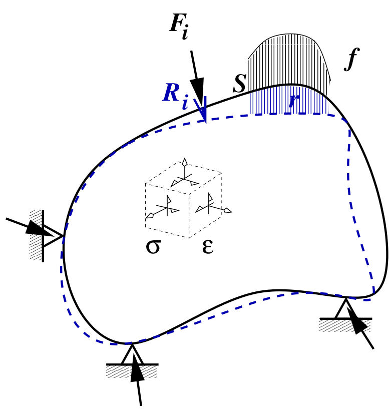
Principle of Virtual Displacements in Structural Dynamics

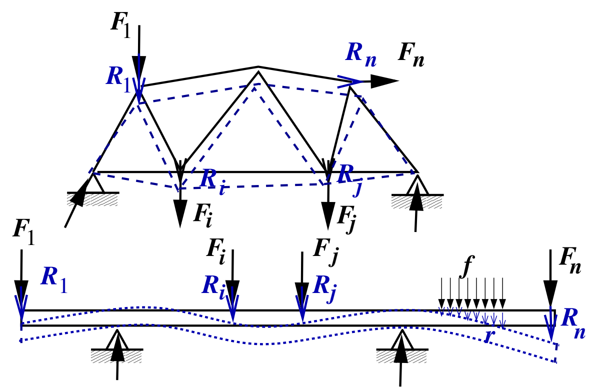
CEE 541. Structural Dynamics
Department of Civil and Environmental Engineering
Duke University

Henri P. Gavin
Fall 2020

1 Strain Energy in Elastic Solids

Consider an elastic object in equilibrium subjected to static forces and displacements.





- F and f are real external forces in equilibrium, acting at points, or over a portion of a surface S,
- R and r are real displacements, admissible with respect to the support conditions, collocated with F and f,
- $\boldsymbol{\sigma}$ are real internal stresses, distributed within the solid volume V, in equilibrium with F & f,
- $\boldsymbol{\epsilon}$ are real internal strains, distributed within the solid volume V, compatible with R and r,

1.1 External Work

The work of external forces increasing from 0 to F and f and pushing through displacements from 0 to R and r is

$$W = \int_0^R F(R')\, dR' + \int_S \int_0^r f(r')\, dr'\, dS \tag{1}$$

where

- the forces F and f depend on displacements R and r
- R' and r' are dummy variables of integration

1.2 Internal Strain Energy

Strain energy is a kind of potential energy arising from stress *and* deformation of elastic solids. In nonlinear elastic solids, the strain energy of stresses increasing from 0 to $\boldsymbol{\sigma}$ and working through strains from 0 to $\boldsymbol{\epsilon}$ is

$$U = \int_V \int_0^{\boldsymbol{\epsilon}} \boldsymbol{\sigma} \cdot d\boldsymbol{\epsilon}' \; dV \tag{2}$$

where

- V is the volume of the solid
- $\boldsymbol{\sigma} = \left\{ \sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \tau_{xy} \quad \tau_{yz} \quad \tau_{xz} \right\}$
- $\boldsymbol{\epsilon} = \left\{ \epsilon_{xx} \quad \epsilon_{yy} \quad \epsilon_{zz} \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{xz} \right\}$
- $\boldsymbol{\epsilon}'$ is a dummy variable of integration

1.3 The Principle of Real Work

In an elastic solid, the work of external forces, W, is stored entirely as *elastic strain energy*, U, within the solid.

$$U = W \tag{3}$$

In *linear* elastic solids:

- Stresses $\boldsymbol{\sigma}$ increase linearly with strains $\boldsymbol{\epsilon}$,

$$\sigma = E\epsilon \quad \ldots \quad \text{and} \quad \ldots \quad \tau = G\gamma$$

- Displacements D and rotations Θ increase linearly with forces F and moments M,

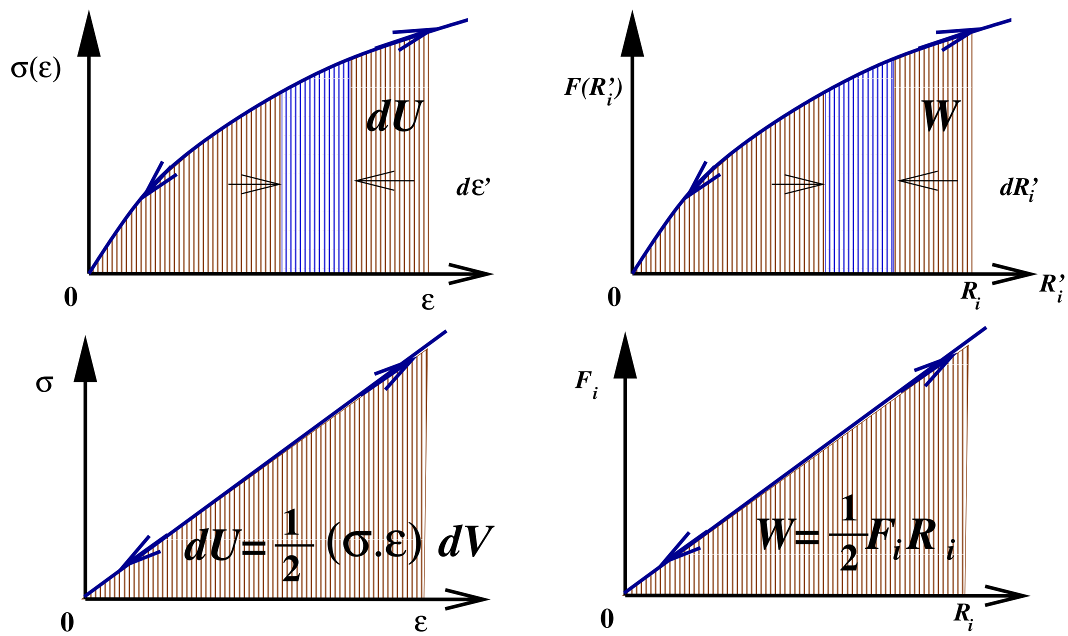
$$F = kD \quad \ldots \quad \text{and} \quad \ldots \quad M = \kappa\Theta$$

- The work of an external force F acting through a displacement D on the solid is

$$W = \frac{1}{2}FD = \frac{1}{2}kD^2 = \frac{1}{2}F^2/k$$

- The work of an external moment M acting through a rotation Θ on the solid is

$$W = \frac{1}{2}M\Theta = \frac{1}{2}\kappa\Theta^2 = \frac{1}{2}M^2/\kappa$$



1.4 Strain energy in slender structural elements

In slender structural elements (bars, beams, or shafts) the internal forces, moments, shears, and torques can vary along the length of each element; so do the displacements and rotations.

The strain energy of spatially-varying internal forces $F(x)$ acting through spatially-varying internal displacements $D(x)$ along a linear elastic prismatic solids is

$$U = \frac{1}{2} \int_l F(x) \cdot \frac{dD(x)}{dx} \, dx = \frac{1}{2} \int_l F(x) D'(x) \, dx \tag{4}$$

The strain energy of spatially-varying internal moments $M(x)$ acting through spatially-varying internal rotations $\Theta(x)$ along linear elastic prismatic solids is

$$U = \frac{1}{2} \int_l M(x) \cdot \frac{d\Theta(x)}{dx} \, dx = \frac{1}{2} \int_l M(x) \Theta'(x) \, dx \tag{5}$$

In slender structural elements, the relation between internal forces and moments and internal displacements and rotations depend on the kind of loading.

