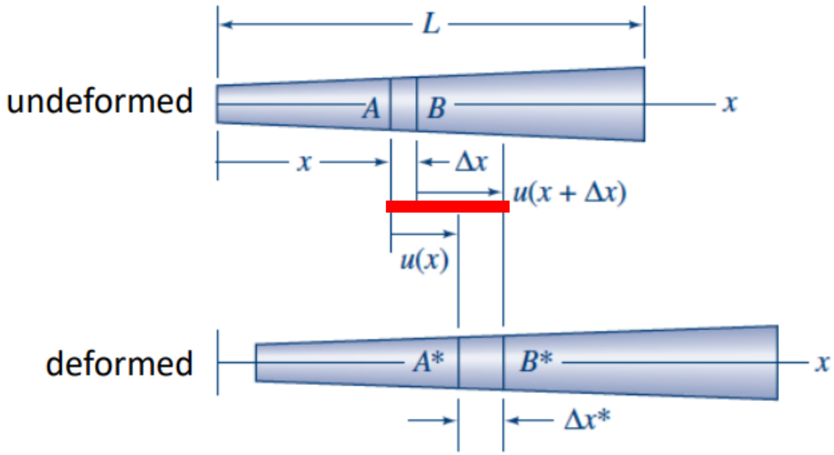
Lecture 21
Computer Project: Axially-deformable Bars

Kinematic assumptions: cross sections, which are plane and are perpendicular to the axis before deformation, remain plane and remain perpendicular to the axis after deformation. In addition, cross sections do not rotate about the axis.

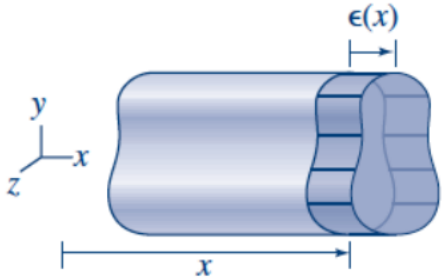


The strain-displacement relationship is …

$$\epsilon_x(x) = \lim_{\Delta x \to 0} \frac{\Delta x^* - \Delta x}{\Delta x} = \frac{du(x)}{dx} = \epsilon(x)$$

$$\mathbf{\Delta x^\star + u(x) = \Delta x + u(x + \Delta x)}$$

$$\mathbf{\Delta x^\star - \Delta x = u(x + \Delta x) - u(x)}$$



HW8: Matlab Project

Consider an elastic bar with Young's modulus, $E = 10$, the cross section area $A = 1$, and the length of the bar $L = 1$. The bar has a built-in boundary condition at $x = 0$, i.e. $u(0) = 0$, and at $x = L$, the internal force $R(L) = 0$ as shown in Fig. 1.

The differential equation that governs the equilibrium of the bar has been derived as follows,

$$\frac{d}{dx}\left(EA\frac{du}{dx}\right) + b(x) = 0, \quad 0 < x < L \ ,$$

where $u(x)$ is the displacement field.

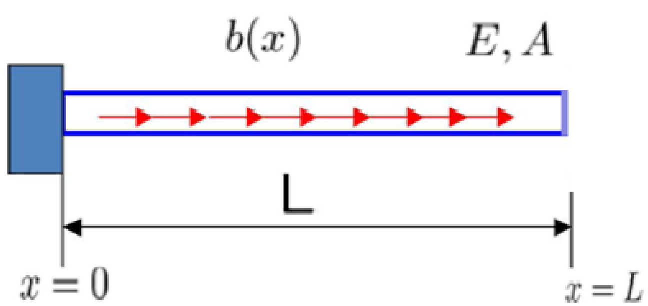
The bar is subjected a distributed load along its span, i.e.

$$b(x) = p\sin\left(\frac{2\pi x}{L}\right),$$

where $p = 1$ with a unit of force per unit length.

