LUSAS plugin for Grasshopper

User Guide: Issue 1

LUSAS
Forge House, 66 High Street, Kingston upon Thames,
Surrey, KT1 1HN, United Kingdom

Tel: +44 (0)20 8541 1999
Fax +44 (0)20 8549 9399
Email: info@lusas.com
http://www.lusas.com

Distributors Worldwide

Table of Contents

LUSAS plugin for Grasshopper

Grasshopper

Grasshopper is a visual programming language and environment that runs within the Rhinoceros 3D computer-aided design application. By using its visual programming, you can algorithmically generate geometry by composing diagrams that link data to functions.

LUSAS plugin

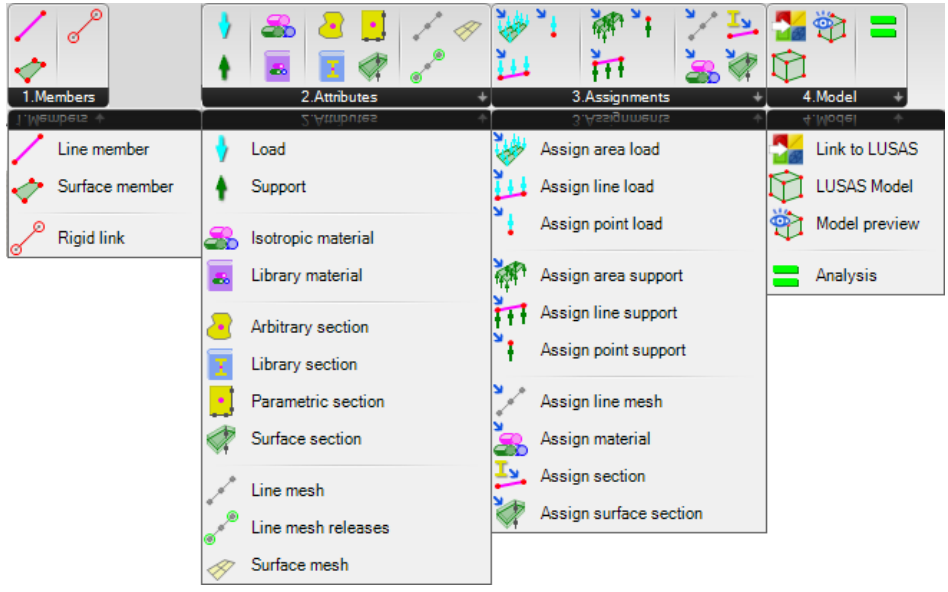
By using the LUSAS Grasshopper official plugin, parametric modelling of any structure is possible, and the models created can be output to LUSAS for further modelling, analysis and design purposes.

The plugin is available from within Rhino's package manager as well as from the food4rhino website at https://www.food4rhino.com/en

LUSAS components

The LUSAS-authored Grasshopper plugin creates a ‘LUSAS’ entry in the main menu and a ‘LUSAS’ tab in Grasshopper’s component ribbon. The LUSAS components are grouped into four named categories: Members, Attributes, Assignments and Model, which are shown with their respective icons.

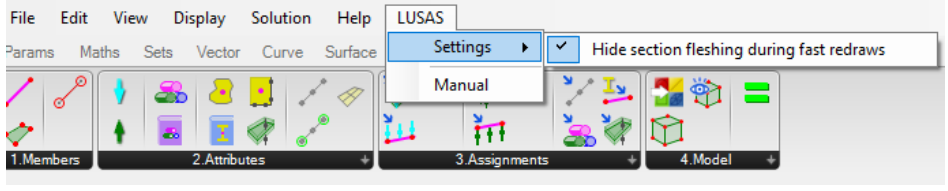
When expanded, each category shows its respective components separated logically into further subcategories. By selecting a component from this menu it can be placed on Grasshopper’s canvas to be used in a script.



LUSAS on the main menu

The LUSAS entry in Grasshopper's main menu holds a sub-menu of global features or settings for the plugin. Currently the available menu items are:

- **Settings** - Displays an option to 'Hide section fleshing during fast redraws', which is 'on' by default.
- **Manual** - Provides a link to the help topics / manual for the plugin.



Model units

All numeric values on LUSAS components are assumed to be in the same units as selected on the LUSAS model component. It is important to decide on the model units before starting a LUSAS Grasshopper script, as the LUSAS model component requires that the length units in the Rhino model match with the selected units on the component. An error will stop the process otherwise.

Note that whilst the units can be changed at any time, any numeric values which have been used as input to other components (for example the values for an isotropic material or the coordinates of geometry) will not be automatically converted.

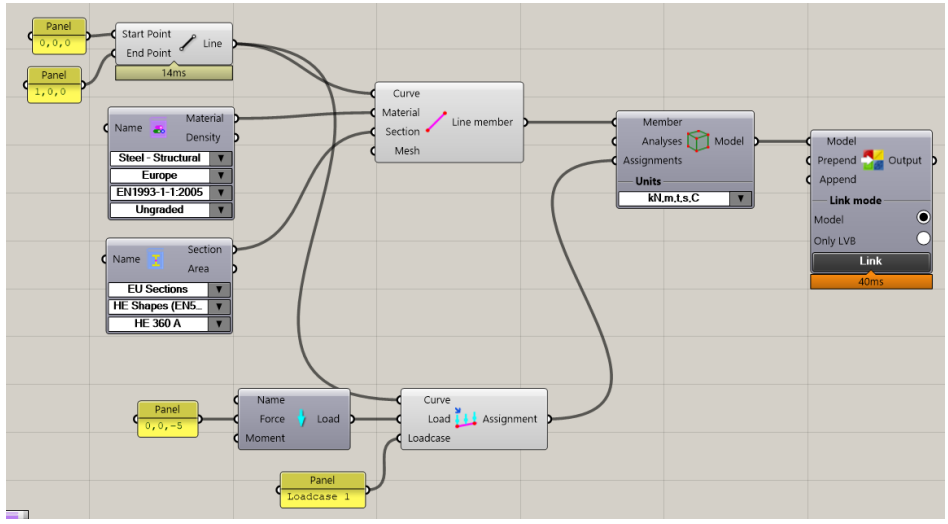
Standard workflow

The standard workflow for a Grasshopper script that creates LUSAS members, attributes and assignments and links them to a model within LUSAS is:

1. Select the desired model units at the beginning and set the same units in the Rhino model.
2. Create structural members (line or surface members), by specifying the geometry using standard Grasshopper components or referencing Rhino model geometry.
3. (Optional) Create attribute definitions (materials, sections etc) and assign them to the structural members.
4. (Optional) Create load and support attribute definitions and assign them to geometry matching a structural member's nodes, lines or surfaces.
5. Assemble all structural members and load/support assignments on the 'LUSAS model' component with the same length units as in the Rhino model.
6. (Optional) Use the 'Model preview' component to preview member sections, loads, supports and rigid links.
7. Use the 'Link to LUSAS' component with the assembled model to link with LUSAS, either by creating a distributable LVB script or creating a model directly.

Simple script

The following image shows a simple script which creates a line member with a material and a section, assigns a line load of -5kN in the Z direction and assembles the model ready to be linked to LUSAS.



Component input parameters

There is a distinction to be made between the two types of input:

1. Generic input types, which can be set with standard Grasshopper components, for example curve/surface geometry, number, text, vector etc.

2. Input types which can be set exclusively using the output of other LUSAS components, for example material, section etc.

