
   offset = length / 8;
   xdev = offset / Sqrt[1 + slope^2];
   ydev = xdev slope;
   ptx1 = ptx0 - xdev;
   ptx2 = ptx0 + xdev;
   pty1 = pty0 - ydev;
   pty2 = pty0 + ydev;
   Return[Graphics[Line[{{ptx1, pty1}, {ptx2, pty2}}]]]];
circle[cpos_, rad_, sl_] :=
  Module[{xdev, ydev, ptx1, pty1, newslope},
   newslope = -1 / sl;
   xdev = rad / Sqrt[1 + newslope^2];
   ydev = xdev newslope;
   ptx1 = cpos[[1]] - xdev;
   pty1 = cpos[[2]] - ydev;
   Return[Graphics[Circle[{ptx1, pty1}, Abs[rad]]]]];
circlesection[rbx_, cbx_, information_, lbx_] :=
  If[rbx ≤ cbx && rbx ≥ lbx,
   Return[circle[information[[1]], information[[2]],
     information[[3]]]], {}];
StaticTimoshenkoDeformation [w_, phi_, TotalLength_,
```

```
Emod_, Iarea_, k_, area_, Gmod_] := Module[{},
Clear[moment, shear, slope, radius, crosssection,
 graphcircle, graphsections];
moment[x_] := Emod Iarea phi'[x];
shear[x_] := k area Gmod (w'[x] - phi[x]);
slope[x_] := w'[x];
radius[x_] := 1 / moment[x];
crosssection[rbx_, cbx_, information_, lbx_] :=
 If[rbx ≤ cbx && rbx ≥ lbx,
  Return[line[information[[1]], information[[2]],
    information[[3]], information[[4]]]], {}];
graphcircle[n_, cpos_, constant_, lbx_, constant2_] :=
 Module[{sep, sections, i},
  sections = 0 Range[n + 1];
  sep = TotalLength / n;
  For[i = 1, i ≤ n + 1, i++,
   Module[{xpos, information},
    xpos = (i - 1) sep + 0.01;
    information = {{xpos, constant w[xpos]},
      constant constant2 radius[xpos],
      constant slope[xpos]};
    sections[[i]] = circlesection[xpos, cpos[[1]],
      information, lbx]]];
  Return[sections]];
graphsections[n_, cbx_, lbx_, constant_] :=
 Module[{sep, sections, i},
  sections = 0 Range[n + 1];
  sep = TotalLength / n;
  For[i = 1, i <= n + 1, i++,
   Module[{information, xpos},
    xpos = (i - 1) sep;
    information = {Tan[Pi / 2 + constant phi[xpos] + 0.01],
      xpos, constant w[xpos], TotalLength};
    sections[[i]] = crosssection[xpos, cbx,
      information, lbx]]];
  Return[sections]];
Manipulate[Show[
  Plot[b w[x], {x, l, u},
   PlotRange → {{-TotalLength / 5, TotalLength 6 / 5},
     {-TotalLength 6 / 5, TotalLength 6 / 5}},
   AxesLabel → {"x", "w[x]"}],
  If[section, line[Tan[Pi / 2 + b phi[u]], u, b w[u],
    TotalLength], Plot[0, {x, 0, 0.01}]],
  If[osc, circle[{u, b w[u]}, b magnify2 radius[u],
```

