CIVL2530: Moments and Forces on Beams

Dorival Pedroso and Samuel Hislop-Lynch

Case 1: Simply supported beam

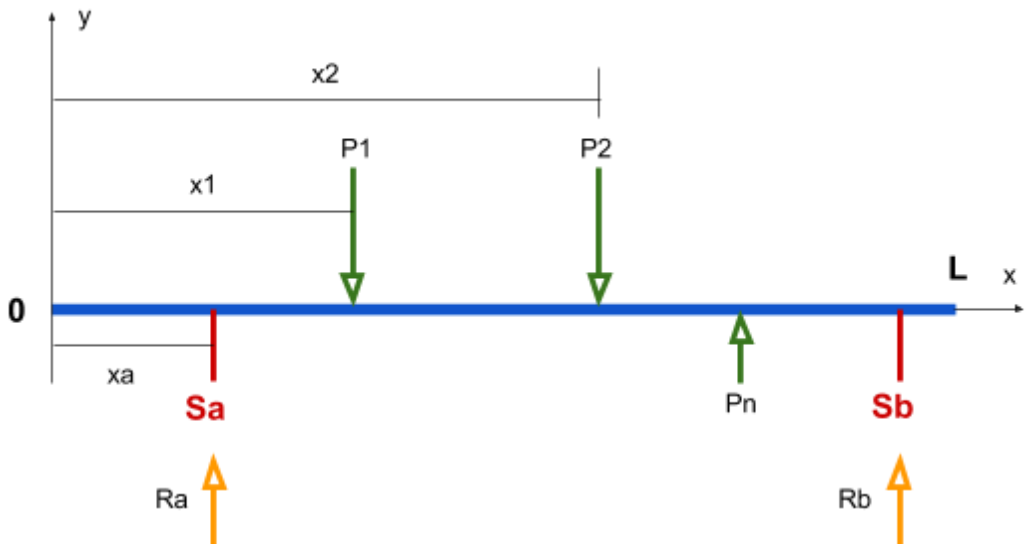


Figure 1. Simply supported beam

To compute the reaction forces on the beam of Figure 1, the principle of moments of forces can be employed. In this principle, the sum of moment around point 0 in the figure has to be zero. Here, we consider that each feature (point load **Pi** or reaction **Ri**) can be uniquely located with its x-distance (**xi**) to the left extremity of the beam. Also, each feature has positive magnitude if pointing in the positive direction of the y-axis. Thus, in the figure, Ra, Rb, and Pn are positive whereas P1 and P2 are negative.

The following formula, derived from the balance of moments, can be used to compute the reaction at support b (remember that, in Figure 1, P1 and P2 are negative numbers) :

$$Rb \cdot (xb - xa) + P1 \cdot (x1 - xa) + P2 \cdot (x2 - xa) + \cdots + Pn \cdot (xn - xa) = 0$$

Thus:

$$Rb = -\left[P1 \cdot (x1 - xa) + P2 \cdot (x2 - xa) + \cdots + Pn \cdot (xn - xa)\right] / (xb - xa)$$

Likewise, the following formula can be used to compute the reaction at support a:

$$Ra \cdot (xa - xb) + P1 \cdot (x1 - xb) + P2 \cdot (x2 - xb) + \cdots + Pn \cdot (xn - xb) = 0$$

Thus:

$$Ra = -\left[P1 \cdot (x1 - xb) + P2 \cdot (x2 - xb) + \cdots + Pn \cdot (xn - xb)\right] / (xa - xb)$$

Note that in the equation above, the signs automatically lead to the right results.

Example 1.

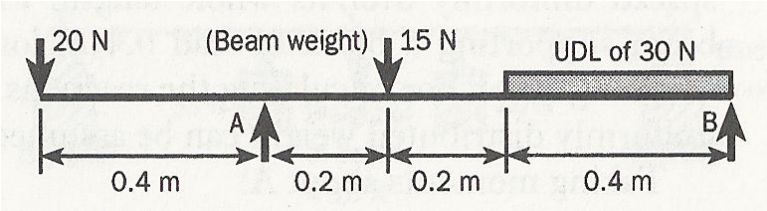


Figure 2. Example in Fig. 5.18 of [Al Nageim et. al 2010], page 82.

In the example of Figure 2, the uniformly distributed load can be converted into a point load located at its centroid. **Note** that 30 N is the resulting equivalent load of the UDL. Usually, the UDL is given in force per metre and not the equivalent load directly. The table below collects all information about the beam in Figure 2 (with **xa=0.4** and **xb=1.2**).

Name	Load value: **Pi**	Pos. from left: **xi**	**xi - xa**	**xi - xb**	**Pi * (xi-xa)**	**Pi * (xi-xb)**
P1	-20.0	0.0	-0.4	-1.2	8.0	24.0
P2	-15.0	0.6	+0.2	-0.6	-3.0	9.0
P3 (udl)	-30.0	1.0	+0.6	-0.2	-18.0	6.0
				Sum =	-13.0	39.0

Thus:

$$Rb = -\sum Pi \cdot (xi - xa) \;/\; (xb - xa) \;=\; -(-13.0)/0.8 \;=\; 16.25$$

$$Ra = -\sum Pi \cdot (xi - xb) \;/\; (xa - xb) \;=\; -(39.0)/-0.8 \;=\; 48.75$$

You can check that the sum of Ra and Rb make sense. In fact, they have to equal the sum of the vertical loads. Thus:

$$Ra + Rb \;=\; 16.25 + 48.75 \;=\; 65.0 \;=\; 20.0 + 15.0 + 30.0$$ **OK**

Example 2.

