Static Analysis of a Sleeper with Mathematica

André Heck

April 2, 1995

Abstract

Showing the power of Mathematica for the static analysis of a sleeper is the main goal of this paper. It is part of a case study for the Engineering Office of the Dutch Railways to investigate the applicability of computer algebra in their engineering practice.

1 Introduction

A classical problem in structural analysis is the study of beams in static equilibrium. The beam analysis is regarded as completed when the internal stresses and the displacements have been determined, i.e., once the shear force, bending moment, slope, and deflection functions have been computed. In this paper we show that Mathematica ([1]) is powerful and flexible enough to solve the symbolic, numerical, and graphical problems concerning beam deflections. The BeamStatics package of Levent Kitiş ([2]), already shows some of the possibilities. However, it does not allow studies of beams on foundations. In this paper we will study such beams. In particular, we will analyze a sleeper, which is modeled as a symmetrically loaded beam with partial elastic support.

2 Acknowledgements

Ing. T. Miedema and drs. G. Kouwijzer from the Engineering Office of the Dutch Railways raised the question of how good Mathematica is as a working environment for studies in beam analysis. The analysis of a sleeper is one of the case studies carried out to arrive at a positive answer to this question.

3 The Problem

We consider a sleeper in static equilibrium under the following conditions (see Figure 1 on the next page):

- the sleeper is a symmetrical system of loads and supports;
- there are only two concentrated loads;
- the support is elastic, but the sleeper is in the middle unsupported.

The task is to describe the bending of the sleeper under these conditions.

4 The Mathematical Model

The mathematical model for the sleeper comes from classical beam analysis. In this section we collect main results of the theory, which can be found in almost any book on beam analysis.

The beam coordinate system is shown in Figure 1, i.e., downward beam deflections are positive. The deflection of the beam is denoted by w. The distance x is measured along the axis of the beam from the origin at the left end of the beam to the right end of the beam at distance L. Figure 2 shows the sign convention for the bending moment M and the shear force Q.

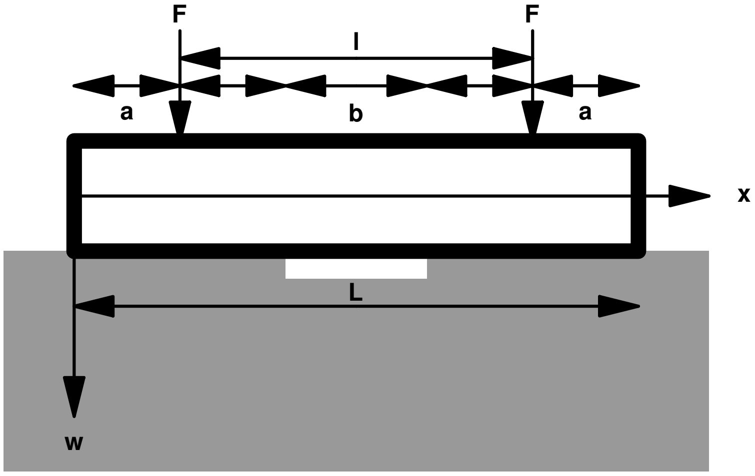

Figure 1: Partially Supported Sleeper

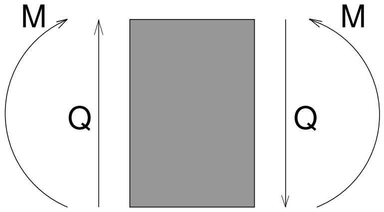

Figure 2: Sign Convention — Positive Shear and Bending Moment

Let H denote the slope of the beam and EI be the flexural rigidity of the beam, which is supposed to be constant along the beam axis.