```
        b slope[u]], Plot[0, {x, 0, 0.01}]],
      If[section, graphsections[number, u, l, b], {}],
      If[osc, graphcircle[number, {u, b w[u]}, b, l,
        magnify2], {}]],
    {u, TotalLength - 0.01, 0.01}, {section, {True, False}},
    {osc, {True, False}}, {l, 0, u - 0.01},
    {magnify2, 1, 1000}, {b, 1, 0.01}, {number, 5}]];
StaticTimoshenkoShear[w_, phi_, TotalLength_, Emod_,
   Iarea_, k_, area_, Gmod_] := Module[{},
   Clear[upperboundshear, lowerboundshear, maxshear,
    upperboundmoment, lowerboundmoment, maxmoment];
   moment[x_] := Emod Iarea phi'[x];
   shear[x_] := k area Gmod (w'[x] - phi[x]);
   upperboundshear =
    MaxValue[{shear[x], 0 < x < TotalLength}, x];
   lowerboundshear = MinValue[{shear[x], 0 < x < TotalLength},
     x];
   maxshear =
    If[Abs[upperboundshear] >= Abs[lowerboundshear],
     upperboundshear, lowerboundshear];
   upperboundmoment =
    MaxValue[{moment[x], 0 < x < TotalLength}, x];
   lowerboundmoment =
    MinValue[{moment[x], 0 < x < TotalLength}, x];
   maxmoment =
    If[Abs[upperboundmoment] ≥ Abs[lowerboundmoment],
     upperboundmoment, lowerboundmoment];
   Manipulate[Show[
     Plot[0, {x, 0, u},
      PlotRange → {{-TotalLength / 5, TotalLength},
        {-1.5, 1.5}},
      AxesLabel → {"x", "scaled magnitude"}],
     Graphics[Arrow[{{u, -1 shear[u] / maxshear},
        {u, 1 shear[u] / maxshear}}]],
     Graphics[Arrow[{{u, -1 moment[u] / maxmoment},
        {u - 0.1, -1 moment[u] / maxmoment}}]],
     Graphics[Circle[{u, 0}, Abs[moment[u] / maxmoment],
       {-Pi / 2, Pi / 2}]]],
    {u, 0.01, TotalLength - 0.01}]];
(*Timoshenko Frequency Plot Visualization*)
```

$\beta_1 = \frac{1}{s\,\chi} + 1;$

$$\beta_2 = \frac{\left(\frac{z}{s} - 1\right)}{\chi};$$

$$a = \sqrt{\frac{z}{2}\left(\beta_1 + \sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{z}}\right)};$$

$$b = \sqrt{\frac{z}{2}\left(-\beta_1 + \sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{z}}\right)};$$

```
HHfreq[z_, s_, χ_] := Evaluate[Sin[a] Sinh[b]];
HCfreq[z_, s_, χ_] :=
  Evaluate[-b Cosh[b] Sin[a] + a Cos[a] Sinh[b]];
CCfreq[z_, s_, χ_] :=
  Evaluate[2 a b - 2 a b Cos[a] Cosh[b] +
     (-a^2 + b^2) Sin[a] Sinh[b]];
CFfreq[z_, s_, χ_] :=
  Evaluate[
   1/(a b) (-2 a^2 (a - b) b^2 (a + b) + 6 a^2 b^2 z α + (a - b) (a + b) z^2 α^2 +
       (a^2 - b^2 - z α) (2 a^2 b^2 + (a - b) (a + b) z α) Cos[a] Cosh[b] +
       a b (z α (3 b^2 + 2 z α) - a^2 (4 b^2 + 3 z α)) Sin[a] Sinh[b])];
FFfreq[z_, s_, χ_] :=
  Evaluate[
   - 1/(a^2 b^2)
    (a (-a^2 + z α) (2 b^2 + z α)
       (b (2 a^2 - z α) (b^2 + z α) (-1 + Cos[a] Cosh[b]) +
         a (a^2 - z α) (2 b^2 + z α) Sin[a] Sinh[b]) +
      b (2 a^2 - z α) (b^2 + z α)
       (a (a^2 - z α) (2 b^2 + z α) Cosh[b] (-Cos[a] + Cosh[b]) +
         Sinh[b] (b (2 a^2 - z α) (b^2 + z α) Sin[a] -
            a (a^2 - z α) (2 b^2 + z α) Sinh[b])))];