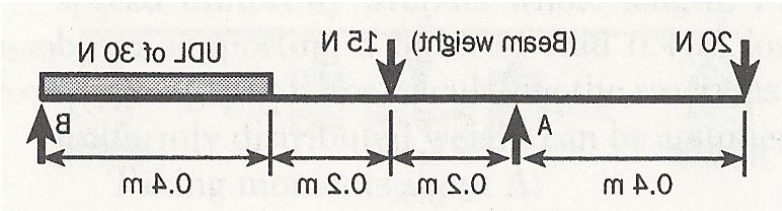


Figure 3. Mirrored version of Example 1.

Let's solve Example 1 again, but now with a mirrored beam around a vertical axis along the right-hand-side (Figure 3). Also, in Figure 3, we will call B as **a** and A as **b**. Furthermore, the numbering of loads will be changed from left to right. The data is collected in the table below (with **xa=0.0** and **xb=0.8**):

Name	Load value: **Pi**	Pos. from left: **xi**	**xi - xa**	**xi - xb**	**Pi * (xi-xa)**	**Pi * (xi-xb)**
P1 (udl)	-30.0	0.2	+0.2	-0.6	-6.0	18.0
P2	-15.0	0.6	+0.6	-0.2	-9.0	3.0
P3	-20.0	1.2	+1.2	+0.4	-24.0	-8.0
				Sum =	-39.0	13.0

Thus:

$$Rb = -\sum Pi \cdot (xi - xa) \ / \ (xb - xa) \ = \ -(-39.0) / 0.8 \ = \ 48.75$$

$$Ra = -\sum Pi \cdot (xi - xb) \ / \ (xa - xb) \ = \ -(13.0) / \ -0.8 \ = 16.25$$

Therefore, the reactions of corresponding supports have the same values! (as expected).

Case 2: Cantilever beam

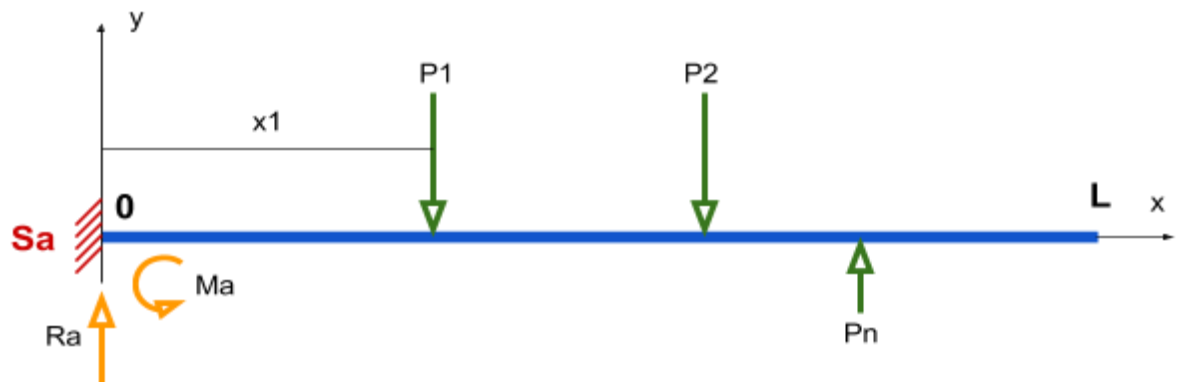


Figure 4. Cantilever beam

The principle to calculate the reactions in a cantilever beam (Figure 4) is the same, of course. Now the balance of moment and the balance of vertical forces shall be used. With the same notation as before, the moment around the support a gives:

$$Ma \ + \ \sum Pi \cdot (xi - xa) \ = \ 0$$

Thus:

$$Ma \ = \ -\sum Pi \cdot (xi - xa)$$

The above equation works whether the fixed support is at the left or at the right of the beam.

From the balance of vertical forces:

$$Ra + \sum Pi = 0$$

Thus:

$$Ra = -\sum Pi$$

Example 3.

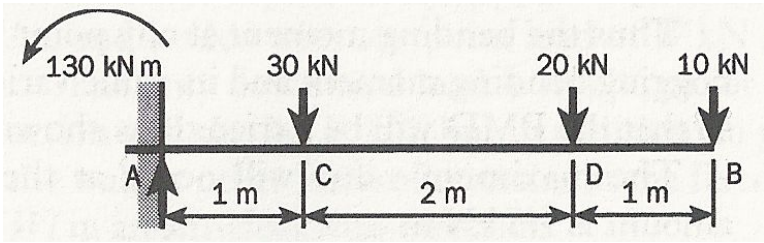


Figure 5. Example 8.1 from [Al Nageim et. al 2010], page 156.

