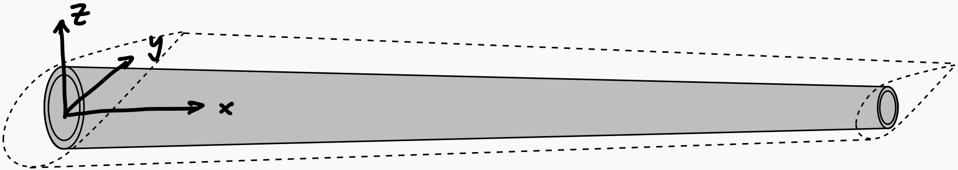
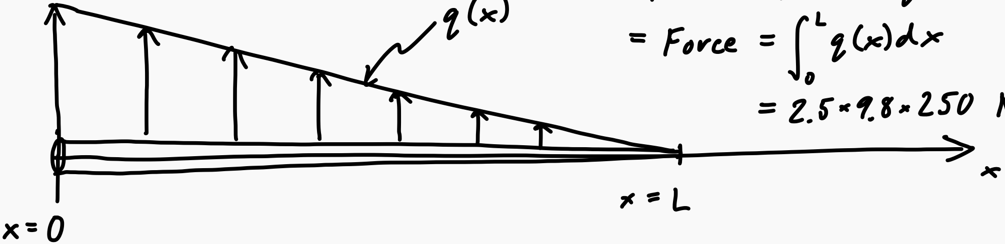
Project #2 Model

Reminder

Objective: minimize spar weight, subject to stress and manufacturing constraints





"Area" under $q(x)$

$= \text{Force} = \int_0^L q(x)\,dx$

$= 2.5 \times 9.8 \times 250$ N

Euler-Bernoulli Beam Theory

We will model the spar using Euler-Bernoulli Beam Theory.

Assumptions:

planar symmetry: longitudinal axis is straight, and cross section of beam has a longitudinal plane of symmetry

cross-section variation: cross section varies smoothly

normality: plan sections that are normal to longitudinal plane before bending remain normal after bending

strain energy: internal strain energy accounts only for bending moment deformations

Euler-Bernoulli Beam Theory (cont.)

linearization: deformations are small enough that nonlinear effects are negligible

material: the material is assumed to be elastic and isotropic

Euler-Bernoulli Beam Theory (cont.)

The displacement of the beam in the vertical direction, the direction of the load, is governed by the 4th-order PDE

$$\frac{d^2}{dx^2}\left(EI_{yy}\frac{d^2 w}{dx^2}\right) = q, \qquad \forall x \in [0, L]$$

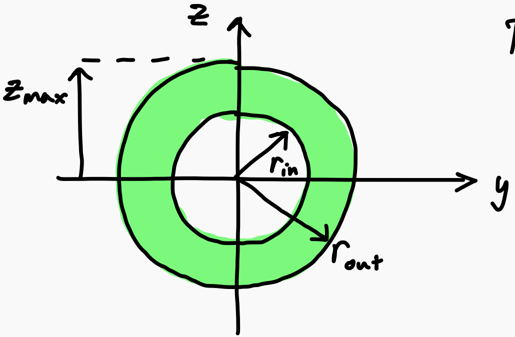
where

- w is the vertical displacement in the z direction;
- $q(x)$ is the applied load;
- E is the elastic, or Young's, modulus, and;
- I_{yy} is the second-moment of area with respect to the y axis.

In particular,

$$I_{yy} = \iint z^2 \, dz dy,$$

with the integral taken over the cross-sectional region. It is assumed that the centroid of the cross section is located at $(y, z) = (0, 0)$.



Tables are available for I_{yy}.

Euler-Bernoulli Beam Theory (cont.)

We will treat the spar like a cantilever beam, for which the boundary conditions are

no vertical → $w(x = 0) = 0,$ $\quad \frac{d^2 w}{dx^2}(x = L) = 0$ ← no stress at tip

or

no angular → $\frac{dw}{dx}(x = 0) = 0$ $\quad \frac{d^3 w}{dx^3}(x = L) = 0.$

displacement at the root

Euler-Bernoulli Beam Theory (cont.)

Once the Euler-Bernoulli equation is solved for w, these displacements can be used to solve for the normal stress as a function of x:

$z_{max} = z_{max}(x)$

i.e. <u>not constant</u>

$$\sigma_{xx}(x) = -z_{\text{max}} E \frac{d^2 w}{dx^2}$$

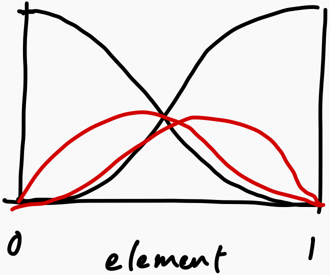
where z_{max} is the maximum height of the cross section (in this case, the outer radius).

- Since we are interested only in the magnitude of σ_{xx}, the negative sign can be ignored.

Finite-Element Discretization

We will discretize the Euler-Bernoulli beam equation using the finite-element method.

