

Guide for learning the basics of SSI for GSA

Welcome to this guide for learning the basics of Smart Structural Interpreter for GSA. I have written this guide upon request by my thesis supervisor, but it is meant for any student who wishes to learn more about parametric designing, and in specific, the coupling between a parametric model in Grasshopper and Oasys GSA.

This guide will cover some of the basics for making a structural model with Grasshopper. It is assumed that the student has knowledge of how both Grasshopper and GSA work. For instance, when a reference is made to a component in Grasshopper is made it is displayed as <component name>. Furthermore, Grasshopper is referenced to as GH and Smart Structural Interpreter as SSI. Geometry refers, for instance, to the lengths and angles of lines. Topology constitutes the relation between the drawn lines (ie. A line can be drawn for point 3 to point 5).

If you have any questions, please post them in the discussion. Alternatively, you can contact me at michael@vantelgen.nl

A beam on two supports

The GH file is included with this document. See file 'parametric beam'.

For this guide, a very simple structural problem is illustrated. The problem is drawn in Figure 1.

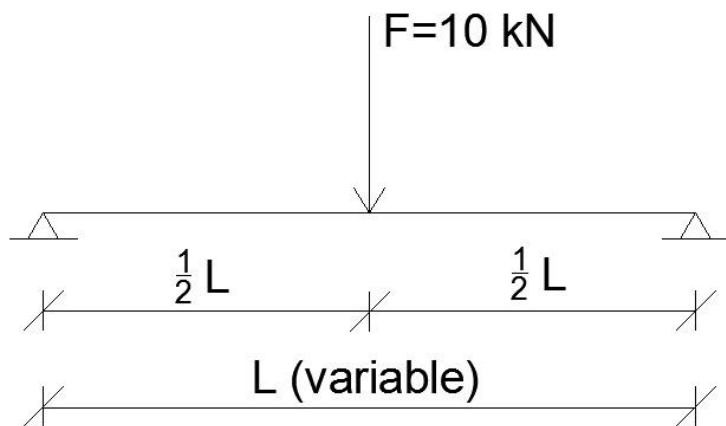


Figure 1

Here we have a beam on two pinned supports, loaded by a point load at $\frac{1}{2} L$ of the beam. The span of the beam is variable: $10 \leq L \leq 20$ (in meters).

Grasshopper

First, the GH model is made. This is the parametric model which can be varied, and subsequently, different models emerge. Since we have only one variable (the length of the span L), the model is quite simple. See Figure 2.

The <slider> determines the span length in meters. However, GSA interprets this '10' in millimeters, so the number must be multiplied by 1000 to get the length in meters. Note that this is due to user preferences; you might want to change it to kN. The <multiplication> components ($A \times B$) multiply the X value for each point. The top <point XYZ> component is located at (0;0;0), the middle one is at (5000;0;0) since its multiplier is set to 500 (half of the difference between millimeters and meters). The bottom multiplier is set to 1000. This results in a point at (10000;0;0).

Then the <line>s are drawn between the right points.