The internal force is defined as $R(x) = \sigma A = EA\epsilon$, i.e.

$$R(x) = EA\frac{du}{dx}$$

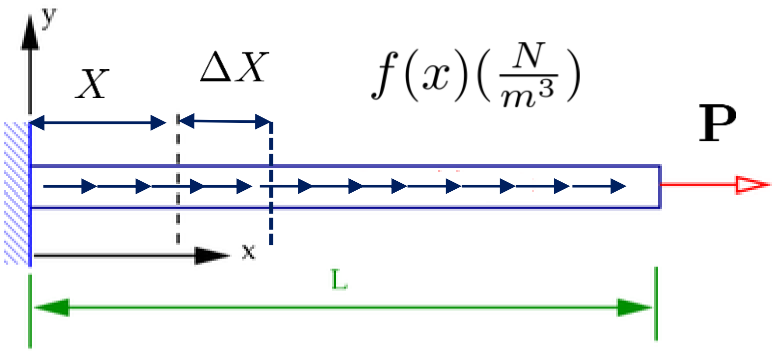


$$u(0) = 0, \quad R(L) = 0$$

Exact solution

$$R(x) = \frac{pL}{2\pi}\cos\left(\frac{2\pi}{L}x\right) - \frac{pL}{2\pi}; \quad u(x) = \frac{pL^2}{4\pi^2 EA}\sin\frac{2\pi x}{L} - \frac{pL}{2\pi EA}x$$

HW8: Matlab Project



$$\mathbf{B.C.: u(0) = 0, \quad R(L) = P}$$

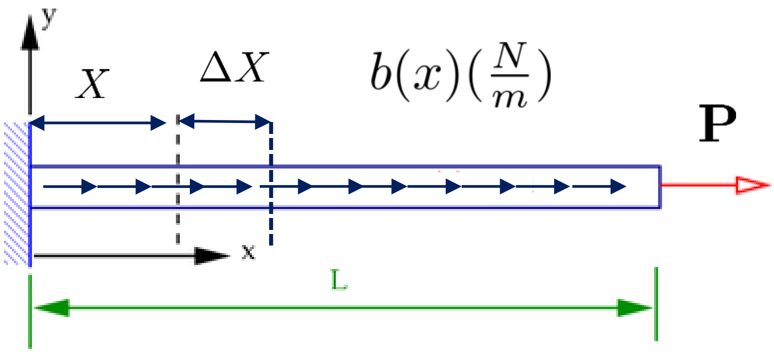
$$(1) \qquad \frac{d\sigma}{dx} + f(x) = 0; \quad (2)\ \frac{d}{dx}E\left(\frac{du}{dx}\right) + f(x) = 0\ .$$

$$\boxed{\text{Let } \ R(x) = EA\frac{du}{dx} \ \rightarrow \quad \frac{d}{dx}R(x) + b(x) = 0\ .}$$

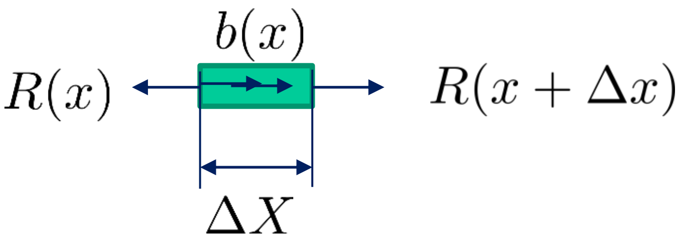
$$\boxed{\frac{d}{dx}\left(AE\frac{du}{dx}\right) + b(x) = 0\ , \ \text{where} \quad b(x) = Af(x) \ \sim \ (F/L)}$$

Today's Lecture Attendance Password is: Matlab Project

HW8



$\mathbf{B.C.: u(0) = 0, \quad R(L) = P}$



$$\sum F_x = 0 \ \rightarrow \ -R(x) + R(x+\Delta x) + b(x)\Delta x = 0$$

$$\frac{R(x+\Delta x) - R(x)}{\Delta x} + b(x) = 0$$

$$\boxed{\text{Let} \quad R(x) = EA\frac{du}{dx} \ \rightarrow \ \frac{d}{dx}R(x) + b(x) = 0 \ .}$$

$$\boxed{\frac{d}{dx}\left(AE\frac{du}{dx}\right) + b(x) = 0; \ \forall \ 0 < x < L, \ \text{where} \quad b(x) \quad \sim \quad (F/L)}$$

