MATH 5485 – Introduction to Numerical Methods I

Fall 2013

Marcel Arndt
(modified by Duane Nykamp and Alexander Shapeev)

Lab Project: Bridge Construction

In this lab project, we investigate the static behavior of bridges built from steel trusses like the one shown in Figure 1. When some loading is applied to the bridge, for example due to the weight of cars going over it, the forces get distributed among all trusses. However, not all trusses will exhibit the same force, rather some will carry more load than others, depending on the design of the bridge.

Heavily loaded trusses are more likely to fail than less loaded trusses. Hence it is of crucial importance for bridge engineers to compute theses forces, identify possible points of failure, and reinforce the structure where necessary to prevent catastrophes. At the same time, one wants the bridge to be as lightweight and efficient as possible to keep material costs low. Numerical simulation allows this to be achieved without performing extensive experiments with real bridges and therefore considerably reduces construction time

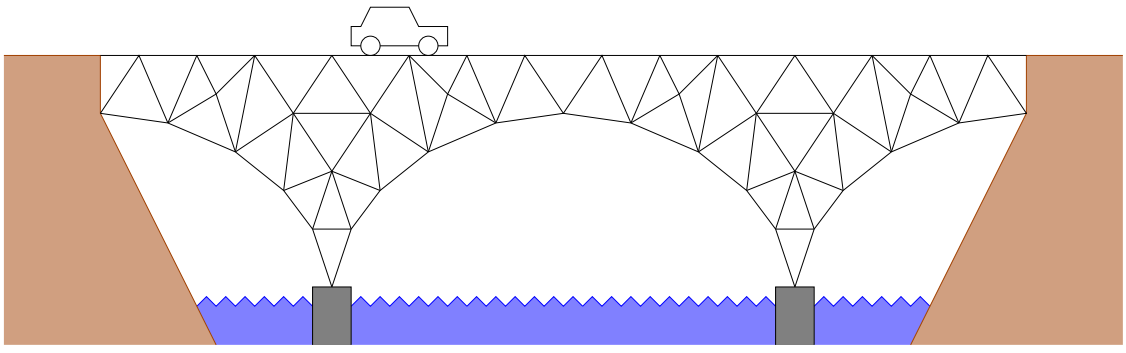

Figure 1: A truss bridge with two foundations over a river valley.

and cost.

In this lab project, you will:

- Learn about the mathematical modeling of the static behavior of truss bridges.
- Learn how to reformulate this model to make it accessible by computers.
- Learn how to implement and numerically solve the problem for a given bridge.
- Build your own bridge which provides the best load distribution and thus is safest!

This text will guide you through the whole process and provide a step-by-step instruction. Work carefully through the text to understand all aspects of modeling and implementation. It is important to follow all details, especially the sometimes subtle indices, otherwise you will run into trouble when it comes to the actual computation later.

You will encounter several tasks in the project. Your job is to perform these tasks. This will finally lead you to a complete solution of the bridge construction project. It is a good idea to work through the entire text first before you start working on the tasks to get an impression of the big picture.

You will need to submit a **term paper** by **Wednesday, October 23, 2013**. This term paper shall include the solutions to all tasks, including the programs you wrote and graphs of your results. Your term paper will be graded and counts as the first midterm. Additionally, submit the input files of the bridge you constructed as will be instructed later. There will be a competition among students: The stability of all bridges will be ranked, and the most stable ones get extra points!

1 Description of the Bridge

The bridges we investigate here are mechanical structures which are built from steel trusses. A truss is a thin straight rod. The trusses are joined together at their endpoints. The junction points are called the nodes. Figure 2 shows a simple mechanical structure with 5 nodes and 7 trusses.

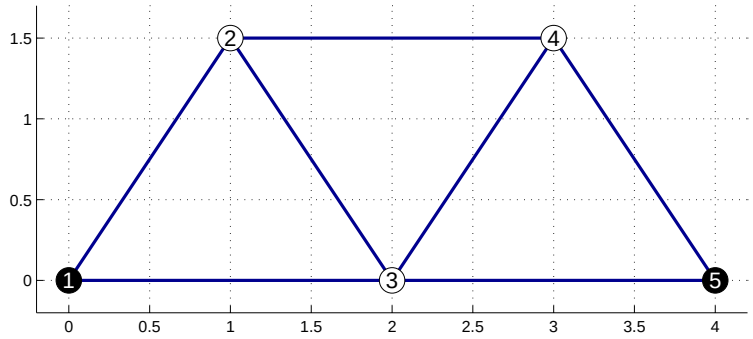

Figure 2: A simple mechanical structure with 5 nodes and 7 trusses. Nodes 1 and 5 are fixed to the ground.