This example is illustrated in Figure 5. The data is collected in the table below (with **xa = 0.0**):

Name	Load value: **Pi**	Pos. from left: **xi**	**xi - xa**	**Pi * (xi-xa)**
P1	-30.0	1.0	1.0	-30.0
P2	-20.0	3.0	3.0	-60.0
P3	-10.0	4.0	4.0	-40.0
				-130.0

Therefore:

$Ma = -130.0$ Counter-clockwise (OK)

And

$Ra = -(-30.0-20.0-10.0) = 60.0$ Point upwards (OK)

Example 4.

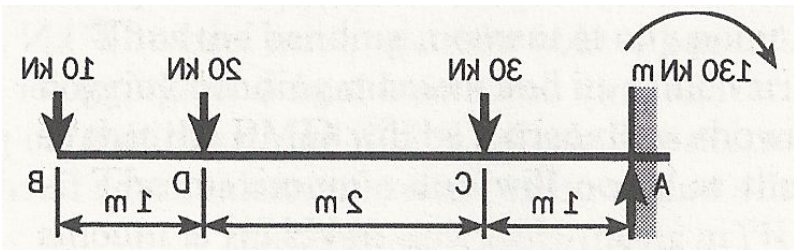


Figure 6. Mirrored version of Figure 5.

Let's solve the previous example again, but now the beam is mirrored around a vertical axis at the right side (Figure 6). Here, we will call A as **b**. In this case, **xb = 4.0.** The data is collected in the next table:

Name	Load value: **Pi**	Pos. from left: **xi**	**xi - xb**	**Pi * (xi-xb)**
P1	-10.0	0.0	-4.0	+40.0
P2	-20.0	1.0	-3.0	+60.0
P3	-30.0	3.0	-1.0	+30.0
				+130.0

Therefore:

$Mb = +130.0$ Clockwise (OK)

And

$Rb = -(-10.0-20.0-30.0) = 60.0$ Point upwards (OK)

Questionnaire: Code development

In this section, we will develop a code in Julia to automatise the computation of reactions on simply supported beams or cantilevers.

Question 1: Create a user defined-type (named **`Support`**) to hold the following information about supports:

1. `Type`: the type of support; e.g. “fixed”, “pin”, “roller”, “none”
2. `X`: the position of the support along beam (from the left; see e.g. xa in Figure 1)
3. `DI`: the degree of indeterminacy of the support
4. `Ry`: the y-reaction @ this support
5. `Mr`: the moment reaction @ this support

Add (lots of) comments! Add comment about the first two data members (Type, X) being input data while the last three data members (DI, Ry, Mr) being computed (or derived) data.

Question 2: Create a function named **`NewSupport`** that creates a `Support` with the following information while keeping the other data equal to zero:

1. L: the length of a beam that will use this support
2. Type: the support type as in Question 1
3. X: the position of the support

This function must return a Support. Within this function, check for cases when the position X is outside the beam; e.g. if X < 0 or X > L. If the support is outside, use the command `throw("support is outside beam")` to stop all computations. Finally, add a comment just before the function definition with a text like this:

```
# NewSupport creates a new support
#   Input:
#     L    -- length of Beam
#     Type -- type of support: fixed, pin, roller, none
#     X    -- position along beam
#   Output:
#     a new Support object
```

Question 3: Create a function named **`SupportSummary`** that prints a nice message with all `Support` data. This function must take the following input parameters:

1. o: the variable holding a previously created `Support`
2. prefix: a text to add before printing each data member of `Support`

Add a comment just before the function definition with a text like this:

```
# SupportSummary prints summary about support
#   Input:
#     o      -- handle of Support object
#     prefix -- a small text to prefix each printed variable
```

For example, the output of the `SupportSummary` function should look like this (with prefix = " "; i.e. 9 spaces):

```
         Type = pin
         X    = 0.0
         DI   = 2
         Ry   = 37.5
         Mr   = 0.0
```

Question 4: Create a user defined-type (named **`PointLoad`**) to hold the following information about point loads:

1. P: the load value considering the convention of Figure 1
2. X: the position of the point load along beam (see Figure 1)

Add comments!

Question 5: Create a function named **`NewPointLoad`** that creates a `PointLoad` with the following information:

1. L: the length of a beam that will have this point load
2. P: the load value (-) pointing downwards, (+) point upwards
3. X: the position of the PointLoad

This function must return a `PointLoad`. In this function, check for cases when the position is outside the beam. "Throw" an error in this case. Add a comment just before the function definition with something like this:

```
#   Input:
#     L -- length of beam
#     P -- point load value (-) means downwards, (+) means upwards
#     X -- position along beam
#   Output:
#     a new PointLoad object
```

Question 6: Create a function named **`PointLoadSummary`** that prints a nice message with all `PointLoad` data. This function must take the following input parameters:

1. o: the variable holding a previously created `PointLoad`
2. prefix: a text to add before printing each data member of `PointLoad`

Add a comment just before the function definition with a text like this:

```
# PointLoadSummary prints a summary of PointLoad data
#   Input:
#     o      -- handle of PointLoad object
#     prefix -- a small text to prefix each printed variable
```

For example, the output of the `PointLoadSummary` function should look like this (with prefix = " "; i.e. 8 spaces):

```
        P = -15.0
        X = 0.6
```

Question 7: Create a user defined-type (named **`Beam`**) to hold the following information about single supported beams or cantilevers:

1. L: length of beam
2. A: cross-sectional area of beam
3. Gamma: material unit weight
4. W: weight of beam
5. Xc: centroid of beam
6. Lsup: the left support (s `Support` type)
7. Rsup: the right support (a `Support` type)
8. Loads: an array of point loads; i.e. an array of `PointLoad`

Add comments! Add comments explaining that L, A and Gamma are essential data, W and Xc are computed data, and Lsup, Rsup, Loads are "supports and loads" data :-).

