Programming of Influence line Diagram for Indeterminate Beam using Muller Breslau Principle

Prepared in partial fulfillment of the

Study Project

Course No. - **CE F266**

Submitted By | **Prepared for**

Samarth Joshi | **DR. S.N. Patel**

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(JUNE,2021)

Acknowledgement

I extend my sincere thanks to **Dr. Shuvendu Patel** for providing the opportunity to undertake this project. His guidance and project have helped to understand Structural Analysis better and in-depth understanding programming of influence line diagram for beams. I am deeply grateful to him for his guidance throughout the project and constant motivation in pursuing the study.

Table of Content

Introduction

The project shows the computation of the influence line diagram using the Muller Breslau principle for indeterminate beams and to build a code to compute values and graphs. According to the **Müller-Breslau principle**, an influence line will be a scaled form of the deflected beam shape, when resistance to the response is removed at the analysis point, and a relative unit deflection is applied for the shear influence line, or a unit rotation is applied for the moment influence line is introduced at the analysis point to determine an influence line. A fictitious roller guide is applied to remove shear resistance, which allows vertical deflection at the analysis point while maintaining resistance to rotation. A fictitious hinge is applied to remove moment resistance. The hinge enables rotation at the analysis point with zero relative displacements. The resulting deflected shape provides the influence line for a unit point load.

There are various approaches to solve the beam's deflection after applying the Muller Breslau principle.

- Castigliano's method
- Moment-area theorem
- Conjugate beam method
- Double Integration
- Force Method

The double integration method is used for calculating bending, and there is another approach used in problems 3 and 4 by taking deflection of the beam as a cubic equation then assigning the boundary condition to the equations to find the coefficients of equations. The second approach makes all the calculations much easier in coding and calculation perspective.

The first four problems contain manual calculation and results in terms of the cubic equation for different intervals and conditions. The final results provide the solution for any user-defined number of supports.

Java and JavaFX are handled for calculation and graph representation. Users can give inputs in terms of length of the beam, distance of support, and distance analysis point. The output will be a graphical line chart representation with thousand intervals and modified to find any position. All calculations were verified using STAAD Pro.

Chapter: 1

Influence line diagram of shear for simply supported beam

To find an influence line diagram of shear force for the point at distance “K” from the fixed end “A “(as shown in fig. 1).

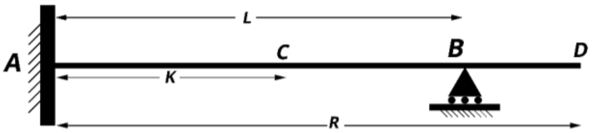


Fig. 1 Cantilever, simply supported Beam

R = Length of Beam,

L = Distance of support from origin from fixed end,

K = Distance of point for which ILD is calculated from fixed end,

EI = flexural rigidity of beam (assumed to be constant)

Solution: The solution will be in three parts for different position of K.

Applying Müller-Breslau principle,

Part 1: K=0,

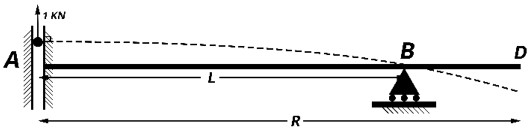


Fig. 2 Influence Line to the shear at C is to the same scale as deflected shape.

When x varies from 0 to L i.e., **$0 \leq x < L$**

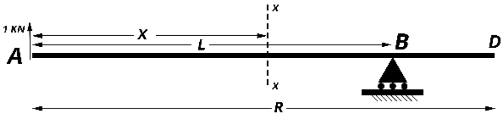


Fig. 3 Moment at x distance from origin

By Double Integration Method,

M_x= - (L-x)

EI $\frac{d^2y}{dx^2}$ = - (L-x)

EI $\frac{dy}{dx}$ = -Lx + $\frac{x^2}{2}$ + C_1 ..... (1)

EI y(x) = -L $\frac{x^2}{2}$ + $\frac{x^3}{6}$ + C_1.x+ C_2 ..... (2)

Putting boundary condition in the equation (1) and (2)

EI$\frac{dy}{dx}$ at x=0 is 0 as there is fixed support. Hence, we get C_1= 0

Again, y(0)=0 as there is fixed support. Hence, we get C_2= $\frac{L^3}{3}$

EI y(x) = -L $\frac{x^2}{2}$ + $\frac{x^3}{6}$ + $\frac{L^3}{3}$ ..... (3)

When x is varies from L to R i.e., $\mathbf{L \leq x < R}$

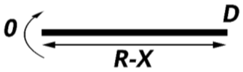


Fig. 4 Moment at x distance from origin

By Double Integration Method,

M_x=0 D

EI $\frac{d^2y}{dx^2}$ = 0

EI $\frac{dy}{dx}$ = C_5 ..... (4)

EI y(x) = C_5.x + C_6 ..... (5)

Using boundary condition,

When x=L, slope in equation (4) is equal to that of equation (1). Hence, we get C_5= - $\frac{L^2}{2}$

When x=L, y(L)=0 as there no displacement in y in equation (5) at a support, C_6= $\frac{L^3}{2}$

EI y(x) = - $\frac{L^2}{2}$x + $\frac{L^3}{2}$ ..... (6)

To find the function of ILD we have to divide the function in equation (3), and (6) by scale factor which is equal to y(L) in any of equation (3) or (6)

Scale factor $= \frac{L^3}{3EI}$

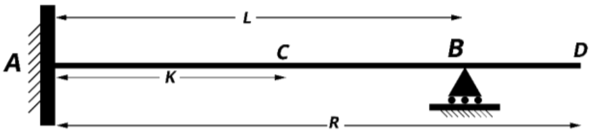
Hence, the **Influence Line Diagram for shear at C** is

$y(x) = (-L\frac{x^2}{2} + \frac{x^3}{6} + \frac{L^3}{3})\frac{3}{L^3}$ for $0< x <L$

$y(x) = (-\frac{L^2}{2}x + \frac{L^3}{2})\frac{3}{L^3}$ for $L< x <R$

Results from Application

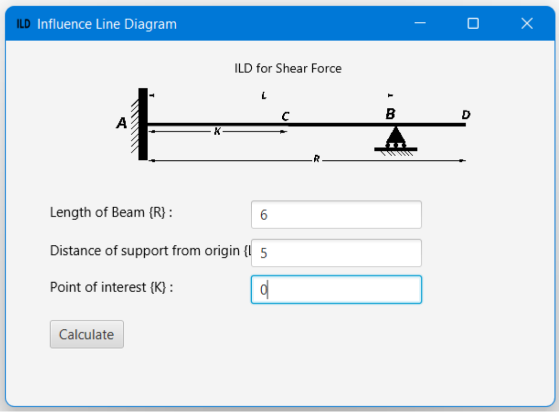
Example : **K=0,**



Length of Beam(R) = 6

Distance between support(L) = 5

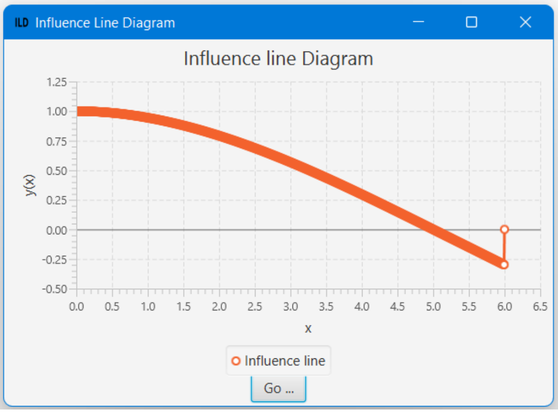
Point of interest(K) = 0



Length of Beam {R} : 6

Distance of support from origin {l 5

Point of interest {K} : 0



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 0.

scale factor = 41.666668

shear force at **0.0 = 1.0**

shear force at 0.006 = 0.9999978

shear force at 0.012 = 0.9999915

Part 2: 0<K<L

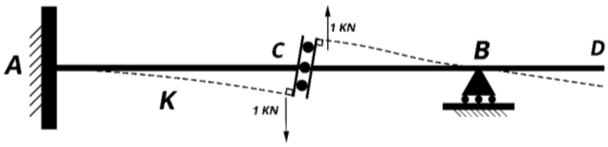


Fig. 5 Influence Line to the shear at C is to the same scale as deflected shape.

When x varies from 0 to K i.e., $\mathbf{0 \leq x < K}$

By Double Integration Method,

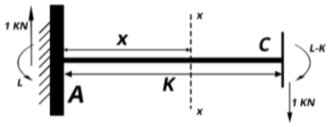


Fig. 6 Moment at x distance from origin

M_x=x-L

$$EI\ \frac{d^2y}{dx^2} = x - L$$

$$EI\,\frac{dy}{dx} = \frac{x^2}{2} - Lx + C_1 \qquad ..... (1)$$

$$EI\ y(x) = \frac{x^3}{6} - L\frac{x^2}{2} + C_1 x + C_2 \qquad ..... (2)$$

Putting boundary condition in the equation (1) and (2)

$EI\frac{dy}{dx}$ at x=0 is 0 as there is fixed support. Hence, we get $C_1=0$.

Again, y(0)=0 as there is fixed support. Hence, we get $C_2=0$.

$$EI\ y(x) = \frac{x^3}{6} - L\frac{x^2}{2} \qquad ..... (3)$$

When x is varies from K to L i.e., $\mathbf{K \leq x < L}$

By Double Integration Method,

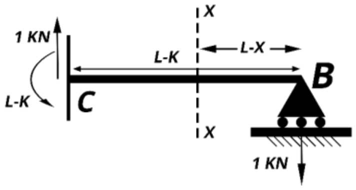


Fig. 7 Moment at x distance from origin

M_x=x-L

EI $\frac{d^2y}{dx^2}$ = x - L

EI $\frac{dy}{dx}$ = $\frac{x^2}{2}$ – Lx + C_3 ..... (4)

EI y(x) = $\frac{x^3}{6}$ - L$\frac{x^2}{2}$ +C_3.x+C_4 ..... (5)

Using boundary condition,

When x=K, slope in equation (1) is equal to that of equation (4). Hence, we get C_3=C_1=0.

When x=L, y(L)=0 as there no displacement in y in equation (5) at roller support, C_4=$\frac{L^3}{3}$.

EI y(x) = $\frac{x^3}{6}$ - L$\frac{x^2}{2}$ + $\frac{L^3}{3}$ ..... (6)

When x is varies from L to R i.e., **L≤ x < R**

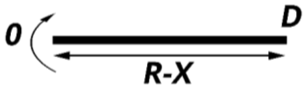


Fig. 8 Moment at x distance from origin

By Double Integration Method,

M_x=0

EI $\frac{d^2y}{dx^2}$ = 0

EI $\frac{dy}{dx}$ = C_5 ..... (7)

EI y(x) = C_5.x + C_6 ..... (8)

Using boundary condition,

When x=L, slope in equation (7) is equal to that of equation (4). Hence, we get C_5= - $\frac{L^2}{2}$

When x=L, y(L)=0 as there no displacement in y in equation (8) at a support, $C_6 = \frac{L^3}{2}$.

$$EI\ y(x) = -\frac{L^2}{2}x + \frac{L^3}{2} \qquad ..... (9)$$

To find the function of ILD we have to divide the function in equation (3), (6) and (9) by scale factor which is equal to addition of y(k) from equation (3) and (6).

$$\text{Scale factor} = \frac{L^3}{3EI}$$

Hence, the **Influence Line Diagram for shear at C** is

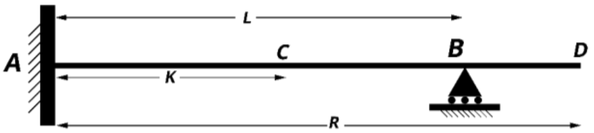
$$y(x) = \left(\frac{x^3}{6} - L\frac{x^2}{2}\right)\frac{3}{L^3} \qquad \text{for } 0 < x < K$$

$$y(x) = \left(\frac{x^3}{6} - L\frac{x^2}{2} + \frac{L^3}{3}\right)\frac{3}{L^3} \qquad \text{for } K < x < L$$

$$y(x) = \left(-\frac{L^2}{2}x + \frac{L^3}{2}\right)\frac{3}{L^3} \qquad \text{for } L < x < R$$

Results from Application

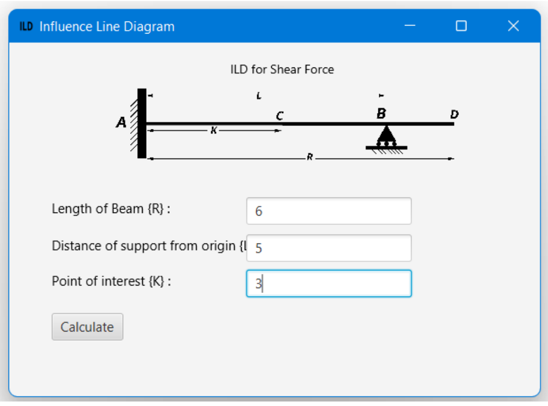
Example : **0<K<L,**



Length of Beam(R) = 6

Distance between support(L) = 5

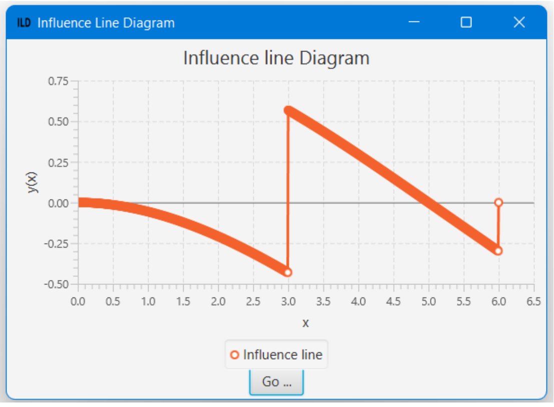
Point of interest(K) = 3



Length of Beam {R} : 6

Distance of support from origin {L 5

Point of interest {K} : 3



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

shear force at x = 2.9880128 is -0.4289827

shear force at **2.9940128 = -0.43049207**

shear force at **3.0000129 = 0.5679967**

shear force at 3.006013 = 0.5664839

Part 3: K=L,

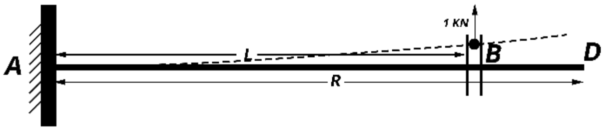


Fig. 9 Influence Line to the shear at C is to the same scale as deflected shape.

When x varies from 0 to L i.e., $\mathbf{0 \leq x < L}$

By Double Integration Method,

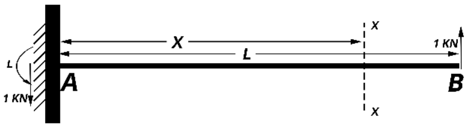


Fig. 10 Moment at x distance from origin

M_x= -(x-L)

$EI\ \frac{d^2y}{dx^2} = -(x-L)$

$EI\frac{dy}{dx} = -\frac{x^2}{2} + Lx + C_1$ ..... (1)

$EI\ y(x) = -\frac{x^3}{6} + L\frac{x^2}{2} + C_1.x + C_2$ ..... (2)

Putting boundary condition in the equation (1) and (2)

$EI\frac{dy}{dx}$ at x=0 is 0 as there is fixed support. Hence, we get C_1=0.

Again, $y(0)=0$ as there is fixed support. Hence, we get $C_2=0$.

$$EI\ y(x) = -\frac{x^3}{6} + L\frac{x^2}{2} \quad ..... (3)$$

When x is varies from L to R i.e., $\mathbf{L \leq x < R}$

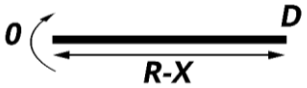


Fig. 11 Moment at x distance from origin

By Double Integration Method,

$M_x=0$

$$EI\ \frac{d^2y}{dx^2} = 0$$

$$EI\frac{dy}{dx} = C_5 \qquad ..... (4)$$

$$EI\ y(x) = C_5.x + C_6 \quad ..... (5)$$

Using boundary condition,

When x=L, slope in equation (4) is equal to that of equation (1). Hence, we get $C_5= \frac{L^2}{2}$

When x=L, y(L)=0 as there no displacement in y in equation (5) at a support, $C_6= -\frac{L^3}{6}$

$$EI\ y(x) = \frac{L^2}{2}x - \frac{L^3}{6} \qquad ..... (6)$$

To find the function of ILD we have to divide the function in equation (3), and (6) by scale factor which is equal to y(L) in any of equation (3) or (6)

$$\text{Scale factor} = \frac{L^3}{3EI}$$

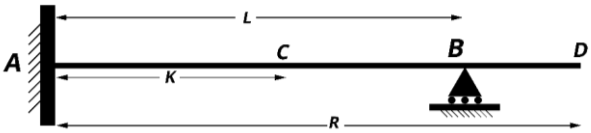
Hence, the **Influence Line Diagram for shear at C** is

$$y(x) = (-\frac{x^3}{6} + L\frac{x^2}{2})\frac{3}{L^3} \quad \text{for } 0< x <L$$

$$y(x) = (\frac{L^2}{2}x - \frac{L^3}{6})\frac{3}{L^3} \quad \text{for } L< x <R$$

Results from Application

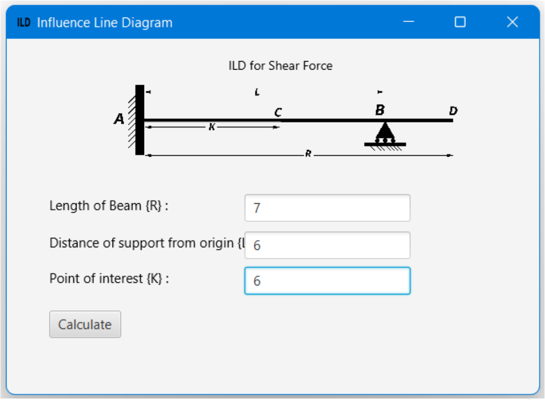
Example : **K=L,**



Length of Beam(R) = 7

Distance between support(L) = 6

Point of interest(K) = 6

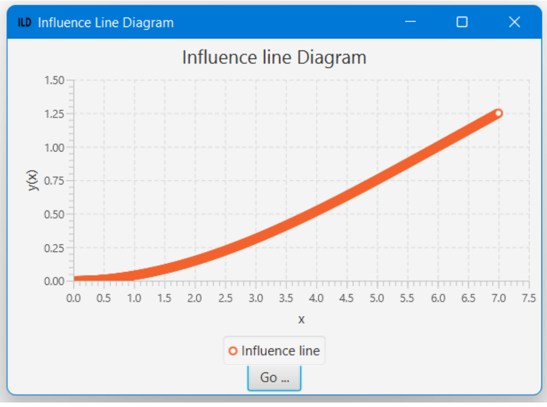


ILD for Shear Force

Length of Beam (R) : 7

Distance of support from origin (l 6

Point of interest (K) : 6



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 6.

scale factor= 41.666668

shear force at 5.9919806 = 0.997995

shear force at **5.9989805 = 0.9997451**

shear force at **6.0059805 = 1.0014951**

shear force at 6.0129805 = 1.0032452

Chapter: 2

Influence line diagram of shear for continuous beam

To find an influence line diagram of shear force for the point at distance "K" from the end "A "(as shown in Fig. 1).