Before we can discuss loading and forces, we need to describe the structure mathematically. First, we define the nodes. Assume our bridge consists of N nodes, numbered $1, 2, \ldots, N$. We denote the position of the i-th node by

$$\vec{x}_i = (x_{i,1}\ x_{i,2}) \in \mathbb{R}^2. \tag{1}$$

(Note that we carry out everything in two-dimensional space here to keep things simple, hence $\vec{x}_i \in \mathbb{R}^2$. However, the generalization to the three-dimensional setting would be straightforward.)

A few nodes rest on a foundation or are fixed to the shoreline to keep the bridge in place and carry the load. We denote these nodes as *fixed*, and the remaining ones as *free*. Thus we need to keep track of the type of the node as well. Let M denote the number of free nodes. As N was the total number of nodes, we have $M \leq N$.

The structure of the bridge from Figure 1 is shown in Figure 3. Here fixed nodes are black, and free nodes are white. For this bridge, we have $N = 47$ and $M = 41$.

Then, there are K trusses, each of which connects two nodes. For each truss, we need to specify these two nodes. Instead of storing the coordinates again (which we already have in form of the vectors $\vec{x}_i$), it is better to store the node numbers, ranging from 1 to N. Hence for $k = 1, 2, \ldots, K$ we introduce two variables

$$t_{k,1}, t_{k,2} \in \{1, 2, \ldots, N\}, \tag{2}$$

meaning that the k-th truss connects nodes $t_{k,1}$ and $t_{k,2}$.

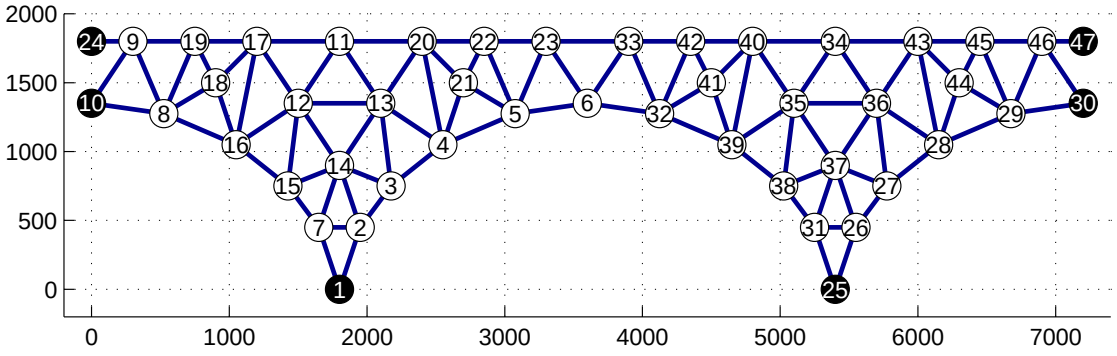

Figure 3: Structure of the bridge from Figure 1. The black nodes are fixed, and the white nodes are free.

Nodes			
i	$x_{i,1}$	$x_{i,2}$	type
1	0.0	0.0	fixed
2	1.0	1.5	free
3	2.0	0.0	free
4	3.0	1.5	free
5	4.0	0.0	fixed

Trusses		
k	$t_{k,1}$	$t_{k,2}$
1	1	2
2	1	3
3	2	3
4	2	4
5	3	4
6	3	5
7	4	5

$$M = 3$$
$$N = 5$$
$$K = 7$$

Table 1: Mathematical description of the simple structure shown in Figure 2.

These variables fully specify the structure. For example, the simple structure displayed in Figure 2 can be described as shown in Table 1.