How to solve this equation by using MatLab

In the lecture these four relations have been combined to obtain a single equation representing equilibrium in terms of the displacement,

$$\frac{d}{dx}\left[EA\frac{du}{dx}\right] + b = 0\,. \tag{5}$$

In order to utilize the solver in MATLAB, one must convert the governing equations into first-order form,

$$\boxed{\frac{d\boldsymbol{y}}{dx} = \boldsymbol{f}(\boldsymbol{y}, x),} \tag{6}$$

where $\boldsymbol{y}$ is a vector of unknown variables, and $\boldsymbol{f}$ is a vector of known functions depending on $\boldsymbol{y}$ and the position x. [1]

$$\begin{aligned} R &= EA\frac{du}{dx}\,. \\ \frac{dR}{dx} + b &= 0, \end{aligned}$$

$$\frac{d}{dx}\begin{bmatrix} u \\ R \end{bmatrix} = \begin{bmatrix} \dfrac{R}{E(x)A(x)} \\ -b(x) \end{bmatrix}$$

By defining,

$$\boxed{\begin{aligned} \boldsymbol{y} &:= \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &&= \begin{bmatrix} u \\ R \end{bmatrix}, \\ \boldsymbol{f}(\boldsymbol{y}, x) &:= \begin{bmatrix} f_1(\boldsymbol{y}, x) \\ f_2(\boldsymbol{y}, x) \end{bmatrix} &&= \begin{bmatrix} \dfrac{y_2}{E(x)A(x)} \\ -b(x) \end{bmatrix}, \end{aligned} \quad (7)}$$

one obtains the desired first-order form,

$$\frac{d}{dx}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{y_2}{E(x)A(x)} \\ -b(x) \end{bmatrix}.$$

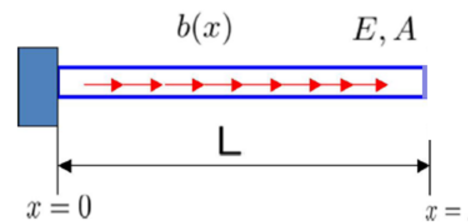
2.2 Boundary condition

In order to apply boundary conditions in the solver in MATLAB, one must define a function which returns a residual of how much the boundary conditions are not satisfied; a residual of zero implies that the boundary conditions are satisfied exactly. The function has the form,

$$\boxed{\boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b))}$$

where $\boldsymbol{g}$ is vector of functions depending on the value of $\boldsymbol{y}$ evaluated at the boundary points $x = a$ and $x = b$ (Here we assume the problem is defined on the interval (a, b)).

For example:



To clarify the form of the function, consider the boundary condition,

$$u(0) = u_0,$$
$$R(L) = R_L,$$

where the displacement is known as u_0 at the end point $x = 0$, and the force is known as R_L at the end point $x = L$. The vector defining the residual of how much the boundary condition is not satisfied is,

$$\begin{bmatrix} u(0) - u_0 \\ R(L) - R_L \end{bmatrix} = 0 \qquad R(L) = EA\frac{du}{dx}\Big|_{x=L}$$

Using the correspondence between u, R and $\boldsymbol{y}$ defined in (7), one defines $\boldsymbol{g}$ as,

Res

$$\boxed{\boldsymbol{g}(\boldsymbol{y}(0), \boldsymbol{y}(L)) \quad := \begin{bmatrix} g_1(\boldsymbol{y}(0), \boldsymbol{y}(L)) \\ g_2(\boldsymbol{y}(0), \boldsymbol{y}(L)) \end{bmatrix} = \begin{bmatrix} y_1(0) - u_0 \\ y_2(L) - R_L \end{bmatrix} . \quad (8)}$$

Use Matlab to find the displacement field and internal force/stress field, and compare them with the statically indeterminate system that has the same dimensions and the same material properties, but with different boundary conditions: $u(0) = u(L) = 0$. Hint:
Go to class Bcourses website and go to the lecture folder, and then download a Matlab-P1 folder that contains the file: bar1d.m. You start your solution there. For all details, please refer to Lecture20F.pdf slide.