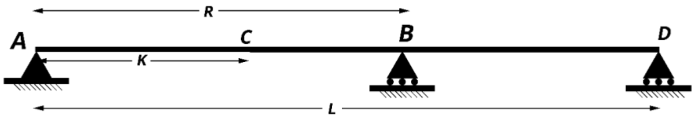


Fig. 1 Continuous three support beam

L = Length of Beam,

R = Distance of middle support from origin from end A,

K = Distance of point for which ILD is calculated from end A,

EI = flexural rigidity of beam (assumed to be constant)

Solution: The solution will be in five parts for different position of K.
Applying Müller-Breslau principle,

PART 1: If 0<K<R,

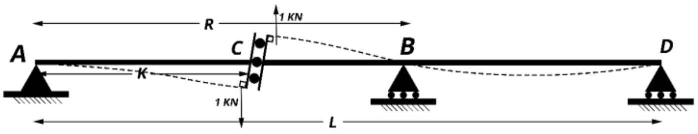


Fig. 2 Removing the support at C and applying the positive shear at C yields the deflected shape. Influence Line to the shear at C is to the same scale as deflected shape.

When x is varies from R to L i.e., $\mathbf{R \leq x < L}$

By Double Integration Method,

$M_x = \frac{R}{L-R}(L\text{-}x)$

$EI\ \frac{d^2y}{dx^2} = \frac{R}{L-R}(L\text{-}x)$

$EI\frac{dy}{dx} = \frac{R}{L-R}(Lx\text{-}\frac{x^2}{2}) + C_5$ ….. (1)

$$\mathrm{EI}\ y(x) = \frac{R}{L-R}\,(L\frac{x^2}{2} - \frac{x^3}{6}) + C_5.x + C_6 \qquad ..... (2)$$

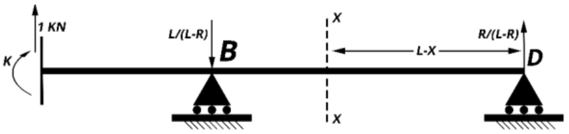


Fig. 3 Moment at x distance from origin

Using boundary condition,

When x=L and x=R the value of y(L)=0 as there no displacement in y in equation (2) at a support.

We get,

$$\frac{R}{L-R}\,(\frac{L^2}{2} - \frac{L^3}{6}) + C_5.L + C_6 = 0$$

$$\frac{R}{L-R}\,(L\,\frac{R^2}{2} - \frac{R^3}{6}) + C_5.L + C_6 = 0$$

Solving the above equations. we get,

$$C_5 = \frac{R}{L-R}\ (L\frac{R^2}{2} - \frac{R^3}{6} - \frac{L^3}{3})$$

$$C_6 = - \frac{R}{L-R}\,(\frac{L^3}{3}) - C_5.L$$

$$\mathrm{EI}\ y(x) = = \frac{R}{L-R}\,(L\frac{x^2}{2} - \frac{x^3}{6}) + C_5.x\ - \frac{R}{L-R}\,(\frac{L^3}{3}) - C_5.L \qquad ..... (3)$$

When x is varies from K to R i.e., $\mathbf{K < x < R}$

By Double Integration Method,

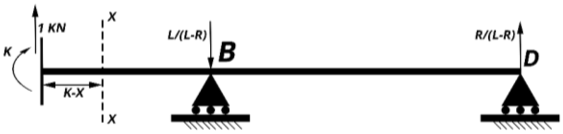


Fig. 4 Moment at x distance from origin

M_x=x

$$\mathrm{EI}\ \frac{d^2y}{dx^2} = x$$

$EI\frac{dy}{dx} = \frac{x^2}{2} + C_3$ ..... (4)

$EI\,y(x) = \frac{x^3}{6} + C_3.x + C_4$ ..... (5)

Using boundary condition,

When x=L, slope in equation (1) is equal to that of equation (4),

we get $C_3 = \frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2}$

When x=L, y(L)=0 as there no displacement in y in equation (7) at roller support,

We get $C_4 = -\frac{R^3}{6} - \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2}\}.R$

$EI\,y(x) = \frac{x^3}{6} + \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2} + C_5 - \frac{R^2}{2}\}.x + -\frac{R^3}{6} - \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2}\}.R$ ..... (6)

When x varies from 0 to K i.e., **$0 \leq x < K$**

By Double Integration Method,

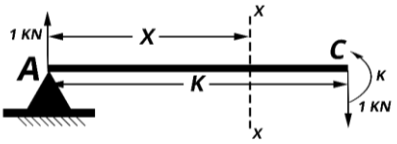


Fig. 5 Moment at x distance from origin

$M_x = x$

$EI\frac{d^2y}{dx^2} = x$

$EI\frac{dy}{dx} = \frac{x^2}{2} + C_1$ ..... (7)

$EI\,y(x) = \frac{x^3}{6} + C_1.x + C_2$ ..... (8)

Putting boundary condition in the equation (7) and (8),

EI y(x)=0 at x=0 as there is a support, we get $C_2 = 0$.

When x=K, slope in equation (4) is equal to that of equation (7), we get $C_1 = C_3$.

$EI\,y(x) = \frac{x^3}{6} + \{\frac{R}{L-R}(L\frac{R^2}{2} - \frac{R^3}{6}) + C_5 - \frac{R^2}{2}\}.x$ ..... (9)

To find the function of ILD we have to divide the function in equation (3), (6) and (9) by scale factor which is equal to addition of y(k) from equation (6) and (9).

$$\text{Scale factor} = \frac{\frac{K^3}{6} + \{\frac{R}{L-R}(L\frac{R^2}{2} - \frac{R^3}{6}) + C_5 - \frac{R^2}{2}\}.K + \frac{K^3}{6} + \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2} + C5 - \frac{R^2}{2}\}.K + -\frac{R^3}{6} - \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2}\}.R}{EI}$$

Hence, the **Influence Line Diagram for shear at C from equation** (3), (6) and (9) is

$$y(x) = \frac{\frac{x^3}{6} + \{\frac{R}{L-R}(L\frac{R^2}{2} - \frac{R^3}{6}) + C_5 - \frac{R^2}{2}\}.x}{\text{Scale factor . EI}} \qquad \text{for } 0< x <K$$

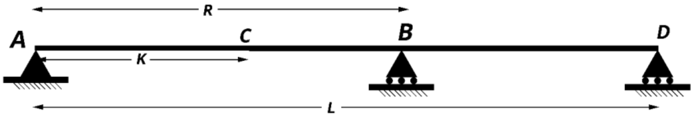
$$y(x) = \frac{\frac{x^3}{6} + \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2} + C5 - \frac{R^2}{2}\}.x - \frac{R^3}{6} - \{\frac{R}{L-R}(LR - \frac{R^2}{2}) + C_5 - \frac{R^2}{2}\}.R}{\text{Scale factor . EI}} \qquad \text{for } K< x <R$$

$$y(x) = \frac{\frac{R}{L-R}(L\frac{x^2}{2} - \frac{x^3}{6}) + C_5 .x - \frac{R}{L-R}(\frac{L^3}{3}) - C_5.L}{\text{Scale factor . EI}} \qquad \text{for } R< x <L$$

$$\text{where } C_5 = \frac{R}{L-R}\ (L\frac{R^2}{2} - \frac{R^3}{6} - \frac{L^3}{3})$$

Results from Application

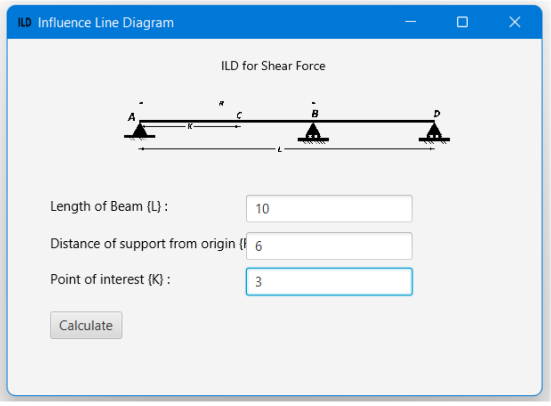
Example : 0<K<R



Distance between support(L) = 10

Length of Beam(R) = 6

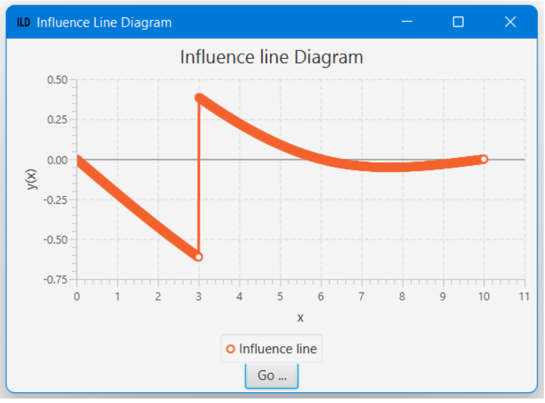
Point of interest(K) = 3



Length of Beam {L} : 10

Distance of support from origin {I 6

Point of interest {K} : 3



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

shear force at 2.9899976 = -0.6107066

shear force at **2.9999976 = -0.61249954**

shear force at **3.0099976 = 0.38571015**

shear force at 3.0199976 = 0.38392207

PART 2: K = R

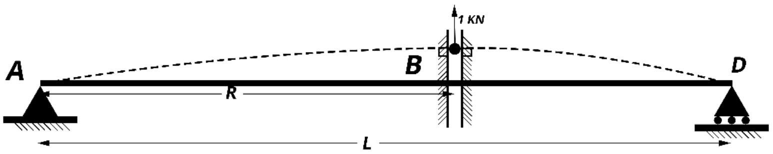


Fig. 6. Placing the roller guide at B and applying the positive shear at B yields the deflected shape. Influence Line to the shear at B is to the same scale as deflected shape

When x is varies from 0 to R i.e., $0 \leq x < R$

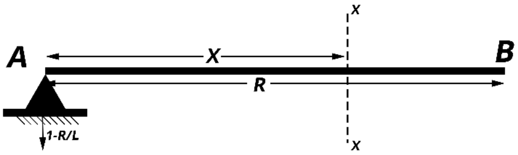


Fig. 7 Moment at x distance from origin

By Double Integration Method,

$M_x = -x.(1-\frac{R}{L})$

$EI \frac{d^2y}{dx^2} = -x.(1-\frac{R}{L})$

$EI \frac{dy}{dx} = -x^2.(1-\frac{R}{L})+C_1$ ..... (1)

$EI\, y(x) = -x^3.(1-\frac{R}{L})+C_1.x+C_2$ ..... (2)

Using boundary condition,

When x=0 the value of y(0)=0 as there no displacement in y in equation (2) as it is at a support.

We get, $C_2=0$.   ..... (3)

When x is varies from R to L i.e., $\mathbf{R< x < L}$

By Double Integration Method,

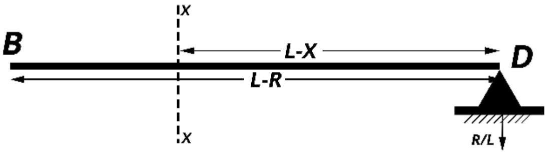


Fig. 8 Moment at x distance from origin

M_x= -(L-x).$\frac{R}{L}$

EI $\frac{d^2y}{dx^2}$ = -(L-x).$\frac{R}{L}$

EI $\frac{dy}{dx}$ = -(Lx $-\frac{x^2}{2}$).$\frac{R}{L}$ + C_3.x   ..... (4)

EI y(x) = -(L$\frac{x^2}{2}-\frac{x^3}{6}$).$\frac{R}{L}$ + C_3.x +C_4 ..... (5)

When x=L the value of y(L)=0 as there no displacement in y in equation (3) as it is at a support.

We get,

C_3.L +C_4 = R.$\frac{L^2}{3}$ ..... (6)

Also,

When x=R, slope in equation (1) is equal to that of equation (4) and also y(x) will be equal in equation (2) and (5),

We get - $\frac{R^3}{6}$ (1 - $\frac{R}{L}$) + C_1R = -(L$\frac{x^2}{2}-\frac{x^3}{6}$).$\frac{R}{L}$ + C_3.x +C_4 ..... (7)

-(Lx $-\frac{x^2}{2}$).$\frac{R}{L}$ + C_3.x = -x^2.(1- $\frac{R}{L}$)+C_1 ..... (8)

From the above equations (3), (6), (7) and (8)

$C_1 = \frac{RL}{3} + \frac{R^3}{6L}$ - $\frac{R^2}{2}$

$C_2 = 0$

$C_3 = \frac{RL^2}{3} - \frac{R^3}{6}$

$C_4 = - \frac{R^3}{6}$

Hence,

$y(x) = -x^3.(1-\frac{R}{L})+(\frac{RL}{3}+\frac{R^3}{6L}-\frac{R^2}{2}).x$ for $0< x <R$ ..... (9)

$y(x) = -\left(\frac{Lx^2}{2}-\frac{x^3}{6}\right)\frac{R}{L} + (\frac{RL^2}{3}-\frac{R^3}{6}).x - \frac{R^3}{6}$ for $R< x <L$ ..... (10)

To find the function of ILD we have to divide the function in equation (9) and (10) by scale factor which is equal to addition of y(k) from equation (9) and (10).

$$\text{Scale factor} = \frac{-R^3.\left(1-\frac{R}{L}\right)+\left(\frac{RL}{3}-\frac{R^3}{6L}-\frac{R^2}{2}\right).R}{EI}$$

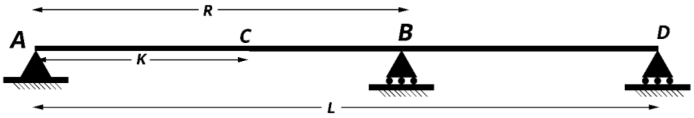
Hence, the **Influence Line Diagram for shear at C from equation** (9), and (10) is

$y(x) = \frac{-x3.(1-\frac{R}{L})+(\frac{RL}{3}+\frac{R^3}{6L}-\frac{R^2}{2}).x}{\text{Scale factor . EI}}$ for $0< x <R$

$y(x) = \frac{-\left(\frac{Lx^2}{2}-\frac{x^3}{6}\right)\frac{R}{L}+(\frac{RL^2}{3}-\frac{R^3}{6}).x-\frac{R^3}{6}}{\text{Scale factor . EI}}$ for $R< x <L$

Results from Application

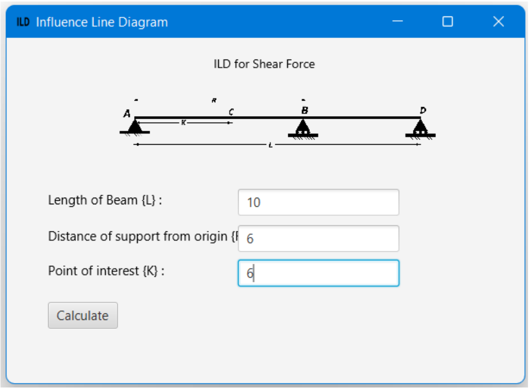
Example : **K=R**



Distance between support(L) = 10

Length of Beam(R) = 6

Point of interest(K) = 3

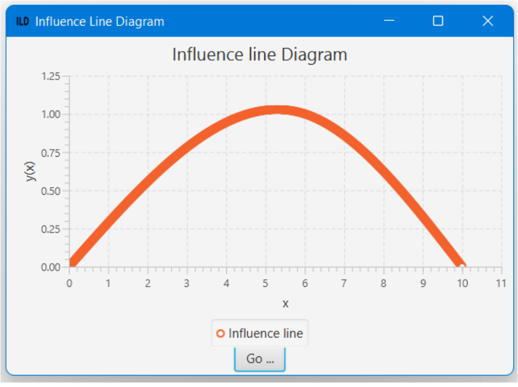


ILD for Shear Force

Length of Beam {L} : 10

Distance of support from origin {F 6

Point of interest {K} : 6



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 6.

shear force at 5.980042 = 1.0016382

shear force at **5.990042 = 1.0008235**

shear force at **6.0000424 = 0.99999565**

shear force at 6.0100427 = 0.9991564

PART 3: R< K <L,

By Similarity we can get ILD by replacing x to (L-x), R to (L-R) and K to (L-K) in the part 2.