The basis of the analysis are the following equations:

$$\begin{aligned} -\frac{\partial Q}{\partial x} &= \mathit{load} - \mathit{reaction\ force} && (1) \\ \frac{\partial M}{\partial x} &= Q(x) && (2) \\ \frac{\partial^2 w}{\partial x^2} &= -\frac{M(x)}{EI} && (3) \end{aligned}$$

From these quations follows

$$EI\frac{\partial^4 w}{\partial x^2} = \mathit{load} - \mathit{reaction\ force} \tag{4}$$

The applied load is conveniently expressed in terms of

- the unit step function

$$\texttt{UnitStep}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

- the Dirac delta function denoted by `DiracDelta` in Mathematica

The contribution of a concentrated load of magnitude F acting at $x = a$ to the load is equal to the product $F \cdot \texttt{DiracDelta}(x - a)$.

The action of an elastic foundation may be taken into account by introducing a restoring force $k \cdot w(x)$ acting in the upward direction. The constant k is called the modulus of the foundation. We can then rewrite Equation 4 as

$$EI \frac{\partial^4 w}{\partial x^4} + k\, w(x) = load \tag{5}$$

If we now introduce β via

$$4\beta^4 = \frac{k}{EI}$$

the equation 5 is transformed to

$$\frac{\partial^4 w}{\partial x^4} + 4\beta^4\, w(x) = \frac{load}{\mathrm{EI}} \tag{6}$$

As we will consider a beam with free ends, the boundary conditions are that the moment and shear force the ends of the beam vanish, i.e.,

$$M(0) = 0, \quad Q(0) = 0, \quad M(L) = 0, \text{ and } Q(L) = 0 \tag{7}$$

5 The Computer Algebra

We consider two cases:

1. the elastically supported beam with two concentrated loads F symmetrically placed at distance l of each other (see Figure 3);
2. the partially supported beam with concentrated loads as shown in Figure 1.

5.1 Beam on Elastic Foundation

We can use Mathematica to sketch the situation:

```
<<Graphics`Arrow`
beam = { Thickness[0.02],
         Line[{ {-4,-1}, {4,-1}, {4,1}, {-4,1}, {-4,-1} }] };
foundation = { GrayLevel[0.8], Rectangle[ {-5,-5}, {5,-1}] };
xArrow = Arrow[{-4,0}, {5, 0}];
wArrow = Arrow[{-4,0}, {-4,-4}];
leftPointLoad  = Arrow[{-2.5,2.5}, {-2.5,1}];
rightPointLoad = Arrow[{ 2.5,2.5}, { 2.5,1}];
LArrow = { Arrow[{0,-2}, {4, -2}], Arrow[{0,-2}, {-4,-2}] };
lArrow = { Arrow[{0, 2}, {2.5,2}], Arrow[{0, 2},{-2.5,2}] };
aArrows = {
  Arrow[{ 2.5,2}, { 4,2}], Arrow[{ 4,2}, { 2.5,2}],
  Arrow[{-2.5,2}, {-4,2}], Arrow[{-4,2}, {-2.5,2}]
        };
text = {
  Text[ FontForm["x",{"Helvetica-Bold",14}], {5.5,0} ],
  Text[ FontForm["w",{"Helvetica-Bold",14}], {-4,-4.5} ],
  Text[ FontForm["L",{"Helvetica-Bold",14}], {0,-1.75} ],
  Text[ FontForm["l",{"Helvetica-Bold",14}], {0,1.7} ],
```

```
  Text[ FontForm["a",{"Helvetica-Bold",14}], {3.25,1.7} ],
  Text[ FontForm["a",{"Helvetica-Bold",14}], {-3.25,1.7} ],
  Text[ FontForm["F",{"Helvetica-Bold",14}], {-2.5,2.8} ],
  Text[ FontForm["F",{"Helvetica-Bold",14}], { 2.5,2.8} ]
      };
Show[Graphics[ { foundation, beam, xArrow, wArrow, leftPointLoad,
  rightPointLoad, LArrow, lArrow, aArrows, text },
  PlotRange -> All ]];
```

It may look overwhelming, but it is merely a specification of the graphics objects and it will produce Figure 3.