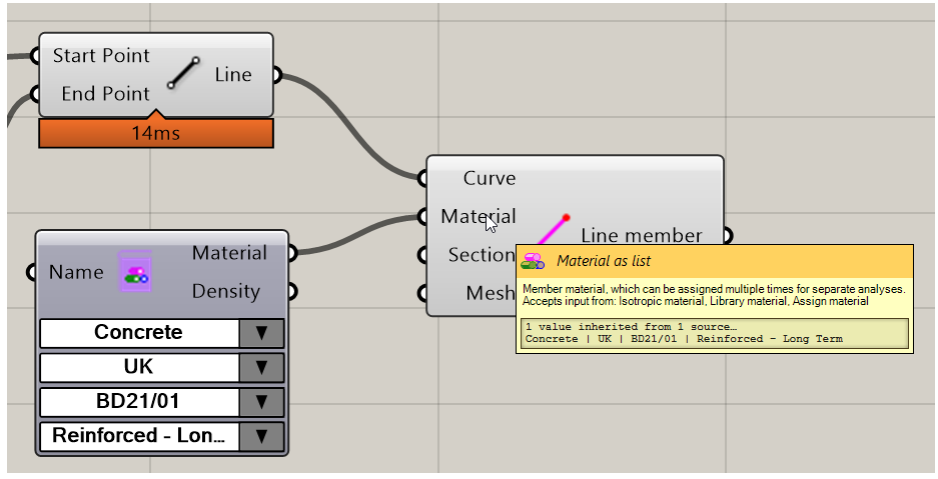
The type of a parameter can be seen by hovering the mouse cursor over the parameter's name.

To prevent confusion, any parameter inputs on LUSAS components of the 2nd type that accept only the input of other LUSAS components have an extended tooltip stating which component outputs can be used to set that parameter's value.

Example

The following example for a line member component shows the two types of different parameter inputs. The 'Curve' parameter is set from a standard Grasshopper component, and the 'Material' parameter is set from a LUSAS component.

A tooltip for the material parameter states which LUSAS components can be connected.

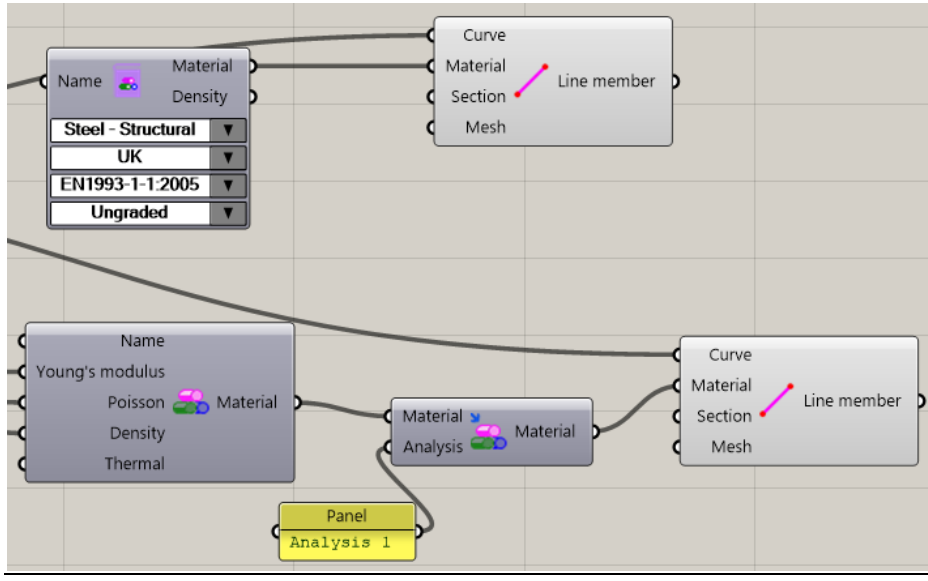


Structural line and surface member components

Some inputs on the structural line and surface member components (for example material, section etc) can be set in two separate ways:

1. By using the output of the attribute definition components, for example Isotropic material, Parametric section etc.

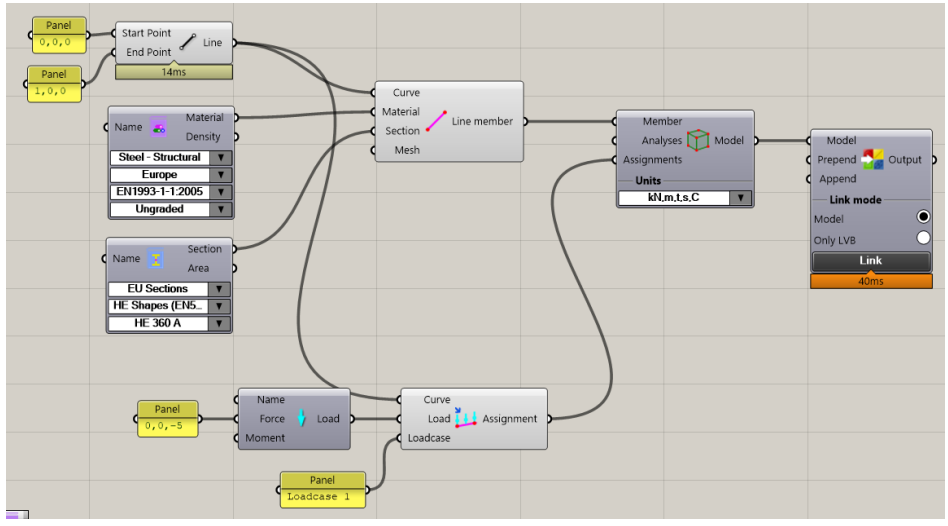
2. By using the intermediate assignment component if additional data like a specific analysis need to be provided, for example Assign material, Assign section etc.



Example

The following image shows the two possible ways of defining the material for a line member:

- by using the output of a material attribute definition component directly
- by using an intermediate assignment component to state a specific analysis.



Load / Support assignment components

Load and support assignment components require the geometry (point, line, surface) as an input. The plugin will automatically find which member the attribute should be assigned to.

Currently, loads and supports can only be assigned to points that are actual nodes of a structural member (curve endpoints, surface vertices) or curves that define a line member or are edges of a surface member. Loads or supports assigned to points or curves within the length of a line member or within a surface member are not supported.

Nurb surface compatibility

Nurb surfaces are exported to LUSAS without any internal trims (holes).

Some types of nurb surfaces are incompatible and will not be exported. Warning messages will appear on the surface member component with an explanation when:

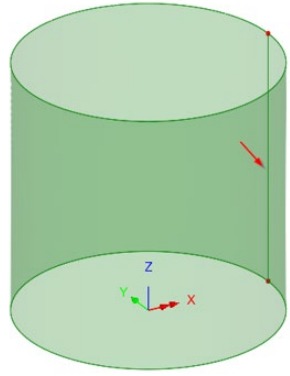
- The surface contains a seam (e.g. cylinder).

- The surface contains a singularity point (e.g. sphere apex).

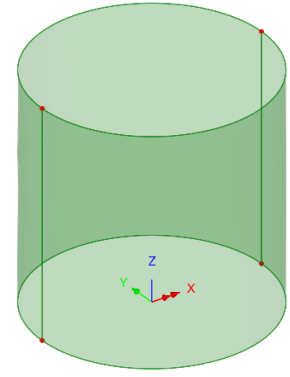
In most cases, problems with nurb surface compatibility can be overcome by subdividing the surface.

Example: Cylinder

Cylinders can be exported successfully by splitting the cylinder in half, so that there is no seam.