Next, when some loading is applied to the bridge, the nodes move a certain amount from their original position, except for the fixed nodes which are tied to the ground. For real structures such as bridges, the nodes do not move much since steel constructions are pretty stiff, but they indeed do move a little bit. We denote the displacement of node i by

$$\vec{u}_i = (u_{i,1}\ u_{i,2}) \in \mathbb{R}^2, \tag{3}$$

that is, node i moves from $\vec{x}_i$ to $\vec{x}_i + \vec{u}_i$ under loading. Since the fixed nodes cannot move, we have

$$\vec{u}_i = 0 \qquad \text{whenever node } i \text{ is fixed.} \tag{4}$$

The node positions $\vec{x}_i$ without loading are quantities given by the bridge, whereas the displacements $\vec{u}_i$ are the main unknowns to be computed at the

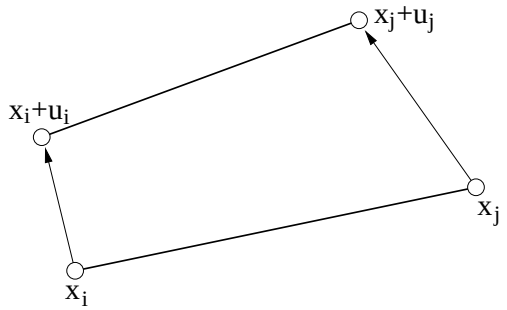

Figure 4: Deformation of nodes i and j from original positions $\vec{x}_i$ and $\vec{x}_j$ to $\vec{x}_i + \vec{u}_i$ and $\vec{x}_j + \vec{u}_j$.

end of the day. Because of (4), we only need to consider the displacements $\vec{u}_i$ of the free nodes i.

2 Mathematical Modeling

Now, lets come to the mathematical modeling. First, we relate the displacement $\vec{u}_i$ of all nodes i to the elongation e_k of truss k. The original length l_k of truss k is

$$l_k := \|\vec{x}_{t_{k,2}} - \vec{x}_{t_{k,1}}\| = \sqrt{(x_{t_{k,2},1} - x_{t_{k,1},1})^2 + (x_{t_{k,2},2} - x_{t_{k,1},2})^2}. \tag{5}$$

Under loading, this length l_k changes to

$$\|(\vec{x}_{t_{k,2}} + \vec{u}_{t_{k,2}}) - (\vec{x}_{t_{k,1}} + \vec{u}_{t_{k,1}})\|. \tag{6}$$

The elongation of truss k is therefore given by

$$\|(\vec{x}_{t_{k,2}} + \vec{u}_{t_{k,2}}) - (\vec{x}_{t_{k,1}} + \vec{u}_{t_{k,1}})\| - \|\vec{x}_{t_{k,2}} - \vec{x}_{t_{k,1}}\|, \tag{7}$$

see Figure 4 for an illustration.

Formula (7) turns out to be too complicated for the subsequent calculations because it leads to nonlinear equations. However for small deformations, we have approximately

$$\|(\vec{x}_{t_{k,2}} + \vec{u}_{t_{k,2}}) - (\vec{x}_{t_{k,1}} + \vec{u}_{t_{k,1}})\| \approx \|\vec{x}_{t_{k,2}} - \vec{x}_{t_{k,1}}\| + \vec{n}_k \cdot (\vec{u}_{t_{k,2}} - \vec{u}_{t_{k,1}}) \tag{8}$$

where

$$\vec{n}_k := \frac{\vec{x}_{t_{k,2}} - \vec{x}_{t_{k,1}}}{\|\vec{x}_{t_{k,2}} - \vec{x}_{t_{k,1}}\|} \tag{9}$$

is the normal vector pointing from node $t_{k,1}$ to node $t_{k,2}$. Thus we define the (approximate) elongation of truss k as

$$\boxed{e_k := \vec{n}_k \cdot (\vec{u}_{t_{k,2}} - \vec{u}_{t_{k,1}}).} \tag{10}$$

The elongation e_k is positive if the truss k is stretched and negative if it is compressed. Equation (10) relates the elongation to the displacement. Note that this relationship is *linear*.

Second, we relate the elongations to forces. This is called the *constitutive relation*. Here we assume that the trusses act like Hookean springs, that is, the force f_k on truss k is proportional to its relative elongation e_k/l_k:

$$\boxed{f_k := \frac{c_k}{l_k} e_k.} \tag{11}$$

This is a reasonable assumption for steel trusses if the displacement is small. The constant $c_k > 0$ is called the *elastic modulus* and describes the stiffness of the truss. Large values of c_k correspond to hard and thick material where you need a strong force to effect a certain elongation, whereas small values of c_k correspond to soft and thin material where already small forces suffice to effect a large elongation. In our application, the trusses all consist of the same material, steel, but we allow for different truss thicknesses. Therefore c_k varies with the thickness of the truss.