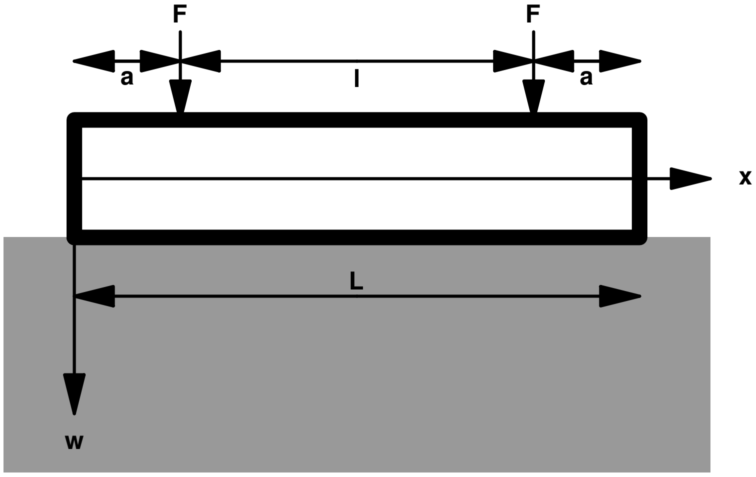

Figure 3: Beam on Elastic Foundation

By the way, this is also how Figure 1 was created.

The differential equation 6 is entered in Mathematica as

```
4 beta^4 w[x] + D[w[x],{x,4}] == load / EI
```

and displays as

```
      4         (4)       load
4 beta  w[x] + w   [x] == ----
                           EI
```

The load is in this case

```
<<Calculus`DiracDelta`
load = F DiracDelta[x-a] +   F DiracDelta[x-a-l];
```

The Mathematica command `DSolve` is powerful enough to solve the differential equation. the result is after some simplification

```
d1 Cos[beta x]       beta x                d2 Sin[beta x]
-------------- + d3 E       Cos[beta x] + -------------- +
    beta x                                     beta x
   E                                          E


      beta x
  d4 E       Sin[beta x] +
```

```
    beta (a - x)
  (E             F (Cos[beta (-a + x)] -

        2 beta (-a + x)
       E                Cos[beta (-a + x)] +

       Sin[beta (-a + x)] +

        2 beta (-a + x)
       E                Sin[beta (-a + x)]) UnitStep[-a + x]

                3
     ) / (8 beta  EI) + (F

     (-Cos[beta (a + l - x)] +

        2 beta (a + l - x)
       E                   Cos[beta (a + l - x)] -

       Sin[beta (a + l - x)] -

        2 beta (a + l - x)
       E                   Sin[beta (a + l - x)])

                                    3  beta (a + l - x)
     UnitStep[-a - l + x]) / (8 beta  E                 EI)
```

We can then define the deflection, slope, bending moment, and shear force functions.

```
W[x_] = %;
H[x_] = D[ W[x], x ];
M[x_] = -EI D[ H[x], x ];
Q[x_] = D[ M[x], x ];
```

The ends of the beam are free so that the bending moment and shear force are at these points equal to zero. This gives us 4 equations in the four unknowns `d1`, `d2`, `d3`, and `d4`, which Mathematica can solve. Even after simplification the results are large: typically formulae that fill one A4 page.

Instead of a we may also use in the formulae the indeterminates l and L via the relation $a = \frac{(L-l)}{2}$. Furthermore, if we take concrete values for the unknowns, says $L = 2600$ mm, $l = 1500$ mm, $EI = 6.33\ 10^{11}\ \mathrm{N\,mm}^2$, $F = 56.25\ 10^3$ N, and $k = 25$ N/mm, then the general solution specializes to

```
{d1 -> 0.537024, d2 -> 0.216867, d3 -> 0.970758,

 d4 -> 0.216867}
```