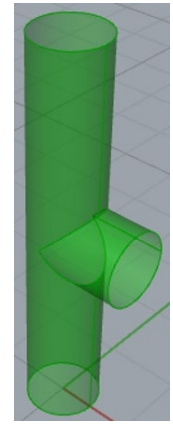
Cylinder prior to splitting with seam highlighted



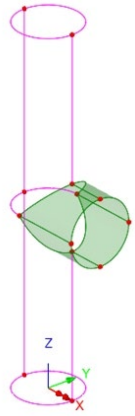
Cylinder after splitting

Example: CHS connections

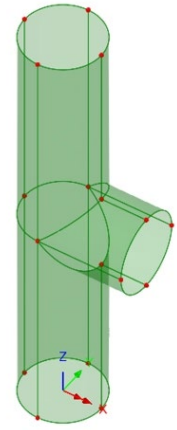
A more complex model of CHS connections can be exported successfully if an appropriate number of subdivisions of the original geometry is made.



CHS connection boundary representations (brep) exported to LUSAS



The horizontal CHS section is subdivided horizontally and vertically.

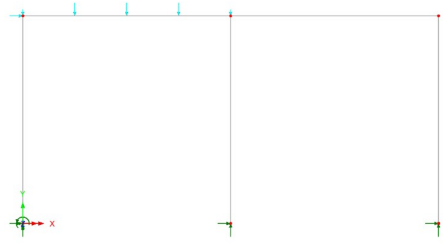


The vertical CHS section is subdivided further to eliminate the cylinder seam and make the export possible for the whole boundary representation.

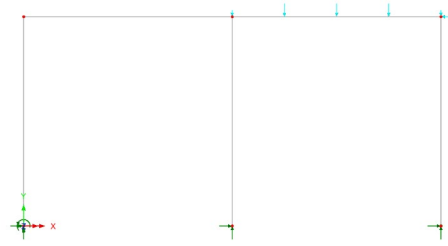
Example script

LUSAS model (for reference)

A model of a simple loaded frame is shown below. The frame consists of 3 columns and 2 beams with different sections. The leftmost column is fixed at the base while the others are pinned. There are two loadcases each containing one concentrated point load and one distributed line load.

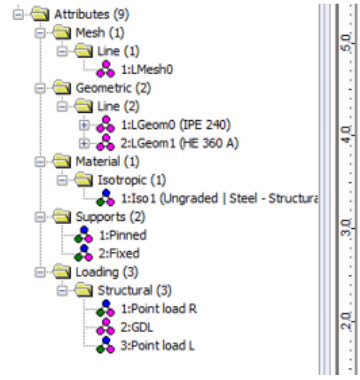


Loadcase 1



Loadcase 2

The corresponding Attributes that would be generated for this model in LUSAS are these:



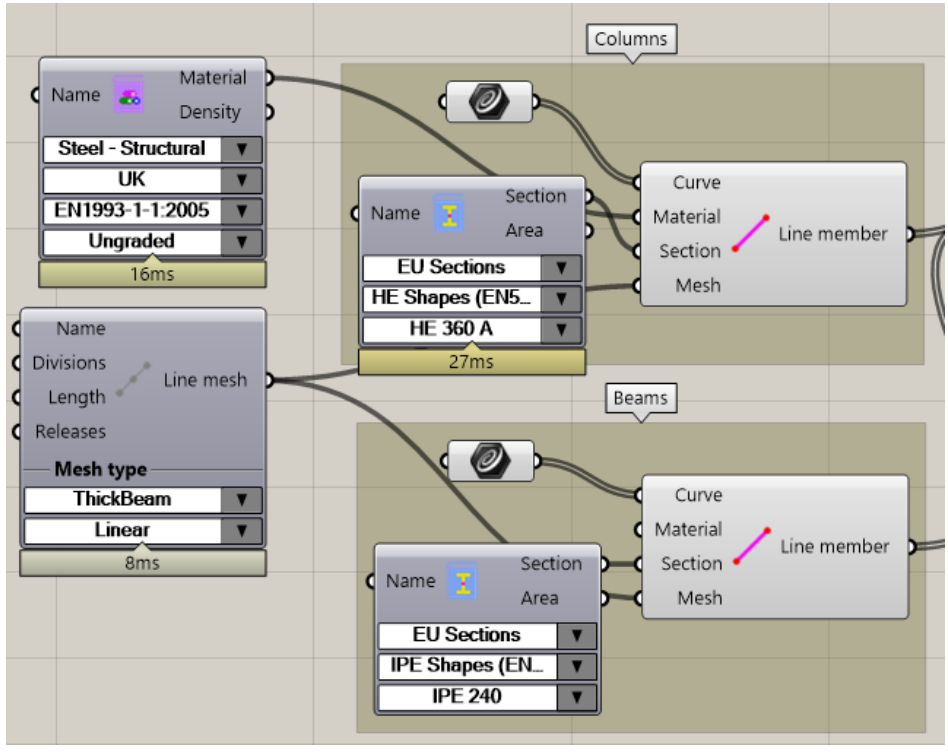
Grasshopper script

This is a step-by-step guide showing the Grasshopper script required to create the model above using LUSAS components. Note that for the sake of brevity the geometry used is always referenced from the Rhino model, but this could also be created parametrically using standard Grasshopper components.

The model consists of 5 line members (3 columns and 2 beams) which share a common mesh and material attribute but have different sections. The geometry is referenced from the Rhino model but could be also set from standard Grasshopper components that create curves.

For this example, a line mesh with default values is used along with a material and sections from the LUSAS libraries.

Defining column and beam members



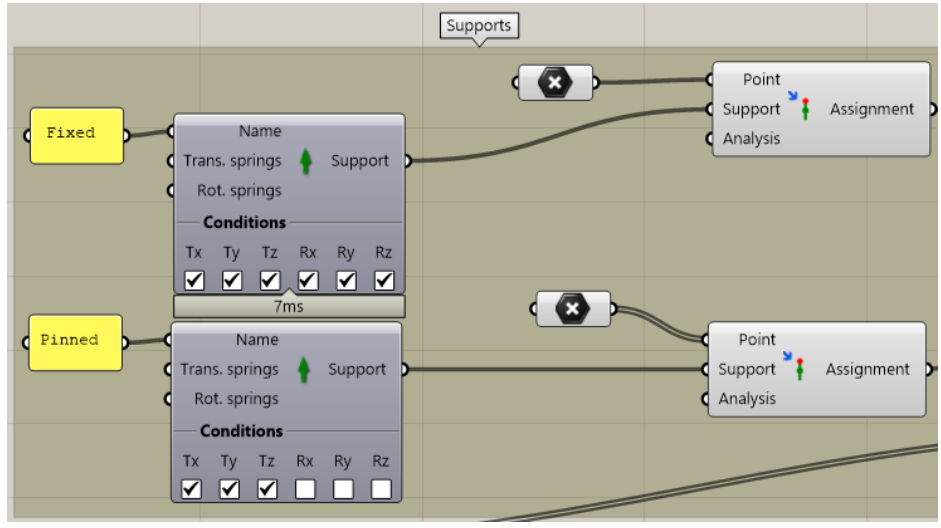
Column and beam members with their geometry, mesh, material and section

Defining supports

To define the supports, the ‘Support’ component is used to create a pinned and a fixed support which is later assigned to the base of the columns using the ‘Assign point support’ component.

For the geometry, point references from the Rhino model are used here; they also could be extracted from the column curves used for the line members.

The analysis parameter is left empty, which signifies that the supports will be assigned to the default base analysis.