$$\frac{d\mathbf{y}}{dx} = \mathbf{f}(\mathbf{y}, x)$$

bvp4c

Solve boundary value problem — fourth-order method

co

Res

$\mathbf{y}$

Syntax

```
sol = bvp4c(odefun,bcfun,solinit)
sol = bvp4c(odefun,bcfun,solinit,options)
```

Description

`sol = bvp4c(odefun,bcfun,solinit)` integrates a system of differential equations of the form $y' = f(x,y)$ specified by `odefun`, subject to the boundary conditions described by `bcfun` and the initial solution guess `solinit`. Use the `bvpinit` function to create the initial guess `solinit`, which also defines the points at which the boundary conditions in `bcfun` are enforced.

`sol = bvp4c(odefun,bcfun,solinit,options)` also uses the integration settings defined by `options`, which is an argument created using the `bvpset` function. For example, use the `AbsTol` and `RelTol` options to specify absolute and relative error tolerances, or the `FJacobian` option to provide the analytical partial derivatives of `odefun`.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% -- Matlab code (BVP4C) to solve Tension-Compression Bar(1D)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function bar1d
% -- Define geometry
L = 1;                   % -- Length of bar

% -- Set solver parameters
nvar  =  2;              % -- Number of variables
np    = 10;              % -- Initial Number of points on [0,L]
xp    = linspace(0,L,np);% -- Initial Points at which to satisfy ODE

% -- Set initial solution for the solver
solinit = bvpinit(xp,zeros(1,nvar));

% -- Set options on the tolerance for the accuracy of the solution
%    Default: RelTol(1e-3), AbsTol(1e-6)
%
%    y'(x) = f(x,y(x)) + res(x)
%
%    norm( res(i)/f(i) ) <= RelTol  and
%    norm( res(i)      ) <= AbsTol
%
options = bvpset('RelTol',1e-3,'AbsTol',1e-6);

% -- Invoke solver bvp4c is a built-in Matlab function
sol     = bvp4c(@bar1d_ode,@bar1d_bc,solinit,options);

% -- Plot solution
bar1d_plot(L,sol);

end
```

y

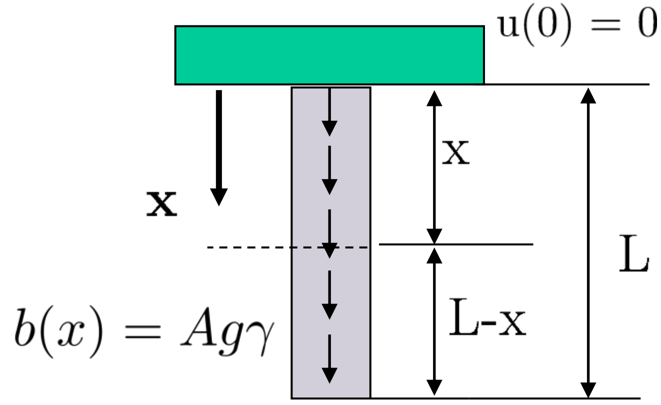
```
function [fxy] = bar1d_ode(x,y)

% -- Define material property and geometry
E = 1;  % Young's Modulus
A = 1;  % Cross-sectional area
L = 1;  % Length of bar

g=9.8;
gamma=1;

% -- Define distributed load
b = gamma*g;

% -- Define function
fxy = [y(2)/(E*A);
                -b];
end
```



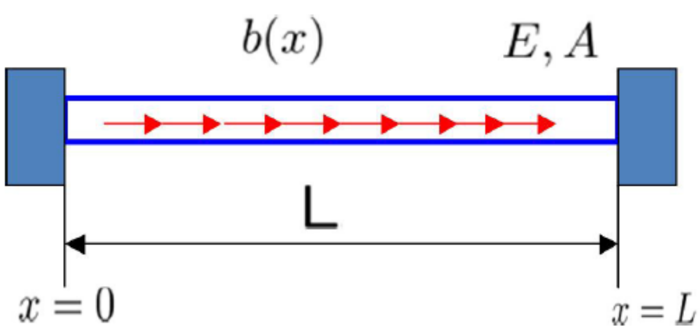
$$\frac{d}{dx}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{y_2}{E(x)A(x)} \\ -b(x) \end{bmatrix}.$$

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [fxy] = bar1d_ode(x,y)

% -- Define material property and geometry
E = 1;  % Young's Modulus
A = 1;  % Cross-sectional area
L = 1;  % Length of bar

% -- Define distributed load
b = sin(pi*x/(2*L));

% -- Define function
fxy = [y(2)/(E*A);
                -b];
end
```