We get,

Scale factor =

$$\frac{\frac{K^3}{6}+\{\frac{(L-R)}{L-(L-R)}(L\frac{(L-R)^2}{2}-\frac{(L-R)^3}{6})+C_5-\frac{(L-R)^2}{2}\}.K+\frac{K^3}{6}+\{\frac{(L-R)}{L-(L-R)}(L(L-R)-\frac{(L-R)^2}{2})+C_5-\frac{(L-R)^2}{2}+C5-\frac{(L-R)^2}{2}\}.K+-\frac{(L-R)^3}{6}-\{\frac{(L-R)}{L-(L-R)}(L(L-R)-\frac{(L-R)^2}{2})+C_5-\frac{(L-R)^2}{2}\}.(L-R)}{EI}$$

Hence, the **Influence Line Diagram for shear at C from equation** (8), (14) and (18) is

$$y(x) = \frac{\frac{(L-x)^3}{6}+\{\frac{(L-R)}{L-(L-R)}(L\frac{(L-R)^2}{2}-\frac{(L-R)^3}{6})+C_5-\frac{(L-R)^2}{2}\}.(L-x)}{\text{Scale factor . EI}} \qquad \text{for } K< x <L$$

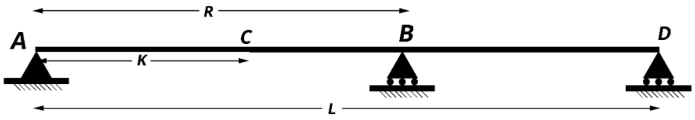
$$y(x) = \frac{\frac{(L-x)^3}{6}+\{\frac{(L-R)}{L-(L-R)}(L(L-R)-\frac{(L-R)^2}{2})+C_5-\frac{(L-R)^2}{2}+C5-\frac{(L-R)^2}{2}\}.(L-x)-\frac{(L-R)^3}{6}-\{\frac{(L-R)}{L-(L-R)}(L(L-R)-\frac{(L-R)^2}{2})+C_5-\frac{(L-R)^2}{2}\}.(L-R)}{\text{Scale factor . EI}} \qquad \text{for } R< x < K$$

$$y(x) = \frac{\frac{(L-R)}{L-(L-R)}(L\frac{(L-x)^2}{2}-\frac{(L-x)^3}{6})+C_5.(L-x)-\frac{(L-R)}{L-(L-R)}(\frac{L^3}{3})-C_5.L}{\text{Scale factor . EI}} \qquad \text{for } 0< x <R$$

where $C_5 = \frac{(L-R)}{L-(L-R)}\ (L\frac{(L-R)^2}{2} - \frac{(L-R)^3}{6} - \frac{L^3}{3})$

Results from Application

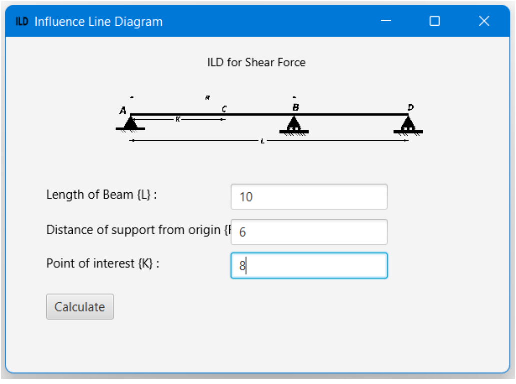
Example : : R< K <L,



Distance between support(L) = 10

Length of Beam(R) = 6

Point of interest(K) = 3



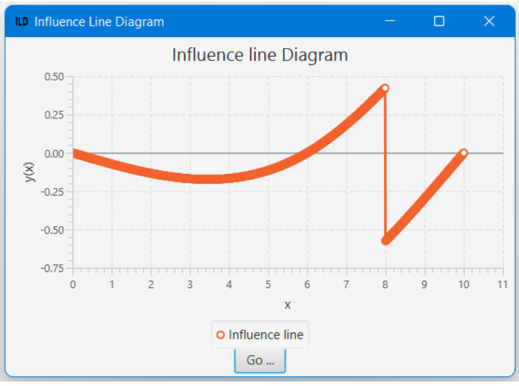
ILD for Shear Force

Length of Beam (L) : 10

Distance of support from origin (I 6

Point of interest (K) : 8

Calculate



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 8.

shear force at 7.9800878 = 0.41978043

shear force at **7.990088 = 0.42239985**

shear force at **8.000088 = -0.5749769**

shear force at 8.010088 = -0.57234997

PART 4: K = L,

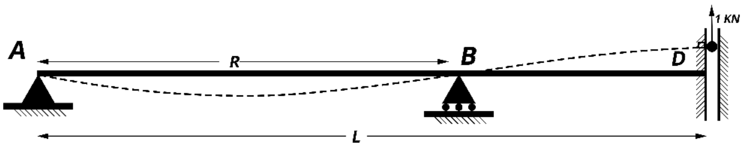


Fig. 9 Placing the roller guide at C and applying the positive shear at C yields the deflected shape. Influence Line to the shear at C is to the same scale as deflected shape.

When x is varies from 0 to R i.e., $\mathbf{0 \leq x < R}$

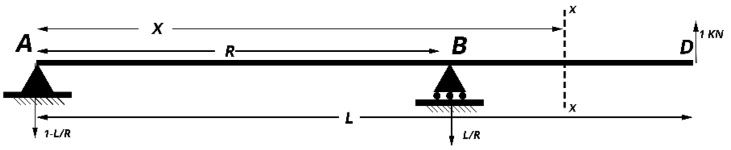


Fig. 10 Moment at x distance from origin

By Double Integration Method,

M_x = -x.(1- $\frac{R}{L}$)

EI $\frac{d^2y}{dx^2}$ = -x.(1- $\frac{R}{L}$)

EI $\frac{dy}{dx}$ = -x^2.(1- $\frac{R}{L}$)+C$_1$ ..... (1)

EI y(x) = -x^3.(1- $\frac{R}{L}$)+C$_1$.x+C$_2$ ..... (2)

Using boundary condition,

When x=0 and x=R the value of y(0)=0 as there no displacement in y in equation (2) as it is at a support.

We get,

C$_1$ = (1 - $\frac{L}{R}$).$\frac{R^2}{6}$

C$_2$ = 0

Hence,

EI y(x) = -x^3.(1- $\frac{R}{L}$)+ (1 - $\frac{L}{R}$)$\frac{R^2x}{6}$ ..... (3)

When x is varies from R to L i.e., $\mathbf{R < x < L}$

By Double Integration Method,

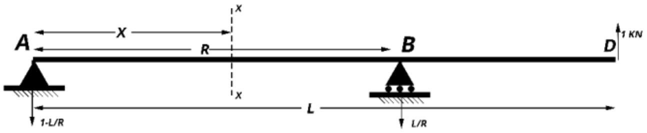


Fig. 11 Moment at x distance from origin

M_x= (L-x)

EI $\frac{d^2y}{dx^2}$ = (L-x)

EI $\frac{dy}{dx}$ = Lx - $\frac{x^2}{2}$ + C_3 ..... (4)

EI y(x) = $(L\frac{x^2}{2} - \frac{x^3}{6})$ + C_3.x +C_4 ..... (5)

When x=R the value of y(L)=0 as there no displacement in y in equation (3) as it is at a support.

We get,

C_3.R +C_4 = - $(L\frac{R^2}{2} - \frac{R^3}{6})$ ..... (6)

Also, When x=R, slope in equation (10) is equal to that of equation (16) and also y(x) will be equal,

We get,

Lx - $\frac{x^2}{2}$ + C_3 = -x^2.(1- $\frac{R}{L}$)+C_1 ..... (7)

From the above equations (6) and (7),

$C_3 = \frac{R^2}{6}$ - 2L$\frac{R}{3}$

$C_4 = L\frac{R^2}{6}$

Hence,

y(x) = - $\frac{x^3}{6}$.(1- $\frac{L}{R}$)+($(1-\frac{L}{R}).\frac{R^2}{6}$).x for 0< x <R ..... (8)

y(x) = $\left(\frac{Lx^2}{2} - \frac{x^3}{6}\right)$+ ($\frac{R^2}{6}$ - 2L$\frac{R}{3}$).x + L$\frac{R^2}{6}$ for R< x <L ..... (9)

To find the function of ILD we have to divide the function in equation (8) and (9) by scale factor which is equal to addition of y(k) from equation (8) and (9).

$$\text{Scale factor} = \frac{\left(\frac{LR^2}{2} - \frac{R^3}{6}\right) + \left(\frac{R^2}{6} - 2L\frac{R}{3}\right).R + L\,\frac{R^2}{6}}{EI}$$

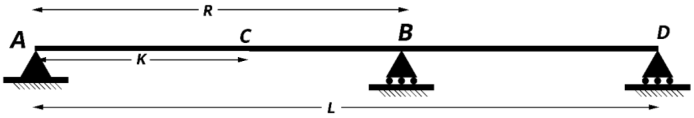
Hence, the **Influence Line Diagram for shear at C from equation** (8), and (9) is

$$y(x) \ = \frac{-\frac{x^3}{6}.(1-\frac{L}{R})+(\ (1-\frac{L}{R}).\frac{R^2}{6}).x}{\text{Scale factor . EI}} \qquad \text{for } \ 0< x <R$$

$$y(x) = \frac{\left(\frac{Lx^2}{2} - \frac{x^3}{6}\right) + \left(\frac{R^2}{6} - 2L\frac{R}{3}\right).x + L\,\frac{R^2}{6}}{\text{Scale factor . EI}} \qquad \text{for } \ R< x <L$$

Results from Application

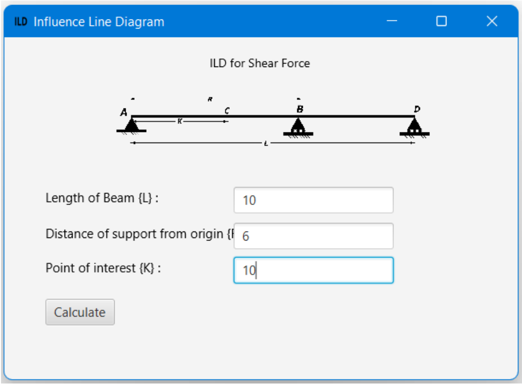
Example : : K=L,



Distance between support(L) = 10

Length of Beam(R) = 6

Point of interest(K) = 10

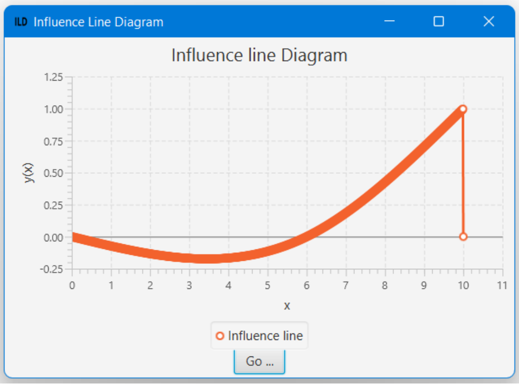


ILD for Shear Force

Length of Beam (L) : 10

Distance of support from origin (I 6

Point of interest (K) : 10



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 10.

shear force at 9.960133 = 0.9880403

shear force at 9.970133 = 0.9910404

shear force at 9.980133 = 0.9940405

shear force at **9.990133 = 0.99704057**

PART 5: If K=0,

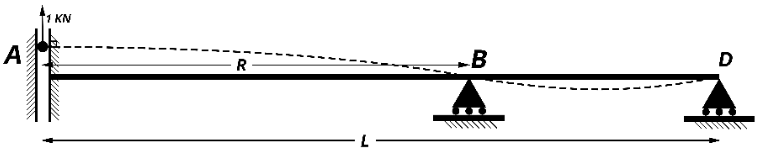


Fig. 12 Deflected shape after applying muller Breslau principle

By Similarity we can get ILD by replacing x to (L-x), R to (L-R) and K to (L-K) in the part 4.

We get,

$$\text{Scale factor} = \frac{\left(\frac{L(L-R)^2}{2} - \frac{(L-R)^3}{6}\right) + (\frac{(L-R)^2}{6} - 2L\frac{(L-R)}{3}).(L-R) + L\,\frac{(L-R)^2}{6}}{EI}$$

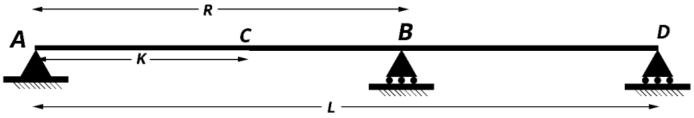
Hence, the **Influence Line Diagram for shear at C from equation** (8), and (9) is

$$y(x) = \frac{-\frac{(L-x)^3}{6}.(1-\frac{L}{(L-R)})+(\,(1-\frac{L}{(L-R)}).\frac{(L-R)^2}{6}).(L-x)}{\text{Scale facto}(L-R)\,.\;EI} \qquad \text{for } R< x <L$$

$$y(x) = \frac{\left(\frac{L(L-x)^2}{2} - \frac{(L-x)^3}{6}\right) + (\frac{(L-R)^2}{6} - 2L\frac{(L-R)}{3}).(L-x) + L\,\frac{(L-R)^2}{6}}{\text{Scale facto}(L-R)\,.\;EI} \qquad \text{for } 0< x <R$$

Results from Application

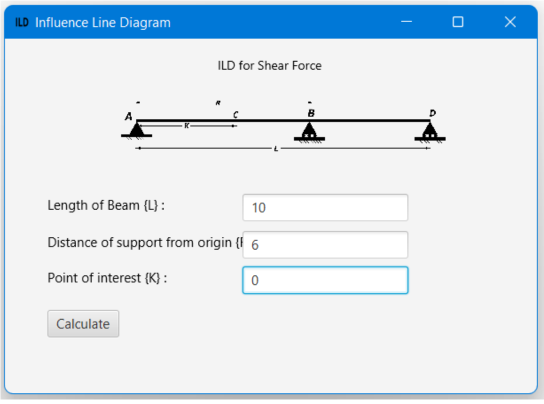
Example : **K=0,**



Distance between support(L) = 10

Length of Beam(R) = 6

Point of interest(K) = 0

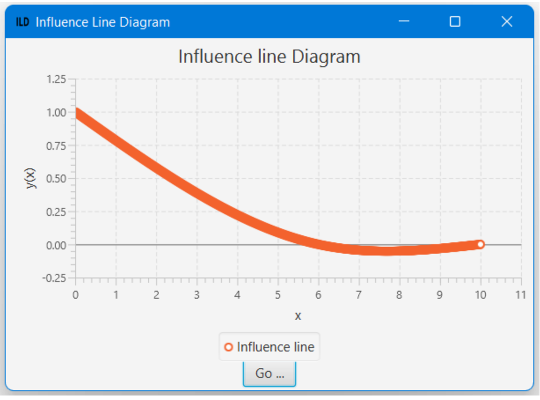


ILD for Shear Force

Length of Beam {L} : 10

Distance of support from origin {l 6

Point of interest {K} : 0



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 0.

Scale Factor = 119.99998

shear force at **0.0 = 1.0**

shear force at 0.01 = 0.9978335

shear force at 0.02 = 0.99566674

shear force at 0.03 = 0.99350053

Chapter: 3

Influence line diagram of moment for simply supported beam

To find an influence line diagram of Bending Moment for the point at distance “K” from the fixed end “A “(as shown in fig. 1).

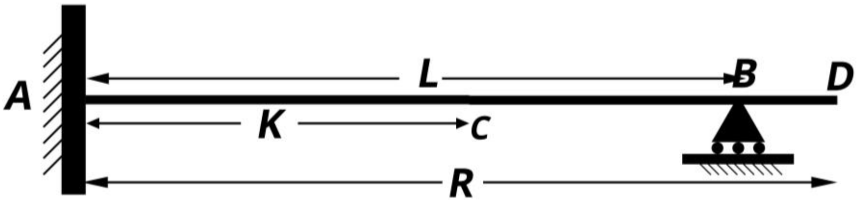


Fig. 1 Cantilever, simply supported Beam

R = Length of Beam,

L = Distance of support from origin from fixed end,

K = Distance of point for which ILD is calculated from fixed end,

EI = flexural rigidity of beam (assumed to be constant)

Solution: The solution will be in three parts for different position of K.

Applying Müller-Breslau principle,

Part 1: K=0

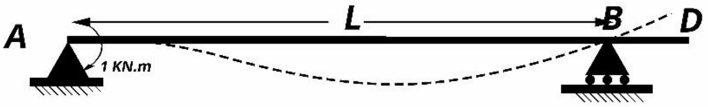


Fig. 2 Influence Line to the bending moment at 0 is to the same scale as deflected shape.