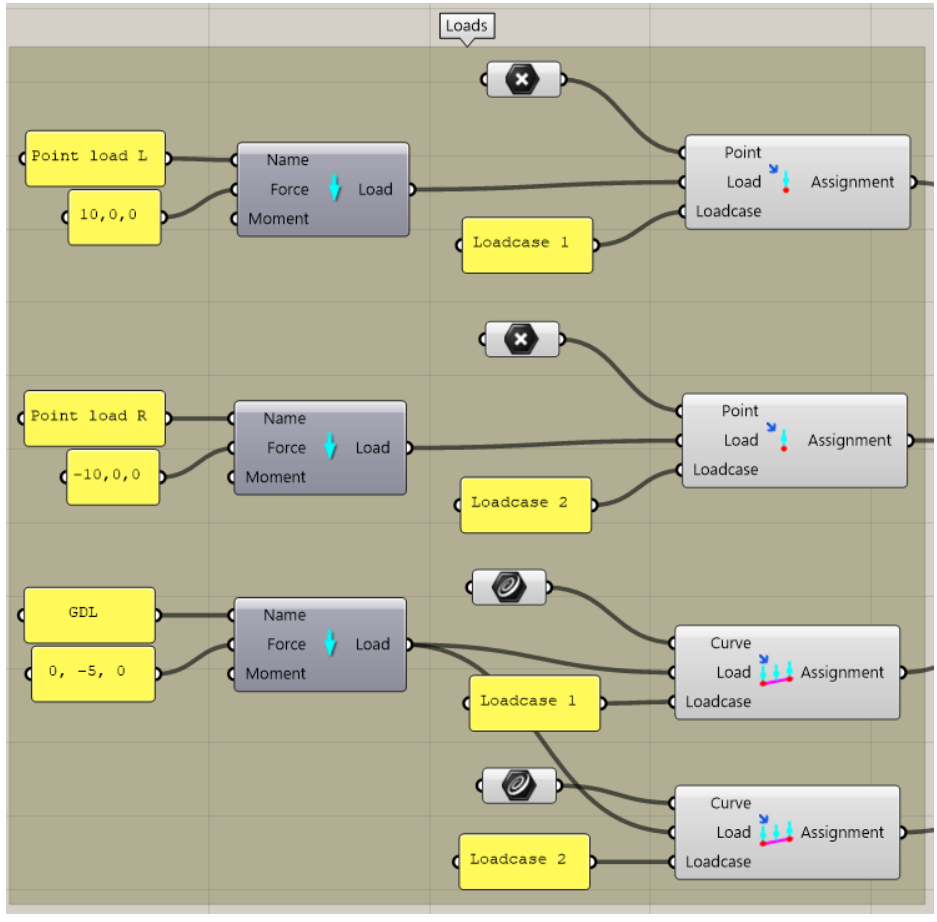
Support definitions at the column bases.

Defining loading

Loads are defined similarly using the Load component to create the 3 different load definitions: a point load of 10kN in the +X direction, a point load of 10kN in the -X direction and a distributed load of 5 in the -Z direction.

The ‘Assign point load’ component is used with referenced points from the Rhino model to assign the point loads; each point load is assigned to a different loadcase.

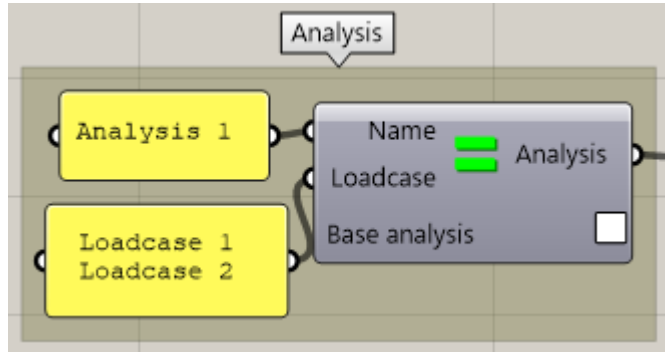
The ‘Assign line load’ component is used with referenced curves to assign the distributed load in each beam for each loadcase.



Load definitions for the beams.

Analysis

The plugin creates an analysis by default called ‘Analysis 1’ with 1 sole loadcase called ‘Loadcase 1’. For this example two loadcases are required, therefore the ‘Analysis’ component is used to define them. If only one analysis is defined in a model, it will be the base analysis by default.

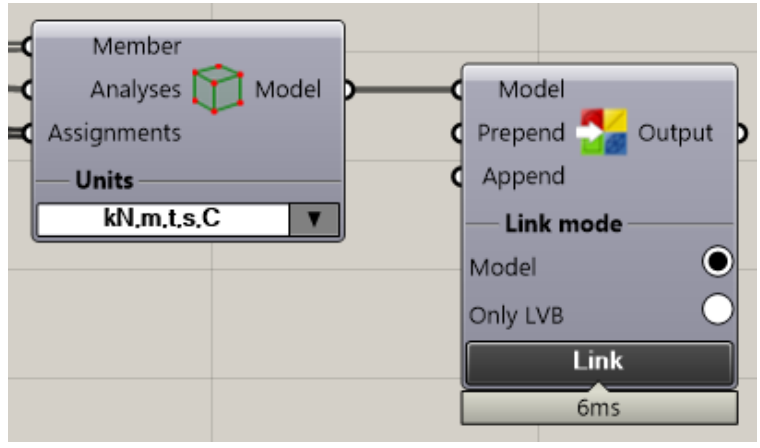


Model analysis and loadcases definition.

Assemble Model data and Link to LUSAS

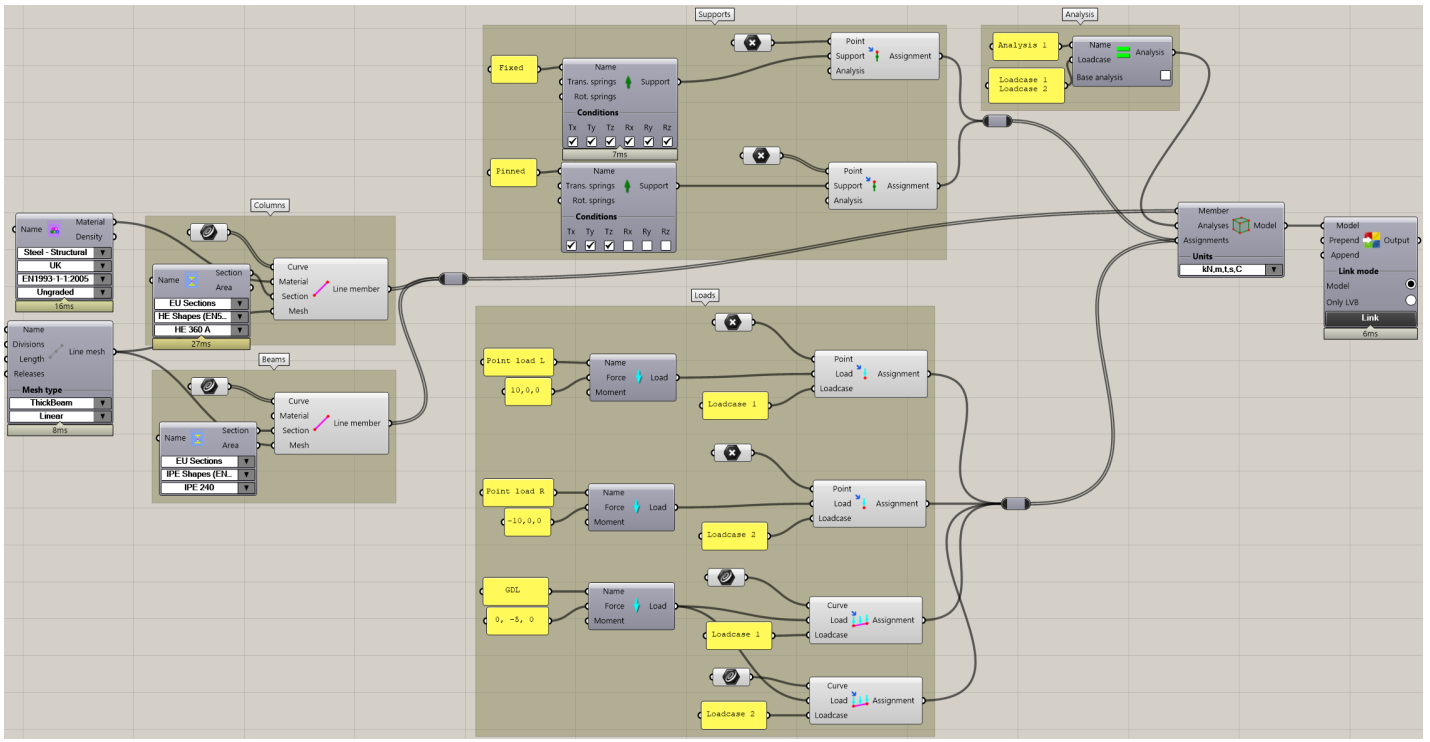
Lastly, the 'Model' component is used to assemble all model data and set the model units. The output of the previously shown components is connected to this component.