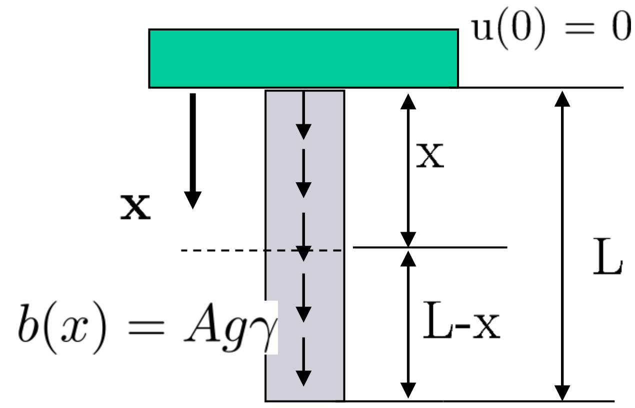
```
function [res] = bar1d_bc(ya,yb)

% -- Boundary Conditions (BC)
%     u: displacement
%     f: force

ua = 0;      % -- Fixed       at x=a
fb = 0;      % -- Zero force at x=b

res= [ya(1)-ua;
      yb(2)-fb];


end
```



$R(L) = 0$

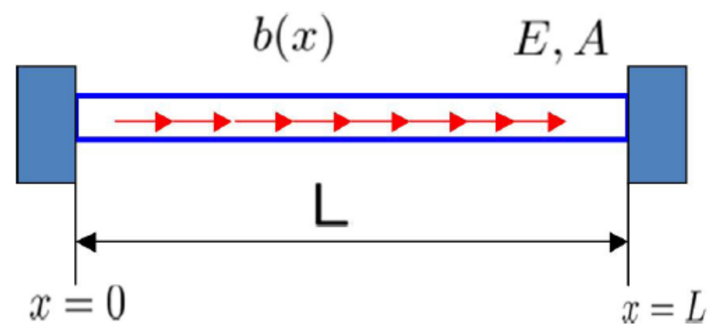
```
function [res] = bar1d_bc(ya,yb)

% -- Boundary Conditions (BC)
%     u: displacement
%     f: force

ua = 0;      % -- Fixed       at x=a
ub = 0;       % -- Fixed       at x=b
%fb = 0;      % -- Zero force at x=b

res= [ya(1)-ua;
      yb(1)-ub];

end
```