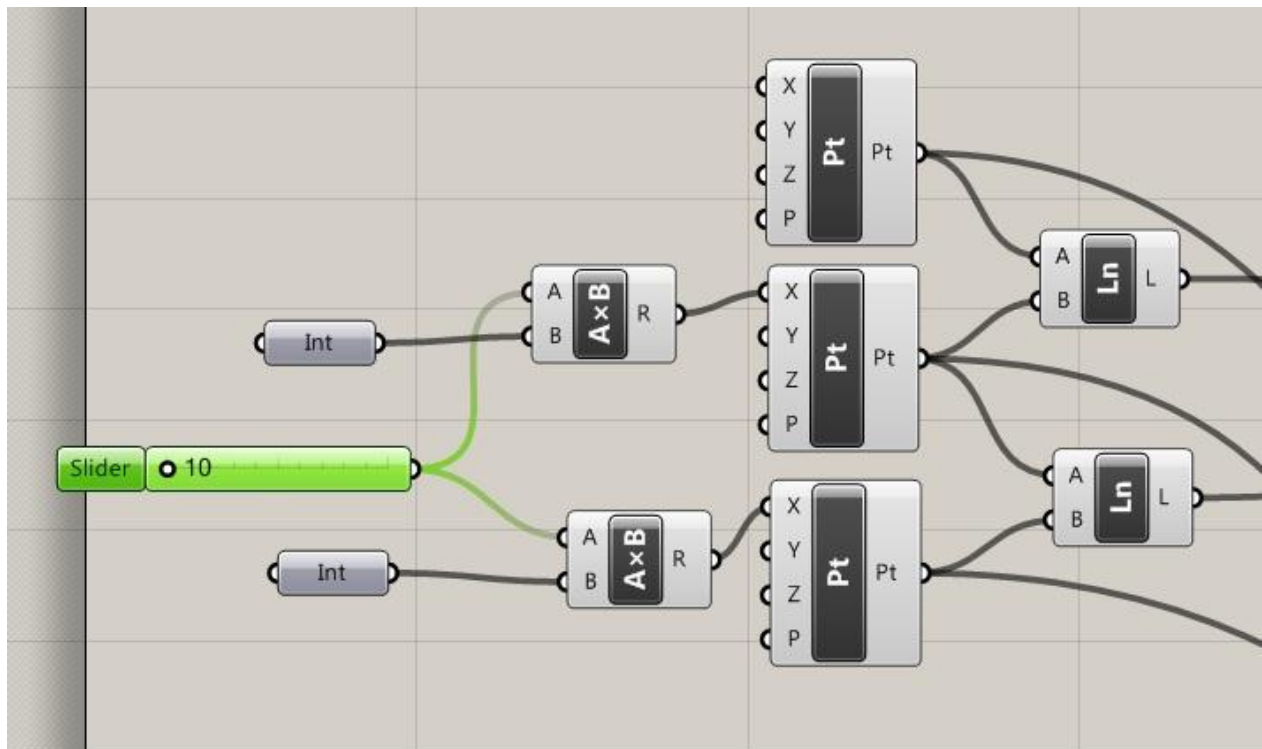


Figure 2

The geometry and topology is now available, and we can make variations in span length. However, GSA doesn't understand yet that the lines are actually beams, and the points are nodes. Also, the point load must be set, and the type and location of the supports must be defined. This is done by the SSI components.

Beams

Let's start with the beams. See Figure 3. Five components are added:

- <ssimat> defines the material type for the beams, in this case it is set to steel
- <sdr catalogueprofile> provides a database of sections which can be selected. In this case, HE300A is selected.
- <ssisp> creates the section properties.
- <ssiba> creates the beam attributes. The 'BT' input is set to 'beams'.
- All information for beams is collected in the <ssibeam> component. Not that the lines are also attached to this component.