When x varies from 0 to L i.e., $\mathbf{0 \leq x < L}$

As we know the M_x is a linear function of x(observation from the above problems and can be proved for any given beam)

By Double Integration it can be proved that deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(0)= D_1 = 0$ ..... (2)

$y(L) = A_1L^3+B_1L^2+C_1L+D_1 = 0$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 1$ at x=0 ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=L ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ L^3 & L^2 & L & 1 \\ 0 & 2 & 0 & 0 \\ 6L & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

The value for the constants is calculated using inverse and multiplication matrix methods in java program.

When x varies from L to R i.e., **$L\leq x <R$**

Let the equation for the deflected beam be,

$y(x)=A_3x^3+B_3x^2+C_3x+D_3$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L)= A_3L^3+B_3L^2+C_3L+D_3 = 0$ ..... (2)

derivatives will be equal at x=K

$3A_3L^2+2B_3L+C_3 = 3A_1L^2+2B_1L+C_1$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at x=L ..... (4)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at x=R ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} L^3 & L^2 & L & 1 \\ 3L^2 & 2L & 1 & 0 \\ 6L & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3A_1K^2 + 2B_1K + C_1 \\ 0 \\ 0 \end{pmatrix}$$

The values for the constant is calculated using java.

After computation of all the constant for the cubic equations, Scale factor is calculated

Scale factor $= (3A_1K^2 + 2B_1K + C_1)$

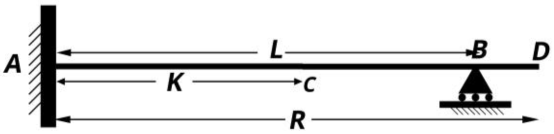
Hence, the **Influence Line Diagram for shear at C from equation**

$y(x)=(A_1x^3+B_1x^2+C_1x+D_1)/(3A_1K^2 + 2B_1K + C_1)$

$y(x)=(A_3x^3+B_3x^2+C_3x+D_3)/(3A_1K^2 + 2B_1K + C_1)$

Results from Application

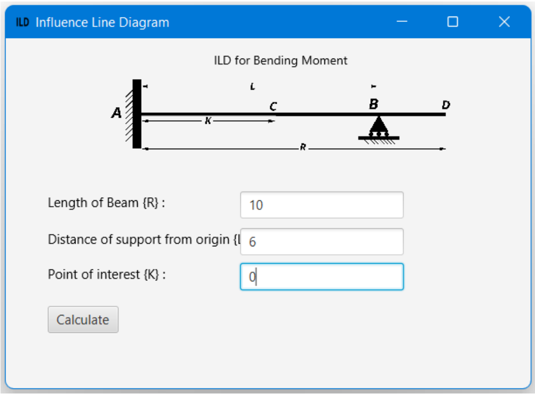
Example : **K=0**



Distance between support(L) $= 6$

Length of Beam(R) $= 10$

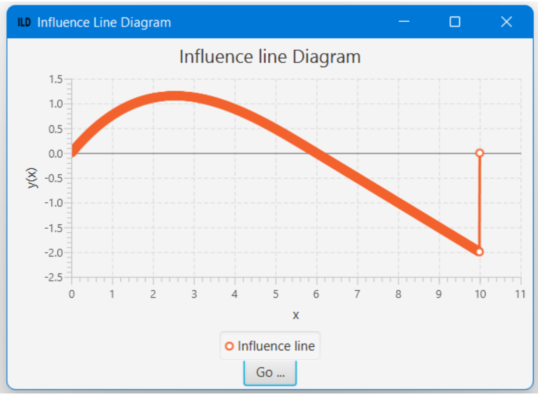
Point of interest(K) $= 0$



Length of Beam {R} : 10

Distance of support from origin {L 6

Point of interest {K} : 0



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 0.

moment at **0.0 = -0.0**

moment at **0.01 = 0.009975013**

moment at 0.02 = 0.019900111

moment at 0.03 = 0.029775374

Part 2: 0<K<L

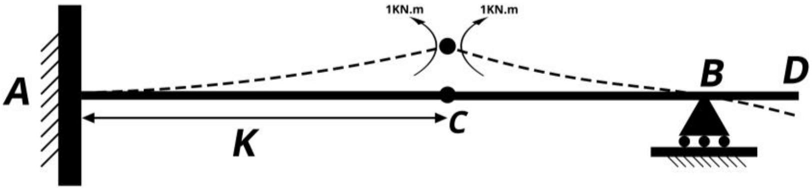


Fig. 3 Influence Line to the bending moment at C is to the same scale as deflected shape.

When x varies from 0 to K i.e., $0 \leq x < K$

As we know the M_x is a linear function of x(observation from the above problems and can be proved for any given beam)

By Double Integration it can be proved that deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(0)= D_1 = 0$ ..... (2)

$\frac{dy}{dx} = 0$ at x=0 ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = \frac{L}{L-K}$ at x=0 ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 1$ at x=K ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 6K & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 6A_1x + 2B_1 \\ 6A_1x + 2B_1 \end{pmatrix}$$

The value for the constants is calculated using inverse and multiplication matrix methods in java program.

When x varies from K to L i.e., $\mathbf{K \leq\ x < L}$

Let the equation for the deflected beam be,

$y(x)=A_2x^3+B_2x^2+C_2x+D_2$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L)= A_2L^3+B_2L^2+C_2L+D_2 = 0$ ..... (2)

displacement will be equal at x=K

$A_2K^3+B_2K^2+C_2K+D_2 = A_1K^3+B_1K^2+C_1K+D_1$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=L ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 1$ at x=K ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} L^3 & L^2 & L & 1 \\ K^3 & K^2 & K & 1 \\ 6L & 2 & 0 & 0 \\ 6K & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{pmatrix} = \begin{pmatrix} 0 \\ A_1K^3 + B_1K^2 + C_1K + D_1 \\ 0 \\ 1 \end{pmatrix}$$

The values for the constant is calculated using java.

When x varies from L to R i.e., $\mathbf{L \leq x < R}$

Let the equation for the deflected beam be,

$y(x)=A_3x^3+B_3x^2+C_3x+D_3$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L)= A_3L^3+B_3L^2+C_3L+D_3 = 0$ ..... (2)

derivatives will be equal at x=K

$3A_3K^2+2B_3K+C_3 = 3A_2K^2+2B_2K+C_2$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at x=L ..... (4)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at x=R ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} L^3 & L^2 & L & 1 \\ 3K^2 & 2K & 1 & 0 \\ 6L & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 3A_2K^2 + 2B_2K + C_2 \\ 0 \\ 0 \end{pmatrix}$$

The values for the constant is calculated using java.

After computation of all the constant for the cubic equations, Scale factor is calculated

Scale factor = $(3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2)$

Hence, the **Influence Line Diagram for shear at C from equation**

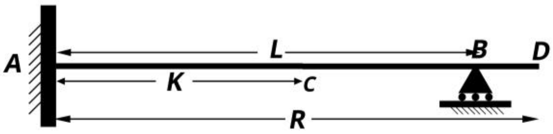
$y(x)=(A_1x^3+B_1x^2+C_1x+D_1)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

$y(x)=(A_2x^3+B_2x^2+C_2x+D_2)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

$y(x)=(A_3x^3+B_3x^2+C_3x+D_3)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

Results from Application

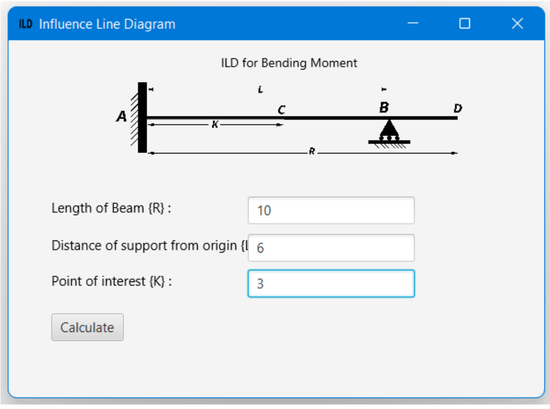
Example : **0<K<L**



Length of Beam(R) = 10

Distance between support(L) = 6

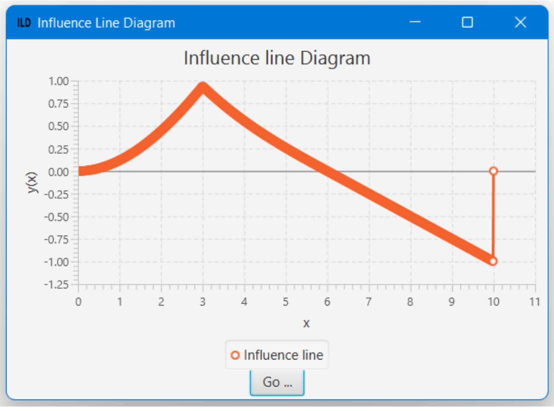
Point of interest(K) = 3



Length of Beam {R} : 10

Distance of support from origin {L 6

Point of interest {K} : 3



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

moment at **2.9999976 = 0.9374986**

moment at **3.0099976 = 0.9331322**

Part 3: K=L

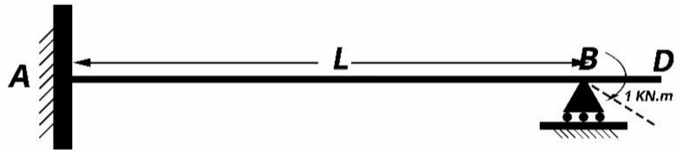


Fig. 4 Influence Line to the bending moment at L is to the same scale as deflected shape.

When x varies from 0 to L i.e., $\mathbf{0 \leq x < L}$

As we know the M_x is a linear function of x(observation from the above problems and can be proved for any given beam)

By Double Integration it can be proved that deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(0) = D_1 = 0$ ..... (2)

$y(L) = A_1L^3+B_1L^2+C_1L+D_1 = 0$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at $x=0$ ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at $x=L$ ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ L^3 & L^2 & L & 1 \\ 0 & 2 & 0 & 0 \\ 6L & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The value for the constants is calculated using inverse and multiplication matrix methods in java program.

When x varies from L to R i.e., **$L \leq x < R$**

Let the equation for the deflected beam be,

$y(x)=A_3x^3+B_3x^2+C_3x+D_3$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L) = A_3L^3+B_3L^2+C_3L+D_3 = 0$ ..... (2)

$\frac{dy}{dx} = 3A_3L^2+2B_3L+C_3 = -1$ at $x=0$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at $x=L$ ..... (4)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at $x=R$ ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} L^3 & L^2 & L & 1 \\ 3L^2 & 2L & 1 & 0 \\ 6L & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

The values for the constant is calculated using java.

After computation of all the constant for the cubic equations, Scale factor is calculated

Scale factor = 1

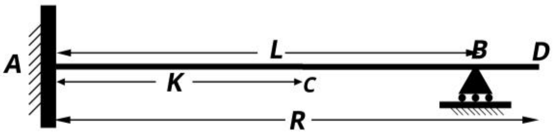
Hence, the **Influence Line Diagram for shear at C from equation**

$y(x)=0$

$y(x)=(A_3x^3+B_3x^2+C_3x+D_3)$

Results from Application

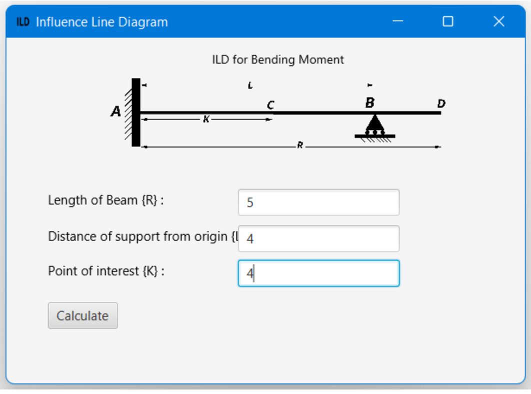
Example : **K=L**



Distance between support(L) = 4

Length of Beam(R) = 5

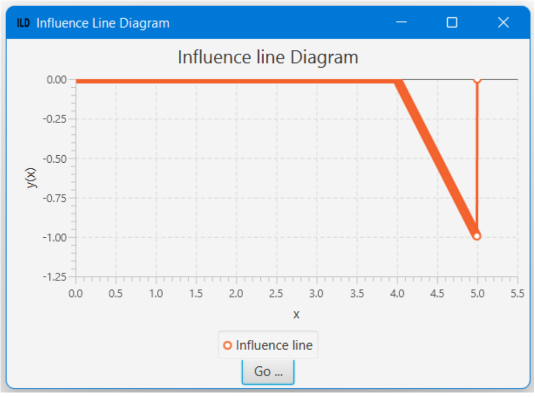
Point of interest(K) = 4



Length of Beam (R) : 5

Distance of support from origin (l 4

Point of interest (K) : 4



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

moment at 4.9800663 = -0.9800663

moment at 4.9850664 = -0.9850664

moment at 4.9900665 = -0.9900665

moment at 4.9950666 = -0.99506664

Chapter: 4

Influence line diagram of moment for continuous beam

To find an influence line diagram of Bending Moment for the point at distance “K” from the fixed end “A “(as shown in fig. 1).

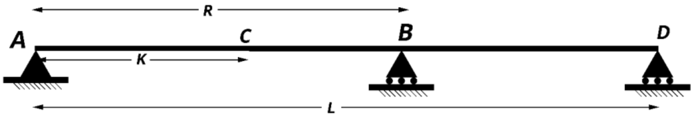


Fig. 1 Cantilever, simply supported Beam

R = Length of Beam,

L = Distance of support from origin from fixed end,

K = Distance of point for which ILD is calculated from fixed end,

EI = flexural rigidity of beam (assumed to be constant)

Solution: The solution will be in three parts for different position of K.

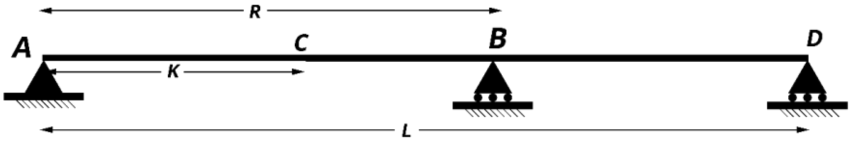
Applying Müller-Breslau principle,

Part 1: K=0

Moment at support will be zero when a unit load act upon beam.

Results from Application

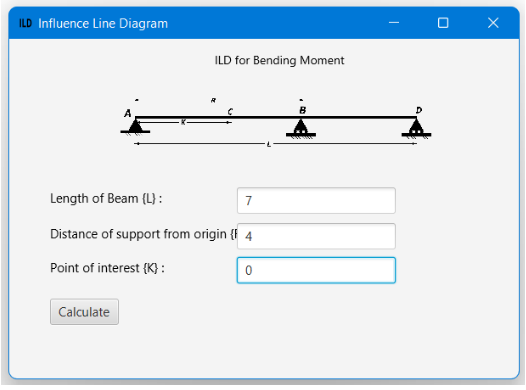
Example : **K=0**



Distance between support(L) = 7

Length of Beam(R) = 4

Point of interest(K) = 0



Influence Line Diagram

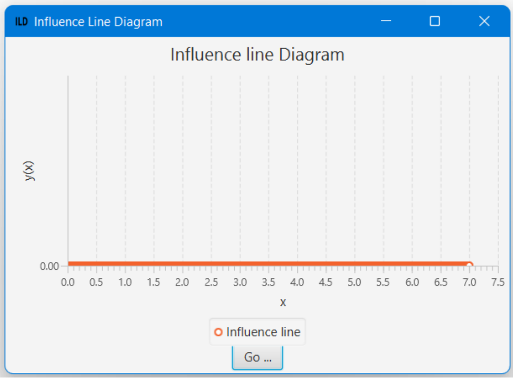
ILD for Bending Moment

Length of Beam {L} : 7

Distance of support from origin {R 4

Point of interest {K} : 0

Calculate



Influence Line Diagram

The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

moment at 0.0 = 0.0

moment at 0.007 = 0.0

moment at 0.014 = 0.0

moment at 0.021000002 = 0.0

Part 2: 0<K<R

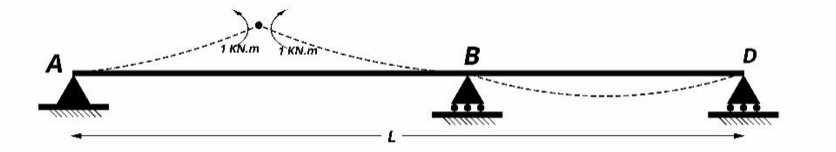


Fig. 2 Influence Line to the bending moment at C is to the same scale as deflected shape.

When x varies from R to L i.e., **$R \leq x < L$**

As we know the M_x is a linear function of x(observation from the above problems and can be proved for any given beam)

By Double Integration it can be proved that deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(R) = A_1R^3+B_1R^2+C_1R+D_1 = 0$ ..... (2)

$y(L) = A_1L^3+B_1L^2+C_1L+D_1 = 0$ ..... (2)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = \frac{R}{K}$ at x=R ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=L ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} L^3 & L^2 & L & 1 \\ R^3 & R^2 & R & 1 \\ 6L & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ R/K \end{pmatrix}$$

The value for the constants is calculated using inverse and multiplication matrix methods in java program.