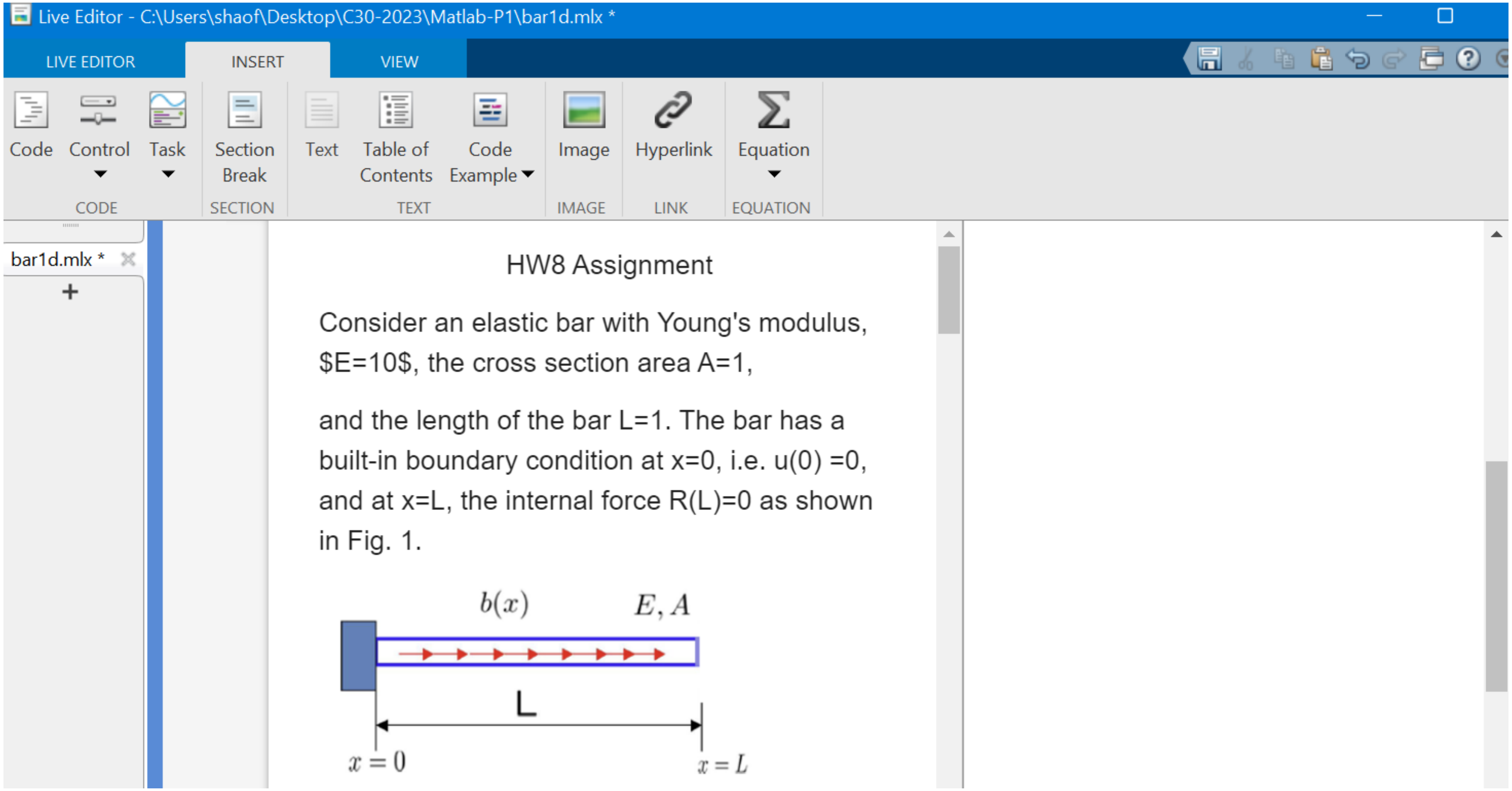
Statically indeterminant problem

HW8.pdf	9:58am	9:58am	86 KB
Matlab-P1.zip	9:53am	9:53am	75 KB

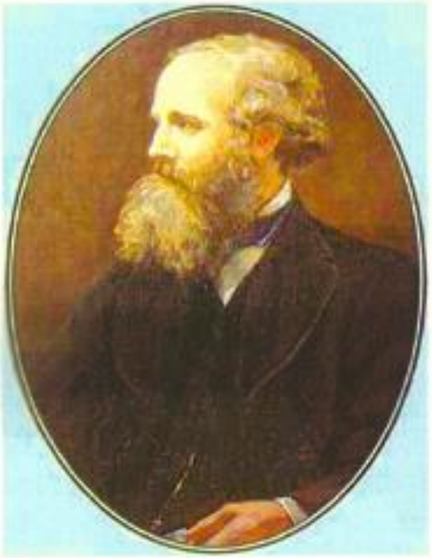
C30-2023 › Matlab-P1

Name	Date modified	Type	Size
bar1d	10/8/2023 9:50 AM	MATLAB Code	4 KB
bar1d_M	10/8/2023 9:51 AM	MATLAB Live Script	42 KB
Fig1	3/16/2023 10:43 AM	PNG File	34 KB

Matlab Live Editor



The person who invented force method



(13 June 1831 – 5 November 1879)

James Clerk Maxwell

Scottish mathematician and physicist, is known for his contribution in electromagnetics theory, (the celebrated Maxwell equations), and his contribution in statistical physics (Maxwellian distribution).

