Experiment #08 - VERTICAL DEFLECTION IN SWIMMING POOL DIVING BOARD

Aim: Finding the vertical deflection in a cantilever beam subjected to variable load and material properties and visualization of it.

Methodology:

Solving governing equation of vertical deflection in swimming pool diving board using Laplace transform.

Laplace Transform

The Laplace Transform of a function f(t),defined for all real numbers t > 0 is the function F(s) defined by:

$$L[f(t)] = \int_0^{\infty} e^{-st} f(t)dt = F(s)$$

Inverse laplace transform

The inverse Laplace Transform of F(s) is defined by:

$$f(t) = L^{-1}[F(s)] = \int_0^{\infty} e^{-st} F(s)dt$$

MATLAB syntax

`laplace(f)`	Returns the Laplace transform of f using the default independent variable t and the default transformation variable s.
`laplace(f, transVar)`	Uses the specified transformation variable transVar instead of s.
`laplace(f, var, transVar)`	Uses the specified independent variable var and transformation variable transVar instead of t and s respectively.
`ilaplace(F)`	Returns the inverse Laplace transform of F using the default independent variable s for the default transformation variable t. If F does not contain s, ilaplace

	uses symvar.
`ilaplace(F, transVar)`	Uses the specified transformation variable transVar instead of t.
`ilaplace(F, var, transVar)`	Uses the specified independent variable var and transformation variable transVar instead of s and t respectively.
`heaviside(t - a)`	To input the heaviside's unit step function H(t − a) .
`dirac(t - a)`	To input the dirac delta function δ(t − a).
`laplace(diff(f(t), t), t, s)`	laplace(diff(f(t), t), t, s)

Example 1:
Write the MATLAB code which computes the Laplace Transform of

$$f(t)=\begin{cases} t^2, & t<2, \\ t-1, & 2<t<3 \\ 7, t>3. & \end{cases}$$

```
clear all
clc
syms t
f=input('Enter the function in terms of t: ');
F=laplace(f );
F=simplify(F)
```

Output

```
Enter the function in terms of t: 2 * (heaviside(t - 0) - heaviside(t - 2)) + (t - 1) *
(heaviside(t - 2) - heaviside(t - 3)) + 7 * (heaviside(t - 3))

F =

(2*s + exp(-2*s) - exp(-3*s) - s*exp(-2*s) + 5*s*exp(-3*s))/s^2
```

Example 2:

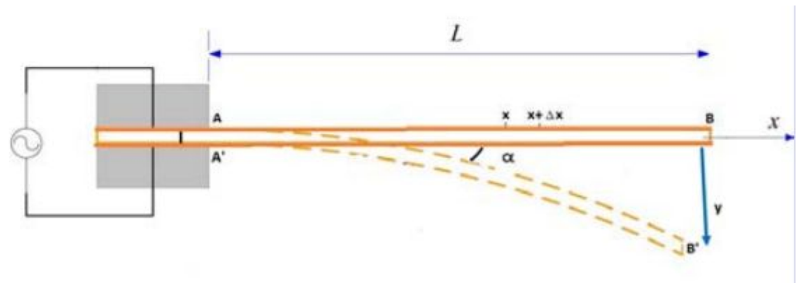
Solve $y'' + 2y' + 10y = 1 + 5(t - 5), y(0) = 1, y'(0) = 2$

```
clc
clear all
syms t s Y
y2=diff(sym('y(t)'),2);
y1=diff(sym('y(t)'),1);
y0=sym('y(t)');
a = input('The Coefficient of D2y = ');
b = input('The Coefficient of Dy = ');
c = input('The Coefficient of y = ');
nh = input('Enter the non homogenous part = ');
eqn=a*y2+b*y1+c*y0-nh;
LTY=laplace(eqn,t,s);
if (a==0)
d = input('The initial value at 0 is ');
LTY=subs(LTY,'laplace(y(t), t, s)','y(0)',Y,d)
else
d = input('The initial value at 0 is ');
e = input('The initial value at 0 is ');
LTY=subs(LTY,{'laplace(y(t), t, s)','y(0)','D(y)(0)'},{Y,d,e})
end
eq=collect(LTY,Y);
Y=simplify(solve(eq,Y));
y=simplify(ilaplace(Y,s,t))
```

```
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In experiment7_2 (line 4)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In experiment7_2 (line 5)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In experiment7_2 (line 6)
The Coefficient of D2y = 1
```

```
The Coefficient of Dy = 2
The Coefficient of y = 10
Enter the non homogenous part = 1+5*dirac(t-5)
The initial value at 0 is 1
The initial value at 0 is 2
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_2 (line 19)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_2 (line 19)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_2 (line 19)

LTY =

10*Y - s - 5*exp(-5*s) + 2*Y*s + Y*s^2 - 1/s - 4


y =

(9*cos(3*t)*exp(-t))/10 + (29*sin(3*t)*exp(-t))/30 + (5*heaviside(t - 5)*exp(5 - t)*sin(3*t
- 15))/3 + 1/10
```