Question 8: Create function named **`NewBeam`** to create a `Beam` type by giving the following information:

1. L: length of beam
2. A: cross-sectional area
3. Gamma: unit weight of beam material
4. selfWeight: a flag indicating if the self weight of the beam should be computed and added as an equivalent point load

The function must return a `Beam`. By default, this function must set the left and right supports as being "pin" and "roller", respectively. Also, this function must add point load to the array Loads if selfWeight == true. Just before the function definition, add a comment such as:

```
# NewBeam returns a new Beam
#   Input:
#     L          -- length of beam
#     A          -- cross sectional area
#     Gamma      -- unity weight of beam material; e.g. 24 kN/m^3
#     selfWeight -- computes self-weight or not
#   Output:
#     a new beam object
```

Question 9: Create function named **`BeamSetSupport`** to set either the left or right support of a Beam type. The input data for this function is as follows:

1. o: the variable holding the Beam type
2. right: a flag telling if the right support is being set
3. Type: a string corresponding to the type of support; e.g. "fixed", "pin", "roller", "none"
4. X: the location of the support

The function does not return anything, but changes the Lsup and Rsup of Beam, accordingly; e.g. by using the `NewSupport` function. Add comments, including a comment just before the function definition like this one:

```
# BeamSetSupport sets support of beam
#   Input:
#     o     -- handle of Beam object
```

```
#      right -- set right support instead of left [default is left]
#      Type  -- type of support
#      X     -- position of support along beam
#    Output:
#      the beam object (o) will be modified
```

Question 10: Create function named **`BeamAddPointLoad`** to append a point load to the Loads member of `Beam`. This function must accept:

1. o: the variable holding the beam object
2. P: the load value
3. X: the position of the load

The function will modify the o object. Add this comment before the function definition:

```
# BeamAddPointLoad adds point load to Beam
#    Input:
#      o -- handle of Beam object
#      P -- point load value (-) means downwards, (+) means upwards
#      X -- position of point load along beam
#    Output:
#      the beam object (o) will be modified
```

Question 11: Create function named **`BeamAddUniDistLoad`** to append a point load to the Beam object with the equivalent point load computed from the given uniformly distributed load (UDL). This function takes as input:

1. o: the variable holding the beam object
2. load: the distributed load value
3. startX: the position (from left, according to the system in Figure 1) of the start of the distributed load on the beam
4. length: the length of the distributed load; startX + length must be on the beam

The function will modify the beam object. Make sure to check for the case when the distributed load may be outside the beam. Throw an error in this case. Compute the equivalent point load and add to the Loads variable of **o**. Add a comment such as:

```
# BeamAddUniDistLoad adds uniformly distributed load (UDL) to Beam
#    Input:
#      o      -- handle of Beam object
#      load   -- distributed load value (e.g. kN/m) (-) points down, (+)
points up
#      startX -- distance from left of UDL
#      length -- length of UDL
#    Output:
#      the beam object (o) will be modified
```

Question 12: Create function named **`BeamReactions`** to calculate the left and right reactions of a simply supported or cantilever beam. This function only takes as input the variable (o) holding the `Beam` object. Add a comment explaining what this function does.

Question 13: Create a function named **`BeamSummary`** that prints a nice message with all `Beam` data. This function must take the following input parameters:

1. o: the variable holding a previously created `Beam`
2. description: a text to add when printing information about this `Beam`

Question 14: Create a function named **`checkFloat`** that compares two Float64 (float-point numbers) by using a small tolerance. This function accepts the following input:

1. name: a string indicating the name of this float-point number
2. tolerance: the tolerance such that $|u - v| \leq tolerance$ is OK
3. u: a float number to be compared with the other one; e.g. coming from a numerical computation
4. v: another float number to be compared with the previous one; e.g. coming from a closed-form (analytical) computation

This function must throw an error if the difference between the u and v values is greater than the given tolerance.

Question 15: Test your "beam code" with the four examples of the previous section. Use the `checkFloat` function to verify the just computed reaction values. For instance, the first example can be solved **and tested** with the following code:

```
# test 1
b1 = NewBeam(1.2, 0.0, 0.0, false)
BeamSetSupport(b1, false, "pin", 0.4)
BeamSetSupport(b1, true, "roller", 1.2)
BeamAddPointLoad(b1, -20.0, 0.0)
BeamAddPointLoad(b1, -15.0, 0.6)
BeamAddPointLoad(b1, -30.0, 1.0)
BeamReactions(b1)
BeamSummary(b1, "test 1")
println()
checkFloat("Ra", 1e-14, b1.Lsup.Ry, 48.75)
checkFloat("Rb", 1e-14, b1.Rsup.Ry, 16.25)
println()
```

Question 16: Solve all problems in the Appendix ;-)

Question 17: Consider the beam illustrated in Figure 7. The three loads on this structure correspond to, respectively (from left to right):

- 35 N is a hanging pipe transversal to the beam direction;
- 15 N is the self-weight of the beam; and
- 20 N is due to hanging electrical cables also transversal to the beam.