Third, we relate the forces to each other. According to Newton's First Law, the bridge is in a stable state if the sum of all forces acting on each free node i vanishes. This is called the *force balance*. In our application, there are two types of forces: internal forces caused by the tension of the trusses, and external forces caused by external loading: vehicles going over the bridge, wind forces, gravitational forces due the weight of the bridge itself and so on. The internal forces are the f_k from above, whereas the external loading is given by the vectors $\vec{f}_i^{ext} = (f_{i,1}^{ext}\ f_{i,2}^{ext}) \in \mathbb{R}^2$ for $i = 1, 2, \ldots, N$.

The directional force exerted by truss k on node i is

$$-f_k\vec{n}_k \quad \text{if } i = t_{k,1} \qquad \text{and} \qquad f_k\vec{n}_k \quad \text{if } i = t_{k,2}, \tag{12}$$

taking into account the direction of the normal vector $\vec{n}_k$. The force balance

at the free node i is thus given by

$$\boxed{\sum_{k:\, t_{k,2}=i} f_k \vec{n}_k - \sum_{k:\, t_{k,1}=i} f_k \vec{n}_k - f_i^{ext} = 0.} \tag{13}$$

Note that the equations (13) are linear conditions on the f_k. At a fixed node, the ground is supposed to absorb all forces, so there is no force balance there.

3 Matrix Notation

Equations (10), (11), and (13) together give linear conditions on the unknown displacements $\vec{u}_i$ of the free nodes. In this section, we will figure out how to translate this linear system into matrix notation.

The unknowns we need to solve for are the 2-vectors $\vec{u}_i$ of displacements for all free nodes i, whereas the fixed nodes are left out due to (4). To enumerate the free nodes, we introduce variables m_i which map overall node numbers to free node numbers. This means we assign a unique free node number m_i with $1 \leq m_i \leq M$ to each free node i. (Recall that M denotes the number of free nodes.) Moreover, we let $m_i = 0$ whenever node i is a fixed node. Thus the mapping not only enumerates the free nodes, but also keeps track of whether a node is free or fixed. This way, we avoid introducing another variable to store the type of the node.

For example, the mapping for the structure given in Figure 2 and Table 1 can be defined as follows:

i	m_i
1	0
2	1
3	2
4	3
5	0

Our unknowns are the 2-vectors $\vec{u}_i = (u_{i,1}\, u_{i,2})$ with $m_i \neq 0$. We now subsume these M unknown 2-vectors to the large common vector $\vec{u} = (u_1 u_2 \dots u_{2M}) \in \mathbb{R}^{2M}$ by means of

$$u_{2m_i-1} = u_{i,1}, \qquad u_{2m_i} = u_{i,2} \qquad \text{for all } i = 1, 2, \dots, N \text{ with } m_i \neq 0.$$

Similarly, we subsume the external forces $\vec{f}_i^{ext}$ to the large common vector $\vec{f}^{ext} = (f_1^{ext}\ f_2^{ext}\ \dots\ f_{2M}^{ext}) \in \mathbb{R}^{2M}$:

$$f_{2m_i-1}^{ext} = f_{i,1}^{ext}, \qquad f_{2m_i}^{ext} = f_{i,2}^{ext} \qquad \text{for all } i = 1, 2, \dots, N \text{ with } m_i \neq 0.$$

Then, we subsume all elongations e_k and forces f_k to large common vectors

$$\vec{e} = (e_1\ e_2\ e_3\ \cdots\ e_K) \in \mathbb{R}^K \qquad \text{and} \qquad \vec{f} = (f_1\ f_2\ f_3\ \cdots\ f_K) \in \mathbb{R}^K. \quad (14)$$

Since relationship (10) between the displacements $\vec{u}_i$ and the elongations e_k is linear, there must be a matrix $B \in \mathbb{R}^{K\times 2M}$ such that

$$\vec{e} = B\vec{u}. \quad (15)$$

The matrix B is called the *incidence matrix*.