- Axial $N_x(x) = E(x)A(x)u'(x)$
- Bending $M_z(x) = E(x)I(x)v''(x)$
- Shear $V_y(x) = G(x)A_{\rm s}(x)v_{\rm s}'(x)$
- Torsion $T_x(x) = G(x)J_(x)\phi'(x)$

Inserting these expressions into the general expressions for internal strain energy above,

	"force"	deformation	strain energy (U)		
Axial	$N_x(x)$	$u'(x)$	$\frac{1}{2}\int_l N_x(x)u'(x)dx$	$\frac{1}{2}\int_l \frac{N_x(x)^2}{E(x)A(x)}dx$	$\frac{1}{2}\int_l E(x)A(x)(u'(x))^2dx$
Bending	$M_z(x)$	$v''(x)$	$\frac{1}{2}\int_l M_z(x)v''(x)dx$	$\frac{1}{2}\int_l \frac{M_z(x)^2}{E(x)I(x)}dx$	$\frac{1}{2}\int_l E(x)I(x)(v''(x))^2dx$
Shear	$V_y(x)$	$v'_s(x)$	$\frac{1}{2}\int_l V_y(x)v'_s(x)dx$	$\frac{1}{2}\int_l \frac{V_y(x)^2}{G(x)A_s(x)}dx$	$\frac{1}{2}\int_l G(x)A_s(x)\left(v'_s(x)\right)^2dx$
Torsion	$T_x(x)$	$\phi'(x)$	$\frac{1}{2}\int_l T_x(x)\phi'(x)dx$	$\frac{1}{2}\int_l \frac{T_x(x)^2}{G(x)J(x)}dx$	$\frac{1}{2}\int_l G(x)J(x)(\phi'(x))^2dx$

$E(x)$ is Young's modulus
$G(x)$ is the shear modulus

$A(x)$ is the cross sectional area of a bar
$I(x)$ is the bending moment of inertia of a beam
$A(x)/\alpha$ is the effective shear area of a beam
$J(x)$ is the torsional moment of inertia of a shaft

$N_x(x)$ is the axial force within a bar
$M_z(x)$ is the bending moment within a beam
$V_y(x)$ is the shear force within a beam
$T_x(x)$ is the torque within a shaft

$u(x)$ is the axial displacement along the bar
$u'(x)$ is the axial displacement per unit length, $du(x)/dx$, the axial strain

$v(x)$ is the transverse bending displacement of the beam
$v'(x)$ is the slope of the displacement of the beam
$v''(x)$ is the rotation per unit length, the curvature, approximately $d^2v(x)/dx^2$

$v_s(x)$ is the transverse shear displacement of the beam
$v'_s(x)$ is the transverse shear displacement per unit length, $dv_s(x)/dx$

$\phi(x)$ is the torsional rotation (twist) of the shaft
$\phi'(x)$ is the torsional rotation per unit length, $d\phi(x)/dx$

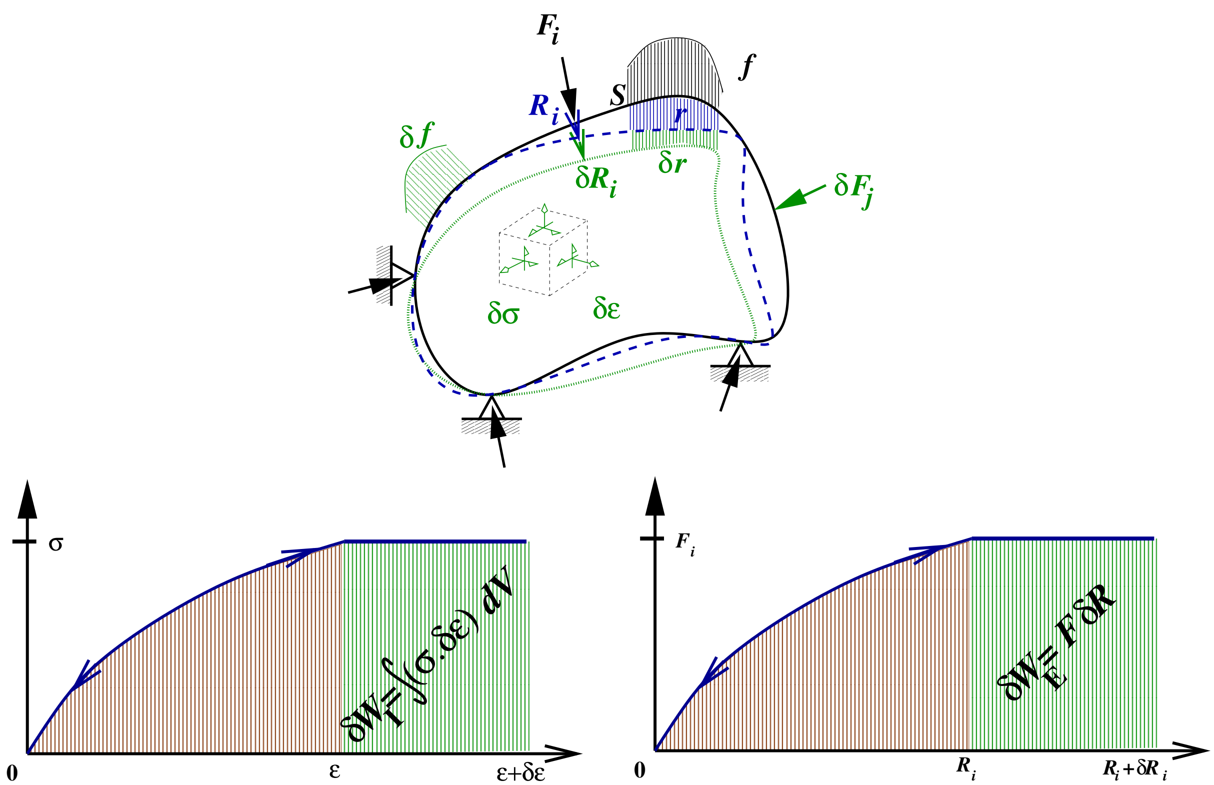
2 Virtual Work in Elastic Solids — The Principle of Virtual Displacements

Now consider a second set of loads, δF, δf, in equilibrium and applied *subsequently* to the loads F and f. The loads δF and δf give rise to displacements δR and δr collocated with forces F and f, and internal stresses $\delta \boldsymbol{\sigma}$ and strains $\delta \boldsymbol{\epsilon}$. In other words, the displacements δR and δr are *admissible* to the kinematic constraints.

Call δF and δf a set of any arbitrary "*virtual*" forces in equilibrium.

Call δR and δr a set of "*virtual*" displacements, collocated with forces F and f, and resulting from forces δF and δf (and therefore kinematically admissible). The displacements δR and δr may also be called *variations of displacements*, admissible to the constraints.

Forces F and f are held constant as loads δF and δf are applied. Stresses $\boldsymbol{\sigma}$, in equilibrium with forces F and f, are therefore also held constant as loads δF and δf are applied. Forces F and f do not increase with displacements δR and δr. Strains $\delta \boldsymbol{\epsilon}$ increase as loads δF and δf are applied.



The *principle of virtual displacements* states that the virtual external work of real external forces (f and F) moving through collocated admissible virtual displacements (δr and δR) equals the internal virtual work of real stresses ($\boldsymbol{\sigma}$) in equilibrium with real forces (f and F) with the virtual strains ($\delta \boldsymbol{\epsilon}$) compatible with the virtual displacements (δr and δR), integrated over the volume of the solid.

$$\begin{aligned} \delta W_{\mathrm{I}} &= \delta W_{\mathrm{E}} \\ \int_V \boldsymbol{\sigma} \cdot \delta \boldsymbol{\epsilon} \; dV &= \int_S f \cdot \delta r \; dS + \sum_i F_i \cdot \delta R_i \end{aligned} \qquad (6)$$

2.1 Work of axial loads and transverse displacements in slender structural elements

In slender solid elements, nonuniform transverse displacements ($dv(x) \neq 0$) induce longitudinal shortening, $de(x)$.

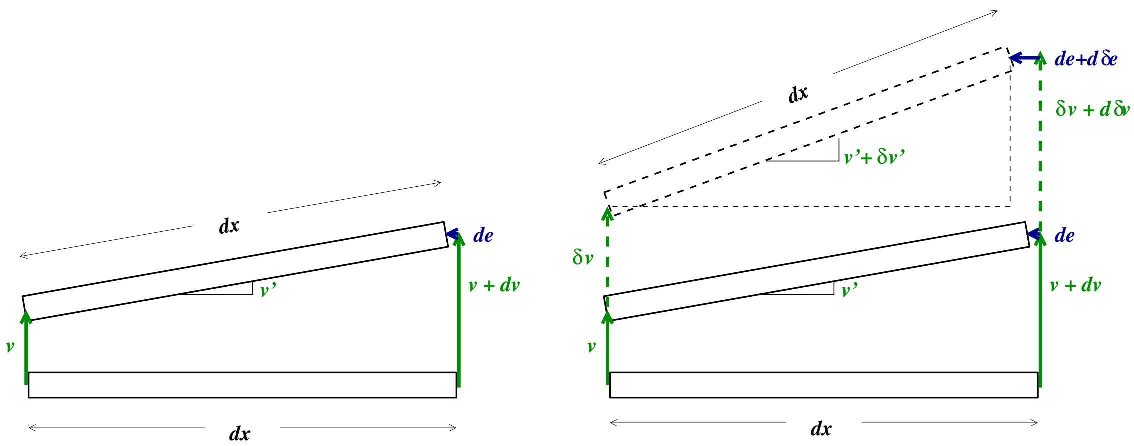


Figure 1. Transverse deformation $v'(x)$ and longitudinal shortening $de(x)$.