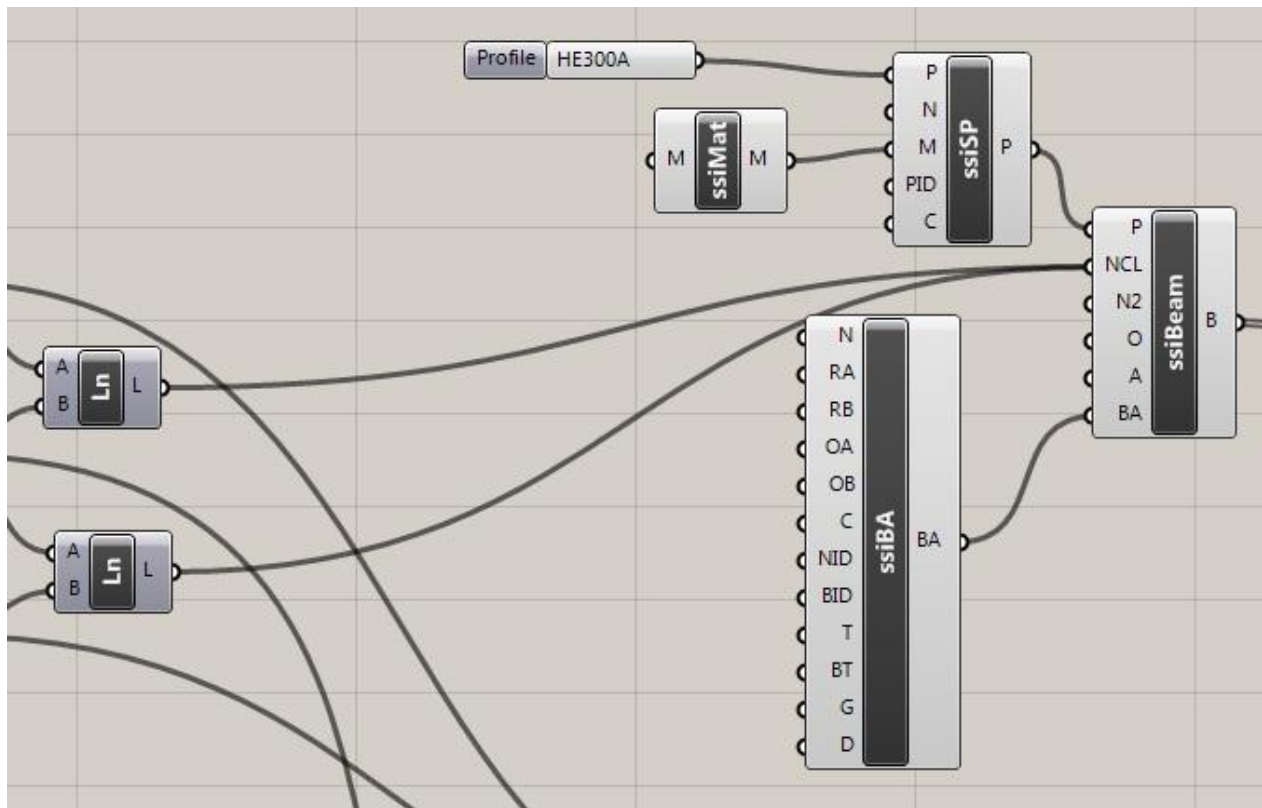


Figure 3

Nodes

Now the nodes can be defined. See Figure 4.

- <ssinode> creates the GSA node from the points in GH. The top one is connected to the outermost points, and the bottom one is connected to the point at the center of the structure.
- <ssinr> creates the proper node restraints for each node. For the pin supports, the translation directions x, y and z are restrained. Therefore their inputs are set to 'true' by the <bool> component. The node at the center of the beam is restrained in rotation directions. This is done to ensure the right kind of connection between the beams. In this case the beam is fully connected on the right and left hand side of the node.
- The <ssina> component defines the node attributes.