Since the hanging features are installed using “hooks”, their position are certain. Also, the location of the self-weight and its magnitude are certain. Nonetheless, the magnitudes of the 35 N and 20 N loads are uncertain. We only know that they are normally distributed with mean of 35 N and 20 N, respectively. The standard deviation of each one is then estimated to be 0.35 and 0.2, respectively. We also assume that the loads are uncorrelated; i.e. the pipes have nothing to do with the electrical cables.

Thus, the questions that a structural engineer should answer are:

1. What is the probability that the reaction at the left support will be greater than 38 N?
2. What is the probability that the reaction at the right support will be greater than 38 N?
3. If 38 N is the maximum capacity of the supports (e.g. of the columns), what is the probability that this structure will fail?

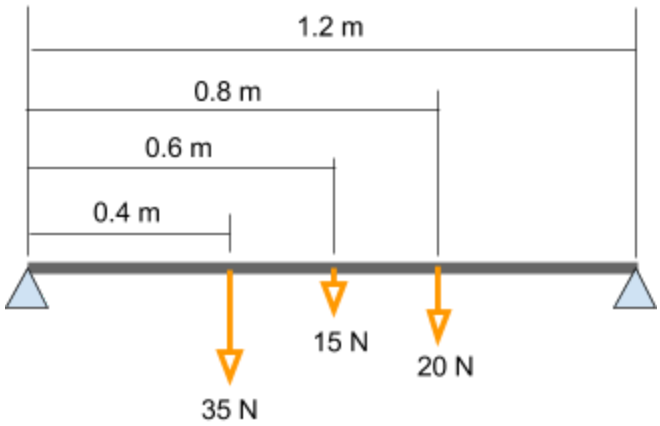


Figure 7. Beam for Question 17.

References

Al Nageim H, Durka F, Morgan W, Williams D (**2010**) *Structural Mechanics: Loads, Analysis Materials and Design of Structural Elements*. 7th Edition. Pearson Prentice Hall. 436p

Appendix A: problems from [Al Nageim 2010], Chapter 5.

1 A uniform rod is in equilibrium under the action of weights as shown in Fig. 5.Q1. Calculate the value of W and the reaction at the fulcrum, ignoring the weight of the rod.

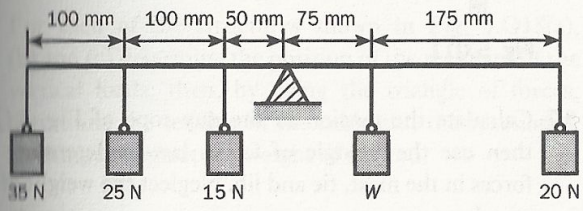


Fig. 5.Q1

2 Calculate the value of x metres so that the uniform rod of Fig. 5.Q2 will balance.

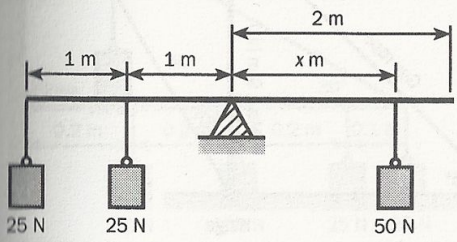


Fig. 5.Q2

3 A uniform rod weighing 100 N and supporting 500 N is hinged to a wall and kept horizontal by a vertical rope (Fig. 5.Q3). Calculate the tension in the rope and the reaction at the hinge.

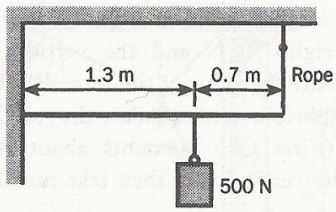


Fig. 5.Q3

4 A uniform rod weighing 40 N is maintained in equilibrium as shown in Fig. 5.Q4. Calculate the distance x and the reaction at the fulcrum.

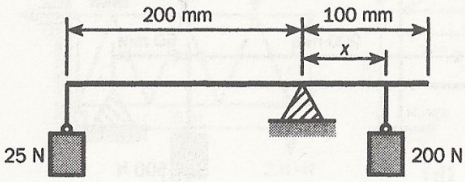


Fig. 5.Q4

5 A uniform rod 0.4 m long weighing 10 N supports loads as shown in Fig. 5.Q5. Calculate the distance x if the rod is in equilibrium.

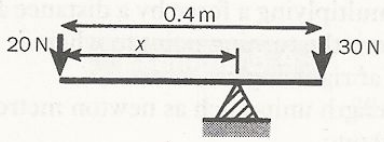


Fig. 5.Q5

6 A uniform horizontal beam 1.3 m long and weighing 200 N is hinged to a wall and supported by a vertical prop as shown in Fig. 5.Q6. Calculate the reactions at the prop and hinge.