Journal
Philosophical Magazine ›
Series 4
Volume 27, 1864 - Issue 182

Enter keywords, authors, DOI etc.

Original Articles

L. *On the calculation of the equilibrium and stiffness of frames*

J. Clerk Maxwell F.R.S.

Pages 294-299 | Published online: 26 May 2009

Download citation

Citations   Metrics   Reprints & Permissions   Get access

Click to increase image size

Select Language ▼
Translator disclaimer

People

Article
LVIII.
econo

L. *On the Calculation of the Equilibrium and Stiffness of Frames. By* J. Clerk Maxwell, *F.R.S., Professor of Natural Philosophy in King's College, London*.*

THE theory of the equilibrium and deflections of frameworks subjected to the action of forces is sometimes considered

Google Scholar   On the calculation of the equilibrium and stiffness

University of California was funded in 1868 !

Articles

Any time
Since 2017
Since 2016
Since 2013
Custom range...

Sort by relevance
Sort by date

L. on the calculation of the equilibrium and stiffness of frames
JC Maxwell - The London, Edinburgh, and Dublin Philosophical …, 1864 - Taylor & Francis
T HE theory of the equilibrium and deflections of frameworks subjected to the aetion of forces is sometimes considered as more eomplieated than it really is, especially in eases in which the framework is not simply stiff, but is strengthened (or weakened as it may be) by additional connecting pieces. I have therefore stated a general method of solving all such questions in the least complicated manner. The method is derived from the principle of ...
☆   99   Cited by 712   Related articles   All 4 versions

L. *On the Calculation of the Equilibrium and Stiffness of Frames.* *By* J. CLERK MAXWELL, *F.R.S., Professor of Natural Philosophy in King's College, London**.

THE theory of the equilibrium and deflections of frameworks subjected to the action of forces is sometimes considered as more complicated than it really is, especially in cases in which the framework is not simply stiff, but is strengthened (or weakened as it may be) by additional connecting pieces.

I have therefore stated a general method of solving all such questions in the least complicated manner. The method is derived from the principle of Conservation of Energy, and is referred to in Lamé's *Leçons sur l'Elasticité*, Leçon 7^{me}, as Clapeyron's Theorem; but I have not yet seen any detailed application of it.

If such questions were attempted, especially in cases of three dimensions, by the regular method of equations of forces, every point would have three equations to determine its equilibrium, so as to give $3s$ equations between e unknown quantities, if s be the number of points and e the number of connexions. There are, however, six equations of equilibrium of the system which must be fulfilled necessarily by the forces, on account of the equality of action and reaction in each piece. Hence if

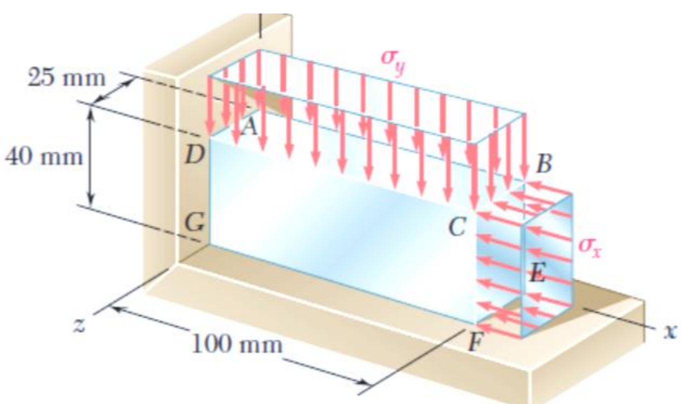
$m \sim e, \ s \sim n$

$$\mathbf{m+r=3n} \ \rightarrow \quad \boxed{e=3s-6,}$$

the effect of any external force will be definite in producing tensions or pressures in the different pieces; but if $e>3s-6$, these forces will be indeterminate. This indeterminateness is got rid

PROBLEM 9.81

The block shown is made of a magnesium alloy, for which $E = 45$ GPa and $\nu = 0.35$. Knowing that $\sigma_x = -180$ MPa, determine (*a*) the magnitude of σ_y for which the change in the height of the block will be zero, (*b*) the corresponding change in the area of the face *ABCD*, (*c*) the corresponding change in the volume of the block.