For your project, it will be important to figure out the precise definition of the incidence matrix B and the normal vectors $\vec{n}_k$. You might find it useful to do this in a pseudo-code style like

```
B = zero-matrix of size K x 2M
for k = 1...K
    n(k,1) = <something>
    n(k,2) = <something>
    add <something> to B(<index1>,<index2>)
    ...
end
```

Be aware that it is extremely important to get the indices completely right, otherwise your implementation will produce garbage later!

The constitutive relation (11) can obviously be written as

$$\vec{f} = C\vec{e} \quad (16)$$

with the diagonal matrix

$$C = \begin{pmatrix} c_1/l_1 & & & \\ & c_2/l_2 & & \\ & & \ddots & \\ & & & c_K/l_K \end{pmatrix} \in \mathbb{R}^{K\times K}. \quad (17)$$

Then, let us come to the force balance.

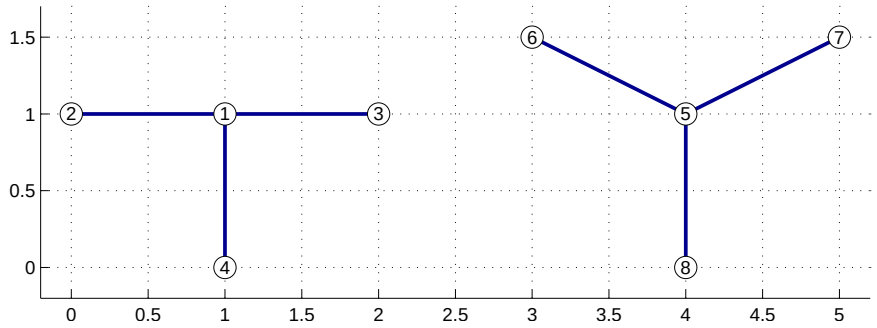

Figure 5: Malformed Nodes. Node 1 is malformed since it has only three trusses (1–2, 1–3, and 1–4), two of which are parallel (1–2 and 1–3). Node 5 is well-formed.

Task 1 *Show that the force balance* (13) *can be written as*

$$B^T \vec{f} = \vec{f}^{ext}, \tag{18}$$

that is the matrix related to the force balance is the transpose of the incidence matrix.

Finally, we put it all together. Recall that we have

$$\begin{aligned} B\vec{u} &= \vec{e}, \\ C\vec{e} &= \vec{f}, \\ B^T \vec{f} &= \vec{f}^{ext}, \end{aligned} \tag{19}$$

so we obtain

$$A\vec{u} = \vec{f}^{ext} \qquad \text{where} \qquad A = B^T C B. \tag{20}$$

This is the linear equation we need to solve!

It turns out that the matrix $A \in \mathbb{R}^{2M \times 2M}$ is regular for most structures. Hence for any external loading $\vec{f}^{ext}$, there exists a unique solution $\vec{u}$. However, in rare cases the matrix happens to become singular. There are two possible reasons for this:

1. Insufficient fixed nodes. A fully connected structure needs at least two fixed nodes to stabilize against translations and rotations (called rigid body motions).

2. Malformed nodes. Avoid free nodes with two or less trusses. For nodes with three trusses only, make sure no two of them are parallel, see Figure 5.

If you encounter a singular matrix A, check whether your structure fulfills these conditions.

4 Implementation

Let us recall what variables are given by the specification of the bridge (input data) and what variables need to be computed (output data). The input data is:

- Numbers M, N, and K
- Node positions $\vec{x}_i \in \mathbb{R}^2$ for $i = 1, 2, \ldots, N$ and their types (free or fixed)
- Truss connectivity $t_{k,1}$ and $t_{k,2}$ for $k = 1, 2, \ldots, K$
- Elastic moduli c_k for $k = 1, 2, \ldots, K$
- External loading f_i^{ext} for $i = 1, 2, \ldots, N$

The output data is:

- Node displacements $\vec{u}_i \in \mathbb{R}^2$ for $i = 1, 2, \ldots, N$
- Truss elongations e_k for $k = 1, 2, \ldots, K$
- Truss forces f_k for $k = 1, 2, \ldots, K$

First, let us discuss the input data. The input data will be read from files. Observe that the list of input data consists of data numbered from 1 to N, associated with the nodes, and data numbered from 1 to K, associated with the trusses, except for the scalar numbers M, N, and K. Thus it makes sense to store all node-based data together in one file, and all truss-based data together in another file.