A relation between dv and de can be derived from the Pythagorean theorem and is quadratic in dv and de.

$$
\begin{aligned}
(dx - de)^2 + (dv)^2 &= (dx)^2 \\
2(de)(dx) - (de)^2 &= (dv)^2 \\
\frac{de}{dx} &\approx \frac{1}{2}\left(\frac{dv}{dx}\right)^2 = \frac{1}{2}(v')^2
\end{aligned}
$$

With additional virtual displacements $\delta v(x)$ a relation for the incremental virtual shortening $d\delta e$ may also be derived from the Pythagorean theorem.

$$
\begin{aligned}
(dx - de - d\delta e)^2 + (dv + d\delta v)^2 &= (dx)^2 \\
2(de)(dx) - 2(de)(d\delta e) + 2(d\delta e)(dx) - (de)^2 - (d\delta e)^2 &= (dv)^2 + 2(dv)(d\delta v) + (d\delta v)^2
\end{aligned}
$$

Subtracting $2(de)(dx) - (de)^2 = (dv)^2$ and dividing by $(dx)^2$ leaves

$$
-2\frac{de}{dx}\frac{d\delta e}{dx} + 2\frac{d\delta e}{dx} - \left(\frac{d\delta e}{dx}\right)^2 = 2(v')(\delta v') + (\delta v')^2
$$

Neglecting higher order terms (assuming virtual displacements are infinitesimal), leaves

$$
\frac{d\delta e}{dx} \approx (v')(\delta v') \tag{7}
$$

The virtual work of a distributed axial compression $P(x)$ (*applied externally*, for example, by gravitational acceleration) acting through virtual shortening displacements $\delta e(x)$ integrated along a slender element is, then,

$$
\delta W_G = \int_l P(x)\frac{d\delta e}{dx}dx = \int_l P(x)\; v'(x)\; \delta v'(x)\; dx \tag{8}
$$

This result can also be obtained by integrating along the arc-length of the deformed element as is done in Tedesco, McDougal, and Ross's textbook, *Structural Dynamics: Theory and Applications*.

3 The Principle of Virtual Displacements for Dynamic Loading

The principle of virtual displacements applies to both static and dynamic forces. Elastic forces $k(x)r(x,t)$ are present in structural systems responding to static or dynamic loads. Forces arising from dynamic effects only include viscous damping forces $c(x)\dot{r}(x,t)$ and inertial forces $m(x)\ddot{r}(x,t)$. Elastic forces, viscous damping forces, and inertial forces can be developed within slender structural elements in response to axial, bending, shear, and torsional deformations.

	real "force"	virtual deformation	internal virtual work ($\delta W_{\rm I}$)	
Axial	$N_x(x,t)$	$\delta u'(x,t)$	$\int_l N_x(x,t)\ \delta u'(x,t)\ dx$	$\int_l EA(x)\ u'(x,t)\ \delta u'(x,t)\ dx$ $\int_l \eta_a A(x)\ \dot{u}'(x,t)\ \delta u'(x,t)\ dx$ $\int_l \rho A(x)\ \ddot{u}(x,t)\ \delta u(x,t)\ dx$
Bending	$M_z(x,t)$	$\delta v''(x,t)$	$\int_l M_z(x,t)\ \delta v''(x,t)\ dx$	$\int_l EI(x)\ v''(x,t)\ \delta v''(x,t)\ dx$ $\int_l \eta_a I(x)\ \dot{v}''(x,t)\ \delta v''(x,t)\ dx$ $\int_l \rho A(x)\ \ddot{v}(x,t)\ \delta v(x,t)\ dx$
Shear	$V_y(x,t)$	$\delta v'_{\rm s}(x,t)$	$\int_l V_y(x,t)\ \delta v'_{\rm s}(x,t)\ dx$	$\int_l GA_{\rm s}(x)\ v'_{\rm s}(x,t)\ \delta v'_{\rm s}(x,t)\ dx$ $\int_l \eta_s A_{\rm s}(x)\ \dot{v}'_{\rm s}(x,t)\ \delta v'_{\rm s}(x,t)\ dx$ $\int_l \rho A(x)\ \ddot{v}(x,t)\ \delta v(x,t)\ dx$
Torsion	$T_x(x,t)$	$\delta\phi'(x,t)$	$\int_l T_x(x,t)\ \delta\phi'(x,t)\ dx$	$\int_l GJ(x)\ \phi'(x,t)\ \delta\phi'(x,t)\ dx$ $\int_l \eta_s J(x)\ \dot{\phi}'(x,t)\ \delta\phi'(x,t)\ dx$ $\int_l \rho J(x)\ \ddot{\phi}(x,t)\ \delta\phi(x,t)\ dx$
Geometric	$P(x)$	$\delta e(x,t)$	$\int_l P(x)\ \delta e(x,t)\ dx$	$\int_l P(x)\ v'(x,t)\ \delta v'(x,t)\ dx$

In this table:

- The internal virtual work of viscous effects is derived assuming linear viscous stress - strain-rate relations: $\sigma = \eta_a \dot{\epsilon}$ and $\tau = \eta_s \dot{\gamma}$. As will be seen later in the course, the damping properties of real structural materials are actually more complicated.
- Rotatory inertia effects are neglected in the virtual work of inertial forces in bending beams.

4 Generalized Coordinates

A dynamic response $r(x,t)$ may be represented as an expansion of products of spatially dependent quantities and time dependent quantities

$$r(x,t) = \sum_k \psi_k(x)\ q_k(t) \tag{9}$$

The functions $\psi_k(x)$ are called *shape-functions*, and the functions $q(t)$ may be called *generalized coordinates*. In order for the above expansion to yield realistic and accurate solutions, the shape functions must at least satisfy the essential boundary conditions. (The shape functions must be *kinematically-admissible*.) Shape functions which also satisfy the natural boundary conditions will yield more accurate solutions. Also, if the shape functions are dimensionless, the generalized coordinates have the same units as the response, which permits a useful interpretation of the generalized coordinates. Further, if the shape functions are kinematically admissible, and the expansion (9) for r is expressed in terms of q, but not $\dot{q}$, then virtual displacements defined as variations in $r(x,t)$ with respect to the set of coordinates $q_k(t)$ are also kinematically admissible

$$\delta r(x,t) = \sum_j \frac{\partial r(x,t)}{\partial q_j(t)}\ \delta q_j(t) = \sum_j \psi_j(x)\ \delta q_j(t)\ ,$$

and the derivatives of r with respect to x and t are

$$r(x,t) = \textstyle\sum_k q_k(t)\psi_k(x) \qquad \dot{r}(x,t) = \sum_k \dot{q}_k(t)\psi_k(x) \qquad \ddot{r}(x,t) = \sum_k \ddot{q}_k(t)\psi_k(x)$$

$$r'(x,t) = \textstyle\sum_k q_k(t)\psi_k'(x) \qquad \dot{r}'(x,t) = \sum_k \dot{q}_k(t)\psi_k'(x) \qquad \ddot{r}'(x,t) = \sum_k \ddot{q}_k(t)\psi_k'(x)$$

$$r''(x,t) = \textstyle\sum_k q_k(t)\psi_k''(x) \qquad \dot{r}''(x,t) = \sum_k \dot{q}_k(t)\psi_k''(x) \qquad \ddot{r}''(x,t) = \sum_k \ddot{q}_k(t)\psi_k''(x)$$