(*a*) $\quad \delta_y = 0 \qquad \varepsilon_y = 0 \qquad \sigma_z = 0$

$$\varepsilon_y = \frac{1}{E}(\sigma_x - \nu\sigma_y - \nu\sigma_z)$$

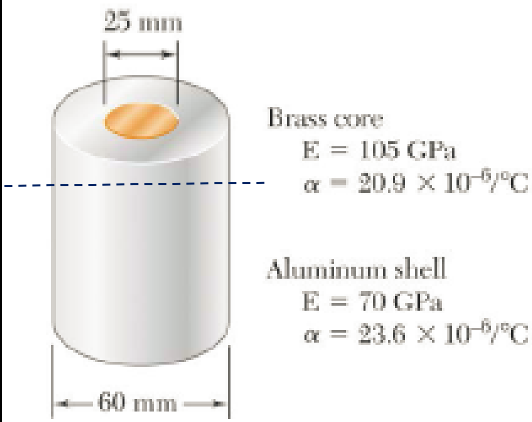
$$\sigma_y = \nu\sigma_x = (0.35)(-180\times10^6)$$

$$= -63\times10^6 \text{ Pa}$$

$\sigma_y = -63.0$ MPa ◀

$$\varepsilon_z = \frac{1}{E}(\sigma_z - \nu\sigma_x - \nu\sigma_y) = -\frac{\nu}{E}(\sigma_x + \sigma_y) = -\frac{(0.35)(-243\times10^6)}{45\times10^9} = +1.890\times10^{-3}$$

$$\varepsilon_x = \frac{1}{E}(\sigma_x - \nu\sigma_y - \nu\sigma_z) = \frac{\sigma_x - \nu\sigma_y}{E} = -\frac{157.95\times10^6}{45\times10^9} = -3.510\times10^{-3}$$

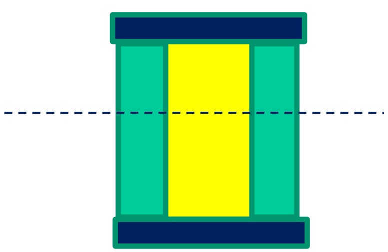


PROBLEM 9.38 $\Delta T = 195 - 15 = 180^o C$

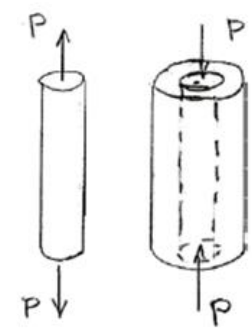
The aluminum shell is fully bonded to the brass core and the assembly is unstressed at a temperature of 15°C. Considering only axial deformations, determine the stress in the aluminum when the temperature reaches 195°C.

$$\Delta_a^T = \alpha_a \Delta T L > \quad \Delta_b^T = \alpha_b \Delta T L$$

$$\sum F_y = 0 \;\rightarrow\; P_b + P_a = 0 \;\rightarrow\; P_b = -P_a =: P$$

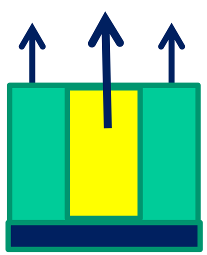


$$f_a = \frac{L}{E_a A_a} \qquad f_b = \frac{L}{E_b A_b}$$



Constrain condition

P_a P_b



$$\Delta_a^T + \Delta_a^P = \Delta_a = \Delta_b = \Delta_b^T + \Delta_b^P$$

$$\rightarrow \quad \alpha_a \Delta T L - f_a P = \alpha_b \Delta T L + f_b P$$

$$P = \frac{(\alpha_a - \alpha_b)\Delta T L}{(f_a + f_b)} \;\rightarrow\; \sigma_b = \frac{P}{A_b}$$