When x varies from K to R i.e., $\mathbf{K \leq \ x < R}$

Let the equation for the deflected beam be,

$y(x)=A_2x^3+B_2x^2+C_2x+D_2$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L) = A_2R^3+B_2R^2+C_2R+D_2 = 0$ ..... (2)

Slope will be equal at x=K

$3A_2R^2+2B_2R+C_2 = 3A_3R^2+2B_3R+C_3$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=R ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 1$ at x=K ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} R^3 & R^2 & R & 1 \\ 3R^2 & 2R & 1 & 0 \\ 6R & 2 & 0 & 0 \\ 6K & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_2 \\ B_2 \\ C_2 \\ D_2 \end{pmatrix} = \begin{pmatrix} 0 \\ A_3R^2 + B_3R + C_3 \\ R/K \\ 1 \end{pmatrix}$$

The values for the constant is calculated using java.

When x varies from 0 to K i.e., **$0 \leq x < K$**

Let the equation for the deflected beam be,

$y(x)=A_3x^3+B_3x^2+C_3x+D_3$ ..... (1)

The above equation will satisfy the following boundary condition

$y(L)= A_3L^3+B_3L^2+C_3L+D_3 = 0$ ..... (2)

displacements will be equal at x=K

$A_1K^3 + B_1K^2 + C_1K + D_1 = A_2K^3 + B_2K^2 + C_2K + D_2$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 0$ at x=0 ..... (4)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 1$ at x=K ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 6K & 2 & 0 & 0 \\ R^3 & R^2 & R & 1 \end{pmatrix} \begin{pmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ A_2K^3 + B_2K^2 + C_2K + D_2 \end{pmatrix}$$

The values for the constant is calculated using java.

After computation of all the constant for the cubic equations, Scale factor is calculated

Scale factor = $(3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2)$

Hence, the **Influence Line Diagram for shear at C from equation**

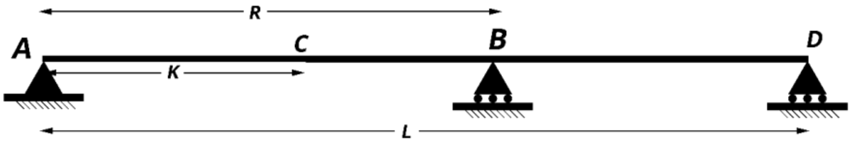
$y(x)=(A_1x^3+B_1x^2+C_1x+D_1)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

$y(x)=(A_2x^3+B_2x^2+C_2x+D_2)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

$y(x)=(A_3x^3+B_3x^2+C_3x+D_3)/((3A_1K^2 + 2B_1K + C_1) - (3A_2K^2 + 2B_2K + C_2))$

Results from Application

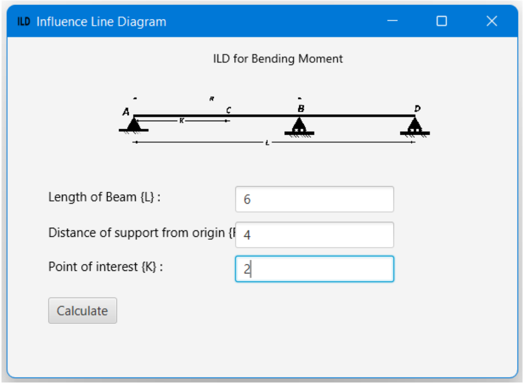
Example : **$0<K<R$**



Distance between support(L) = 7

Length of Beam(R) = 4

Point of interest(K) = 0

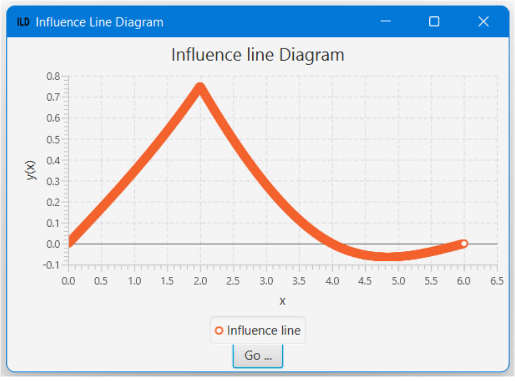


ILD for Bending Moment

Length of Beam (L) : 6

Distance of support from origin (I 4

Point of interest (K) : 2



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 2.

moment at 1.992006 = 0.7463404

moment at 1.998006 = **0.7490867**

moment at 2.004006 = **0.7478314**

moment at 2.010006 = 0.74458665

Part 3: K=R

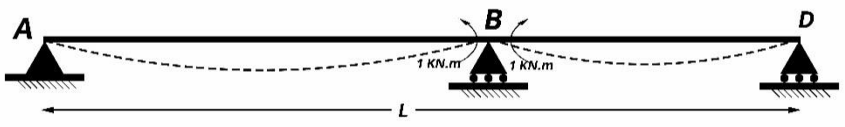


Fig. 3 Influence Line to the bending moment at L is to the same scale as deflected shape.

When x varies from 0 to L i.e., $\mathbf{0 \leq x < L}$

As we know the M_x is a linear function of x(observation from the above problems and can be proved for any given beam)

By Double Integration it can be proved that deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(0)= D_1 = 0$ ..... (2)

$y(R) = A_1R^3+B_1R^2+C_1R+D_1 = 0$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=0 ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = 0$ at x=R ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ R^3 & R^2 & R & 1 \\ 0 & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1/2 \\ 1 \end{pmatrix}$$

The value for the constants is calculated using inverse and multiplication matrix methods in java program.

When x varies from L to R i.e., **L≤ x <R**

Let the equation for the deflected beam be,

$y(x)=A_3x^3+B_3x^2+C_3x+D_3$ ..... (1)

The above equation will satisfy the following boundary condition

$y(R)= A_3R^3+B_3R^2+C_3R+D_3 = 0$ ..... (2)

$y(L)= A_3L^3+B_3L^2+C_3L+D_3 = 0$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = -1/2$ at x=L ..... (4)

$\frac{d^2y}{dx^2} = 6A_3x+2B_3 = 1$ at x=R ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} R^3 & R^2 & R & 1 \\ L^3 & L^2 & L & 1 \\ 6L & 2 & 0 & 0 \\ 6R & 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} A_3 \\ B_3 \\ C_3 \\ D_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1/2 \\ 1 \end{pmatrix}$$

The values for the constant is calculated using java.

After computation of all the constant for the cubic equations, Scale factor is calculated

Scale factor = $3A_1K^2+2B_1K+C_1$ - $3A_3K^2+2B_3K+C_3$

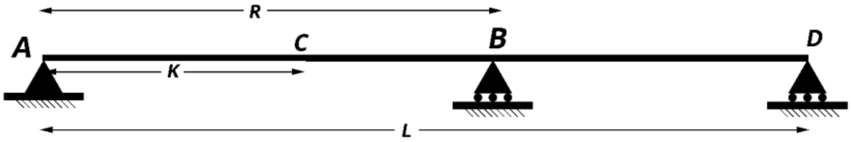
Hence, the **Influence Line Diagram for shear at C from equation**

$y(x)=(A_1x^3+B_1x^2+C_1x+D_1)/(\,(3A_1K^2 + 2B_1K + C_1) - (3A_3K^2 + 2B_3K + C_3))$

$y(x)=(A_3x^3+B_3x^2+C_3x+D_3)/(\,(3A_1K^2 + 2B_1K + C_1) - (3A_3K^2 + 2B_3K + C_3))$

Results from Application

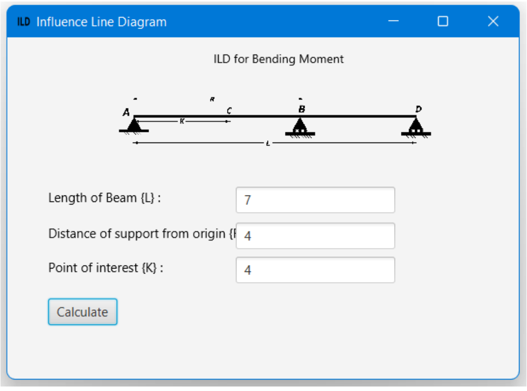
Example : **K=R**



Distance between support(L) = 7

Length of Beam(R) = 4

Point of interest(K) = 4

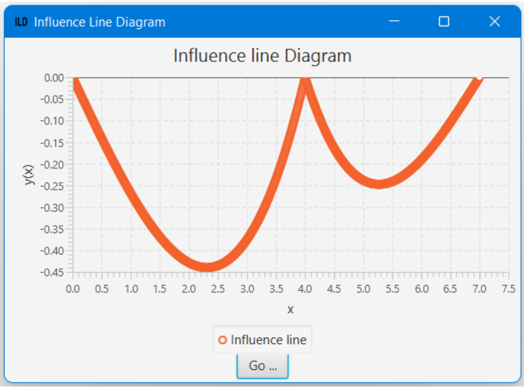


ILD for Bending Moment

Length of Beam {L} : 7

Distance of support from origin {I 4

Point of interest {K} : 4



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 4.

moment at 3.989989 = -0.0056991545

moment at 3.996989 = **-0.0017186564**

moment at 4.003989 = **-0.0017068036**

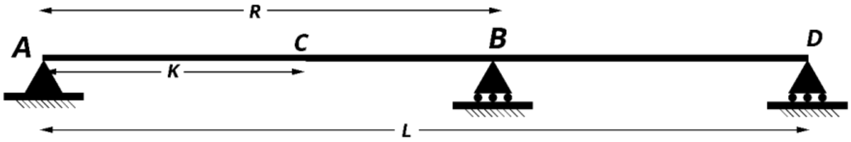
moment at 4.010989 = -0.004683083

Part 4: R<K<L

Replacing x by L-x in part 2 we will get solution for the part 4.

Results from Application

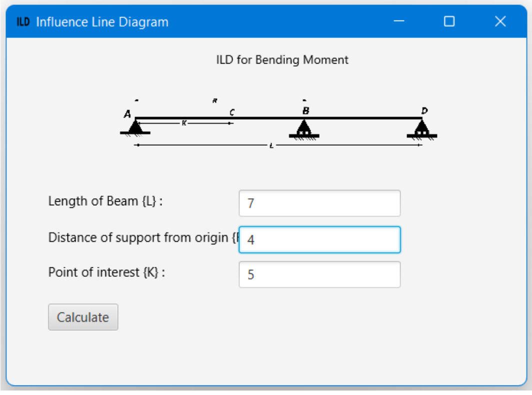
Example : R<K<L



Distance between support(L) = 7

Length of Beam(R) = 4

Point of interest(K) = 5

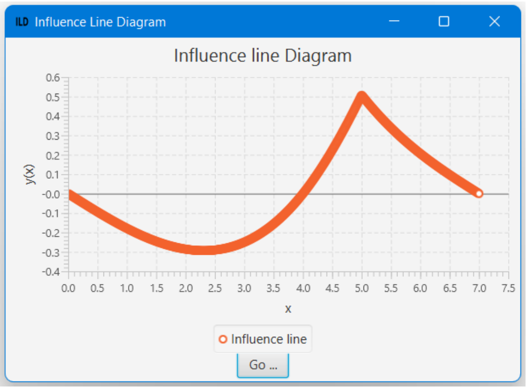


ILD for Bending Moment

Length of Beam {L} : 7

Distance of support from origin {I 4

Point of interest {K} : 5



The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 5.

moment at 4.990985 = 0.50236374

moment at **4.997985 = 0.50668925**

moment at **5.004985 = 0.50604004**

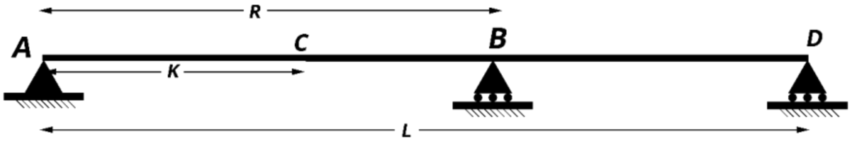
moment at 5.011985 = 0.5033843

Part 5: K=L

Moment at support will be zero when a unit load act upon beam.

Results from Application

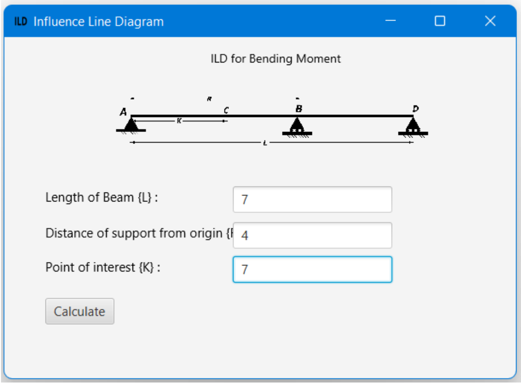
Example :



Distance between support(L) = 7

Length of Beam(R) = 4

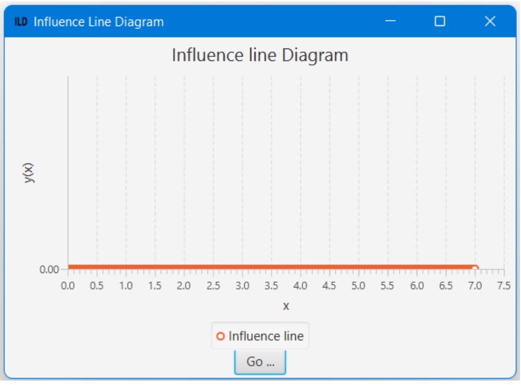
Point of interest(K) = 7



Length of Beam {L} : 7

Distance of support from origin {F 4

Point of interest {K} : 7



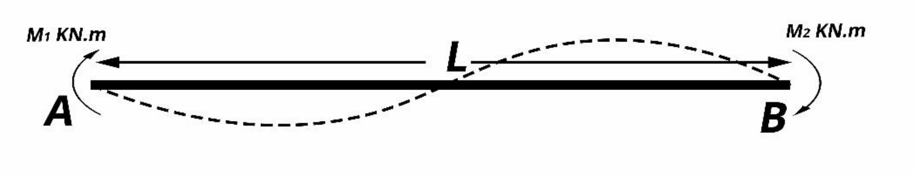
The values are calculated at different interval to plot the ILD as per shown in figure. The application print all the displacement values in command prompt for small intervals. Values are mentioned below as per application output in command prompt for the point nearby 3.

```
moment at 6.9789762 = 0.0

moment at 6.985976 = 0.0

moment at 6.992976 = 0.0

moment at 6.999976 = 0.0
```

Chapter: 5

Interpolation Function

To find arbitrary beam's deflection with known moment (using **slope deflection method)**, after applying Muller Breslau principle.



We can prove that Internal Moment will be a linear function of x

By Double Integration we can say deflection of beam is either a cubic function or a linear function

Let the equation for the deflected beam be,

$y(x)=A_1x^3+B_1x^2+C_1x+D_1$ ..... (1)

The above equation will satisfy the following boundary condition

$y(0)= D_1 = y_1$ ..... (2)

$y(L) = A_1L^3+B_1L^2+C_1L+D_1 = y_2$ ..... (3)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = M_1$ at x=0 ..... (4)

$\frac{d^2y}{dx^2} = 6A_1x+2B_1 = M_2$ at x=L ..... (5)

The equation (2), (3), (4), & (5) we can solve for y(x) using matrix method

$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ L^3 & L^2 & L & 1 \\ 0 & 2 & 0 & 0 \\ 6L & 2 & 0 & 0 \end{pmatrix}\begin{pmatrix} A_1 \\ B_1 \\ C_1 \\ D_1 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ M_1 \\ M_2 \end{pmatrix}$$

From the above matrix we can find coefficient of cubic equation hence we can find beam’s deflection equation after applying Muller Breslau Principle.

It can be concluded that we can find ILD for any beam after finding out the boundary conditions using methods such as Stiffness Matrix.

Application and Programming

Java is a programming language and a platform. Java is a high level, robust, object-oriented, and secure programming language. Java developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java. As previously mentioned, the code created on Java for performing the computations such as matrix multiplication using arrays and any such mathematical operations. It is essential to have prior knowledge of multi-dimensional arrays to understand mathematical solutions.
Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means organizing software as a combination of different types of objects that incorporate both data and behavior. Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules.
Basic concepts of OOPs are:

- Object
- Class
- Inheritance
- Polymorphism
- Abstraction
- Encapsulation

Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

There are two types of platforms software-based and hardware-based. Java provides a software-based platform.

The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on top of other hardware-based platforms. It has two components:

- Runtime Environment
- API (Application Programming Interface)

Java code can be executed on multiple platforms, for example, Windows, Linux, Sun Solaris, Mac/OS, etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms, i.e., Write Once and Run Anywhere (WORA).

JavaFX

The front end of the application is made on JavaFX to create a simple user interface. JavaFX is a Java library used to develop Desktop applications and Rich Internet Applications (RIA). JavaFX is intended to replace swing in Java applications as a GUI framework. However, it provides more functionalities than swing. Like Swing, JavaFX also provides its components and does not depend upon the operating system. It is lightweight and hardware-accelerated. It supports various operating systems, including Windows, Linux, and Mac OS. The applications built in JavaFX can run on multiple platforms, including Web, Mobile and Desktops.

Scene Graph - It is the starting point of constructing a JavaFX application. It is a hierarchical tree of nodes that represent all the visual elements of the user interface. It also has the capability of handling events. In general, a scene graph is defined as a collection of nodes. Each node has its separate id, style, and volume. Every node of a scene graph can only have a single parent and zero or more children. All the implementation on a scene graph is applied to its node. There are various classes present in javafx.scene package that is used for creating, modifying and applying some transformations on the node.