The model output is then connected to the 'Link to LUSAS' component which is responsible for creating a LUSAS model (or LVB script) with the specified model data.



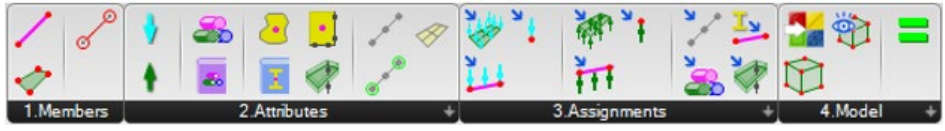
Model assembly and linking to LUSAS.

The complete script containing all components along with their required connections is shown in the following image.



LUSAS component reference

LUSAS component groups



These comprise:

- **Members**
- **Attributes**
- **Assignments**
- **Model**

Members

Structural members (LUSAS line and surface features with attributes) are called line and surface members respectively in this reference guide.

Line member(s)

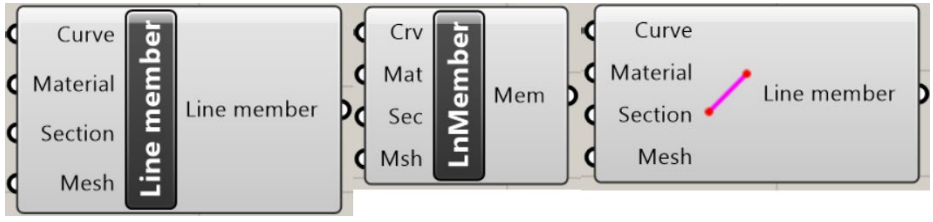
Line members have a ‘Curve’ input (one or multiple), which is a data type native to Rhino. This curve is converted to a curve type compatible with LUSAS.

The ‘Mesh’ input accepts the output from both the ‘Line mesh’ and the ‘Assign line mesh’ components. In the case of the former, a beta angle of 0 degrees will be assumed.

The ‘Material’ and ‘Section’ inputs accept the output of their respective attribute definition components either directly or through their respective assignment component.

The output of the component can also be cast as a native Rhino 'Curve' (in essence the same as the input curve) so that it can also be fed to geometry components as an input.

Inputs	Outputs
Curve(s) – *native Rhino type*	Line member(s)
(Material) – *output of the (Assign) Material component*	
(Section) – *output of the (Assign) Section component*	
(Mesh) – *output of the (Assign) Line mesh component*	



The line member component in all possible display modes.

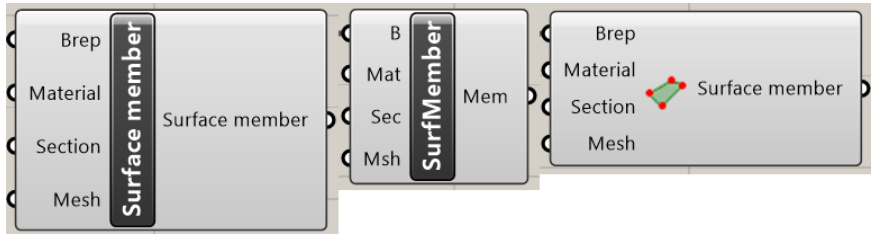
Surface member(s)

Surface members have a boundary representation (brep) input (one or multiple), which is a data type native to Rhino potentially consisting of many surfaces. Each surface will be converted to either a flat or a NURBS surface in LUSAS depending on the geometry in Rhino.

The output of the component can be cast as one or multiple native Rhino Surfaces (in essence the same as the surfaces in the input brep) so that it can also be fed to geometry components as an input.

The component will validate if the surface can be exported and show a warning message otherwise.

Inputs	Outputs
Brep(s) – *native Rhino type* (Material) – *output of the (Assign) Material component* (Section) – *output of the (Assign) Surface section component* (Mesh) – *output of the (Assign) Surface mesh component*	Surface member(s)

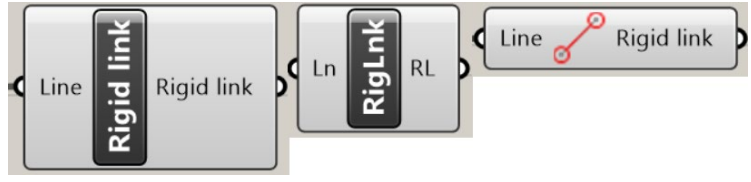


The surface member component in all possible display modes.

Rigid link

Rigid links are defined with native Rhino Lines which will connect their end points with a rigid joint in the resulting LUSAS model. The component's output must be connected to the 'Members' input of the 'Model' component.

Inputs	Outputs
Line(s) – *native Rhino type*	Rigid link(s)



The rigid link component in all possible display modes.

Attributes

Attributes have their own definition component and will be assigned to members with their respective assignment components where necessary. Attributes that require no additional assignment data in LUSAS (or in which the assignment data will be

assumed) can be connected directly from the attribute definition component to the corresponding member.

Line mesh

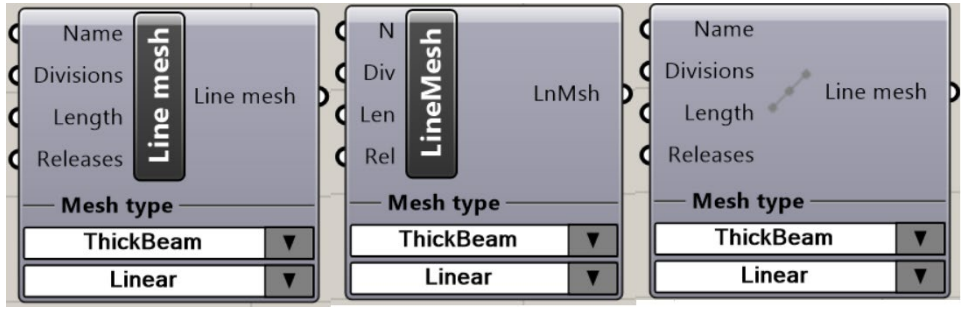
The line mesh component allows for the definition of line mesh attributes. It features two custom dropdown lists for selecting the line mesh type and order. The available line mesh types are the following: none, bar, thick beam (default), thick cross section beam, thick beam with torsional warping, thick cross section beam with torsional warping, thin beam.

The ‘Divisions’ input optionally specifies the number of mesh divisions. If left undefined, the default spacing option in LUSAS is assumed.