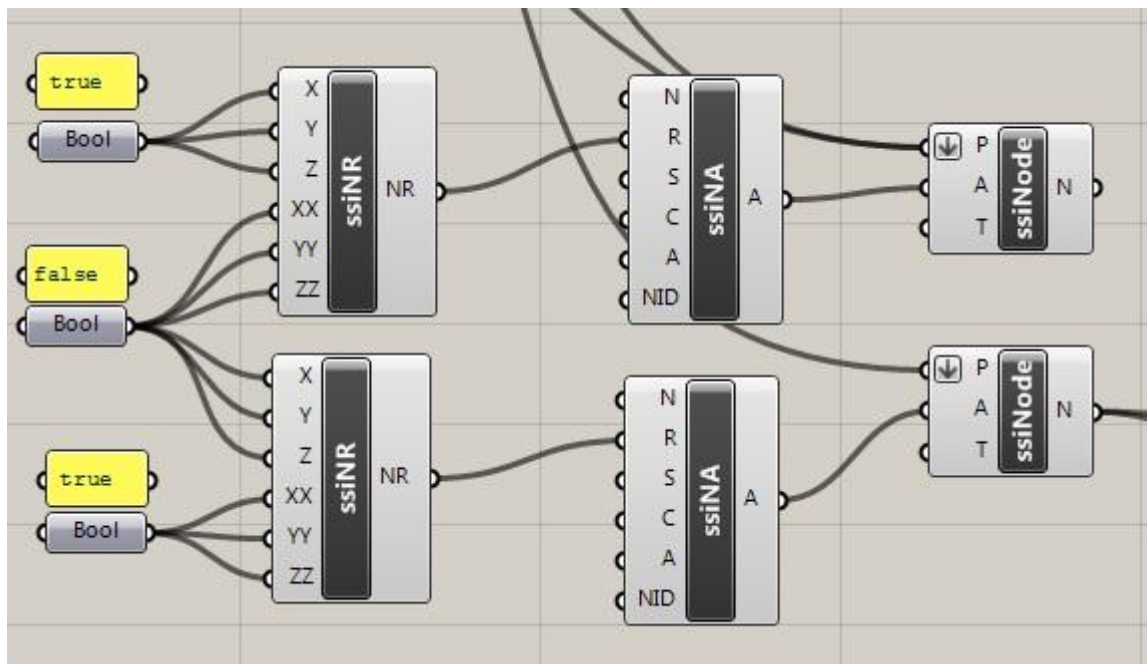


Figure 4

Load case

We have to define the load case next (see Figure 5).

On the top left we see the node which was also displayed in Figure 4. Not that it is connected to the <ssinl> component. This component makes a node load for the attached node. The direction and magnitude of this load is defined by the <num> and the <vec xyz> component. The <num> is set to -10000, so GSA knows that the load is 10 kN (10000 N) in the negative direction (downwards). The vector component then makes the vector using this value for z. Note that the values for x and y are set to 0 by default.

The load case is made by the <ssilc> component. The name of the component is defined by the <string> component which is renamed as 'point load'. The load case ID is set to 1. The load case must be set for each node to which it must apply. Here, it only applies to the single node at the center.

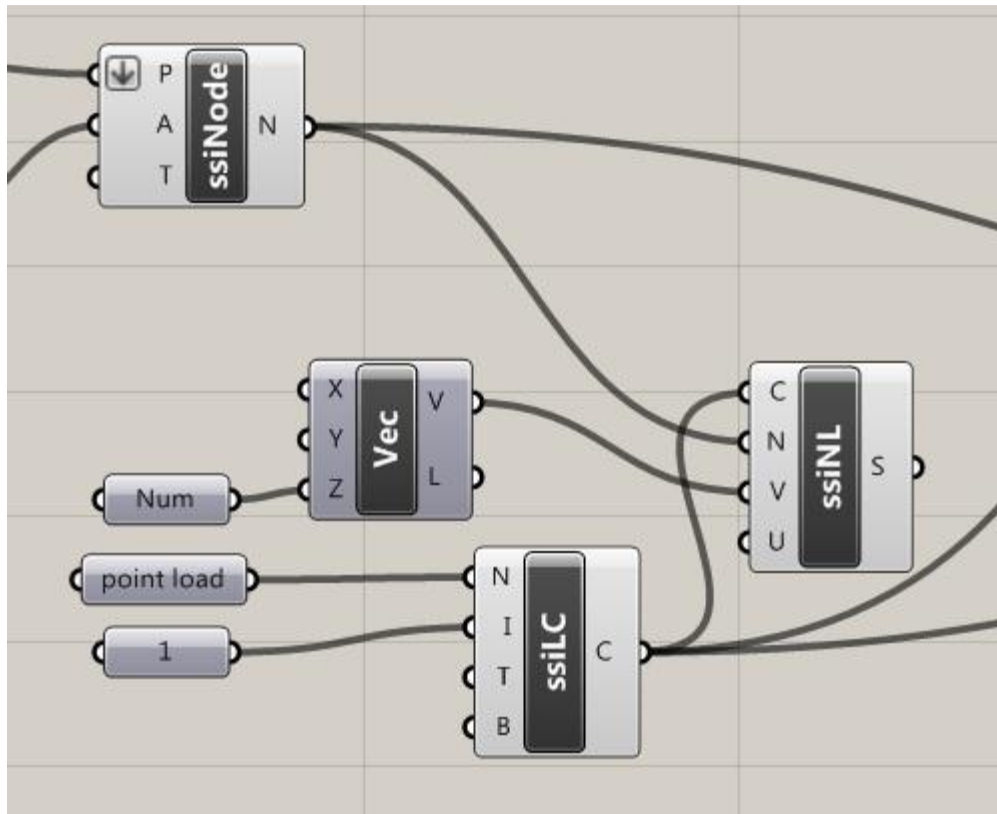


Figure 5

Structural analysis

Since the model is now properly defined, a structural analysis can be made. It is chosen to perform a static analysis, since the model is very simple. See Figure 6.