Mathematical Model



Assumptions:

- Stress is proportional to strain. Thus, the equation is valid only for beams that are not stressed beyond the elastic limit.
- The curvature is always small.
- Any deflection resulting from the shear deformation of the material or shear stresses is neglected.
- For the deflected shape of the beam, the slope α at any point is defined as tan α = dy/dx . Assuming tan α = α we can write α = dy/dx .
- The curvature of a plane curve at a point can be expressed as: $\frac{1}{\rho} = \frac{y''}{(1+y'^2)^{3/2}}$
- In the elastic curve of beam dy/dx is very small so we can neglect its higher order terms. $1/\rho = d^2y/d^2x$
- From the theory of elasticity, if x is the distance of the section from the left end of the beam then $1/\rho = M(x)/EI$ where is the M-Bending moment, E-Modulus of Elasticity and I-Moment of inertia of the cross section.
- $d^2y/d^2x = -M(x)/EI$ - is the governing equation for an elastic curve.

When a beam supports a distributed load w(x) then dM/dx = V(Shear force) and , dV/dx = −w
Therefore $d^4y/d^4x = w(x)/EI$,
subjected to the boundary conditions $y(0) = y'(0) = y''(L) = y'''(L) = 0$.
Note. $y''(L) = 0$ because there is no bending moment and $y'''(L) = 0$ because there is no shear at that point.

MATLAB code:

```
clc;clear all

syms x s C D Y

y4 = diff(sym('y(x)'), 4);
y0 = sym('y(x)');

L = input('Enter the length of the beam: ');
E = input('Enter Modulus of elasticity: ');
I = input('Enter Moment of inertia of the cross section: ');
```

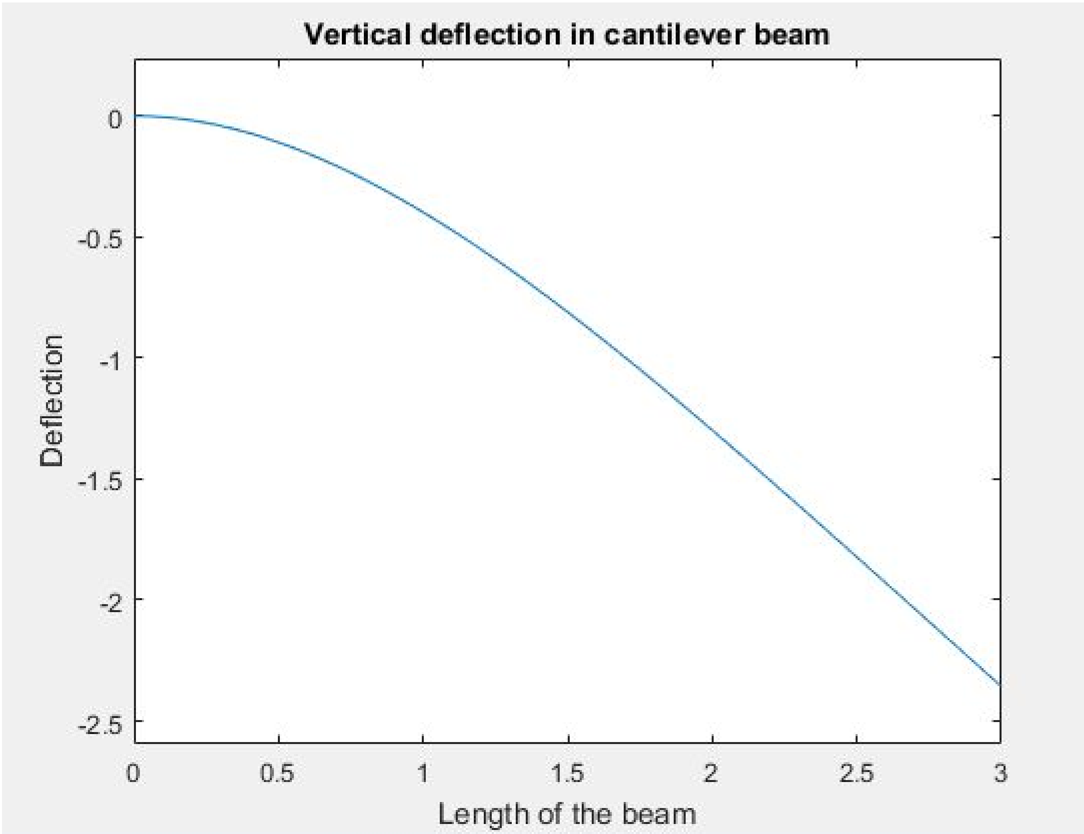
```
w = x% input('Enter distributive load w(x): ');

eqn = E*I*y4 - w;
LTY = laplace(eqn, x, s)

% a = input('Enter y(0): ');
% b = input('Enter y''(0): ');
% c = input('Enter y''''(L): ');
% d =  input('Enter y''''''(L): ');
a = 0; b = 0; c = 0; d = 0;

LTY = subs(LTY, {'laplace(y(x), x, s)', 'y(0)', 'D(y)(0)', 'D(D((y)))(0)',
'D(D(D((y))))(0)'}, {Y, a, b, C, D})

eq = collect(LTY, Y)
Y = simplify(solve(eq, Y))
y = simplify(ilaplace(Y, s, x))

eq1 = subs(diff(y, x, 2), x, L);
eq2 = subs(diff(y, x, 3), x, L);
[C, D] = solve(eq1, eq2)

def = subs(y);
gen = subs(def, heaviside(x - L), 0)

ezplot(-gen, [0, L])
title('Vertical deflection in cantilever beam')
xlabel('Length of the beam')
ylabel('Deflection')
```