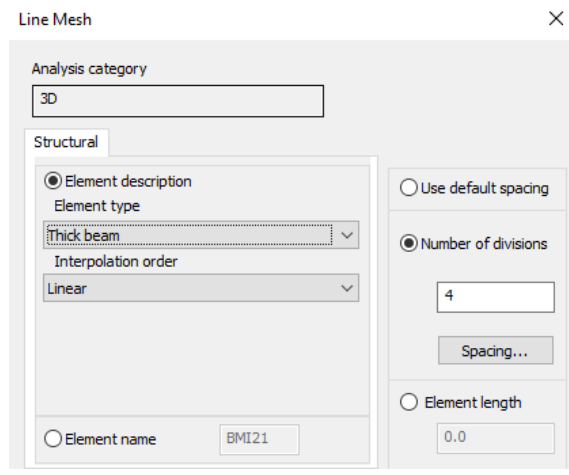
The ‘Length’ input optionally defines a fixed element length for the mesh. The component only allows one of the two inputs to be specified at any time and reports an error otherwise.

The ‘Releases’ input can optionally be used to specify releases for the line mesh, by taking the output of the ‘Releases’ component as input. Note that not all line mesh types support end releases. This is validated by the component and an error message is shown if necessary.

Inputs	Outputs
(Name) – *single line text*	Line mesh
(Divisions) – *numeric, > 0, default: 4*	
(Length) – *numeric, ≥ 0, default: 0*	
(Releases) – *output of the Line mesh releases component*	



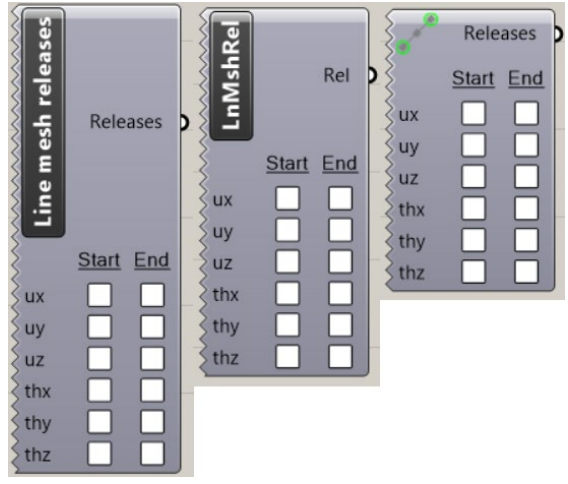
The line mesh component in all possible display modes.



Line mesh dialog in LUSAS.

Line mesh releases

The releases component allows the definition of releases for the 'Line mesh' component. By checking any of the custom checkbox controls, the respective degree of freedom for the start/end of the line mesh can be released.



The line mesh releases component in all possible display modes.

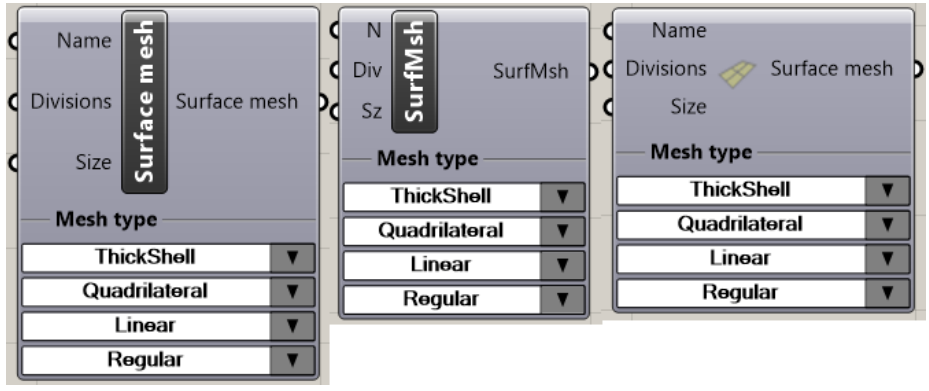
Surface mesh

The surface mesh component allows for the definition of surface mesh attributes. It features three custom dropdown lists for selecting the surface mesh type, shape and order and one dropdown list for selecting between regular, regular (allow irregular) and irregular mesh. The available surface mesh types are the following: none, thick shell (default), thin shell, flat thin shell, space membrane.

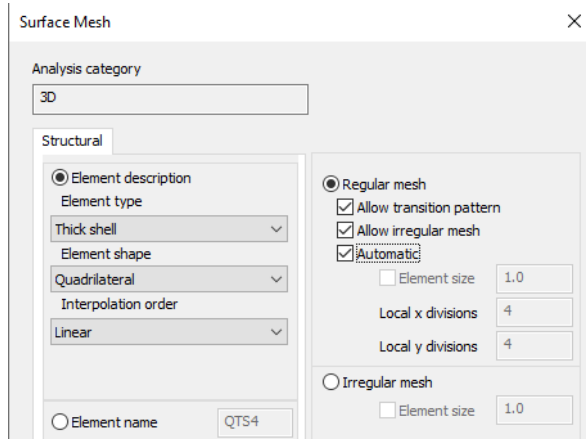
The 'Divisions' input optionally specifies the number of mesh divisions in both directions. The 'Size' input optionally defines a fixed element size for the mesh. If both inputs are left undefined, the 'Automatic' option will be set on the Surface mesh dialog within LUSAS. The component only allows one of the two inputs to be specified at any time and reports an error otherwise.

If an irregular mesh is specified, only the 'Size' input is valid, and an error is shown if any input is connected to the 'Divisions' input.

Inputs	Outputs
(Name) – *single line text*	Surface mesh
(Divisions) – *numeric, > 0, default: 4*	
(Size) – *numeric, ≥ 0, default: 0*	



The surface mesh component in all possible display modes.

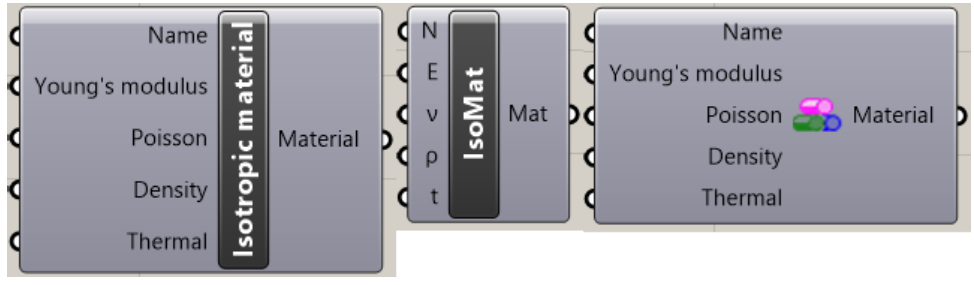


Surface mesh dialog in LUSAS.

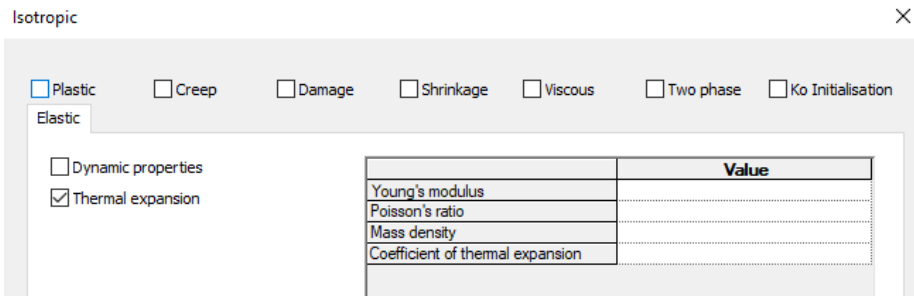
Isotropic material

Isotropic material inputs match the inputs in Modeller's Isotropic dialog.

Inputs	Outputs
(Name) – *single line text*	Isotropic material
Young's modulus – *numeric, > 0*	
Poisson ratio – *numeric, > 0*	
Density – *numeric, > 0*	
(Thermal expansion coefficient) – *numeric, > 0, default: 0*	



The isotropic material component in all possible display modes.