The <ssis> component executes the structural analysis. With this component the supplied parametric model is sent to GSA for analysis, and optionally, the relevant data is retrieved from GSA. If the 'C' input is set to true, the model can be seen in GSA. This is extremely useful for problem solving, since all data is then visible in GSA.

If desired, the relevant data can be retrieved from GSA. Here we are interested in:

- The shear force in direction Z
- Bending moments in direction Y
- Node displacement for the beam at the center

The <ssiquerycurveelemfor> component inquires the model for beam forces. In this case, the 'Fz' and 'Myy' inputs are set to 'true'. (Note that in this case, it is done by setting it directly in the component itself. Of course it can also be done by adding a <bool> component set to 'true' and connecting it)

The <ssiquernodedisp> inquires the model for node displacements.

For both components input 'C' is the load case as seen in Figure 5.

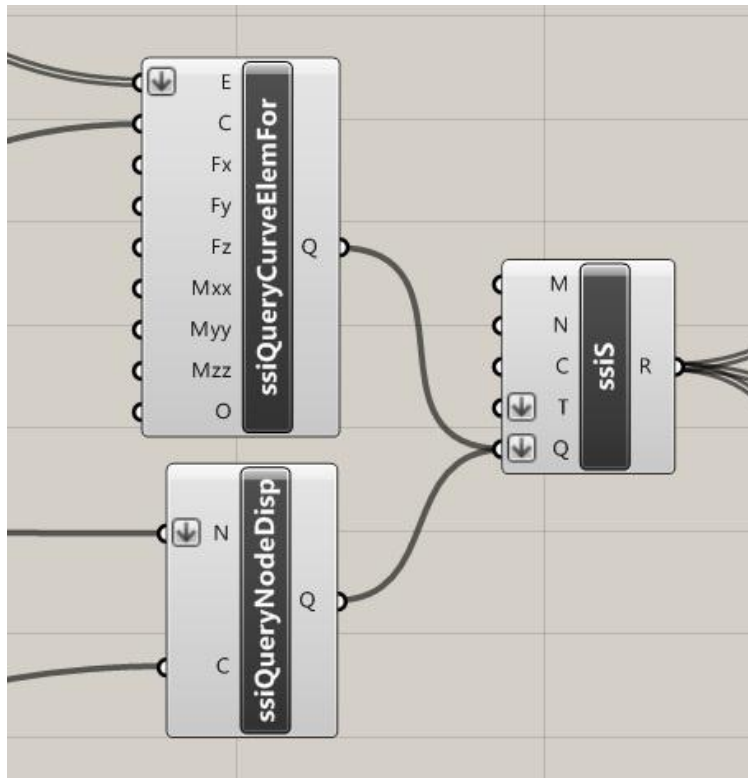


Figure 6

Displaying the results

Next, the relevant result must be extracted from the solver (see Figure 7). The solver provides a list from which the correct results must be extracted. This is done with the <cull pattern> component. There are three streams, so three Booleans are defined in the cull list. For the shear forces the pattern is true, false, false.

The <ssiRes1dSetDcompose> retrieves the values for the shear force. Since this result is in N, the data is divided by 1000 by the <int> and <divide> components. The results are displayed in the panel component.

The same is done for the bending moments, only now the cull list is false, true, false.

The <ssiResNodeSetDcompose> retrieves the node displacement at the center of the beam. This data is actually displaced as a vector, so in order to see the displacement in Z direction, the vector must be decomposed.