ContourPlotVisualization [s_, χ_, equation_] := Module[{},
   Clear[fix];
   f = equation;
   fix = {s, χ};
   Which[
    fix[[1]] == 0,
```

```
      temp = ContourPlot[f[z, s, fix[[2]]] == 0, {z, 0, 10},
        {s, 0, 10}, PlotRange → All],
      fix[[2]] == 0,
      temp = ContourPlot[f[z, fix[[1]], g] == 0, {z, 0, 10},
        {g, 0, 10}, PlotRange → All]];
    Return[temp]
   ];
timcrosssection[pos_, angle_] := Module[{line1},
    line1 =
     Line[{{pos[[1]] - 0.2 Sin[angle],
        pos[[2]] - 0.2 Cos[angle]},
       {pos[[1]] + 0.2 Sin[angle],
        pos[[2]] + 0.2 Cos[angle]}}];
    Return[line1]];
TimoshenkoVibrationVisualization[s_, χ_, temp_,
    equation_, modes_] := Module[{},
    f = equation;
    roots = omegafinder[s, χ, f];
    placeholder = Range[Length[modes]];
    For[i = 1, i ≤ Length[modes], i++,
     placeholder[[i]] = {}];
    repetition[constant_] := Module[{},
      z = Abs[roots[[constant]]];
```

$$\beta_1 = \frac{1}{s\,\chi} + 1;$$

$$\beta_2 = \frac{\left(\frac{z}{s} - 1\right)}{\chi};$$

$$a = \sqrt{\frac{z}{2}\left(\beta_1 + \sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{z}}\right)};$$

$$b = \sqrt{\frac{z}{2}\left(-\beta_1 + \sqrt{\beta_1{}^2 - \frac{4\,\beta_2}{z}}\right)};$$

$$B_1 = \frac{\left(z - a^2\right) A_2}{a};$$

$$B_2 = \frac{\left(a^2 - z\right) A_1}{a};$$

$$B_3 = \frac{\left(z + b^2\right) A_4}{b};$$

```
B4 = (z + b^2) A3 / b;
w[x_] := A1 Sin[a x] + A2 Cos[a x] + A3 Sinh[b x] +
   A4 Cosh[b x];
phi[x_] := B1 Sin[a x] + B2 Cos[a x] + B3 Sinh[b x] +
   B4 Cosh[b x];
(*Identifying the current omega^2 value that
 we are considering*)
conditions = Switch[temp,
   HH, {w[0] == 0, phi'[0] == 0, w[1] == 0},
   HC , {w[0] == 0, phi'[0] == 0, w[1] == 0},
   CC, {w[0] == 0, w'[0] == 0, w[1] == 0},
   CF, {w[0] == 0, w'[0] == 0, phi'[1] == 0},
   FF, {phi'[0] == 0, w'[0] + phi[0] == 0, phi'[1] == 0}];
(**)
eqn1 = conditions[[1]];
eqn2 = conditions[[2]];
eqn3 = conditions[[3]];
eqn4 = Integrate[z w[x]^2 - z phi[x]^2 / s, {x, 0, 1}] == 1;
solution = Solve[eqn1 && eqn2 && eqn3 && eqn4,
   {A1, A2, A3, A4}];
set1 = {Re[solution[[1]][[1]][[2]]],
   Re[solution[[1]][[2]][[2]]],
   Re[solution[[1]][[3]][[2]]],
   Re[solution[[1]][[4]][[2]]]};
set2 = {Re[solution[[2]][[1]][[2]]],
   Re[solution[[2]][[2]][[2]]],
   Re[solution[[2]][[3]][[2]]],
   Re[solution[[2]][[4]][[2]]]};
test = Abs[set1];
cellindex = Position[test, Max[test]];
index = cellindex[[1]][[1]];
If[set1[[index]] == test[[index]], coeff = set1,
 coeff = set2];
δ = (z - a^2) coeff[[2]] / a;
ϵ = (a^2 - z) coeff[[1]] / a;
ϕ = (z + b^2) coeff[[4]] / b;
```