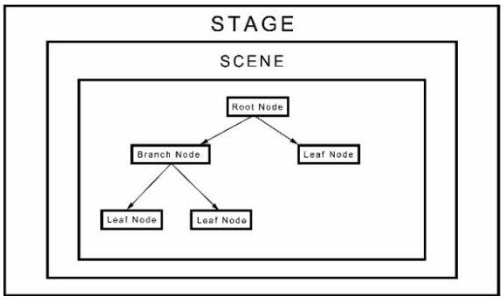


Fig. 1 JavaFX hierarchical tree

JavaFX **Scene Builder** is used to design an interface, enabling you to drag and drop graphical user interface (GUI) components onto a JavaFX scene. It can help you to quickly prototype interactive applications that connect GUI components to the application logic.

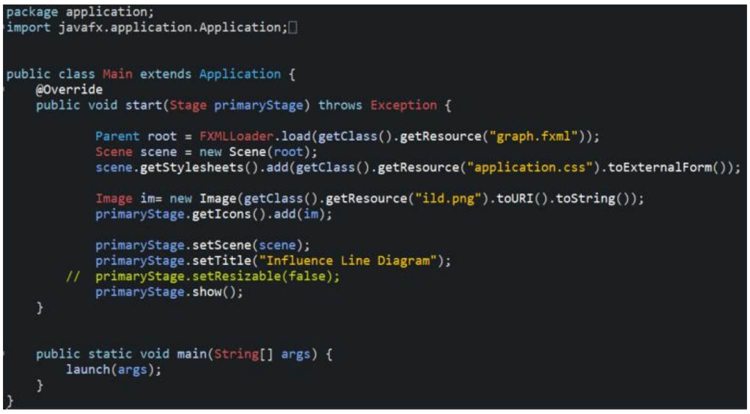
Eclipse is used as an integrated development environment that is used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment.

Classes and Objects

For Chapter 1-4:

There are total of six JavaFX projects, and each project consist of six to seven different classes in an src folder–

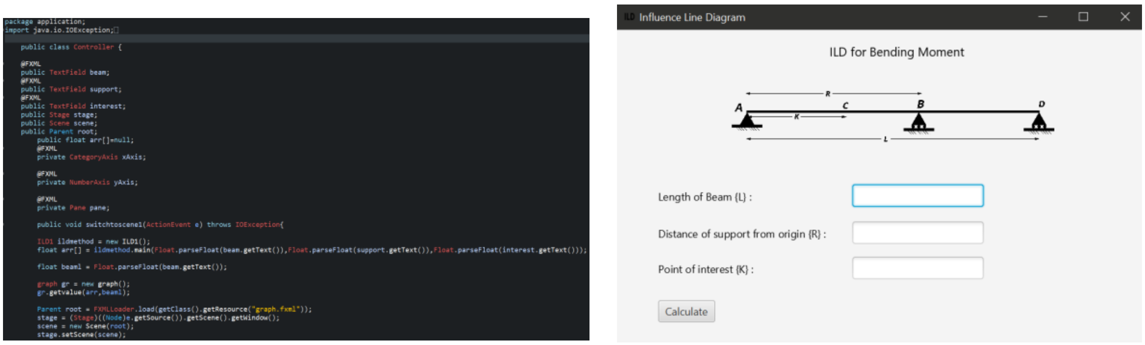
1. **Main.java** – To launch the program and set up basics prerequisites to launching an application.



```java
package application;
import javafx.application.Application;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {

            Parent root = FXMLLoader.load(getClass().getResource("graph.fxml"));
            Scene scene = new Scene(root);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());

            Image im= new Image(getClass().getResource("ild.png").toURI().toString());
            primaryStage.getIcons().add(im);

            primaryStage.setScene(scene);
            primaryStage.setTitle("Influence Line Diagram");
        //  primaryStage.setResizable(false);
            primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Fig. 2 Main.java

2. **Controller.java** – The main function of the class is to control actions like switching scenes when the button is pressed. To perform a function, the specific onaction id must be assigned in Scene Builder. Whenever the step has been performed a set of actions can be done using the controller class.



```java
package application;
import java.io.IOException;

    public class Controller {

    @FXML
    public TextField beam;
    @FXML
    public TextField support;
    @FXML
    public TextField interest;
    public Stage stage;
    public Scene scene;
    public Parent root;
        public float arr[]=null;
        @FXML
        private CategoryAxis xAxis;

        @FXML
        private NumberAxis yAxis;

        @FXML
        private Pane pane;

        public void switchtoscene1(ActionEvent e) throws IOException{

        ILD1 ildmethod = new ILD1();
        float arr[] = ildmethod.main(Float.parseFloat(beam.getText()),Float.parseFloat(support.getText()),Float.parseFloat(interest.getText()));

        float beaml = Float.parseFloat(beam.getText());

        graph gr = new graph();
        gr.getvalue(arr,beaml);

        Parent root = FXMLLoader.load(getClass().getResource("graph.fxml"));
        stage = (Stage)((Node)e.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
```

Influence Line Diagram

ILD for Bending Moment

Length of Beam {L} :

Distance of support from origin {R} :

Point of interest {K} :

Fig. 3 controller.java class and application GUI

3. **graph.java** – When all the calculation is computed in ILD.java it will return array of float values to graph.java where the graphical representation is done.

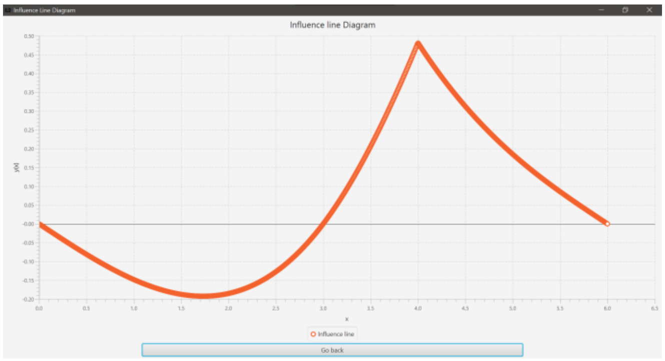


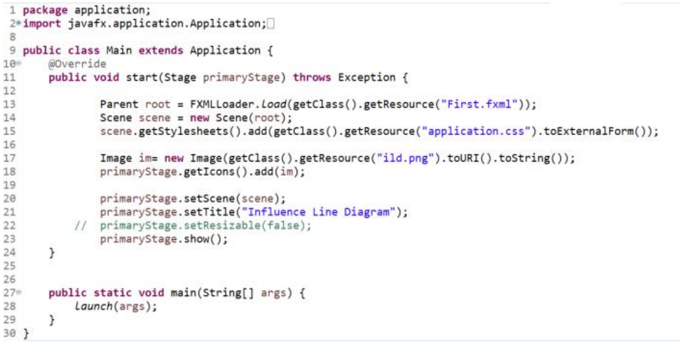
Fig. 4 ILD after inserting input in previous class

4. **application.css and graph.fxml -** JavaFX FXML, an XML-based language that provides the structure for building a user interface separate from the application logic of your code. JavaFX has a rich set of extensions to CSS to support features such as colour derivation, property lookup, and multiple background colours and borders for a single node. These features add significant new power for developers and designers and are described in detail in this document. JavaFX Scene Builder is used to design an interface, which enables you to drag and drop graphical user interface (GUI) components onto a JavaFX scene by creating CSS and fxml files.
5. **ILD.java or ILD1.java** – All the calculative and mathematical operations part is in this class. This class will return output which will be used to create a graph by graph.java class.
6. **Inverse.java** – It is a class that takes a n*n matrix as input and will return inverse if matrix using mathematics.

Beam with n number of support

The first four problems mentioned in the report are explicit formula based. This section will be focused on the explanation of beam with n number of supports. The problem is divided into four parts Moment, Shear, Shear at Support, and Moment at Support. Each part is divided into three subparts depending upon whether both ends are pinned, left support Fixed, and both support fixed. The classes and objects used for this application are -

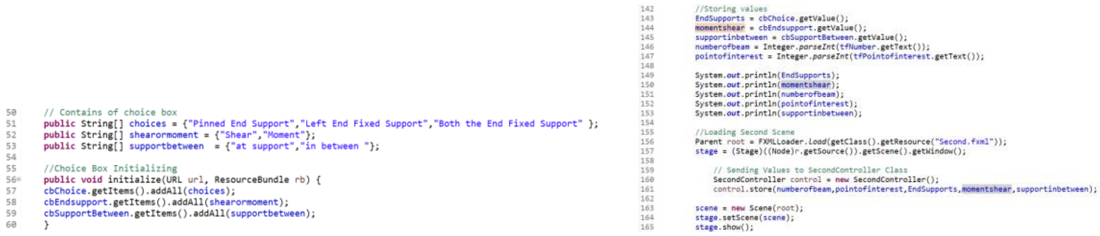
1. **Main.java -** To launch the First.fxml and set up basics prerequisites to launch an application.



```java
package application;
import javafx.application.Application;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws Exception {

            Parent root = FXMLLoader.load(getClass().getResource("First.fxml"));
            Scene scene = new Scene(root);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());

            Image im= new Image(getClass().getResource("ild.png").toURI().toString());
            primaryStage.getIcons().add(im);

            primaryStage.setScene(scene);
            primaryStage.setTitle("Influence Line Diagram");
        //  primaryStage.setResizable(false);
            primaryStage.show();
    }


    public static void main(String[] args) {
        launch(args);
    }
}
```

Fig. 1 Main.java class

2. **FirstController.java** – The First.fxml consist two text field, and three choice box and one button with FirstController.java class as controller. The controller class initialize entries of choice box on the launch of scene as shown in figure.

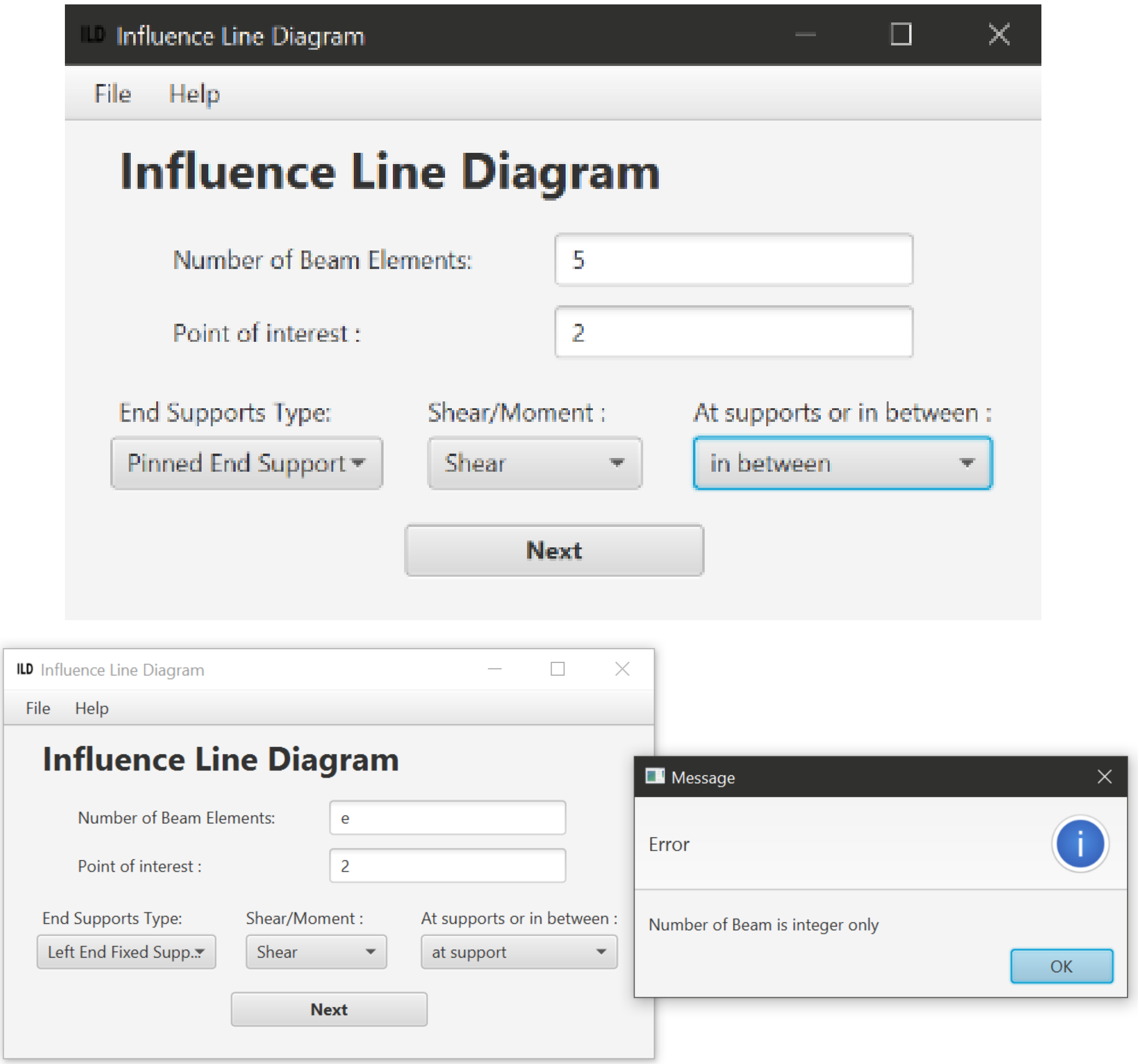


```java
    // Contains of choice box
    public String[] choices = {"Pinned End Support","Left End Fixed Support","Both the End Fixed Support" };
    public String[] shearormoment = {"Shear","Moment"};
    public String[] supportbetween  = {"at support","in between "};

    //Choice Box Initializing
    public void initialize(URL url, ResourceBundle rb) {
    cbChoice.getItems().addAll(choices);
    cbEndsupport.getItems().addAll(shearormoment);
    cbSupportBetween.getItems().addAll(supportbetween);
    }
```

```java
        //Storing values
        EndSupports = cbChoice.getValue();
        momentshear = cbEndsupport.getValue();
        supportinbetween = cbSupportBetween.getValue();
        numberofbeam = Integer.parseInt(tfNumber.getText());
        pointofinterest = Integer.parseInt(tfPointofinterest.getText());

        System.out.println(EndSupports);
        System.out.println(momentshear);
        System.out.println(numberofbeam);
        System.out.println(pointofinterest);
        System.out.println(supportinbetween);

        //Loading Second Scene
        Parent root = FXMLLoader.load(getClass().getResource("Second.fxml"));
        stage = (Stage)((Node)r.getSource()).getScene().getWindow();

            // Sending Values to SecondController Class
            SecondController control = new SecondController();
            control.store(numberofbeam,pointofinterest,EndSupports,momentshear,supportinbetween);

        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
```

Fig.2 FirstController.java class

After button is pressed the on action method “switchtosecond” will run. The text fields only take integers as input else it will show alert box with errors. Also choice box must be selected else error alert will displayed.



Influence Line Diagram

Influence Line Diagram

Number of Beam Elements: 5

Point of interest : 2

End Supports Type: Pinned End Support

Shear/Moment : Shear

At supports or in between : in between

Influence Line Diagram

Influence Line Diagram

Number of Beam Elements: e

Point of interest : 2

End Supports Type: Left End Fixed Supp..

Shear/Moment : Shear

At supports or in between : at support

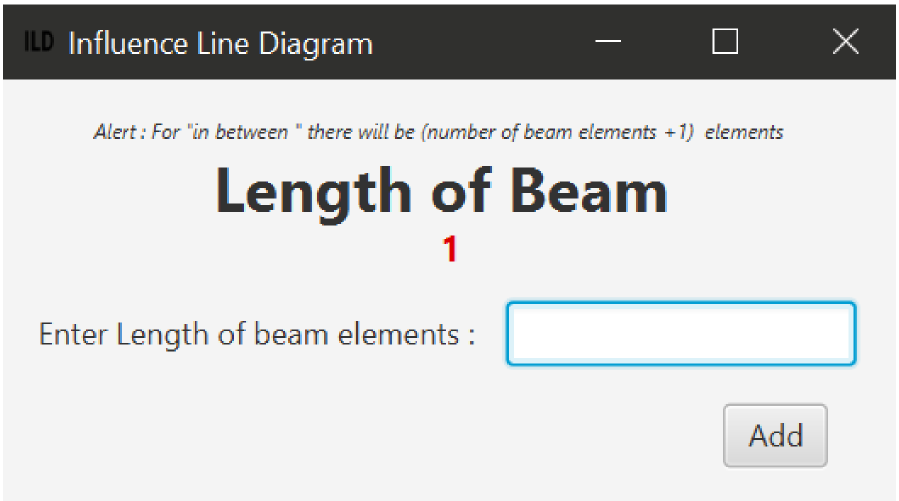
Message

Error

Number of Beam is integer only

Fig.3 GUI of application with n number of supports

3. **Secondcontroller.java** – If the switchtosecond method runs successfully, it will launch Second. The fxml scene contains a Textfield and button. User must enter values for each beam element starting from left to rightward. After entering all values add button and textfield will disappear and button named calculate will appear. On clicking calculate button, it will bring to the final graph scene using the action "switchtograph" method along with the values that the user entered previously.

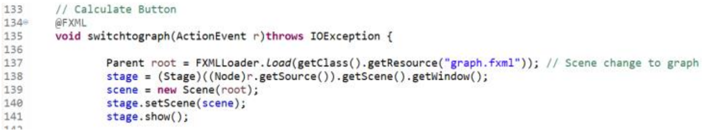


Alert : For "in between " there will be (number of beam elements +1) elements

Length of Beam

1

Enter Length of beam elements :