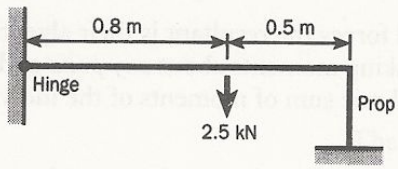


Fig. 5.Q6

7 Calculate the reaction at the prop and hinge in Fig. 5.Q7. The uniform beam weighs 200 N.

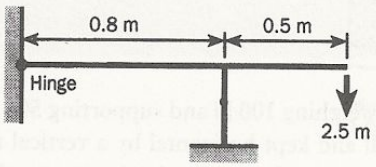


Fig. 5.Q7

8 A compound lever is shown in Fig. 5.Q8. Each of the horizontal rods weighs 50 N and the vertical rod weighs 40 N. Calculate the value of the force W which is required for equilibrium, and calculate the reactions at the fulcrums. (First take moments about A to find the force in the vertical rod, then take moments about B.)

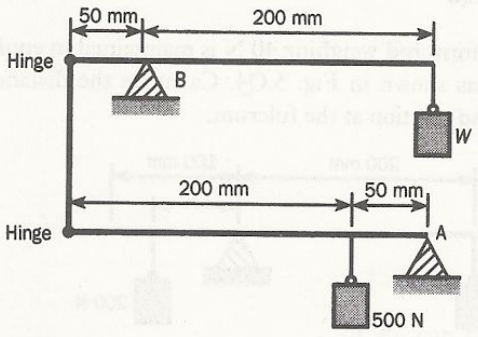


Fig. 5.Q8

9 Solve Exercises 1 to 13 of Chapter 4, with the aid of the principle of moments.

10 Determine the tension in the chain and the reaction at the hinge (Fig. 5.Q10). The uniform rod weighs 500 N.

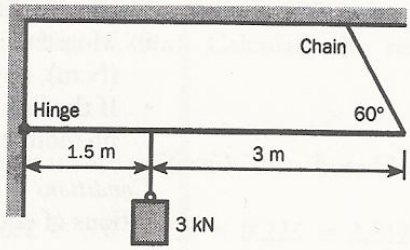


Fig. 5.Q10

11 The horizontal cable in Fig. 5.Q11 is attached to the rod halfway along its length. The uniform rod weighs 250 N. Calculate the tension in the cable and the reaction at the hinge.

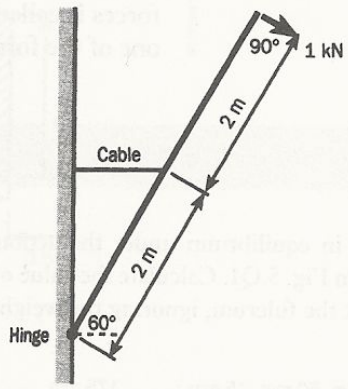


Fig. 5.Q11

12 Calculate the tension in the guy rope of Fig. 5.Q12, then use the triangle of forces law to determine the forces in the mast, tie and jib. Neglect the weight of the members.

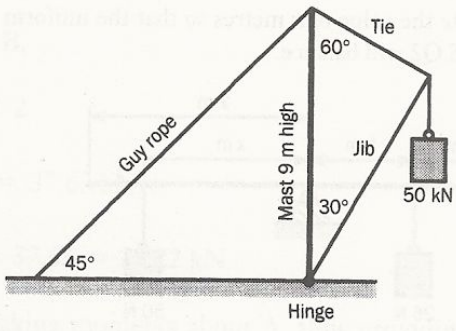


Fig. 5.Q12

13 A uniform rod weighing 50 N is loaded as shown in Fig. 5.Q13. Calculate the tension in the cable and the reaction at the hinge.

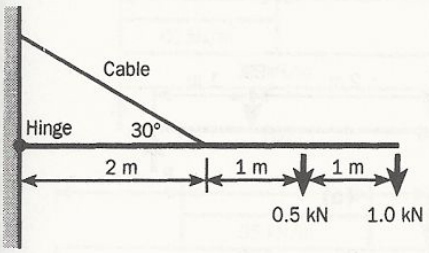


Fig. 5.Q13

14 Two uniform rods each 3 m long and weighing 250 N are connected by a link bar (Fig. 5.Q14). Calculate the tensions in the link bar and chain neglecting their weights.

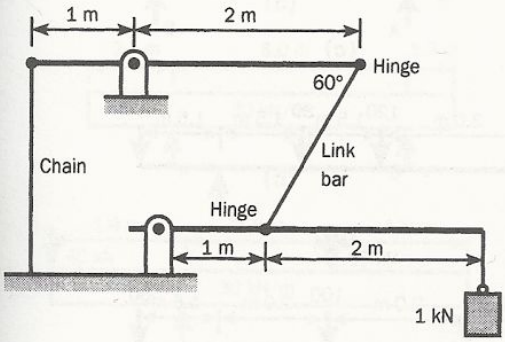
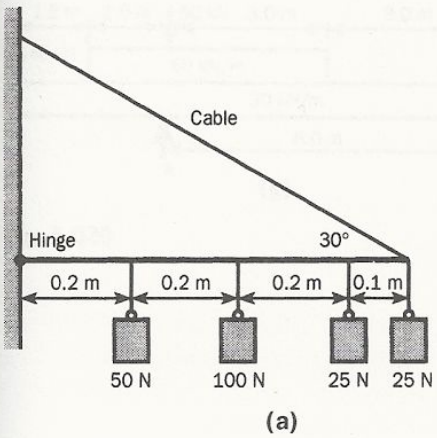


Fig. 5.Q14

15 For each of the structures shown in Fig. 5.Q15(a), (b) and (c) determine the position of the resultant of the vertical loads; then, by using the triangle of forces, determine the tension in the cable and the reaction at the hinge. Neglect the weight of the rod.