```
      η = ((z + b^2) coeff[[3]]) / b;
      tempw[x_, t_] := Cos[Abs[Sqrt[z]] t]
         (coeff[[1]] Sin[a x] + coeff[[2]] Cos[a x] +
           coeff[[3]] Sinh[b x] + coeff[[4]] Cosh[b x]);
      tempphi[x_, t_] :=
       Cos[Abs[Sqrt[z]] t]
         (δ Sin[a x] + ϵ Cos[a x] + ϕ Sinh[b x] + η Cosh[b x]);
      Return[{{tempw[x, t]},
         {tempphi[x, t]}}]];
    placeholder[[1]] = repetition[modes[[1]]];
    If[Length[modes] > 1,
     For[i = 2, i ≤ Length[modes], i++,
      placeholder[[i]] = repetition[modes[[i]]] +
         placeholder[[i - 1]]]];
    temporary1 = placeholder[[Length[modes]]][[1]][[1]];
    temporary2 = placeholder[[Length[modes]]][[2]][[1]];
    Ω[x_, t_] := Evaluate[temporary1];
    Ψ[x_, t_] := Evaluate[temporary2];
    Manipulate[Module[{},
      graph1 = Plot[Ω[x, t], {x, 0, 1},
         PlotRange → {{0, 1.5}, {-0.5, 0.5}},
         AxesLabel → {"x", "y"}];
      line1 = timcrosssection[{0, Ω[0, t]}, mag Ψ[0, t]];
      line2 = timcrosssection[{0.2, Ω[0.2, t]},
         mag Ψ[0.2, t]];
      line3 = timcrosssection[{0.4, Ω[0.4, t]},
         mag Ψ[0.4, t]];
      line4 = timcrosssection[{0.6, Ω[0.6, t]},
         mag Ψ[0.6, t]];
      line5 = timcrosssection[{0.8, Ω[0.8, t]},
         mag Ψ[0.8, t]];
      line6 = timcrosssection[{1, Ω[1, t]}, mag Ψ[1, t]];
      Show[graph1,
       Graphics[{line1, line2, line3, line4, line5, line6}]]
     ]
     , {t, 0, 16 Pi}, {mag, 1}]
   ];

Manipulate[Module[{},
  Which[
   StaticEulercompute,
```

```
Module[{graphcell}, (*Static Bernoulli-Euler Module*)
 Clear[function];
 function = StaticEulercenterline[Emod, Iarea,
   TotalLength, Lsupport, Rsupport, t, b];
 graphcell = StaticEulerGraphs[function, TotalLength];
 Which[
  stateulerdisplacement, Show[graphcell[[1]]],
  stateulerslope, Show[graphcell[[2]]],
  stateulertheta, Show[graphcell[[3]]],
  stateulermoment, Show[graphcell[[4]]],
  stateulershear, Show[graphcell[[5]]],
  stateulersections, StaticEulerSections[function,
   TotalLength],
  stateulerloadvalues,
  StaticEulerShear[function, TotalLength]
 ]],
VibEulercompute,
Module[{}, (*Vibrational Bernoulli-Euler Module*)
 Clear[function];
 function = VibEulerCenterline[type, mode];
 Which[
  vibeulervisualization,
  VibEulerVisualization[function],
  vibeulerdeformation, VibEulerDeformation[function],
  vibeulershear, VibEulerShear[function]
 ]],
StatTimcompute,
Module[{graphcell}, (*Static Timoshenko Module*)
 Clear[functions, fun1, fun2];
 functions = StaticTimoshenkoCenterline[Emod,
   Gmod, area, k, Iarea, TotalLength, Lsupport,
   Rsupport, t, b];
 fun1[x_] := Evaluate[functions[[1]]];
 fun2[x_] := Evaluate[functions[[2]]];
 graphcell = StaticTimoshenkoGraphs[fun1, fun2,
   TotalLength, Emod, Iarea, k, area, Gmod];
 Which[
  stattimdisplacement, Show[graphcell[[1]]],
  stattimshearingangle, Show[graphcell[[2]]],
  stattimmoment, Show[graphcell[[3]]],
  stattimshear, Show[graphcell[[4]]],
  stattimslope, Show[graphcell[[5]]],
  stattimradius, Show[graphcell[[6]]],
  stattimdeformation,
```