Internal virtual work can also be expressed in terms of generalized virtual displacements. For example in the elastic bending of a beam, the work of moments $(EIv''(x,t))$ moving through virtual rotations $(\delta v''(x,t)\ dx)$ in terms of generalized coordinate displacements $q_k(t)$ and virtual displacements $\delta q_j(t)$ is

$$\begin{aligned}
\delta W_{\mathrm{I}} &= \int_l EI(x)\ v''(x,t)\ \delta v''(x,t)\ dx \\
&= \int_l EI(x)\ \sum_k \psi_k''(x)\ q_k(t)\ \sum_j \psi_j''(x)\delta q_j(t)\ dx \\
&= \sum_j \sum_k \left[\int_l EI(x)\ \psi_j''(x)\ \psi_k''(x)\ dx\right]\ q_k(t)\ \delta q_j(t)
\end{aligned} \tag{10}$$

The work of transverse inertial forces $(\rho A\ddot{v}(x,t)\ dx)$ moving through virtual displacements $(\delta v(x,t))$ in terms of generalized coordinate accelerations $\ddot{q}_k(t)$ and virtual displacements $\delta q_j(t)$ is

$$\begin{aligned}
\delta W_{\mathrm{I}} &= \int_l \rho A(x)\ \ddot{v}(x,t)\ \delta v(x,t)\ dx \\
&= \int_l \rho A(x)\ \sum_k \psi_k(x)\ddot{q}_k(t)\ \sum_j \psi_j(x)\delta q_j(t)\ dx \\
&= \sum_j \sum_k \left[\int_l \rho A(x)\ \psi_j(x)\ \psi_k(x)\ dx\right]\ddot{q}_k(t)\ \delta q_j(t)
\end{aligned} \tag{11}$$

The work of external forces $f(x,t)$ and $F(t)$ moving through collocated virtual displacements $\delta v(x,t)$ can be expressed in terms of virtual displacements of generalized coordinates, $\delta q_j(t)$.

$$\begin{aligned}\delta W_{\rm E} &= \int_l f(x,t)\cdot \delta v(x,t)\ dx + \sum_i F_i \cdot \delta v(x,t_i) \\ &= \int_l f(x,t)\cdot \sum_j \psi_j(x)\ \delta q_j(t)\ dx + \sum_i F_i \cdot \sum_j \psi_j(x_i)\ \delta q_j(t) \\ &= \sum_j \left[\int_l f(x,t)\cdot \psi_j(x)\ dx\right] \delta q_j(t) + \sum_j \left[\sum_i F_i \cdot \psi_j(x_i)\right] \delta q_j(t) \end{aligned} \qquad (12)$$

The external virtual work of axial compression $P(x)$ moving through virtual end shortening $(v'(x,t)\delta v'(x,t)\ dx)$ in terms of generalized coordinate displacements $q_k(t)$ and virtual displacements $\delta q_j(t)$ is

$$\begin{aligned}\delta W_{\rm E} &= \int_l P(x)\cdot v'(x,t)\ \delta v'(x,t)\ dx \\ &= \int_l P(x)\cdot \sum_k \psi_k'(x)\ q_k(t) \sum_j \psi_j'(x)\ \delta q_j(t)\ dx \\ &= \sum_j \sum_k \left[\int_l P(x)\ \psi_j'(x)\ \psi_k'(x)\ dx\right] q_k(t)\ \delta q_j(t) \end{aligned} \qquad (13)$$

By setting the internal virtual work equal to the external virtual work, and factoring out the independent and arbitrary variations δq_j, equations (10), (11), (12), and (13), result in

$$\Big(\ [M]\ddot{q}(t) + [K_{\rm E}]q(t) - [K_{\rm G}]q(t) - f(t)\ \Big)\cdot \Big(\delta q(t)\Big) = 0$$

Noting that each variation $\psi_j \delta q_j$ is be arbitrary, and the set of variations $j = 1, 2, ...$ must be independent, not only must the dot product equal zero, but each term within the inner product must be zero. Therefore, the term on the left of the inner product must evaluate to the zero-vector. This is an important concept in the principle of virtual work and in the calculus of variations. It's application results in the matrix equations of motion,

$$[M]\ \ddot{q}(t) + [K_{\rm E}]\ q(t) - [K_{\rm G}]\ q(t) = f(t)$$

where the j, k term of the mass matrix is,

$$M_{jk} = \int_l \rho A(x)\ \psi_j(x)\ \psi_k(x)\ dx$$

the j, k term of the elastic stiffness matrix is,

$$K_{{\rm E}jk} = \int_l EI(x)\ \psi_j''(x)\ \psi_k''(x)\ dx$$

the j, k term of the geometric stiffness matrix is,

$$K_{{\rm G}jk} = \int_l P(x)\ \psi_j'(x)\ \psi_k'(x)\ dx$$

and the j-th element of the forcing vector is the inner product of the forcing with the j-th shape function,

$$f_j = \int_l f(x)\cdot \psi_j(x)\ dx + \sum_i F_i \cdot \psi_j(x_i)$$

From the above relations, it is clear that $M_{ij} = M_{ji}$ (the mass matrix is symmetric), $K_{ij} = K_{ji}$ (the stiffness matrices is symmetric), and that $[M]$ and $[K]$ are positive definite, provided that the set of shape functions are linearly independent.

5 Choice of Shape Function

In the set of shape functions described by

$$\psi_j(x) = \sin\left(\frac{2j-1}{2}\pi x\right)$$

$\psi_j(0) = 0$ and $\psi'(1) = 0$. The figure below shows a set of the first four shape functions ($j \in (1, 2, 3, 4)$) on the left and the set of the first seven shape functions on the right ($j \in (1, 2, 3, 4, 5, 6, 7)$).

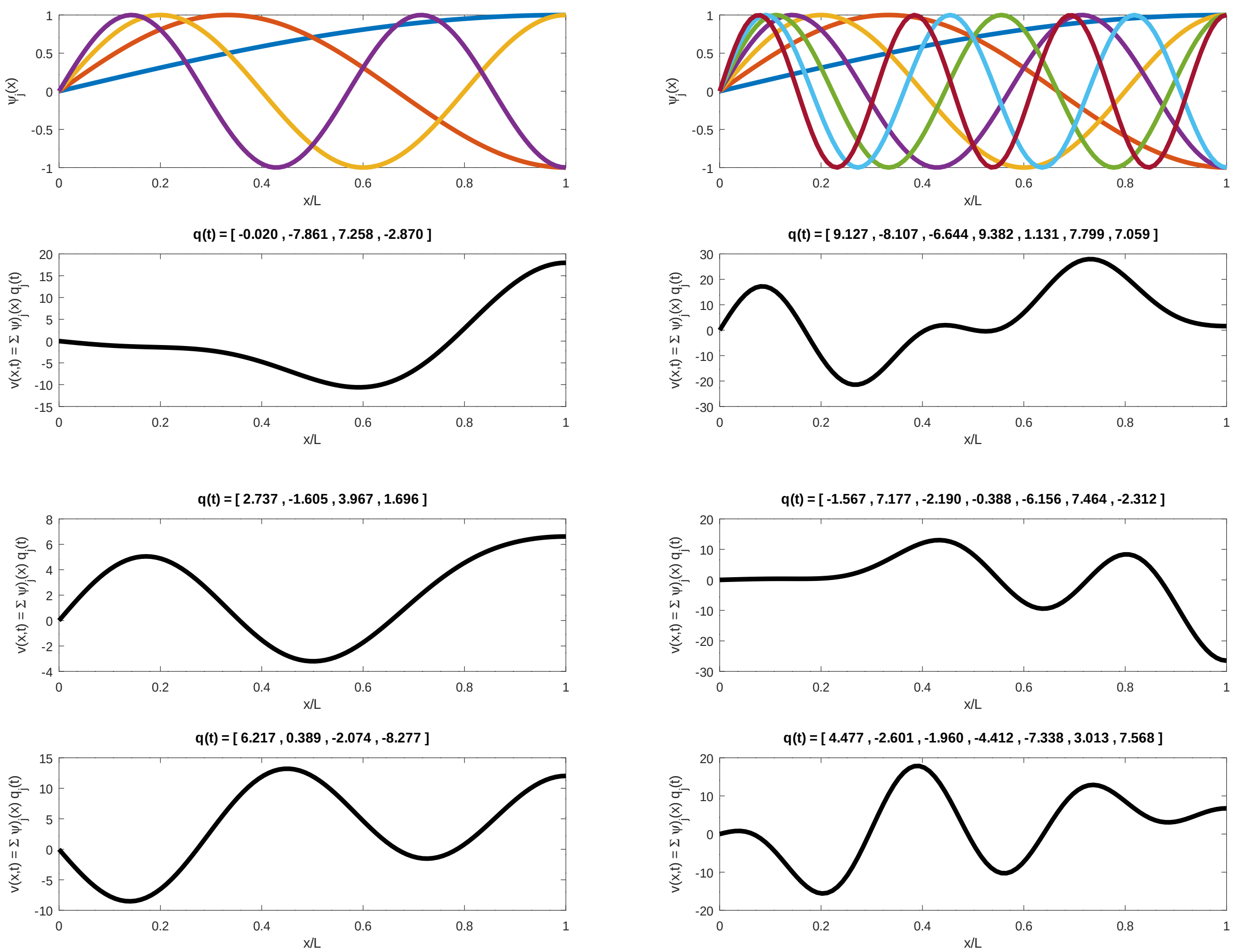
The curves in black show examples of weighted sums of the basis functions.

$$v(x,t) = q_1(t)\psi_1(x) + q_2(t)\psi_2(x) + ... + q_N(t)\psi_N(x)$$

in which the "weights" correspond to the time-dependent generalized coordinates $q_j(t)$. So one may think of the black curves as snapshots of vibrational shapes taken at various instances in time.