For easy human readability, we choose a plain text format. For the node-based data, we agree on the following format. For each node (free or fixed),

there is one line in the file, ordered by node numbers. Each line consists of (in this order):

$x_{i,1}$

$x_{i,2}$

0 if node is free or 1 if node is fixed

$f_{i,1}^{ext}$

$f_{i,2}^{ext}$

The entries are separated by at least one blank. The file is named as (name of bridge)`.nodes`.

The trusses are stored in a similar way. Each line in the file (name of bridge)`.trusses` consists of the entries

$t_{k,1}$

$t_{k,2}$

c_k

Note that we do not need to specify the numbers M, N, and K explicitly. N is implicitly given by the number of lines in the nodes file, M by the number of lines in the nodes file whose third entry is 0, and K by the number of lines in the trusses file.

For example, the simple structure from Figure 2 and Table 1 is stored as: File `simple.nodes`:

```
0.0  0.0  1  0.0  0.0
1.0  1.5  0  0.0  0.0
2.0  0.0  0  0.0 -2.0
3.0  1.5  0  0.0  0.0
4.0  0.0  1  0.0  0.0
```

File `simple.trusses`:

```
1  2  1
1  3  1
2  3  1
```

```
2  4  1
3  4  1
3  5  1
4  5  1
```

(Do not deviate from the format defined here, since otherwise it would become difficult for the grader to compare the different solutions later.)

Task 2 *Write a MatLab function* `readbridge` *with the name of the bridge as input argument. The function shall read the bridge data from the two files and compute the mapping m_i. It shall return the following variables:*

`x`	*(matrix of size $N \times 2$)*
`map`	*(vector of size N)*
`fext`	*(matrix of size $N \times 2$)*
`t`	*(vector of size $K \times 2$)*
`c`	*(vector of size K)*
`M, N, K`	*(scalars)*

Note: The built-in MatLab function `fscanf` *is useful here.*

Now we have all necessary input data, so we can come to the actual computation.

Task 3 *Write a MatLab program* `bridge` *which first calls* `readbridge` *to get the input data. Then compute the normal vectors $\vec{n}_i$ and the matrix B. Assemble the matrix A from* (20) *and the right hand side $\vec{f}^{ext}$. Solve the linear system for $\vec{u}$ using one of your own solvers developed as homework. Make sure your solver stops with an error message if the matrix A happens to be singular. Compute the elongations $\vec{e}$ and the forces $\vec{f}$ from* (15) *and* (16)*, respectively.*

After having determined the unknowns, we need to interpret the results. Instead of printing large tables of numbers, it is more useful to extract certain quantities. Here the following quantities are of interest:

1. Maximal force: $\max_k |f_k|$

2. Maximal elongation: $\max_k |e_k|$

3. Maximal relative elongation: $\max_k |e_k|/l_k$

Task 4 *Extend your program from Task 3 to determine these quantities.*

It is reasonable to assume that a truss will fail if its relative elongation exceeds a certain value. Thus the relative elongation is especially useful to identify possible points of failure of the bridge. However, we do not only want to discover that our bridge might fail under some given loading. We rather want to know which truss is endangered so we can reinforce it by increasing its thickness or by adding more trusses nearby to better balance the load. To this end, we visualize our solution. The graph shall show:

1. The nodes at their deformed positions $\vec{x}_i + \vec{u}_i$.

2. The trusses with color coding of the relative elongation: Blue means minimal elongation, green to yellow medium elongation, and red to brown maximal elongation.

Task 5 *Write a MatLab function* `plotbridge` *which plots the bridge as stated above and add it to your main program from Task 3.*

Now you are ready to actually compute a bridge by yourself!