We define a new plot routine to plot the graphs of the beam functions in our downward oriented coordinate system. This shows a bit of Mathematica's programming facilities.

```
myPlot[ function_, domain_, options___ ] :=
Module[ {picture, range, markers},
picture = Plot[ function, domain, options,
  DisplayFunction -> Identity ];
range = FullOptions[ picture, PlotRange ];
```

```
range = { range[[1]], { - range[[2,2]], - range[[2,1]] } };
picture = Plot[ -function, domain, PlotRange -> range, options,
  DisplayFunction -> Identity ];
markers = FullOptions[ picture, Ticks ];
markers = {
  Partition[Flatten[
    Map[ ( If[ NumberQ[#[[2]]], {#[[1]],#[[2]]},{} ] )&,
    markers[[1]] ] ], {2} ],
  Partition[Flatten[
    Map[ ( If[ NumberQ[#[[2]]], {#[[1]],-#[[2]]},{} ] )&,
    markers[[2]] ] ], {2} ]
          };
Show[ picture,  Ticks -> markers, PlotRange -> range, options,
  DisplayFunction -> $DisplayFunction   ]
];
```

So, this procedure is used to graph the deflection, slope, bending moment, and shear force of the beam.

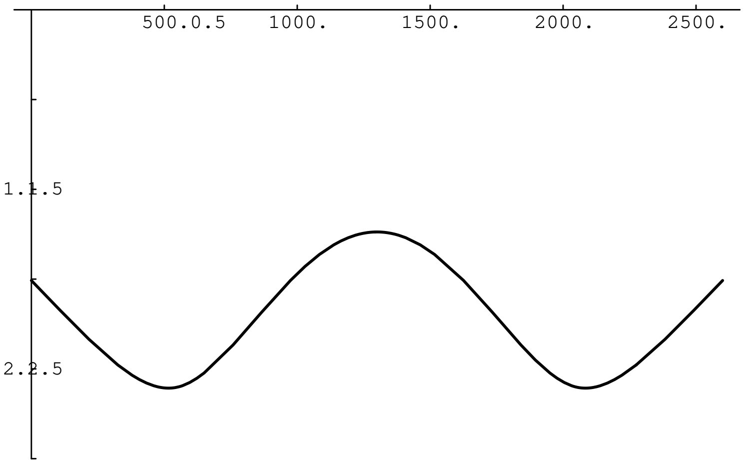

Figure 4: Deflection of Supported Beam

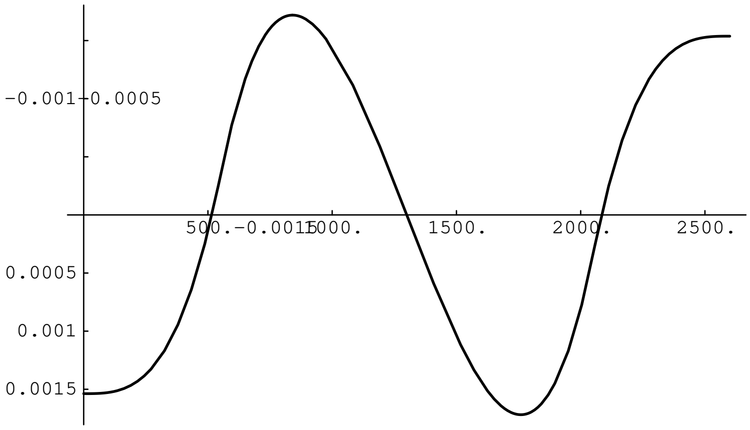

Figure 5: Slope of Supported Beam

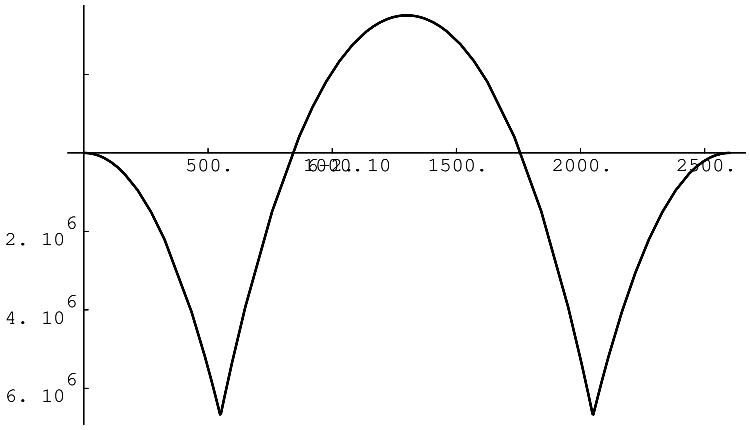

Figure 6: Bending Moment of Supported Beam

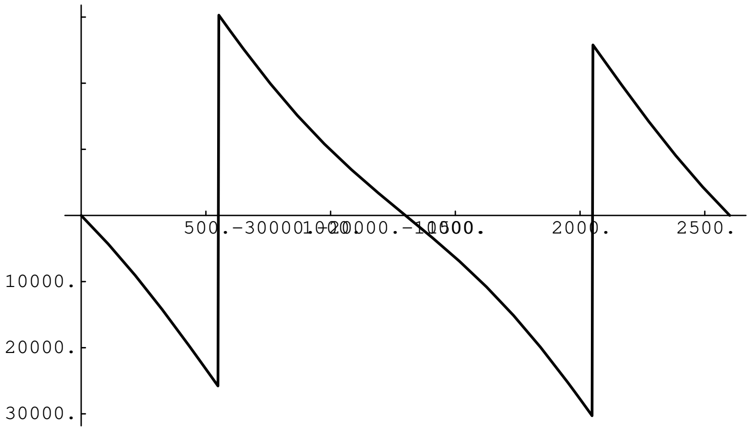

Figure 7: Shear Force of Supported Beam

Notice that in the solution method we have nowhere used the symmetry of the problem. Therefore, the symmetry shown in the graphs supports the correctness of the solution.

Having solved the general problem with indeterminates instead of concrete values allows us to analyze the dependency of the solution on certain parameters. For example, we could look at the dependency of the deflection on the stiffness of the foundation, i.e., on the parameter β. We could make a surface plot or have an animation showing this dependency. On the next page is a picture consisting of four graphs from an animation of the bending moment in which β ranges from 0.0015 to 0.0035 $\mathrm{N/mm^3}$. Going from left to right, and from top to bottom, you see that when the foundation becomes more stiff, the bending moment reduces and gets flatter in the middle.

5.2 Partially Supported Beam

We come back to our original problem of a partially supported beam as shown in Figure 1. As we will see that gap in the middle of the foundation is designed such that the stress in the middle of the beam is reduced. In this case, the modulus k of the foundation is given in Mathematica as

```
4 EI beta^4 ( UnitStep[a + (l-b)/2 - x] + UnitStep[x - a - (l+b)/2 ] )
```