```
      StaticTimoshenkoDeformation [fun1, fun2, TotalLength,
       Emod, Iarea, k, area, Gmod],
      stattimshearload, StaticTimoshenkoShear [fun1,
       fun2, TotalLength, Emod, Iarea, k, area, Gmod]
     ]],
    VibTimcompute,
    Module[{}, (*Vibrational Timoshenko Module*)
     Which[
      HingedHinged, ContourPlotVisualization [s, χ, HHfreq],
      HingedClamped, ContourPlotVisualization [s, χ,
       HCfreq],
      ClampedClamped, ContourPlotVisualization [s, χ,
       CCfreq],
      ClampedFree, ContourPlotVisualization [s, χ, CFfreq],
      FreeFree, ContourPlotVisualization [s, χ, FFfreq]
     ]],
    VibTimcompute2,
    Module[{}, (*Vibrational Timoshenko Module-
      Deformation*)
     Which[
      HingedHinged,
      Quiet[TimoshenkoVibrationVisualization [s, χ,
        HH, HHfreq, modes]],
      HingedClamped,
      Quiet[TimoshenkoVibrationVisualization [s, χ,
        HC, HCfreq, modes]],
      ClampedClamped,
      Quiet[TimoshenkoVibrationVisualization [s, χ,
        CC, CCfreq, modes]],
      ClampedFree,
      Quiet[TimoshenkoVibrationVisualization [s, χ,
        CF, CFfreq, modes]],
      FreeFree,
      Quiet[TimoshenkoVibrationVisualization [s, χ,
        FF, FFfreq, modes]]
     ]]]],
  Item[
   "Please look through whichever drop-down menu best
     suits your needs before proceeding to check
     a compute box; thanks!  (Otherwise the code
     will throw you a bunch of errors due to lack
     of proper variable definitions)", Alignment → Left],
  OpenerView[{"Static Variables",
```

```
Column[
 {Item[
   "For the static case for either beam, please
     input the necessary variables (described
     below their respective boxes). Some
     variables are also marked as for
     Timoshenko only; consequently, if
     you would like to analyze deformation
     given by Timoshenko theory, then you
     would need to assign said variables
     (you can still assign said variables
     but only consider Bernoulli-Euler as
     well, the code won't throw you any
     errors).  Once you have identified
     the proper variables, please feel
     free to check a necessary graph in
     Static-Graphs before proceeding to
     hit the appropriate Compute button
     (StatEulercompute--Bernoulli-Euler
     beams and StatTimCompute--Timoshenko
     beams)", Alignment → Left],
  Control[{Emod}], Item["Elastic Modulus, E (Pa)",
   Alignment → Left],
  Item["----------------------------",
   Alignment → Left],
  Control[{Iarea}],
  Item["Second Moment of Area, I (m^4)",
   Alignment → Left],
  Item["----------------------------",
   Alignment → Left],
  Control[{TotalLength}],
  Item["Beam Length, L (m)", Alignment → Left],
  Item["----------------------------",
   Alignment → Left],
  Control[{Lsupport}],
  Item["Left End Support", Alignment → Left],
  Item["Pick one of the following expressions:
Fixed: Fix
Simple (Pin/Roller): S
Free: Free", Alignment → Left],
  Item["----------------------------",
   Alignment → Left],
```