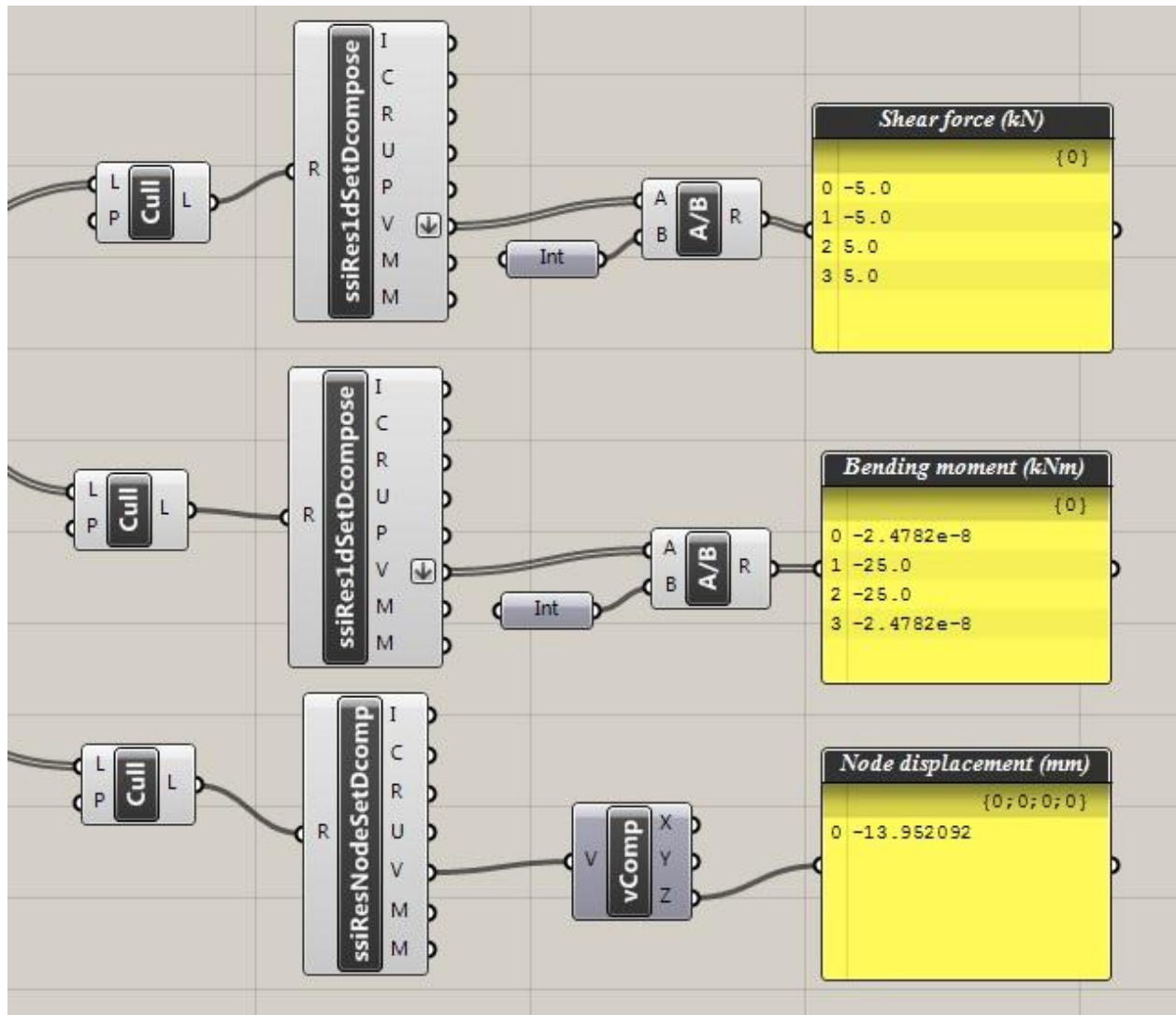


Figure 7

Conclusion

This is a very simple example of the method which must be used. It is strongly advised to test the different components for yourself, as well as adding other variables.