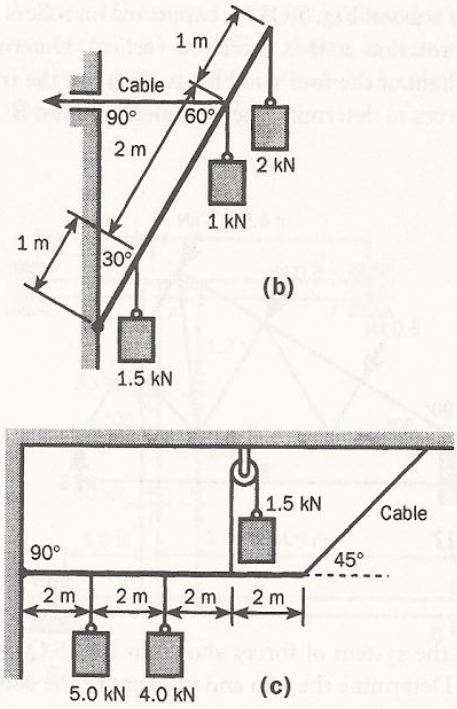


Fig. 5.Q15

16 The weights of the structures shown in Fig. 5.Q16(a), (b), (c) and (d) may be neglected. In each case the reaction at A is horizontal, and B is a hinge. Determine the position of the resultant of the vertical loads; then, by applying the principle of the triangle of forces, determine the reactions at A and B.

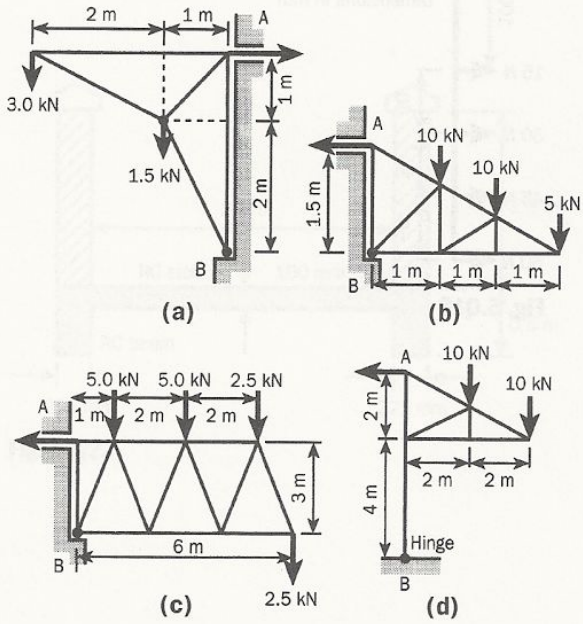


Fig. 5.Q16

17 The roof truss of Fig. 5.Q17 is supported on rollers at B and the reaction at B is therefore vertical. Determine the resultant of the four wind loads, then use the triangle of forces to determine the reactions at A and B.

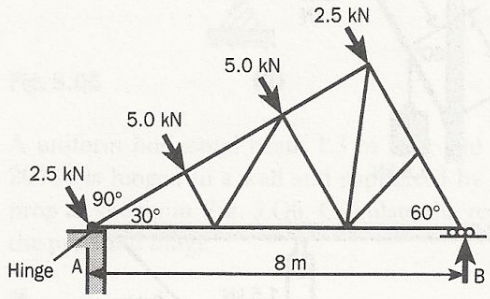


Fig. 5.Q17

18 Reduce the system of forces shown in Fig. 5.Q18 to a couple. Determine the arm and moment of the couple.

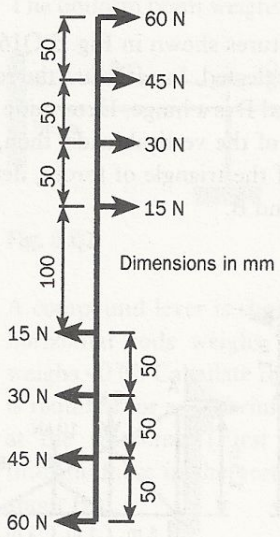


Fig. 5.Q18

19 Calculate the reactions at A and B due to the given point loads in Fig. 5.Q19(a) to (e). The loads are in kilonewtons (kN).