```
      Control[{Rsupport}],
      Item["Right End Support", Alignment → Left],
      Item["Pick one of the following expressions:
Fixed: Fix
Simple (Pin/Roller): S
Free: Free", Alignment → Left],
      Item["----------------------------",
       Alignment → Left],
      Control[{t}],
      Item[
       "***t=Applied traction forces as a function
          of x (N/m)", Alignment → Left],
      Item[
       "For Syntax purposes, a function of x can be
          defined as:
f[x_]= {function would go here}
Note: You do not have to consider an
          end condition at the end of the beam,
          just input a general function from
          {x|x=[0,inf)}", Alignment → Left],
      Item["----------------------------",
       Alignment → Left],
      Control[{b}],
      Item[
       "***b=Applied body forces as a function of x (N/m)",
       Alignment → Left],
      Item[
       "For Syntax purposes, a function of x can be
          defined as:
f[x_]= {function would go here}
Note: You do not have to consider an
          end condition at the end of the beam,
          just input a general function from
          {x|x=[0,inf)}", Alignment → Left],
      Item["----------------------------",
       Alignment → Left],
      Control[{Gmod}],
      Item[
       "***Gmod = Shear Modulus (Timoshenko only;
          else can leave blank) (Pa)",
```

```
        Alignment → Left],
       Item["----------------------------",
        Alignment → Left],
       Control[{area}],
       Item[
        "***Cross Sectional area of your beam (Timoshenko
          only; else can leave blank) (m^2)"],
       Item["----------------------------",
        Alignment → Left],
       Control[{k}],
       Item[
        "***Timoshenko Shear Coefficient for your
          beam (normal rectangular cross-sections
          = 5/6 (Timoshenko only; else can
          leave blank)"],
       Item["----------------------------",
        Alignment → Left]
      }]}],
   OpenerView[{"Static Graphs",
     Column[{
       OpenerView[{"Static Bernoulli-Euler",
         Column[
          {Item[
            "Make sure that only one graph is checked
              at a time (specifically the
              one you want to see).
 Furthermore, make sure all variables have been
              inputted before checking a
              graph (they won't spawn
              properly otherwise)",
            Alignment → Left], Item["----------------",
            Alignment → Left],
           Control[{stateulerdisplacement, {False, True}}],
           Item[
            "Identifies displacement of the beam
              centerline", Alignment → Left],
           Item["----------------", Alignment → Left],
           Control[{stateulerslope, {False, True}}],
           Item[
            "Identifies the slope of the tangent line
              to the beam centerline at
              all points", Alignment → Left],
           Item["----------------", Alignment → Left],
```

```
            Control[{stateulertheta, {False, True}}],
            Item[
             "Identifies the angle associated with
               the slope of the tangent-line",
             Alignment → Left], Item["----------------",
             Alignment → Left],
            Control[{stateulermoment, {False, True}}],
            Item[
             "Identifies the Internal Moment for the beam",
             Alignment → Left], Item["----------------",
             Alignment → Left],
            Control[{stateulershear, {False, True}}],
            Item[
             "Identifies the Internal Shear for the beam",
             Alignment → Left], Item["----------------",
             Alignment → Left],
            Control[{stateulersections, {False, True}}],
            Item[
             "Permits visualization of the deformed
               centerline in addition to
               various cross-sections
               (user-determined)"],
            Item["----------------", Alignment → Left],
            Control[{stateulerloadvalues, {False, True}}],
            Item[
             "Graphs Shear and Moment diagrams along
               the beam cross-section",
             Alignment → Left], Item["----------------",
             Alignment → Left]
           }]}],
        OpenerView[{"Static Timoshenko",
          Column[
           {Item[
             "Make sure that only one graph is checked
               at a time (specifically the
               one you want to see).
 Furthermore, make sure all variables have been
               inputted before checking a
               graph (they won't spawn
               properly otherwise)",
             Alignment → Left], Item["----------------",
             Alignment → Left],
            Control[{stattimdisplacement, {False, True}}],
            Item[
```