- solution is represented using Hermite-cubic shape functions
- finite-element equations result from the minimization of the potential energy functional



Matlab Implementation

This finite-element discretization of the beam equation is implemented by the (top-level) Matlab function

stores $w, \frac{dw}{dx}$

```
function [u] = CalcBeamDisplacement(L, E, Iyy, force, Nelem)
% Estimate beam displacements using Euler-Bernoulli
% Inputs:
%   L - length of the beam
%   E - longitudinal elastic modulus
%   Iyy - moment of area with respect to the y axis
%   force - force per unit length along the axis x
%   Nelem - number of finite elements to use
% Outputs:
%   u - displacements at each node along the beam
%-----------------------------------------------
```

Nelem + 1 arrays

represents $q(x)$

1-d arrays

Nelem = 3 $\Rightarrow$ 4 nodes

Matlab Implementation (cont.)

Once the `u` displacements are known, they can be passed to `CalcBeamSress` to obtain the stress:

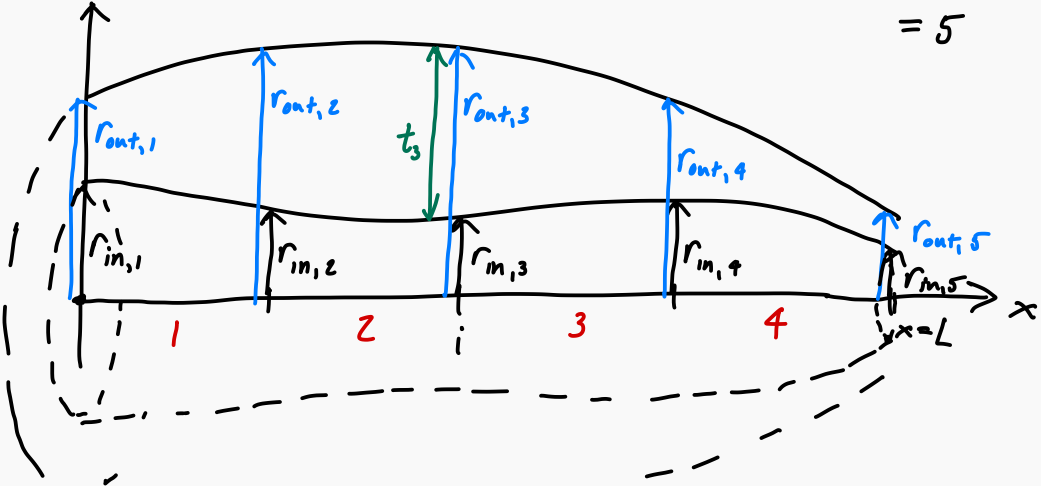
```
function [sigma] = CalcBeamStress(L, E, zmax, u, Nelem)
% Compute stress in beam using Euler-Bernoulli
% Inputs:
%   L - length of the beam
%   E - longitudinal elastic modulus
%   zmax - maximum height of the beam at each node
%   u - displacements at each node along the beam
%   Nelem - number of finite elements to use
% Outputs:
%   sigma - stress at each node in the beam
%-----------------------------------------
```

1-d arrays

Design Variables

$N_{elem} = 4$

$N_{node} = N_{elem} + 1$
$= 5$



What is the objective? Weight, mass, or volume

What are the constraints?

geometry : node i $\quad \underbrace{\overbrace{r_{out,i} - r_{in,i}}^{t_i} \geq 0.0025m}_{\underline{\underline{A}}\,\underline{r} \leq \underline{b}}$

$$0.0125m \leq r_{out,i} \leq 0.05m$$

$$0.01m \leq r_{in,i} \leq 0.0475m$$

stress : node i $\quad \dfrac{\sigma_i}{\sigma_{max}} - 1 \equiv c_i(r) \leq 0$

! Make a function that extracts r into r_{in}, r_{out}

Design Variables

$\underline{r}$: how many ? $\quad nvar = 2*(Nelem + 1)$

how are they stored ?

2 options that make sense

#1:

$$\underline{r} = \begin{bmatrix} r_{in,1} \\ r_{out,1} \\ r_{in,2} \\ r_{out,2} \\ \vdots \\ r_{in,Nelem+1} \\ r_{out,Nelem+1} \end{bmatrix}$$

#2

$$\underline{r} = \begin{bmatrix} r_{in,1} \\ r_{in,2} \\ \vdots \\ r_{in,Nelem+1} \\ r_{out,1} \\ r_{out,2} \\ \vdots \end{bmatrix}$$

let's work backwards

- $\hat{c}_i(r) = \sigma_i / \sigma_{max} - 1$, $\forall$ i nodes in FE mesh
- compute $\sigma_i(r)$: call calcBeamStress
- call calcBeamDisplacement, "evaluate" z_{max}
- compute $I_{yy_i}(r)$ $\forall$ i nodes in FE mesh
- (optional) extract r_{in}, r_{out} from input array (or r_{in}, t, or...)

Debugging Steps (stages)

1) check force dist. $q(x)$: plot it vs x
   ↳ check integral

opt. 2) check that r_{in}, r_{out} arrays are extractly correctly from r

3) check that mass/volume works for the nominal design

4) optimize mass (vol) without the stress constraints (and maybe without $Ax \leq b$)

5) check that the nominal design satisfies all the constraints