```
// Calculate Button
@FXML
void switchtograph(ActionEvent r)throws IOException {

        Parent root = FXMLLoader.load(getClass().getResource("graph.fxml")); // Scene change to graph
        stage = (Stage)((Node)r.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
```

Fig. 4 Secondcontroller.java

4. **Graph.java** – Graph will call Inbetween.java and Atsupport.java class where all the data is transfer and final calculation is carried out. The explanation for the calculation part is mentioned in the next section.

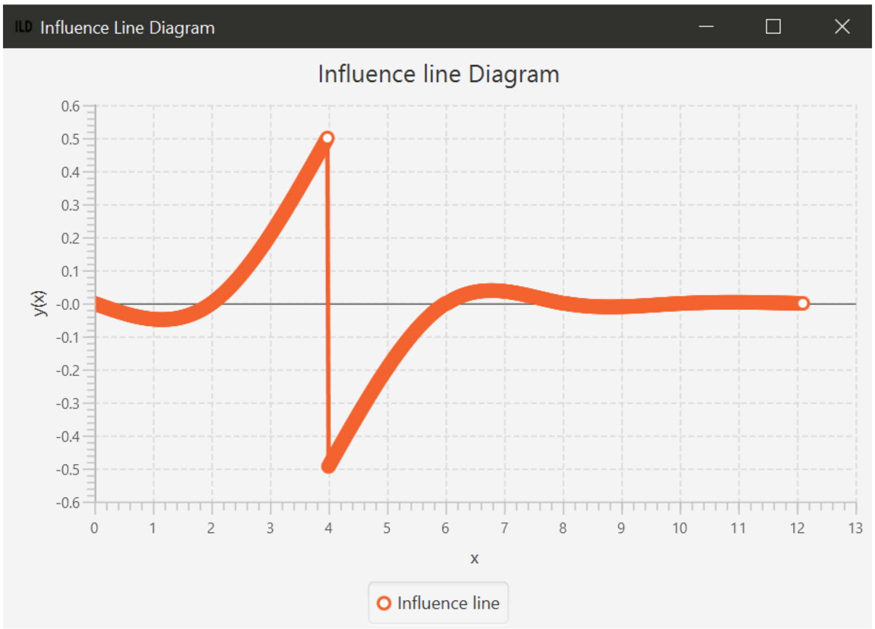


Fig. 5 ILD after giving input in previous parts

5. **Atsupport.java & Inbetween.java** - It consist of 12 different part that is explained under next section.

6. **Fxml** - Scene Builder is used to design interface, tool that enables you to drag and drop graphical user interface (GUI) components onto a JavaFX scene by creating changes in CSS and fxml files.

Now, further explanation will be using an example (all reasons of parts and subparts will be similar). Let there is four support and, the problem is to find an influence line diagram for a moment at any point of interest, say two.

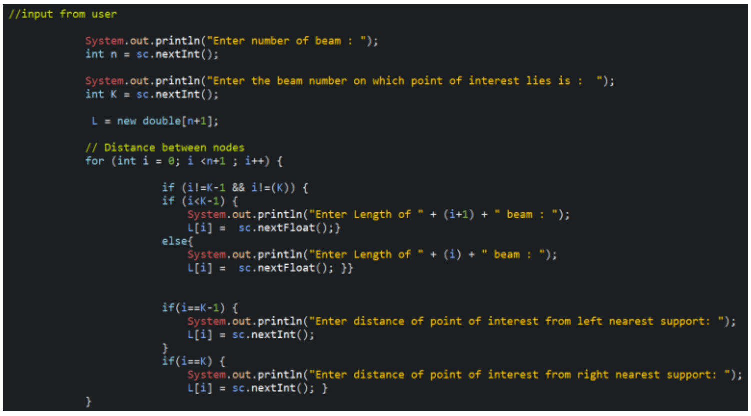
At first, there are user inputs that are required-

1) Number of beams - 4

2) Point of Interest -2

3) Length between each pair of nearest nodes or length of members- L[1], L[2], L[3], L[4]

4) whether ends are pinned or fixed.



```
//input from user

            System.out.println("Enter number of beam : ");
            int n = sc.nextInt();

            System.out.println("Enter the beam number on which point of interest lies is :  ");
            int K = sc.nextInt();

             L = new double[n+1];

            // Distance between nodes
            for (int i = 0; i <n+1 ; i++) {

                        if (i!=K-1 && i!=(K)) {
                        if (i<K-1) {
                            System.out.println("Enter Length of " + (i+1) + " beam : ");
                            L[i] =  sc.nextFloat();}
                        else{
                            System.out.println("Enter Length of " + (i) + " beam : ");
                            L[i] =  sc.nextFloat(); }}


                        if(i==K-1) {
                            System.out.println("Enter distance of point of interest from left nearest support: ");
                            L[i] = sc.nextInt();
                        }
                        if(i==K) {
                            System.out.println("Enter distance of point of interest from right nearest support: ");
                            L[i] = sc.nextInt(); }
            }
```

Fig. 1 Inputs from user

As shown in the figure, there will be four elements and five nodes after applying the Muller Breslau principle, and it will consist of 11 forces and 11 displacement variables. When the beam deflects, the internal pin will allow a single deflection. However, the slope of each connected member will be different. Also, a slope at the roller will occur. The global axis set to the leftmost point of the beam.

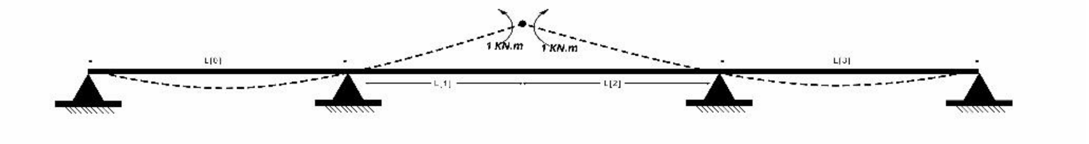


Fig. 2 Four pinned support continuous beam

To find beams deflection - after applying the Muller Breslau principle, we are using the stiffness matrix method to find slopes and vertical displacement at every single node, then will apply double integration on each element using boundary conditions to calculate the elements deflection equation.

First Step, Member Stiffness matrix is computed using the formula mentioned below in the figure -

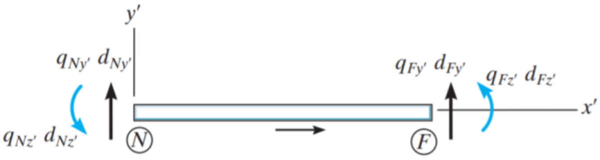
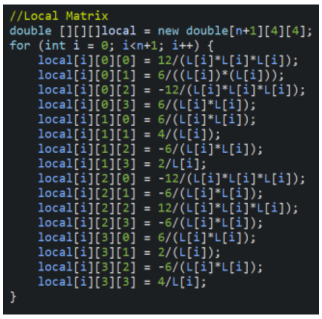


Fig. 3 Any arbitrary element

$$\begin{bmatrix} q_{Ny'} \\ q_{Nz'} \\ q_{Fy'} \\ q_{Fz'} \end{bmatrix} = \begin{array}{cccc} N_{y'} & N_{z'} & F_{y'} & F_{z'} \\ \left[\begin{array}{cccc} \frac{12EI}{L^3} & \frac{6EI}{L^2} & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{4EI}{L} & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ \frac{6EI}{L^2} & \frac{2EI}{L} & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{array}\right] \end{array} \begin{bmatrix} d_{Ny'} \\ d_{Nz'} \\ d_{Fy'} \\ d_{Fz'} \end{bmatrix}$$



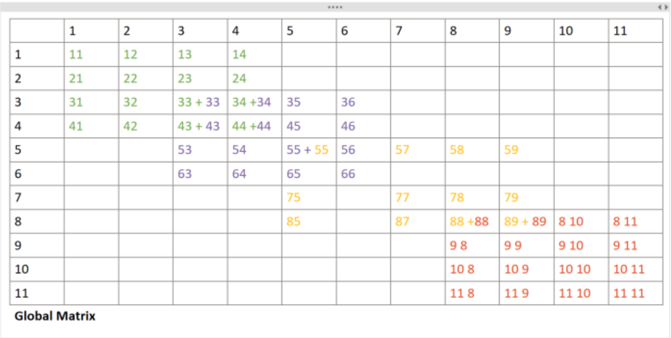
```
//Local Matrix
double [][][]local = new double[n+1][4][4];
for (int i = 0; i<n+1; i++) {
    local[i][0][0] = 12/(L[i]*L[i]*L[i]);
    local[i][0][1] = 6/((L[i])*(L[i]));
    local[i][0][2] = -12/(L[i]*L[i]*L[i]);
    local[i][0][3] = 6/(L[i]*L[i]);
    local[i][1][0] = 6/(L[i]*L[i]);
    local[i][1][1] = 4/(L[i]);
    local[i][1][2] = -6/(L[i]*L[i]);
    local[i][1][3] = 2/L[i];
    local[i][2][0] = -12/(L[i]*L[i]*L[i]);
    local[i][2][1] = -6/(L[i]*L[i]);
    local[i][2][2] = 12/(L[i]*L[i]*L[i]);
    local[i][2][3] = -6/(L[i]*L[i]);
    local[i][3][0] = 6/(L[i]*L[i]);
    local[i][3][1] = 2/(L[i]);
    local[i][3][2] = -6/(L[i]*L[i]);
    local[i][3][3] = 4/L[i];
}
```

Fig. 4 Member Stiffness matrix or Local matrix

Second Step, to form Global Stiffness matrix –

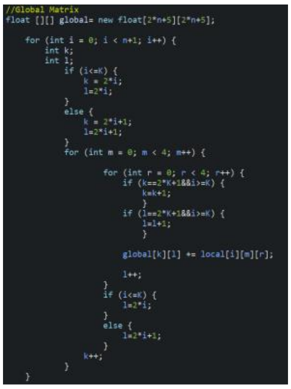
After the structure stiffness matrix is determined, the loads at the nodes of the beam can be related to the displacements using the structure stiffness equation.

F=K*D ……….()



	1	2	3	4	5	6	7	8	9	10	11
1	11	12	13	14							
2	21	22	23	24							
3	31	32	33 + 33	34 +34	35	36					
4	41	42	43 + 43	44 +44	45	46					
5			53	54	55 + 55	56	57	58	59		
6			63	64	65	66					
7					75		77	78	79		
8					85		87	88 +88	89 + 89	8 10	8 11
9								9 8	9 9	9 10	9 11
10								10 8	10 9	10 10	10 11
11								11 8	11 9	11 10	11 11

Global Matrix

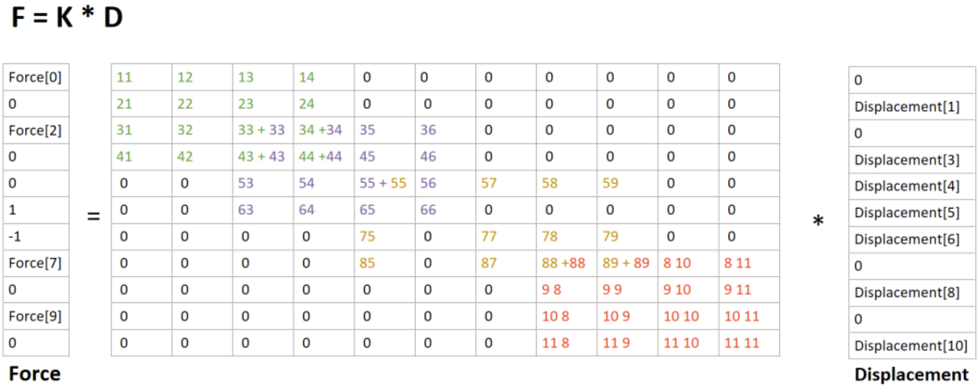


```
//Global Matrix
float [][] global= new float[2*n+5][2*n+5];

    for (int i = 0; i < n+1; i++) {
        int k;
        int l;
            if (i<=K) {
                k = 2*i;
                l=2*i;
            }
            else {
                k = 2*i+1;
                l=2*i+1;
            }
            for (int m = 0; m < 4; m++) {

                    for (int r = 0; r < 4; r++) {
                        if (k==2*K+1&&i>=K) {
                            k=k+1;
                            }
                        if (l==2*K+1&&i>=K) {
                            l=l+1;
                            }

                        global[k][l] += local[i][m][r];

                        l++;
                    }
                    if (i<=K) {
                        l=2*i;
                    }
                    else {
                        l=2*i+1;
                    }
                k++;
            }
    }
```

Fig. 5 Global Stiffness matrix

Here Q and D are column matrices that represent both the known and unknown forces(both vertical forces and moments) and displacements(both slope and vertical displacements).

Partitioning the stiffness matrix into the known and unknown elements of load and displacement, we have



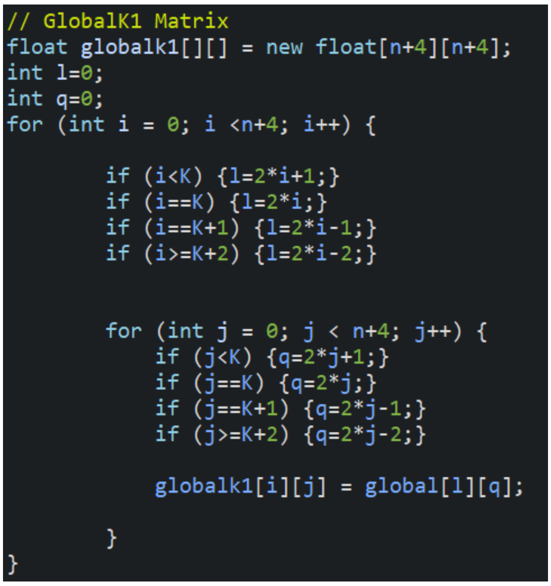
F = K * D

Force[0]		11	12	13	14	0	0	0	0	0	0	0		0
0		21	22	23	24	0	0	0	0	0	0	0		Displacement[1]
Force[2]		31	32	33 + 33	34 +34	35	36	0	0	0	0	0		0
0		41	42	43 + 43	44 +44	45	46	0	0	0	0	0		Displacement[3]
0		0	0	53	54	55 + 55	56	57	58	59	0	0		Displacement[4]
1	=	0	0	63	64	65	66	0	0	0	0	0	*	Displacement[5]
-1		0	0	0	0	75	0	77	78	79	0	0		Displacement[6]
Force[7]		0	0	0	0	85	0	87	88 +88	89 + 89	8 10	8 11		0
0		0	0	0	0	0	0	0	9 8	9 9	9 10	9 11		Displacement[8]
Force[9]		0	0	0	0	0	0	0	10 8	10 9	10 10	10 11		0
0		0	0	0	0	0	0	0	11 8	11 9	11 10	11 11		Displacement[10]
Force														**Displacement**

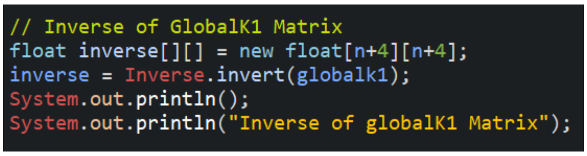
Fig.6 Global Stiffness matrix

Third Step, Code for computing displacements, solution of the fig.6 -

There will be either known force or known displacement in above matrix equation. There is an analogy that displacements are either zero or unknown. Then unknown displacements are calculated by diminishing the global matrix to smaller matrix GlobalK1 matrix.



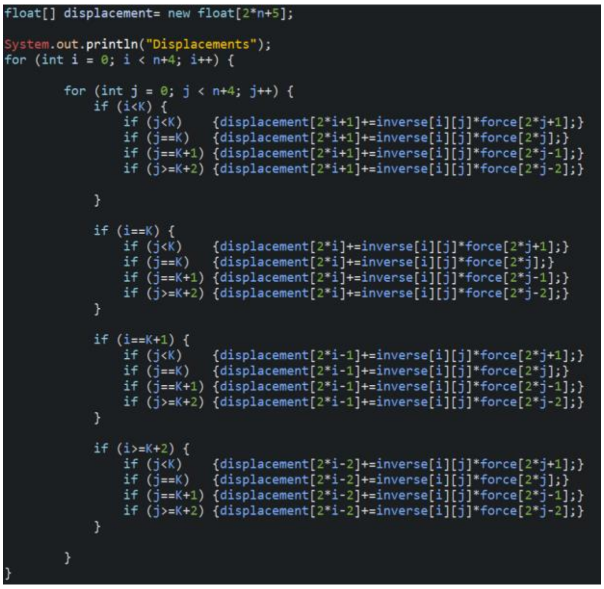
```
// GlobalK1 Matrix
float globalk1[][] = new float[n+4][n+4];
int l=0;
int q=0;
for (int i = 0; i <n+4; i++) {

        if (i<K) {l=2*i+1;}
        if (i==K) {l=2*i;}
        if (i==K+1) {l=2*i-1;}
        if (i>=K+2) {l=2*i-2;}


        for (int j = 0; j < n+4; j++) {
            if (j<K) {q=2*j+1;}
            if (j==K) {q=2*j;}
            if (j==K+1) {q=2*j-1;}
            if (j>=K+2) {q=2*j-2;}

            globalk1[i][j] = global[l][q];