Task 6 *Download the data for the bridge shown in Figure 1 from*
`http://www.shapeev.com/teaching/MATH5485/bridge/bridge.nodes`
`http://www.shapeev.com/teaching/MATH5485/bridge/bridge.trusses`
Run your program to compute the above mentioned quantities and create the corresponding graphs for the following loading regimes:

1. $f^{ext}_{11,2} = -0.1$, *all other* $f^{ext}_{i,j} = 0$ *(vertical point load)*

2. $f^{ext}_{23,2} = -0.1$, *all other* $f^{ext}_{i,j} = 0$ *(vertical point load)*

3. $f^{ext}_{i,2} = -0.01$ *for* $i = 9, 11, 17, 19, 20, 22, 23, 33, 34, 40, 42, 43, 45, 46$, *all other* $f^{ext}_{i,j} = 0$
   (uniform vertical load on the whole road surface)

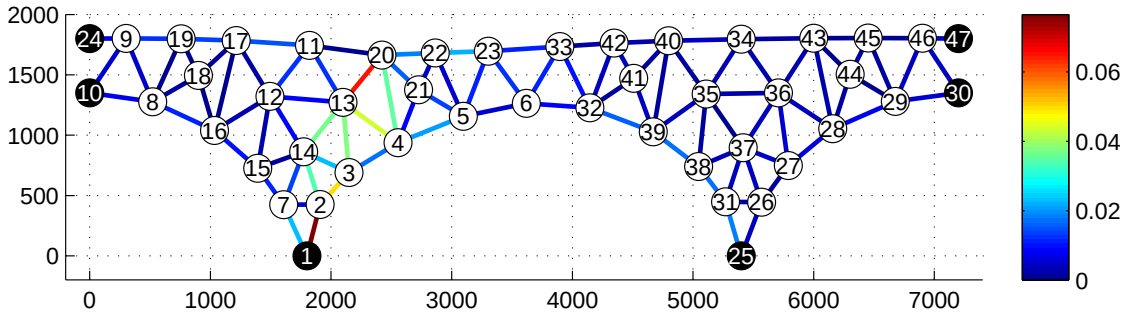

Figure 6: Bridge with loading $f^{ext}_{20,2} = -0.1$.

4. $f^{ext}_{42,1} = 0.1$*, all other* $f^{ext}_{i,j} = 0$ *(horizontal point load, e.g. caused by wind)*

Describe and interpret your results.

Figure 6 shows how the bridge from Figure 1 looks like under the loading $f^{ext}_{20,2} = -0.1$.

5 Bridge Optimization

After having computed a given bridge, we now climb up to the next step. The ultimate interest of bridge engineers is not to evaluate a given bridge, but to design a new one which is as stable and efficient as possible. Of course there are conditions to be met: The bridge must fit with the geometry of the river valley, and there is only a limited amount of material.

Task 7 *Build your own bridge according to the rules described below and submit the two input files (in the format discussed above) as will be instructed later.*

You may build your own bridge from scratch, or you may start with the bridge given here and improve it. For example, one can read off from Figure 6 that the truss between nodes 1 and 2 is highly endangered: The brown color indicates the highest relative elongation, see the color bar. So it makes sense to improve the structure at this point by increasing the elastic modulus of the truss 1–2 or adding more nodes and trusses close to it.

Your bridge must comply with the following rules:

1. At most 100 nodes, i.e. $N \leq 100$.
2. The amount of material for the trusses is limited to $\sum_{k=1}^{K} c_k l_k \leq 100000$.
3. There must be nodes at (0 1800) and (7200 1800). Along the straight line between these two nodes, there must be additional nodes with distance 600 or less. These nodes shall all be connected by trusses to form the road surface.
4. The only possible points for fixed nodes are: (compare Figure 3)
   - $(0\ y)$ with $1350 \leq y \leq 1800$ (left shoreline)
   - $(7200\ y)$ with $1350 \leq y \leq 1800$ (right shoreline)
   - (1800 0) (left foundation in the river)
   - (5400 0) (right foundation in the river)
5. No intersecting trusses.

There will be a competition among students: The stability of all bridges will be ranked, and the most stable ones get extra points! Stability is measured as follows. The following loads are applied to the bridge:

1. Vertical point load $\vec{f}_i^{ext} = (0\ -0.1)$, individually for each node i on the road surface.
2. Horizontal point load $\vec{f}_i^{ext} = (0.1\ 0)$, individually for each node i on the road surface.

For each loading regime, the maximum relative elongation is determined. Then the maximum of these values is taken over all loading regimes. The bridge with the smallest maximum value is best. Don't forget to submit this number for your bridge.

Good luck and have fun!