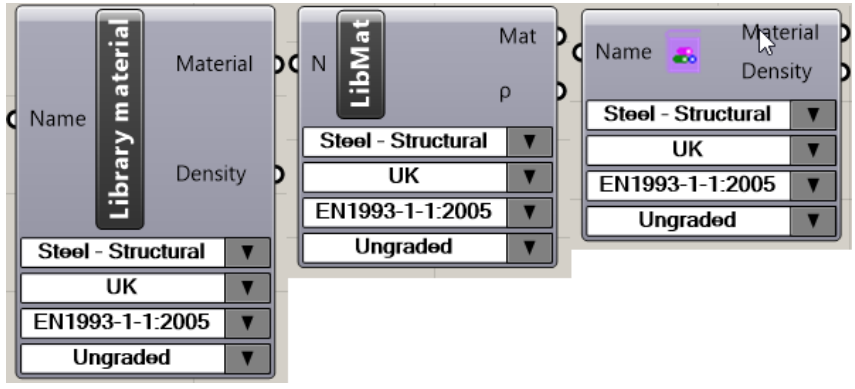
Isotropic material dialog in Modeller.

Library material

With the library material component you can select from the available materials in the LUSAS material library using the custom dropdown menus.

The component outputs the material density (converted to model units) in case it can be used for other calculations.

Inputs	Outputs
(Name) – *single line text*	Isotropic material Density – *numeric*



The library material component in all possible display modes.

Parametric section

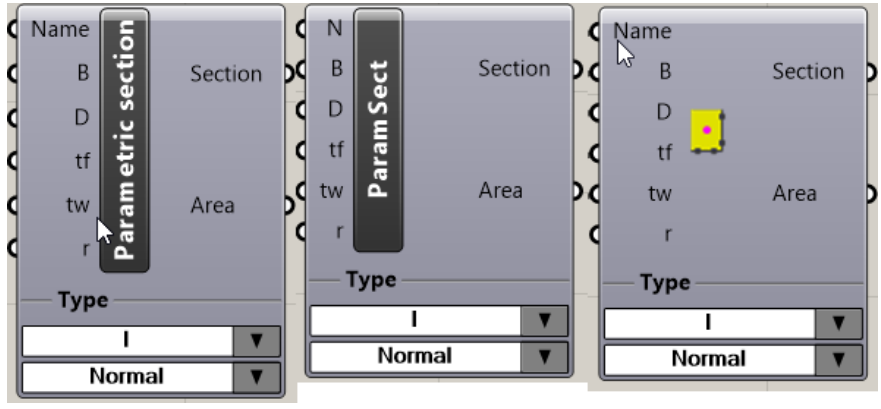
The parametric section component features two custom dropdown menus where you can select the type and subtype of the parametric section. When the selection changes, the component inputs apart from the 'Name' input are updated to reflect the selected type and subtype, while any already connected inputs will be preserved only for parameters that match in name and order with the previous selection. For example, in the scenario where the selection is changed from rectangular section (B, D) to rectangular hollow section (B, D, t, ri, to), the inputs to B, D parameters will be preserved.

The component also shows descriptive tooltips for each parameter and supports optional parameters, for example for the I section the r (radius) parameter.

Although most of the parametric sections have an available preview shape, the component will show an informative message if a preview is not possible.

The component also outputs the section area (converted to model units) in case it can be used for other calculations.

Inputs	Outputs
(Name) – *single line text*	Parametric section
Required parameters e.g. B, D – *numeric, > 0*	Area – *numeric*
(Optional parameters e.g. r) – *numeric, > 0, default: 0*	



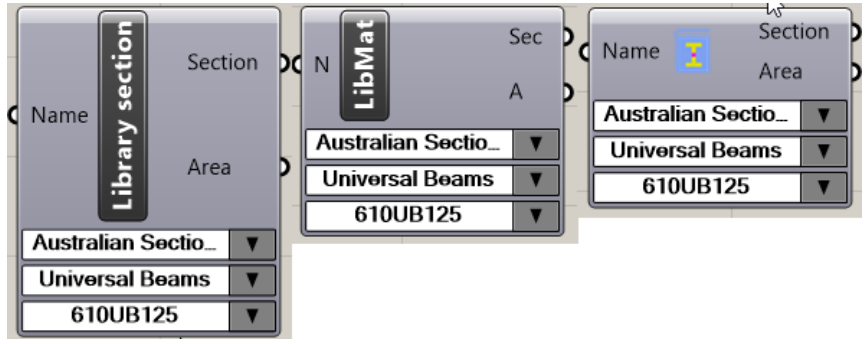
The parametric section component in all possible display modes; I section.

Library section

With the library section component you can select from the available sections in the LUSAS section library using the custom dropdown lists.

For any section type that does not have an available preview shape, the component will show an informative message.

Inputs	Outputs
(Name) – *single line text*	Library section Area – *numeric*



The library section component in all possible display modes.

Arbitrary section

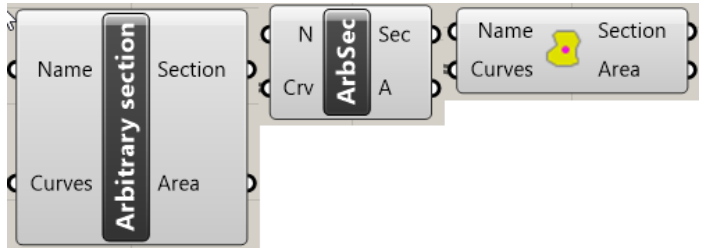
The arbitrary section component accepts a list of curves as an input, which define the shape of an arbitrary section. These curves need to form one or more closed loops in the XY plane, which will be trimmed in Modeller to create the section shape with any internal holes.

The component validates that the curves are planar in the XY plane and that they form closed loops. Otherwise, an error is reported, and no output is provided.

Section preview is by default available for any arbitrary section.

The component also outputs the section area (converted to model units) in case it can be used for other calculations.

Inputs	Outputs
(Name) – *single line text*	Arbitrary section
Curves – *native Rhino type*	Area – *numeric*

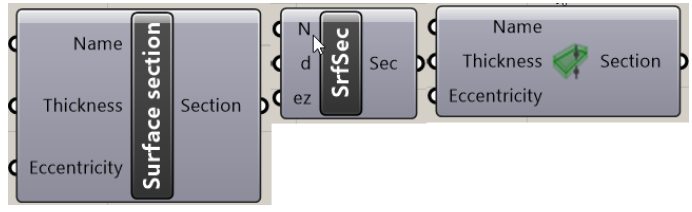


The arbitrary section component in all possible display modes.

Surface section

The surface section component is the equivalent of the surface geometric attribute in LUSAS. It allows for the definition of a surface's thickness and eccentricity.

Inputs	Outputs
(Name) – *single line text*	Parametric section
Thickness – *numeric, > 0*	
(Eccentricity) – *numeric, default: 0*	

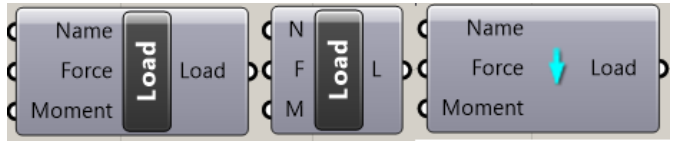


The surface section component in all possible display modes.

Load

The 'Load' component allows the definition of load attributes, by specifying a force and a moment 3D vector for the load values. The moment load values will be ignored if the load is assigned to a surface member.