Note that since all the shape functions satisfy $\psi_j(0) = 0$ and $\psi'(1) = 0$, then so must the weighted sum of those shape functions.

Note also that the use of a larger set of shape functions permits more complicated vibrational shapes.



It is *essential* that selected shape functions is kinematically admissible with respect to the essential boundary conditions (the structural supports). The analytically "correct" shape function is both kinematically admissible and *satisfies equilibrium.* Equations of motion resulting from the use of kinematically admissible shape functions that do not satisfy equilibrium will provide approximate solutions, which, in many cases are within the errors implied by other fundamental assumptions.

The true equations of motion for a particular system are unique. The principle of virtual displacements provides a means to derive approximate equations of motion. The accuracy of the PVD approximation depends on the set of shape functions used, $[\psi_1(x), ..., \psi_N(x)]$.

Since $v(x, t)$ has units of length, if $\psi(x)$ is unitless, then the coordinate $q(t)$ must have a unit of length, and if $\psi(x)$ has units of length, then $q(t)$ is unitless (like a rotation).

6 Examples

6.1 Example 1: a single generalized coordinate, choice of two shape functions

Consider the vibration of a cantilever beam with a *point* end-mass (assuming that the rotatory inertia of the end mass is negligible). And consider the choice between two similar shape functions,

$$\psi(x) = \frac{3}{2}\left(\frac{x}{L}\right)^2 - \frac{1}{2}\left(\frac{x}{L}\right)^3 \qquad \text{or} \qquad \psi(x) = 1 - \cos\left(\frac{\pi}{2}\frac{x}{L}\right)$$

The cubic shape function is the *static* displacement of a cantilever beam with a concentrated tip load, which would seem like a reasonable guess for the deformed shape in this problem. Note that the static displacements satisfy equilibrium for a static load; they do not necessarily satisfy equilibrium for a dynamic load.

The (1-cosine) shape function is a reasonable guess, since it is smooth and satisfies the essential boundary conditions. But the (1-cosine) does not satisfy internal equilibrium for static or dynamic loads.

Regardless of the choice of shape function, the internal virtual work is the work of inertial forces moving through collocated virtual displacements plus the work of internal bending moments moving through virtual rotations.

$$\begin{aligned}
\delta W_{\text{INT}} &= \int_0^L m\ddot{v}(x,t)\ \delta v(x,t)\ dx + M\ddot{v}(L,t)\ \delta v(L,t) + \int_0^L EIv''(x,t)\ \delta v''(x,t)\ dx \\
&= m\int_0^L (\psi(x))^2 dx\ \ddot{q}(t)\ \delta q(t) + M(\psi(L))^2\ \ddot{q}(t)\ \delta q(t) + EI\int_0^L (\psi''(x)^2)\ dx\ q(t)\ \delta q(t)
\end{aligned}$$

and the work of the external force moving through its collocated virtual rotation is

$$\delta W_{\text{EXT}} = F(t)\delta v(L,t) = F(t)\ \psi(L)\ \delta q(t)$$

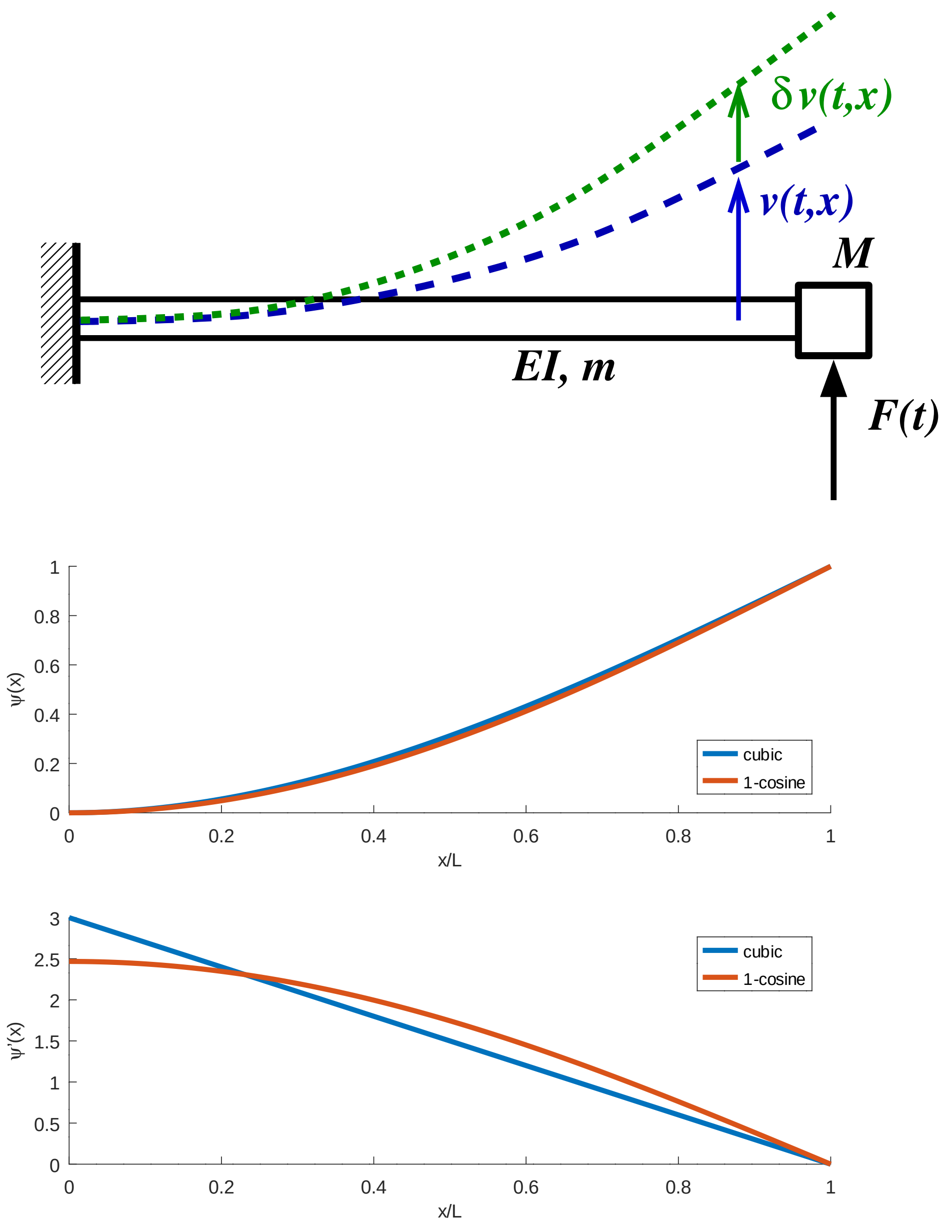


Figure 2. Two similar shape functions assumed for the displaced shape of a vibrating cantilever beam with an end mass. Differences are clearer in the curvature $\psi''(x)$ (bending moments $M(x) = EI\psi''(x)$) of the system. The cubic shape function corresponds to the triangular-shaped bending moment of a cantilever beam with a *static* end-load.