and the differential equation becomes

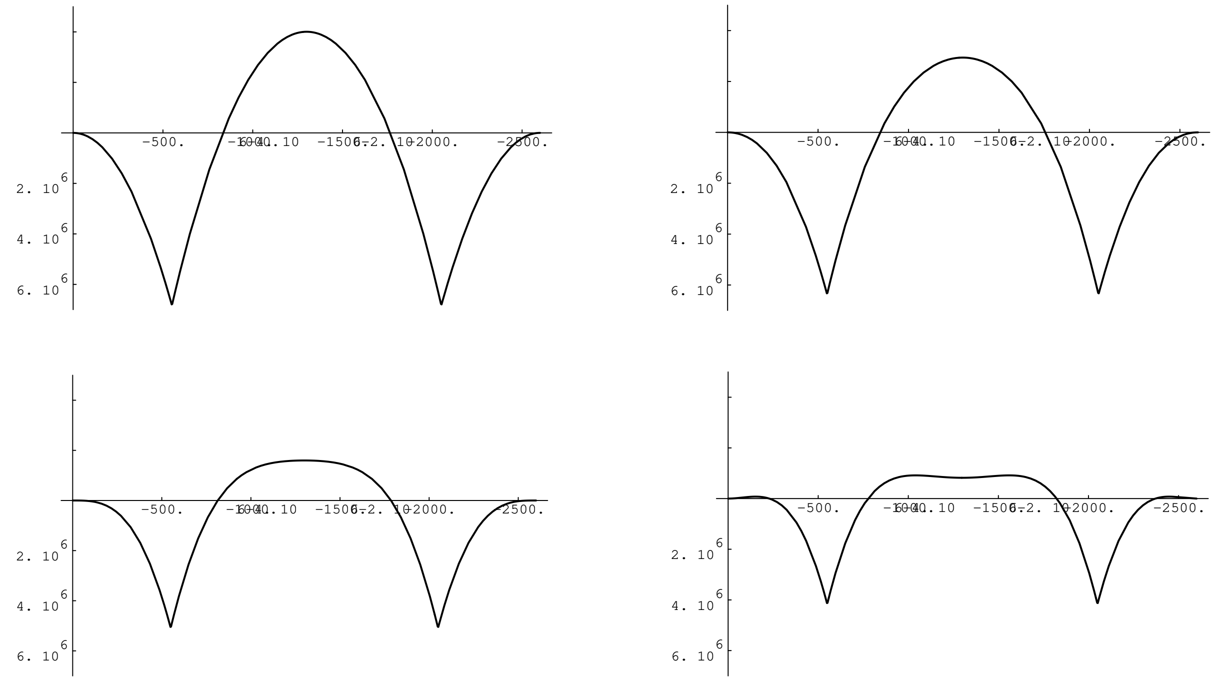

Figure 8: Bending Moments for Different β

```
      4               b   l
4 beta  UnitStep[a - - + - - x] w[x] +
                      2   2

        4               b   l              (4)
   4 beta  UnitStep[-a - - - - + x] w[x] + w    [x] ==
                        2   2

  F (DiracDelta[-a + x] + DiracDelta[-a - l + x])
  -----------------------------------------------
                        EI
```

Mathematica will not be able to solve all by itself the differential equation with this modulus of the foundation. But this does not mean that we have to give up. Actually, we can split the horizontal range into three regions for which we already know the solution:

1. $0 \le x \le a + \frac{l-b}{2}$
2. $a + \frac{l-b}{2} \le x \le a + \frac{l+b}{2}$
3. $a + \frac{l+b}{2} \le x \le L$

In the first and third region we are just considering a supported beam. For this case we already found the general solution in the previous section. We apply the replacement rule

```
{UnitStep[x-a-l]->0, d1->d11, d2->d12, d3->d13, d4->d14}
```

in the general solution to obtain the following deflection $w_1(x)$ in the left region:

```
d11 Cos[beta x]        beta x
--------------- + d13 E       Cos[beta x] +
     beta x
    E

  d12 Sin[beta x]        beta x
  --------------- + d14 E       Sin[beta x] +
       beta x
      E

    beta (a - x)
  (E             F (Cos[beta (-a + x)] -

        2 beta (-a + x)
       E                Cos[beta (-a + x)] +

       Sin[beta (-a + x)] +

        2 beta (-a + x)
       E                Sin[beta (-a + x)]) UnitStep[-a + x]

                3
     ) / (8 beta  EI)
```

To obtain the deflection function $w_3(x)$ in the right region we just apply the replacement rule

```
{UnitStep[x-a-l]->1, d1->d31, d2->d32, d3->d33, d4->d34}
```

to the general solution of a beam on an elastic support.