```
 "Identifies the displacement of the beam
   centerline as per Timoshenko
   theory", Alignment → Left],
Item["----------------", Alignment → Left],
Control[{stattimshearingangle , {False, True}}],
Item[
 "Identifies the shear angle of the beam
   cross-section", Alignment → Left],
Item["----------------", Alignment → Left],
Control[{stattimmoment, {False, True}}],
Item[
 "Identifies the internal moments in the beam",
 Alignment → Left], Item["----------------",
 Alignment → Left],
Control[{stattimshear, {False, True}}],
Item["Identifies the internal shear in the beam",
 Alignment → Left], Item["----------------",
 Alignment → Left],
Control[{stattimslope, {False, True}}],
Item[
 "Identifies the slope of the beam centerline",
 Alignment → Left], Item["----------------",
 Alignment → Left],
Control[{stattimradius, {False, True}}],
Item[
 "Identifies the radius of the associated
   osculating circle for the beam",
 Alignment → Left], Item["----------------",
 Alignment → Left],
Control[{stattimdeformation , {False, True}}],
Item[
 "Permits visualization of the deformed
   centerline of the beam along
   with the rotating cross-sections
   (unless your computer has
   a decent amount of processing
   power, I would not recommend
   manipulating any of the
   sliders in the resultant
   output; it will slow down
   your computer considerably.)"],
Item["----------------", Alignment → Left],
Control[{stattimshearload , {False, True}}],
Item[
```

```
            "Identifies the shear/moment values along
              the beam centerline (once
              again, unless your computer
              has a decent amount of
              processing power, I would
              not recommend manipulating
              any of the sliders in the
              resultant output; it will
              slow down your computer
              considerably.)"],
           Item["----------------", Alignment → Left]
          }]}]}]
  }],
  OpenerView[{"Vibrational Bernoulli-Euler",
    Column[
     {Item[
       "Please make sure that you input the mode and
         type first before proceeding to check
         the VibEulerCompute box; thanks!",
       Alignment → Left],
      Control[{type}],
      Item[
       "***Possible inputs include: FreeFree,
         FixedFixed, FixedFree, FixedPinned,
         PinnedPinned"],
      Item["----------------", Alignment → Left],
      Control[{mode}],
      Item[
       "***What vibrational modes you are attempting
         to visualize.  Note, currently only
         modes up through 10 are available for each.
Just input the number mode (1-10) that you are
         attempting to visualize.  Also, we
         are currently unable to visualize
         the FreeFree Rigid Body mode of 0"],
      Item["----------------", Alignment → Left],
      Control[{vibeulervisualization, {False, True}}],
      Item["Shows the vibrating centerline",
       Alignment → Left],
      Item["----------------", Alignment → Left],
      Control[{vibeulerdeformation, {False, True}}],
      Item[
       "Shows the vibrating centerline accompanied
         with the cross-section as well",
```

```
        Alignment → Left],
       Item["----------------", Alignment → Left],
       Control[{vibeulershear, {False, True}}],
       Item[
        "Shows the varying Shear/Moment along the
          beam centerline", Alignment → Left],
       Item["----------------", Alignment → Left]
      }]}],
  OpenerView[
   {"Vibrational Timoshenko-Frequency Visualization",
    Column[
     {Item[
        "Note that for these you should check the
          VibTimcompute box; thanks.
Also, choose the condition first before inputting
          the constants as required.  Furthermore,
          for the one you would like to fix,
          please input 0; for the others you
          may input whatever you like."],
       Control[{HingedHinged, {False, True}}],
       Control[{HingedClamped, {False, True}}],
       Control[{ClampedClamped, {False, True}}],
       Control[{ClampedFree, {False, True}}],
       Control[{FreeFree, {False, True}}],
       Control[{s}],
       Control[{χ}]
      }]}],
  OpenerView[{"Vibrational Timoshenko-Deformation",
    Column[
     {Item[
        "Note that for these you should check the
          VibTimcompute2 box; thanks.
Also, choose the condition first before inputting
          the constants as required."],
       Control[{HingedHinged, {False, True}}],
       Control[{HingedClamped, {False, True}}],
       Control[{ClampedClamped, {False, True}}],
       Control[{ClampedFree, {False, True}}],
       Control[{FreeFree, {False, True}}],
       Control[{s}],
       Control[{χ}],
       Control[{modes}]
      }]}],
  {StaticEulercompute, {False, True}},
```