Setting $\delta W_{\rm INT} = \delta W_{\rm EXT}$, factoring out the arbitrary virtual coordinate $\delta q(t)$, and solving the integrals gives for each of the candidate shape functions gives, for the cubic shape function,

$$\left(\frac{33}{140} mL + M\right) \ddot{q}(t) + \frac{3EI}{L^3} q(t) = F(t)$$

and for the (1-cosine) shape function

$$\left(\frac{3\pi - 8}{2\pi} mL + M\right) \ddot{q}(t) + \frac{\pi^4 EI}{32 L^3} q(t) = F(t)$$

with natural frequency for the cubic shape function

$$\omega_{\rm n} = \sqrt{\frac{3EI/L^3}{33mL/140 + M}}$$

and for the (1-cosine) shape function

$$\omega_{\rm n} = \sqrt{\frac{\pi^4 \; EI/(32L^3)}{(3\pi - 8)mL/(2\pi) + M}}$$

The shape function giving the lower natural frequency is more accurate.

As shown in the figure below, the cubic shape function gives a slightly more accurate dynamic model as compared to the (1-cosine) function, by about one percent for mass ratios from 0 to 5, which could be close to the difference of including or neglecting the rotational inertia of the end-mass.

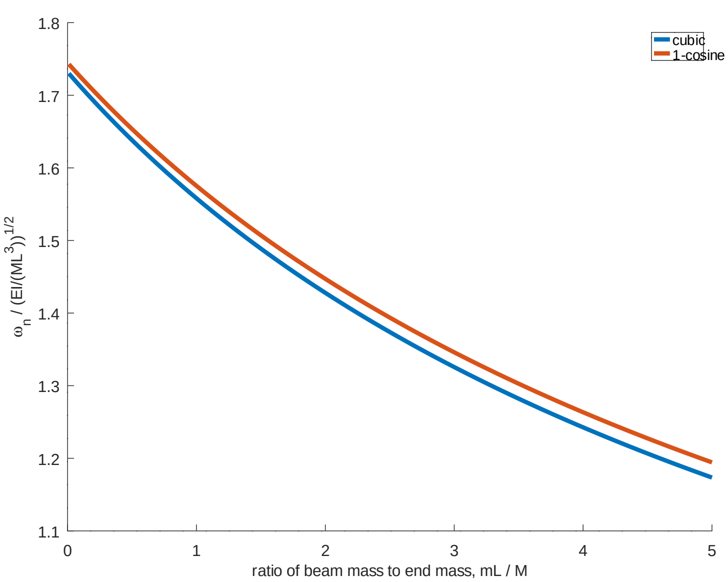
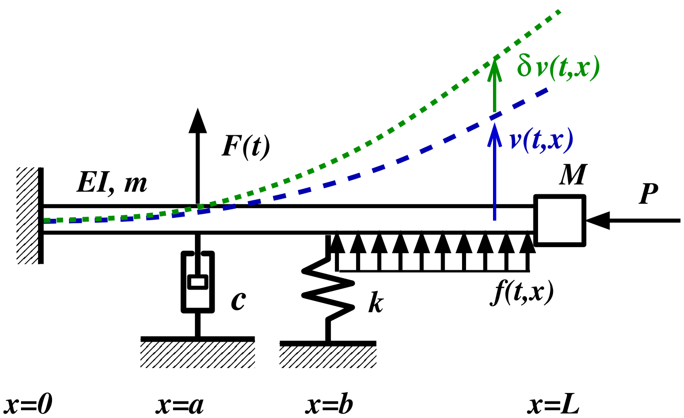


Figure 3. Natural frequencies of a beam with an end mass as a function of the ratio of the beam mass to the end mass, using two choices for the shape function.

6.2 Example 2: a single generalized coordinate

In this example, the essential boundary conditions are $v(t,0) = 0$ and $v'(t,0) = 0$, so any shape function used in this problem must also satisfy $\psi_k(0) = 0$ and $\psi_k'(0) = 0$. In this first example, we will consider a single (dimensionless) shape function, such as, $\psi(x) = (x/L)^2$, $\psi(x) = (x/L)^3$, or $\psi_(x) = 1 - \cos(\pi x/(2L))$. Just to keep this simple for now, we choose $\psi(x) = (x/L)^3$. Forces and associated virtual displacements are tabulated below.



Element	Real Internal Force	Virtual Internal Displacement
M	$M\ddot{v}(L,t) = M\psi(L)\ddot{q}(t) = M\ddot{q}(t)$	$\delta v(L,t) = \psi(L)\delta q(t) = \delta q(t)$
c	$c\dot{v}(a,t) = c\psi(a)\dot{q}(t) = c(a/L)^3\dot{q}(t)$	$\delta v(a,t) = \psi(a)\delta q(t) = (a/L)^3\delta q(t)$
k	$kv(b,t) = k\psi(b)q(t) = k(b/L)^3 q(t)$	$\delta v(b,t) = \psi(b)\delta q(t) = (b/L)^3\delta q(t)$
EI, m	$EIv''(x,t) = EI\psi''(x)q(t) = EI \cdot 6x/L^3 \cdot q(t)$	$\delta v''(x,t) = \psi''(x)\delta q(t) = 6x/L^3 \cdot \delta q(t)$
EI, m	$m\ddot{v}(x,t) = m\psi(x)\ddot{q}(t) = m(x/L)^3 \cdot \ddot{q}(t)$	$\delta v(x,t) = \psi(x)\delta q(t) = (x/L)^3\delta q(t)$
	Real External Force	Virtual Displacement
F(t)	$F(t)$	$\delta v(a,t) = \psi(a)\delta q(t) = (a/L)^3\delta q(t)$
f(t,x)	$f(x,t)$	$\delta v(x,t) = \psi(x)\delta q(t) = (x/L)^3\delta q(t)$
P	P	$v'(x,t)\delta v'(x,t) = 9x^4/L^6\ q(t)\ \delta q(t)$

Equating the work of real internal forces moving through internal virtual displacements, with real external forces moving through collocated virtual displacements,

$$M\ddot{q}\ \delta q + c((a/L)^3)^2\ \dot{q}\ \delta q + k((b/L)^3)^2\ q\ \delta q + \int_0^L EI((6x/L^3))^2\ dx\ q\ \delta q + \int_0^L m((x/L)^3)^2\ dx\ \ddot{q}\ \delta q$$
$$= F(t)(a/L)^3\ \delta q + \int_b^L f(x,t)(x/L)^3\ dx\ \delta q + \int_0^L P(9x^4/L^6)\ dx\ q\ \delta q$$

Evaluating the definite integrals, factoring out the (arbitrary) virtual coordinate δq, specifying that the distributed dynamic force is uniform with intensity f_o, and grouping terms, the equation of motion for this system is

$$\left(M + \frac{1}{6}mL\right)\ddot{q}(t) + c\left(\frac{a}{L}\right)^6 \dot{q}(t) + \left(k\left(\frac{b}{L}\right)^6 + 12\frac{EI}{L^3} - \frac{9}{5L}P\right) q(t) = \left(\frac{a}{L}\right)^3 F(t) + \frac{1}{4}\frac{L^4 - b^4}{L^3} f_o(t)$$

Note that this equation of motion is dimensionally homogeneous (as it should be).