The midsection is the special case of an unsupported beam. In this case the differential equation is

```
 (4)       F (DiracDelta[-a + x] + DiracDelta[-a - l + x])
w   [x] == -----------------------------------------------
                                 EI
```

and can be easily solved by Mathematica; the result is

```
{{w[x] -> C[1] + x C[2] + x  C[3] + x  C[4] +

     3       2            2      3
  -(a  F)   a  F x   a F x    F x
 (------- + ------ - ------ + ----) UnitStep[-a + x] +
   6 EI      2 EI     2 EI    6 EI

             3             2                 2      3
  -(F (a + l) )   F (a + l)  x   F (-a - l) x    F x
 (------------- + ------------ + ------------- + ----)
      6 EI            2 EI           2 EI        6 EI
```

Applying the replacement rule

```
{UnitStep[x-a]->1, UnitStep[x-a-l]->0,
 C[1]->d21,C[2]->d22, C[3]->d23, C[4]->d24}
```

we obtain the following deflection function $w_2(x)$ for the middle region.

```
                                                   3
                      2         3   F (-a + x)
d21 + d22 x + d23 x   + d24 x   + -----------
                                     6 EI
```

By differentiation we can also compute the slopes, bending moments, and shear forces in the threwe regions. We are left with twelve unknowns d_{ij}, where $i = 1, 2, 3$ and $j = 1, 2, 3, 4$ in the solutions. We can calculate these unknowns from the four boundary conditions of free beam ends and the eight conditions that come from matching the different deflection functions and higher derivatives as continuous functions. The commands are shown below, but we have omitted the lengthy list of simplification rules. These rules are of the type `DiracDelta[a] -> 0`, and `UnitStep[-a] -> 0` to let Mathematica know that we understand a to be a positive real number.

```
eq[1] = Simplify[ m1[0]==0 /. simplificationrules ];

eq[2] = Simplify[ m3[L]==0 /. simplificationrules ];

eq[3] = Simplify[ q1[0] == 0 /. simplificationrules ];

eq[4] = Simplify[ q3[L] == 0 /. simplificationrules ];

eq[5] = Simplify[ w1[a+(l-b)/2] == w2[a+(l-b)/2] /.
    simplificationrules ];

eq[6] =  Simplify[ h1[a+(l-b)/2] == h2[a+(l-b)/2] /.
    simplificationrules  ]

eq[7] =  Simplify[ m1[a+(l-b)/2] == m2[a+(l-b)/2] /.
    simplificationrules ];

eq[8] =  Simplify[ ( q1[a+(l-b)/2] == q2[a+(l-b)/2] ) /.
    simplificationrules ];

eq[9] = Simplify[ ( w2[a+l/2+b/2] == w3[a+l/2+b/2] ) /.
    simplificationrules ];

eq[10] = Simplify[ ( h2[a+l/2+b/2] == h3[a+l/2+b/2] ) /.
    simplificationrules  ];

eq[11] = Simplify[ ( m2[a+l/2+b/2] == m3[a+l/2+b/2] ) /.
    simplificationrules ];

eq[12] = Simplify[ q2[a+l/2+b/2] == q3[a+l/2+b/2] /.
    simplificationrules ];
```

If we take $b = 600$ mm and the same data as in the previous section, Mathematica can solve the twelve equations in the d_{ij}'s. The deflection and the bending moment are shown below, together with the result without the gap. As you can clearly see from these figures, the stress is reduced in the middle section if the gap in the foundation is introduced.

In the design phase of course we don't know a "good" value for the gap width b. But it is rather easy to graphically find such a value: solve numerically the problem for several values of b, plot the function, and in an animation step through the graphs to find the best choice for b. This is what we have done in the following Mathematica code:

```
equations = Table[eq[i],{i,12}];
```

```
L=2600; l=1500; EI=6.33 10^(11); F=56.25 10^3; k=25;
beta=(k/(4 EI))^(1/4); a=(L-l)/2;

Do[ (
solutions = NSolve[ equations,
{d11,d12,d13,d14,d21,d22,d23,d24,d31,d32,d33,d34} ][[1]];
p1 = a+(l-b)/2; p2 = a+(l+b)/2;
W1 = w1[x] /. solutions;
W2 = w2[x] /. solutions;
W3 = w3[x] /. solutions;
Clear[W]; Clear[M];
W[x_ /; Evaluate[x<=p1] ] = W1;
W[x_ /; Evaluate[x>p1 && x<p2] ] = W2;
W[x_ /; Evaluated[x>=p2] ] = W3;
M1 = m1[x] /. solutions;
M2 = m2[x] /. solutions;
M3 = m3[x] /. solutions;
M[x_ /; Evaluate[x<=p1] ] = M1;
M[x_ /; Evaluate[x>p1 && x<p2] ] = M2;
M[x_ /; Evaluate[x>=p2] ] = M3;
plot[b] = myPlot[ W[x], {x,0,L}, PlotRange->{0,5.5},
  DisplayFunction->Identity];
mplot[b] = myPlot[ M[x], {x,0,L}, PlotRange -> {-5 10^6,7 10^6},
  DisplayFunction->Identity]), {b,0,1000,100} ];
```