        }
}
```



```
// Inverse of GlobalK1 Matrix
float inverse[][] = new float[n+4][n+4];
inverse = Inverse.invert(globalk1);
System.out.println();
System.out.println("Inverse of globalK1 Matrix");
```

Fig. 8 Inverse function inverses any matrix

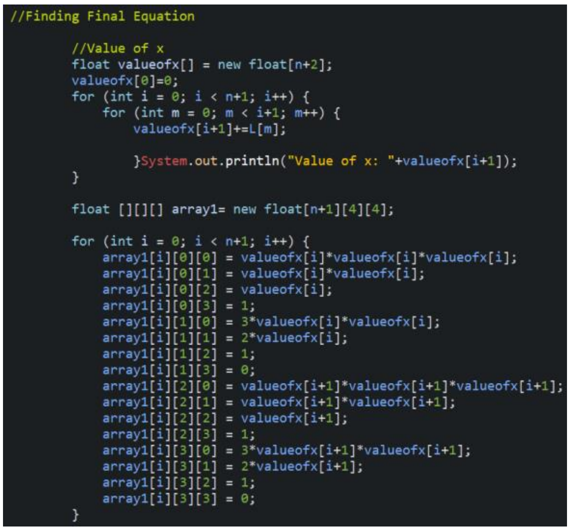
Fig. 7 GloblK1 Matrix and its Inverse



```
float[] displacement= new float[2*n+5];

System.out.println("Displacements");
for (int i = 0; i < n+4; i++) {

        for (int j = 0; j < n+4; j++) {
            if (i<K) {
                if (j<K)    {displacement[2*i+1]+=inverse[i][j]*force[2*j+1];}
                if (j==K)   {displacement[2*i+1]+=inverse[i][j]*force[2*j];}
                if (j==K+1) {displacement[2*i+1]+=inverse[i][j]*force[2*j-1];}
                if (j>=K+2) {displacement[2*i+1]+=inverse[i][j]*force[2*j-2];}

            }

            if (i==K) {
                if (j<K)    {displacement[2*i]+=inverse[i][j]*force[2*j+1];}
                if (j==K)   {displacement[2*i]+=inverse[i][j]*force[2*j];}
                if (j==K+1) {displacement[2*i]+=inverse[i][j]*force[2*j-1];}
                if (j>=K+2) {displacement[2*i]+=inverse[i][j]*force[2*j-2];}
            }

            if (i==K+1) {
                if (j<K)    {displacement[2*i-1]+=inverse[i][j]*force[2*j+1];}
                if (j==K)   {displacement[2*i-1]+=inverse[i][j]*force[2*j];}
                if (j==K+1) {displacement[2*i-1]+=inverse[i][j]*force[2*j-1];}
                if (j>=K+2) {displacement[2*i-1]+=inverse[i][j]*force[2*j-2];}
            }

            if (i>=K+2) {
                if (j<K)    {displacement[2*i-2]+=inverse[i][j]*force[2*j+1];}
                if (j==K)   {displacement[2*i-2]+=inverse[i][j]*force[2*j];}
                if (j==K+1) {displacement[2*i-2]+=inverse[i][j]*force[2*j-1];}
                if (j>=K+2) {displacement[2*i-2]+=inverse[i][j]*force[2*j-2];}
            }

        }
}
```

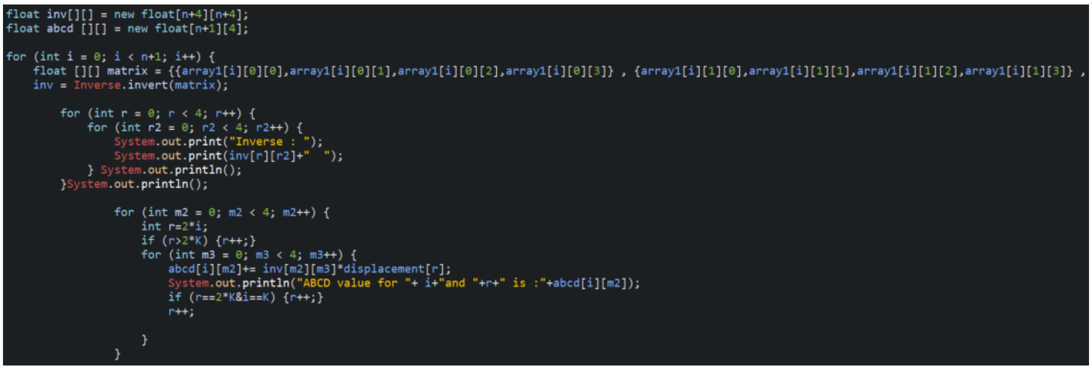
Fig. 9 Displacements by matrix multiplication of inverse and force

From the above matrix equation, we compute displacements which we can use in the above result of ILD problem 4 to compute the bending of beam.



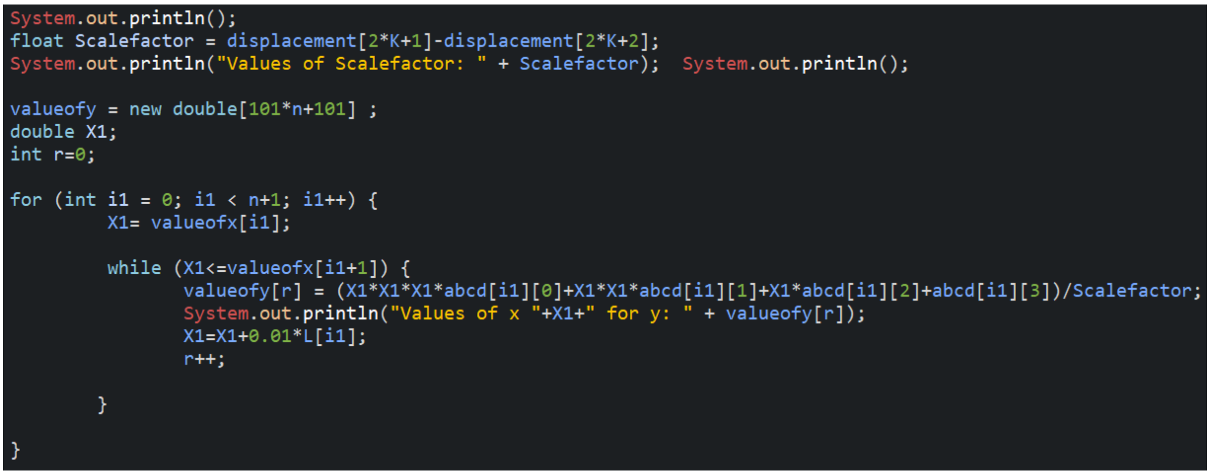
```
//Finding Final Equation

        //Value of x
        float valueofx[] = new float[n+2];
        valueofx[0]=0;
        for (int i = 0; i < n+1; i++) {
            for (int m = 0; m < i+1; m++) {
                valueofx[i+1]+=L[m];

                }System.out.println("Value of x: "+valueofx[i+1]);
        }

        float [][][] array1= new float[n+1][4][4];

        for (int i = 0; i < n+1; i++) {
            array1[i][0][0] = valueofx[i]*valueofx[i]*valueofx[i];
            array1[i][0][1] = valueofx[i]*valueofx[i];
            array1[i][0][2] = valueofx[i];
            array1[i][0][3] = 1;
            array1[i][1][0] = 3*valueofx[i]*valueofx[i];
            array1[i][1][1] = 2*valueofx[i];
            array1[i][1][2] = 1;
            array1[i][1][3] = 0;
            array1[i][2][0] = valueofx[i+1]*valueofx[i+1]*valueofx[i+1];
            array1[i][2][1] = valueofx[i+1]*valueofx[i+1];
            array1[i][2][2] = valueofx[i+1];
            array1[i][2][3] = 1;
            array1[i][3][0] = 3*valueofx[i+1]*valueofx[i+1];
            array1[i][3][1] = 2*valueofx[i+1];
            array1[i][3][2] = 1;
            array1[i][3][3] = 0;
        }
```

Fig. 10 Formulating Final equation variable as function of global x



```
float inv[][] = new float[n+4][n+4];
float abcd [][] = new float[n+1][4];

for (int i = 0; i < n+1; i++) {
    float [][] matrix = {{array1[i][0][0],array1[i][0][1],array1[i][0][2],array1[i][0][3]} , {array1[i][1][0],array1[i][1][1],array1[i][1][2],array1[i][1][3]} ,
    inv = Inverse.invert(matrix);

        for (int r = 0; r < 4; r++) {
            for (int r2 = 0; r2 < 4; r2++) {
                System.out.print("Inverse : ");
                System.out.print(inv[r][r2]+"  ");
            } System.out.println();
        }System.out.println();

                for (int m2 = 0; m2 < 4; m2++) {
                    int r=2*i;
                    if (r>2*K) {r++;}
                    for (int m3 = 0; m3 < 4; m3++) {
                        abcd[i][m2]+= inv[m2][m3]*displacement[r];
                        System.out.println("ABCD value for "+ i+"and "+r+" is :"+abcd[i][m2]);
                        if (r==2*K&i==K) {r++;}
                        r++;

                    }
                }
```

Fig. 11 Computing coefficients of individual elements



```
System.out.println();
float Scalefactor = displacement[2*K+1]-displacement[2*K+2];
System.out.println("Values of Scalefactor: " + Scalefactor);  System.out.println();

valueofy = new double[101*n+101] ;
double X1;
int r=0;

for (int i1 = 0; i1 < n+1; i1++) {
         X1= valueofx[i1];

         while (X1<=valueofx[i1+1]) {
                valueofy[r] = (X1*X1*X1*abcd[i1][0]+X1*X1*abcd[i1][1]+X1*abcd[i1][2]+abcd[i1][3])/Scalefactor;
                System.out.println("Values of x "+X1+" for y: " + valueofy[r]);
                X1=X1+0.01*L[i1];
                r++;

        }

}
```

Fig. 12 Equation of Influence line using the above calculated coefficient

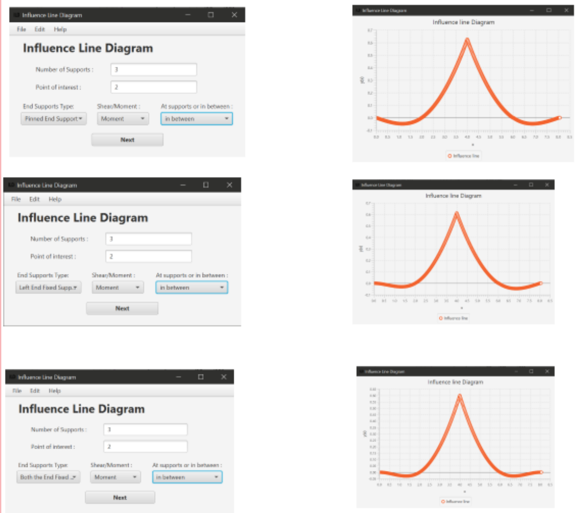
Results

All possible cases are shown below with values that are used to calculate the output.

Solutions

Tuesday, July 20, 2021 12:02 AM

In Between
Moment



Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Pinned End Support
Shear/Moment : Moment
At supports or in between : in between

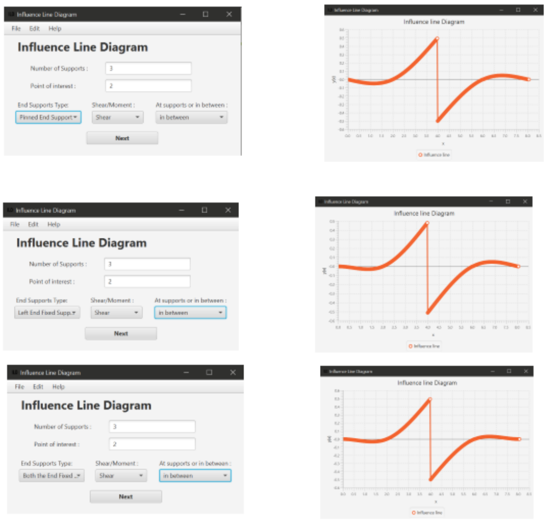
Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Left End Fixed Supp.
Shear/Moment : Moment
At supports or in between : in between

Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Both the End Fixed
Shear/Moment : Moment
At supports or in between : in between

In Between
Shear



Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Pinned End Support
Shear/Moment : Shear
At supports or in between : in between

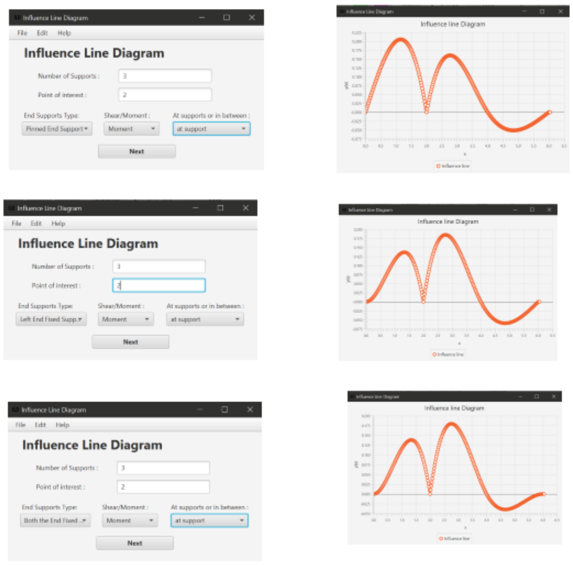
Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Left End Fixed Supp.
Shear/Moment : Shear
At supports or in between : in between

Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Both the End Fixed
Shear/Moment : Shear
At supports or in between : in between

At Support
Moment



Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Pinned End Support
Shear/Moment : Moment
At supports or in between : at support

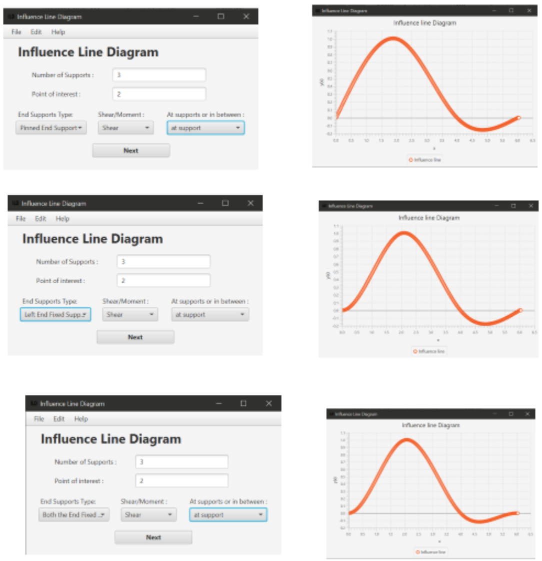
Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Left End Fixed Supp.
Shear/Moment : Moment
At supports or in between : at support

Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Both the End Fixed
Shear/Moment : Moment
At supports or in between : at support

At Support
Shear



Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Pinned End Support
Shear/Moment : Shear
At supports or in between : at support

Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Left End Fixed Supp.
Shear/Moment : Shear
At supports or in between : at support

Influence Line Diagram

Number of Supports : 3
Point of interest : 2
End Supports Type: Both the End Fixed
Shear/Moment : Shear
At supports or in between : at support

Summary

The **influence line diagram** is an essential part of analysis while designing structures like bridges, crane rails, conveyor belts, floor girders, and other structures where loads will move along their span. The influence lines show where a load will create the maximum effect for any of the functions studied.

The central principle used across the project is the Muller Breslau Principle which gives the result directly without calculating individually for different positions of unit load. Influence lines based on Maxwell reciprocal work theorem. Also, the Double Integration method is used to find the bending of the beam with known boundary conditions.

Java and **JavaFX** are used to create the backend and front end of the application, respectively.

The project aims to develop an interactive user application that can compute graphs for influence lines for any indeterminate beam using the Muller Breslau principle. Initial approaches were based on a manual calculation to formulate an equation that can produce ILD values for corresponding positions, but it is not easy to calculate manually for high indeterminacy. Hence, the Stiffness matrix method used to compute the displacements and then, with the help of the result in ILD problem five, the deflection of the beam is computed.

The first four problems consist of manual calculation to find functions in terms of the position of unit load. The section "Application and Programming" in which the ILD for any number of supports is calculated using the stiffness matrix method due to high indeterminacy.

The results are verified using STAAD Pro and problems from the book mentioned in references.

References

1. R.C Hibbeler, Structural Analysis, fourth edition, Prentice-Hall, NJ, 1998
2. Dr. Moujalli Hourani. 2002. Mathematical Model of Influence Lines for Indeterminate Beams. 7.841.1 - 7.841.14
3. Kamal B. Rojiani and Robert Schottler. 2002. Java Applet for Teaching Influence Line Analysis. 7.772.1 - 7.772.10
4. Yasha Zeinali and Brett A. 2017. Framework for Flexural Rigidity Estimation in Euler-Bernoulli Beams Using Deformation Influence Lines. 1-19
5. Herbert Schildt. 2014. Java: The Complete Reference, Ninth Edition
6. Richard M. Barker and Jay A. Puckett. 2013. Design of Highway Bridges An LRFD Approach Third Edition
7. Herbert Schildt. 2015. Introducing JavaFX 8 Programming