The natural frequency of this system is

$$\omega_{\rm n} = \sqrt{\frac{k\left(\frac{b}{L}\right)^6 + 12\frac{EI}{L^3} - \frac{9}{5L}P}{M + \frac{1}{6}mL}}$$

In this equation the term $(9Pq(t))/(5L)$ is moved to the left hand side of the equation, as it is a function of position $q(t)$. The coefficient $(9P)/(5L)$ is called the *geometric stiffness* of this system. The negative sign on this term shows that the axial compressive force P is destabilizing for this system. Under the condition

$$k\left(\frac{b}{L}\right)^6 + 12\frac{EI}{L^3} - \frac{9}{5L}P = 0$$

the natural frequency would go to zero, and the system would buckle. So the critical axial buckling load for the system is

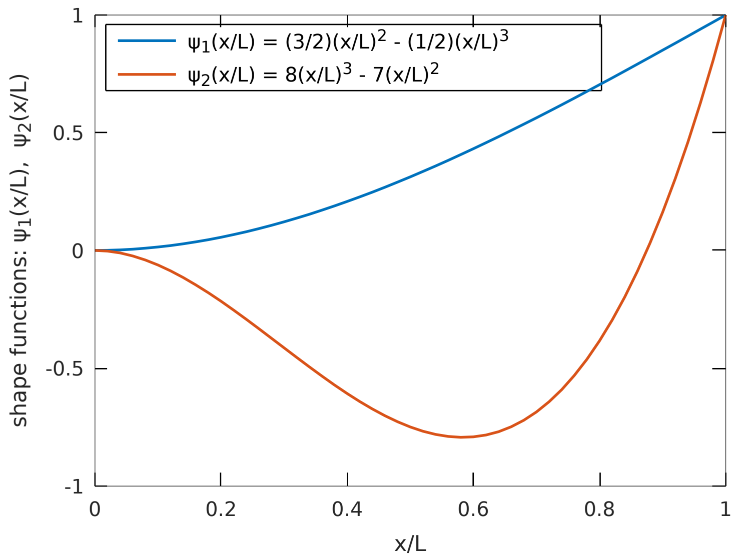
$$P_{\rm cr} = \left(k\left(\frac{b}{L}\right)^6 + 12\frac{EI}{L^3}\right)\left(\frac{5L}{9}\right)$$

Dynamical responses of complex systems require complex mathematical descriptions. The simple approximation $v(x,t) = (x/L)^3 q(t)$ used here could be passable for a simple cantilever beam. But in this example if the spring stiffness k were much higher than EI/L^3 the dynamic response at $x = b$ would have a very small amplitude compared to responses the domains $x < b$ and $x > b$. This kind of response is not captured by the approximation $\psi(x) = (x/L)^3$. In fact, the nature of the free dynamic response in systems such as the one in this example depends on the relative values of the physical parameters, EI/L^3, Mg/L, mg, P/L, k, etc. More complex mathematical models for $v(x,t)$ are required to describe the dynamic responses of complex systems such as this. The next example shows an extension in this direction, in which $v(x,t)$ is modeled by the superposition of two shape functions, and two generalized coordinates.

6.3 Example 3: the same example with two shape functions and two generalized coordinates

In this example, the displaced shape is expressed as the sum of two (independent and kinematically admissible) shape functions, $\psi_1(x)$ and $\psi_2(x)$

$$v(x,t) = \left[\frac{3}{2}\left(\frac{x}{L}\right)^2 - \frac{1}{2}\left(\frac{x}{L}\right)^3\right] q_1(t) + \left[8\left(\frac{x}{L}\right)^3 - 7\left(\frac{x}{L}\right)^2\right] q_2(t)$$



Generalized coordinates associated with dimensionless shape functions have the same physical dimensions as the response variables, which is generally desirable. Shape functions that resemble the actual dynamic responses correspond to more realistic dynamic models. Actual dynamic responses must adhere to essential and natural boundary conditions. So as a first requirement, shape function approximations must adhere to the essential boundary conditions. Shape functions that also adhere to the natural boundary conditions correspond to more realistic models. Mass, and stiffness matrices derived from sets of linearly independent shape functions are positive definite (assuming the system has no rigid body modes). Mass and/or stiffness matrices derived from sets of mutually orthogonal shape functions are numerically well conditioned. Because of this, models derived from sets of mutually orthogonal shape functions are more precise over a broader frequency range.

In this example, $\psi_1(x)$ corresponds to the static deflection of a cantilever beam with a point load at $x = L$; $\psi_2(x)$ has an inflection point and a zero-crossing.

The application of the principle of virtual displacements in which the responses are an expansion of n (admissible and linearly independent) shape functions result in n dimensional matrix equations of motion. Examples of mass and stiffness matrices for higher dimensional approximations are given in equations (10), (11), (12), and (13). This problem is slightly more complex as it involves a spring, a damper, and a concentrated mass.

Applying the principle of superposition, expressions for the internal and external virtual work corresponding to each of these various components may be taken individually.

Work of the inertial force of the distributed mass of the beam, $m\ddot{v}(x,t)dx$, moving through virtual displacements $\delta v(x,t)$

$$\delta W_{\mathrm{I}} = \sum_j \sum_k \left[\int_l m(x) \ \psi_j(x) \ \psi_k(x) \ dx \right] \ddot{q}_k(t) \ \delta q_j(t)$$

Work of the inertial force of the point mass of the beam, $M\ddot{v}(L,t)$ moving through virtual displacements $\delta v(L,t)$

$$\delta W_{\mathrm{I}} = \sum_j \sum_k \left[\int_l M\delta(x-L) \ \psi_j(x) \ \psi_k(x) \ dx \right] \ddot{q}_k(t) \ \delta q_j(t)$$

Work of the bending moments distributed along the beam, $EIv''(x,t)$ moving through virtual rotations distributed along the beam $\delta v''(x,t) \ dx$

$$\delta W_{\mathrm{I}} = \sum_j \sum_k \left[\int_l EI(x) \ \psi_j''(x) \ \psi_k''(x) \ dx \right] q_k(t) \ \delta q_j(t)$$

Work of the spring force, $kv(b,t)$, moving through virtual displacements $\delta v(b,t)$

$$\delta W_{\mathrm{I}} = \sum_j \sum_k \left[\int_l k\delta(x-b) \ \psi_j(x) \ \psi_k(x) \ dx \right] q_k(t) \ \delta q_j(t)$$

Work of the damper force, $c\dot{v}(a,t)$, moving through virtual displacements $\delta v(a,t)$

$$\delta W_{\mathrm{I}} = \sum_j \sum_k \left[\int_l c\delta(x-a) \ \psi_j(x) \ \psi_k(x) \ dx \right] \dot{q}_k(t) \ \delta q_j(t)$$

Work of the dynamic point force, $F(t)$, moving through virtual displacements $\delta v(a,t)$

$$\delta W_{\mathrm{E}} = \sum_j \left[\int_l F(t)\delta(x-a) \ \psi_j(x) \ dx \right] \ \delta q_j(t)$$

Work of the dynamic distributed Force, $f(t) \ dx$, moving through virtual displacements $\delta v(x,t)$

$$\delta W_{\mathrm{E}} = \sum_j \left[\int_b^L f(x,t) \ \psi_j(x) \ dx \right] \ \delta q_j(t)$$

Work of the uniform axial force, P, moving through distributed virtual end shortening $v'(x,t)\delta v'(x,t)dx$

$$\delta W_{\mathrm{E}} = \sum_j \sum_k \left[\int_l P(x) \ \psi_j'(x) \ \psi_k'(x) \ dx \right] q_k(t) \ \delta q_j(t)$$

Each j, k term within the square brackets corresponds to the j, k term of a mass, damping, or stiffness matrix. In these derivations, $\delta(x - a)$ is the Dirac delta function, which has the defining property,