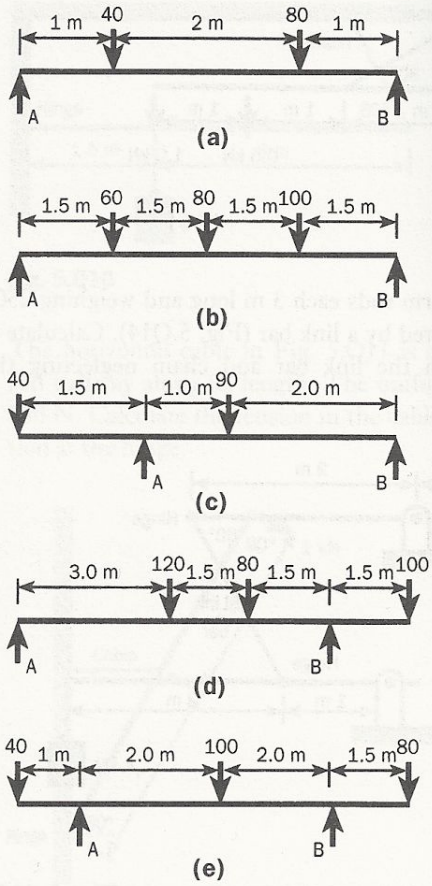


Fig. 5.Q19

20 Calculate the reactions at A and B due to the given system of loads in Fig. 5.Q20(a) to (g).

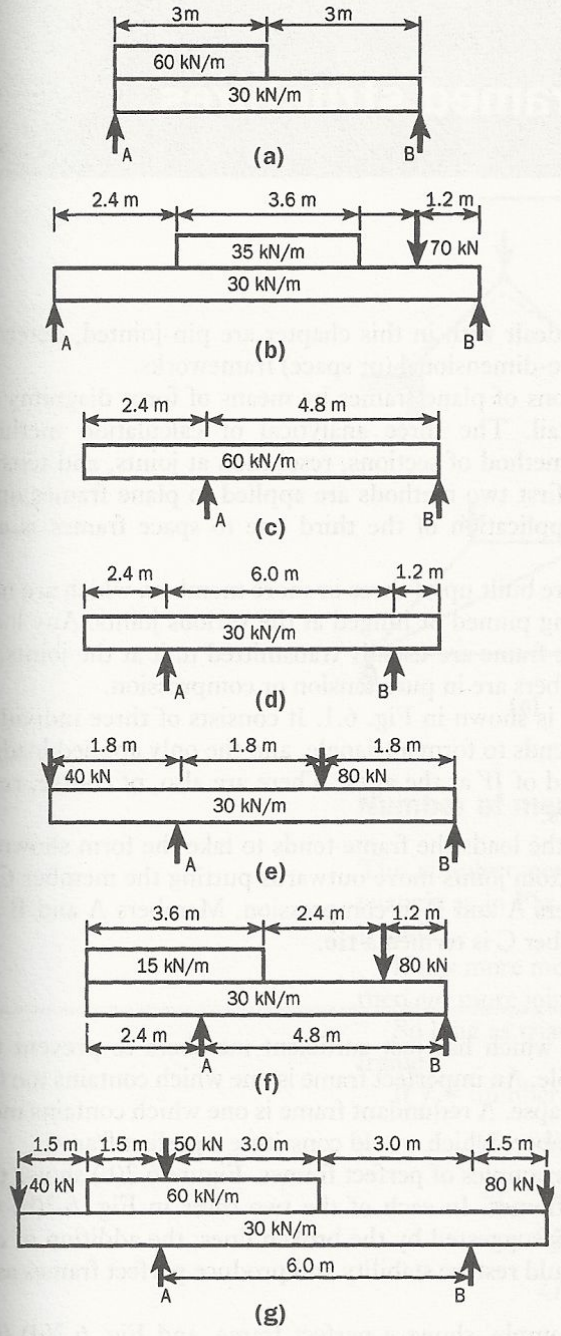


Fig. 5.Q20

21 A beam AB carries two point loads and is supported by two beams CD and EF as shown in plan in Fig. 5.Q21. Neglecting the weights of the beams, calculate the reactions at C, D, E and F.

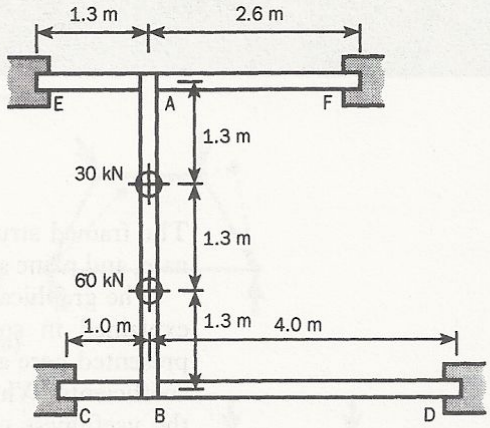


Fig. 5.Q21

22 A reinforced concrete bridge between two buildings spans 8 m. The cross-section of the bridge is shown in Fig. 5.Q22. The brickwork weighs 20 kN/m^3, the reinforced concrete weighs 23 kN/m^3, and the stone coping weighs 0.5 kN/m. The floor of the bridge has to carry a uniformly distributed load of 1.5 kN/m^2 in addition to its own weight. Calculate the force on the supporting buildings from one end of each beam.

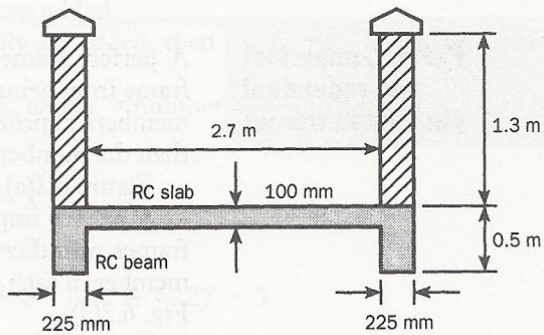


Fig. 5.Q22