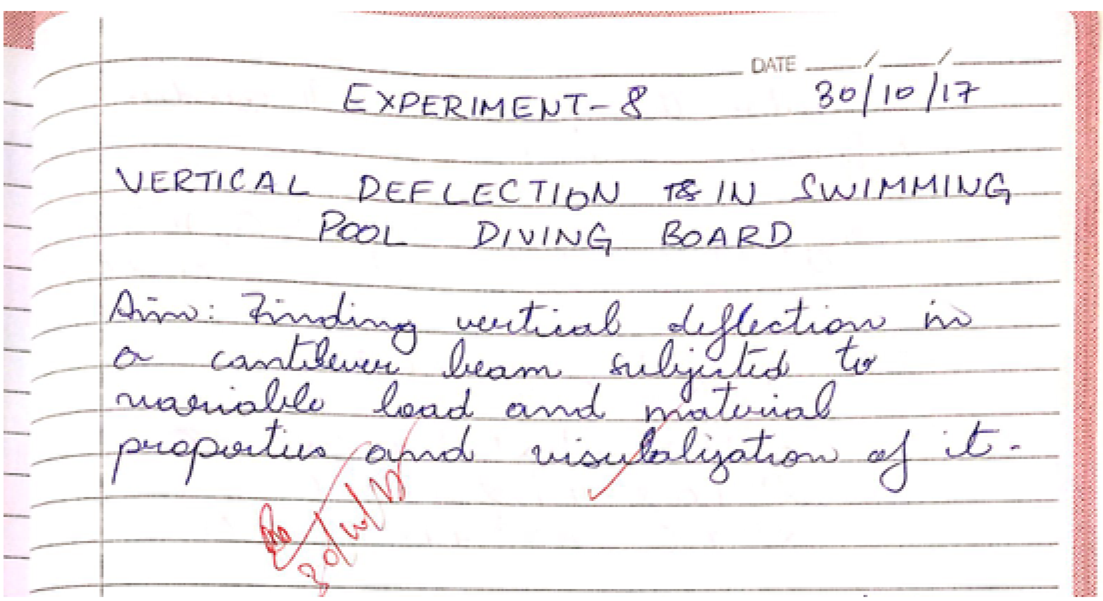
Output:

```
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In experiment7_3 (line 5)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In experiment7_3 (line 6)
Enter the length of the beam: 3
Enter Modulus of elasticity: 2.1*(10^11)
Enter Moment of inertia of the cross section: 4.5*(10^-11)

w =
```

```
x


LTY =

(189*s^4*laplace(y(x), x, s))/20 - (189*D(D(D(y)))(0))/20 - (189*s^3*y(0))/20 - 1/s^2 -
(189*s*D(D(y))(0))/20 - (189*s^2*D(y)(0))/20

Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_3 (line 22)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_3 (line 22)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_3 (line 22)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
```

```
  In experiment7_3 (line 22)
Warning: Support of character vectors that are not valid variable names or define a number
will be removed in a future
release. To create symbolic expressions, first create symbolic variables and then use
operations on them.
> In sym>convertExpression (line 1586)
  In sym>convertChar (line 1491)
  In sym>tomupad (line 1243)
  In sym (line 199)
  In sym/subs>@(x)sym(x) (line 156)
  In sym/subs>normalize (line 156)
  In sym/subs>mupadsubs (line 147)
  In sym/subs (line 135)
  In experiment7_3 (line 22)

LTY =

(189*Y*s^4)/20 - (189*C*s)/20 - (189*D)/20 - 1/s^2


eq =

((189*s^4)/20)*Y - (189*D)/20 - (189*C*s)/20 - 1/s^2


Y =

(189*C*s^3 + 189*D*s^2 + 20)/(189*s^6)


y =

(x^2*(x^3 + 189*D*x + 567*C))/1134


C =

20/21


D =

-10/21


gen =

(x^2*(x^3 - 90*x + 540))/1134

>>
```




DATE 30/10/17

EXPERIMENT-8

VERTICAL DEFLECTION ~~TS~~ IN SWIMMING POOL DIVING BOARD

Aim: Finding vertical deflection in a cantilever beam subjected to variable load and material properties and visualization of it.

30/10/17