$$\int_l g(x)\delta(x - a)\ dx = g(a)$$

The evaluation of the associated derivatives and integrals can be easily carried out using symbolic manipulation software like Mathematica, Maple, or Wolfram-α.

```
: login-teer-12    Sun Sep 02 14:11:35   ~
: maple
    |\^/|     Maple 2017 (X86 64 LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2017
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.

## INPUT THE SHAPE FUNCTION EQUATIONS ....................

> p1 := (3/2)*(x/L)^2 - (1/2)*(x/L)^3;
                                           2     3
                                        3 x     x
                                  p1 := ---- - ----
                                           2      3
                                        2 L    2 L

> p2 := 8*(x/L)^3 - 7*(x/L)^2;
                                           3      2
                                        8 x    7 x
                                  p2 := ---- - ----
                                          3      2
                                         L      L


## EVALUATE DERIVITIVES .................................


> dp1 := diff(p1,x);
                                                  2
                                         3 x   3 x
                                  dp1 := --- - ----
                                          2       3
                                         L     2 L

> ddp1 := diff(dp1,x);
                                          3     3 x
                                 ddp1 := ---- - ---
                                           2     3
                                          L     L

> dp2 := diff(p2,x);
                                               2
                                        24 x    14 x
                                 dp2 := ----- - ----
                                          3       2
                                         L       L

> ddp2 := diff(dp2,x);
                                         48 x    14
                                 ddp2 := ---- - ----
                                           3      2
                                          L      L
```

```
## EVALUATE INTEGRALS ..................................



## MASS MATRIX TERMS ....................................

> m11 := int(m*p1*p1,x=0..L);
                                          33 m L
                                  m11 := ------
                                           140

> m12 := int(m*p1*p2,x=0..L);
                                           37 m L
                                  m12 := - ------
                                            420

> m22 := int(m*p2*p2,x=0..L);
                                          29 m L
                                  m22 := ------
                                           105


> M11 := eval(M*p1*p1,x=L);
                                    M11 := M

> M12 := eval(M*p1*p2,x=L);
                                    M12 := M

> M22 := eval(M*p2*p2,x=L);
                                    M22 := M


## STIFFNESS MATRIX TERMS ...............................

> EI11 := int(EI*ddp1*ddp1,x=0..L);
                                           3 EI
                                  EI11 := ----
                                             3
                                            L

> EI12 := int(EI*ddp1*ddp2,x=0..L);
                                           3 EI
                                  EI12 := ----
                                             3
                                            L

> EI22 := int(EI*ddp2*ddp2,x=0..L);
                                          292 EI
                                  EI22 := ------
                                             3
                                            L
```

```
> k11 := eval(k*p1*p1,x=b);

> k11 := simplify(k11);
                                4          2
                             k b  (3 L - b)
                      k11 := ---------------
                                       6
                                    4 L

> k12 := eval(k*p1*p2,x=b);

> k12 := simplify(k12);
                           4
                        k b  (3 L - b) (8 b - 7 L)
                 k12 := --------------------------
                                         6
                                      2 L

> k22 := eval(k*p2*p2,x=b);

> k22 := simplify(k22);
                               4            2
                            k b  (7 L - 8 b)
                     k22 := -----------------
                                       6
                                      L

## DAMPING MATRIX TERMS ................................

> c11 := eval(c*p1*p1,x=a);

> c11 := simplify(c11);
                                4          2
                             c a  (3 L - a)
                      c11 := ---------------
                                       6
                                    4 L

> c12 := eval(c*p1*p2,x=a);

> c12 := simplify(c12);
                           4
                        c a  (3 L - a) (8 a - 7 L)
                 c12 := --------------------------
                                         6
                                      2 L

> c22 := eval(c*p2*p2,x=a);

> c22 := simplify(c22);
                               4            2
                            c a  (8 a - 7 L)
                     c22 := -----------------
                                       6
                                      L
```

```
## EXTERNAL FORCING TERMS ...............................


> F1 := eval(F*p1,x=a);

> F1 := simplify(F1);
                                     2
                                  F a  (3 L - a)
                            F1 := --------------
                                          3
                                       2 L

> F2 := eval(F*p2,x=a);

> F2 := simplify(F2);
                                    2
                                 F a  (8 a - 7 L)
                           F2 := ----------------
                                         3
                                        L

> f1 := int(fo*p1,x=b..L);

> f1 := simplify(f1);
                                     4        3    4
                              fo (3 L  - 4 L b  - b )
                        f1 := -----------------------
                                           3
                                        8 L


> f2 := int(fo*p2,x=b..L);

> f2 := simplify(f2);
                                   4        3      4
                              fo (L  - 7 L b  + 6 b )
                      f2 := - ---------------------------
                                           3
                                        3 L


## GEOMETRIC STIFFNESS TERMS ............................

> P11 := int(P*dp1*dp1,x=0..L);
                                        6 P
                                 P11 := ---
                                        5 L

> P12 := int(P*dp1*dp2,x=0..L);
                                       41 P
                               P12 :=  ----
                                       20 L

> P22 := int(P*dp2*dp2,x=0..L);
                                       188 P
                                P22 := -----
                                       15 L
```

The resulting equations of motion in terms of generalized coordinates, $q_1(t)$ and $q_2(t)$ are

$$
\begin{bmatrix} \frac{33}{140}mL + M & -\frac{37}{420}mL + M \\ -\frac{37}{420}mL + M & \frac{29}{105}mL + M \end{bmatrix} \begin{bmatrix} \ddot{q}_1(t) \\ \ddot{q}_2(t) \end{bmatrix}
$$

$$
+ \frac{ca^4}{L^6} \begin{bmatrix} \frac{(3L-a)^2}{4} & \frac{(3L-a)(8a-7L)}{2} \\ \frac{(3L-a)(8a-7L)}{2} & (8a-7L)^2 \end{bmatrix} \begin{bmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \end{bmatrix}
$$

$$
+ \frac{kb^4}{L^6} \begin{bmatrix} \frac{(3L-b)^2}{4} & \frac{(3L-b)(8b-7L)}{2} \\ \frac{(3L-b)(8b-7L)}{2} & (8b-7L)^2 \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}
$$

$$
+ \frac{EI}{L^3} \begin{bmatrix} 3 & 3 \\ 3 & 292 \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix}
$$

$$
- \frac{P}{L} \begin{bmatrix} \frac{6}{5} & \frac{41}{20} \\ \frac{41}{20} & \frac{188}{15} \end{bmatrix} \begin{bmatrix} q_1(t) \\ q_2(t) \end{bmatrix} = \begin{bmatrix} \frac{a^2(3L-a)}{2L^3} & \frac{3L^4-4Lb^3-b^4}{8L^3} \\ \frac{a^2(8a-7L)}{L^3} & -\frac{L^4-7Lb^3+6b^4}{3L^3} \end{bmatrix} \begin{bmatrix} F(t) \\ f_o(t) \end{bmatrix} \quad (14)
$$

With the scaling of the dimensionless shape functions, $\psi_1(L) = \psi_2(L) = 1$, $q_1(t)$ and $q_2(t)$ are the values of $v(L,t)$ corresponding to $\psi_1(x)$ and $\psi_2(x)$. With the dimensionless formulation of the shape functions, every term in this equation has units of force.

This example is an introduction to methodologies that are invoked later in the course.

For more complex geometries, for example, beams with tapered sections, the derivation can become very complex, and the analysis is more easily carried out numerically.

The same example, computed with matlab, is:

```
%% DEFINE NUMERICAL VALUES FOR CONSTANTS

EI =  1e7;    % flexural rigidity       N.m^2
k  =  1e2;    % concentrated stiffness  N/m
m  =  1e0;    % distributed mass        kg/m
M  =  1e1;    % lumped mass             kg
c  =  0.1;    % spring damping rate     N/m/s
L  =  10;     % overall length          m
a  =   3;     % location of damper      m
b  =   5;     % location of spring      m

dx = 0.01;            % increment of length along the beam
 x = [ 0 : dx : L ];  % x-axis
xa = round(a/dx);
xb = round(b/dx);
```

```
%% INPUT THE SHAPE FUNCTION EQUATIONS ...................

p1 = (3/2)*(x/L).^2 - (1/2)*(x/L).^3;

p2 = 8*(x/L).^3 - 7*(x/L).^2;


%% EVALUATE DERIVITIVES ................................

dp1  = cdiff(p1)/dx;
ddp1 = cdiff(dp1)/dx;

dp2  = cdiff(p2)/dx;
ddp2 = cdiff(dp2)/dx;



%% EVALUATE INTEGRALS ..................................


%% MASS MATRIX TERMS ...................................

m11 = trapz(m*p1.*p1)*dx;
m12 = trapz(m*p1*p2)*dx;
m22 = trapz(m*p2*p2)*dx;

M11 = M*p1(end)*p1(end);
M12 = M*p1(end)*p2(end);
M22 = M*p2(end)*p2(end);


%% STIFFNESS MATRIX TERMS ..............................

EI11 = trapz(EI*ddp1.*ddp1)*dx;
EI12 = trapz(EI*ddp1.*ddp2)*dx;
EI22 = trapz(EI*ddp2.*ddp2)*dx;

k11 = k*p1(xb)*p1(xb);
k12 = k*p1(xb)*p2(xb);
k22 = k*p2(xb)*p2(xb);


%% DAMPING MATRIX TERMS ...............................

c11 = c*p1(xa)*p1(xa);
c12 = c*p1(xa)*p2(xa);
c22 = c*p2(xa)*p2(xa);


%% EXTERNAL FORCING TERMS ..............................

F1 = F*p1(xa);
F2 = F*p2(xa);

f1 = trapz(fo*p1(xb:L))*dx;
f2 = trapz(fo*p2(xb:L))*dx;


%% GEOMETRIC STIFFNESS TERMS ...........................

P11 = trapz(P*dp1.*dp1)*dx;
P12 = trapz(P*dp1.*dp2)*dx;
P22 = trapz(P*dp2.*dp2)*dx;
```