```
{VibEulercompute , {False, True}},
{StatTimcompute , {False, True}},
{VibTimcompute , {False, True}},
{VibTimcompute2 , {False, True}},
ControlPlacement → Left]
```

Acknowledgments

The work of Evan Hemingway was supported by a Berkeley Fellowship from the University of California at Berkeley and a U.S. National Science Foundation Graduate Research Fellowship.

References

[1] A. Authorlast, "Article Title," *Journal Title*, **Volume**(Issue), 2005 pp. #–#. Linked URL or doi number if available, without http:// preceding.

[2] A. Authorlast1 and B. Authorlast2, "Article Title," *Journal Title*, **Volume**(Issue), 2005 pp. #–#. Linked URL or doi number if available, without http:// preceding.

[3] A. Authorlast1, B. Authorlast2, and C. Authorlast3, "Article Title," *Journal Title*, **Volume**(Issue), 2005 pp. #–#. Linked URL or doi number if available, without http:// preceding.

[4] A. Authorlast, *Book Title*, *n*th ed., Publisher Location: Publisher Name, 2005.

[5] A. Authorlast1 and B. Authorlast2, *Book Title*, *n*th ed., Publisher Location: Publisher Name, 2005.

[6] A. Authorlast1, B. Authorlast2, and C. Authorlast3, *Book Title*, *n*th ed., Publisher Location: Publisher Name, 2005.

[7] A. Authorlast, "Paper Title," in *Conference Proceedings Title (Conference Acronym and Year)*, Conference Location (A. Authorlast, ed.), Publisher Location: Publisher Name, Publication Date pp. #–#. Linked URL or doi number if available, without http:// preceding.

[8] A. Authorlast1, B. Authorlast2, and C. Authorlast3, "Paper Title," in *Conference Proceedings Title (Conference Acronym and Year)*, Conference Location (A. Authorlast, ed.), Publisher Location: Publisher Name, Publication Date pp. #–#. Linked URL or doi number if available, without http:// preceding.

[9] A. Authorlast1, B. Authorlast2, and C. Authorlast3, "Paper Title," in *Conference Proceedings Title (Conference Acronym and Year)*, Conference Location (A. Authorlast, ed.), Publisher Location: Publisher Name, Publication Date pp. #–#. Linked URL or doi number if available, without http:// preceding.

[10] A. Authorlast. "Website Title." (Last updated date or date visited in three-character Month Day, Year format) Linked URL or doi number if available, without http:// preceding.

[11] E. W. Weisstein. "Entry Title" from Wolfram *MathWorld*—A Wolfram Web Resource. Linked URL, without http:// preceding. [Wolfram resource format]

[12] *Software Title*. Version number, Publisher Location: Publisher Name, Publication Date.

About the Authors

Prithvi Akella
Department of Mechanical Engineering
University of California at Berkeley,
Berkeley, CA 94720-1740
USA
Email: prithviakella@berkeley.edu

Evan Hemingway
Department of Mechanical Engineering
University of California at Berkeley,
Berkeley, CA 94720-1740
USA
Email: evanhem@berkeley.edu

Oliver M O'Reilly
Department of Mechanical Engineering
University of California at Berkeley,
Berkeley, CA 94720-1740
USA
Tel: 510-642-0877
Fax: 510-643-5599
Email: oreilly@berkeley.edu