Now, the plots of the deflection function and bending moment can be easily drawn.

```
Show[ {plot[0],plot[600]}, DisplayFunction->$DisplayFunction ];
Show[ {mplot[0],mplot[600]}, DisplayFunction->$DisplayFunction ];
```

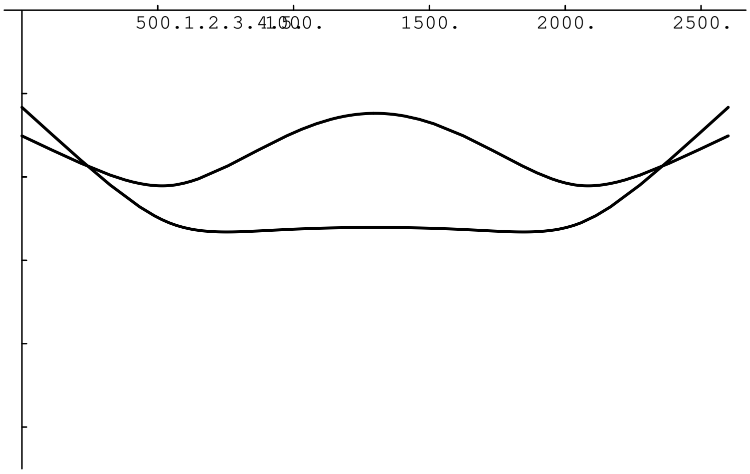

Figure 9: Deflection for $b = 0$ and $b = 600$

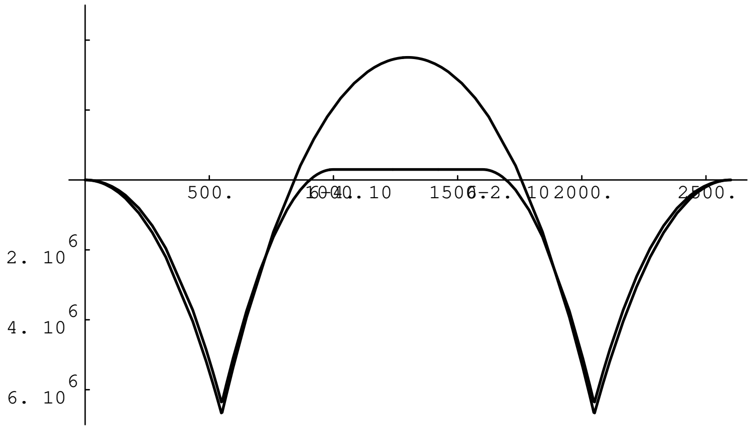

Figure 10: Bending Moment for $b = 0$ and $b = 600$

6 Conclusion

From the example of a sleeper it is clear that Mathematica is a convenient working environment in beam analysis. It speeds up calculations, allows you to keep parameters in problems and solutions, enables graphical investigation, to name a few of the features. In short, it enriches the set of tools that an engineer can use in practice.

7 References

[1] S. Wolfram, em Mathematica: A System for Doing Mathematics by Computer, 2nd edition, Addison-Wesley, 1991.

[2] Levent Kitiş, *A Mathematica Package for the State Analysis of Straight Euler-Bernoulli Beams*, Mathsource, Wolfram Research, URL: http://www.wri.com/