Inputs	Outputs
(Name) – *single line text*	Load
(Force) – *3D numeric vector, default: (0,0,0)*	
(Moment) – *3D numeric vector, default: (0,0,0)*	

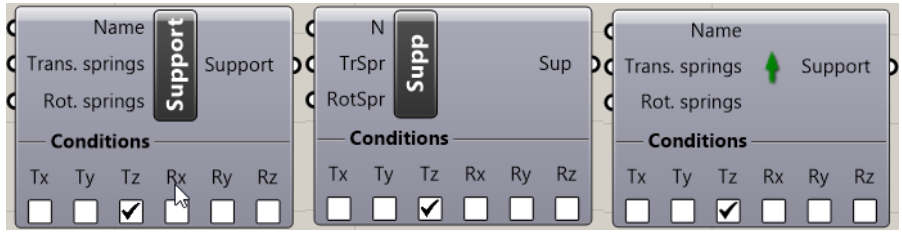


The load component in all possible display modes.

Support

The support component allows for the definition of a support attribute. Select the fixed degrees of freedom by enabling the respective checkbox controls. The 'Translation/Rotation springs' inputs each accept a 3D vector for the spring values. Any non-zero spring value will override the 'Conditions' controls and will force the respective degree of freedom to use the specified spring value.

Inputs	Outputs
(Name) – *single line text*	Support
Translation springs – *3D vector, numeric, > 0, default: (0,0,0)*	
Rotation springs – *3D vector, numeric, > 0, default: (0,0,0)*	



The support component in all possible display modes.

Assignments

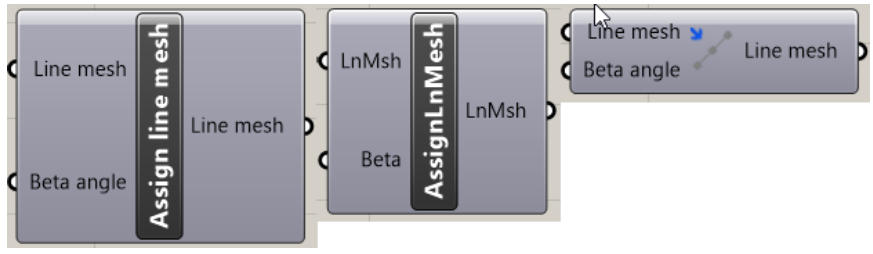
The assignment components sit between the attribute definition components and the inputs of member of model definition.

Assign line mesh

The line mesh assignment component accepts a line mesh and a beta angle as inputs, allowing you to customise the beta angle for specific members (in degrees). Use of this component is optional; the output of the line mesh component itself can be directly connected to a line member and assume a beta angle of 0.

This component will also check if the definition of a beta angle is accepted by LUSAS for the specific line mesh type. For example the following line mesh types do not accept a beta angle and will produce an error on the component: none, bar.

Inputs	Outputs
Line mesh – *output of the Line mesh component* Beta angle – *numeric*	Line mesh

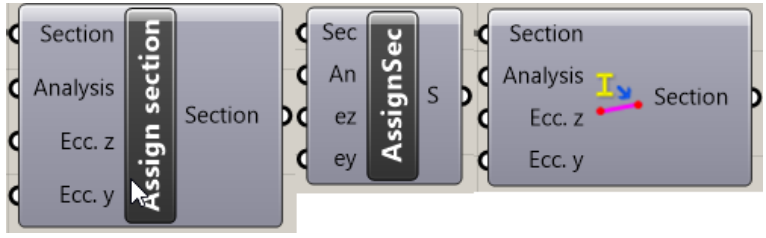


The assign line mesh component in all possible display modes.

Assign section/material

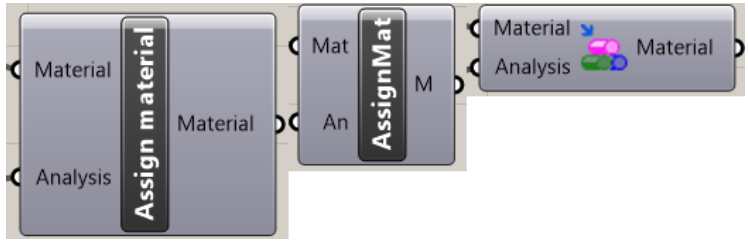
The section and material assignment components allow for specifying a specific analysis and or optional eccentricity values in the case of a section. Their use is optional, i.e. the output of the section or material components can be directly assigned to members assuming the base analysis of the model.

Inputs	Outputs
Section – *output of any Section component*	Section
Analysis – *single line text*	
(Ecc. z) – *numeric, default: 0*	
(Ecc. y) – *numeric, default: 0*	



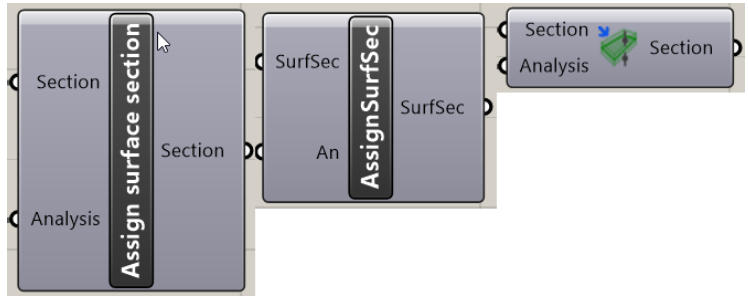
The assign section component in all possible display modes.

Inputs	Outputs
Material – *output of any Material component*	Material
Analysis – *single line text*	



The assign material component in all possible display modes.

Inputs	Outputs
Section – *output of the Surface section component*	Surface section
Analysis – *single line text*	



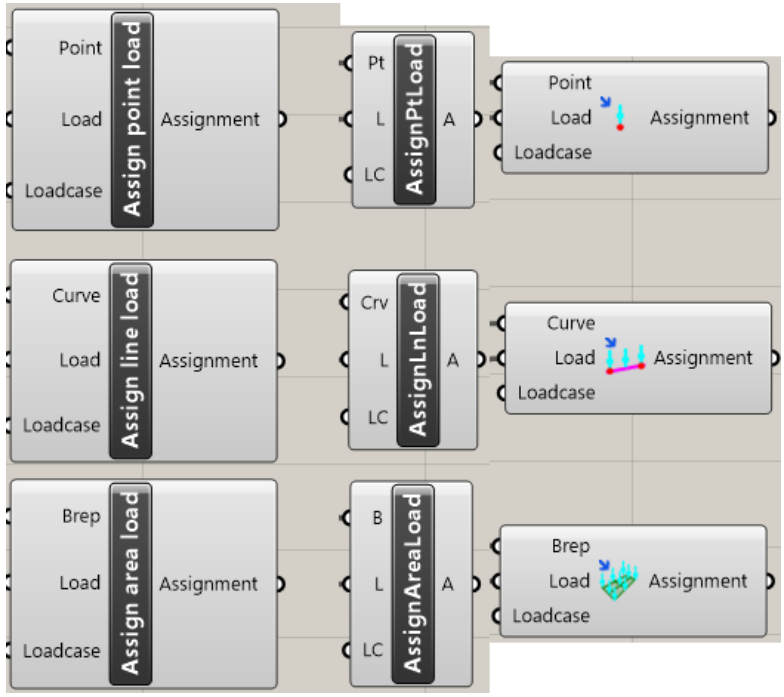
The assign surface section component in all possible display modes.

Assign point/line/area load

Loads are assigned by specifying the underlying geometry (points, curves or breps) which must match that of a corresponding line/surface member. This is necessary for point loads in order to know which point of a line member to assign them to, and also for line loads assigned to the edges of surface members.

Load assignment components also have the ‘Loadcase’ input, where you must specify a valid loadcase name, as is defined via an ‘Analysis’ component connected to the ‘Model’ component.

Inputs	Outputs
Point(s)/Curve(s)/Brep(s) – *native Rhino types*	Point/Line/Area load assignment
Load – *output of the Load component*	
Loadcase – *single line text*	



All assign load components in all possible display modes.

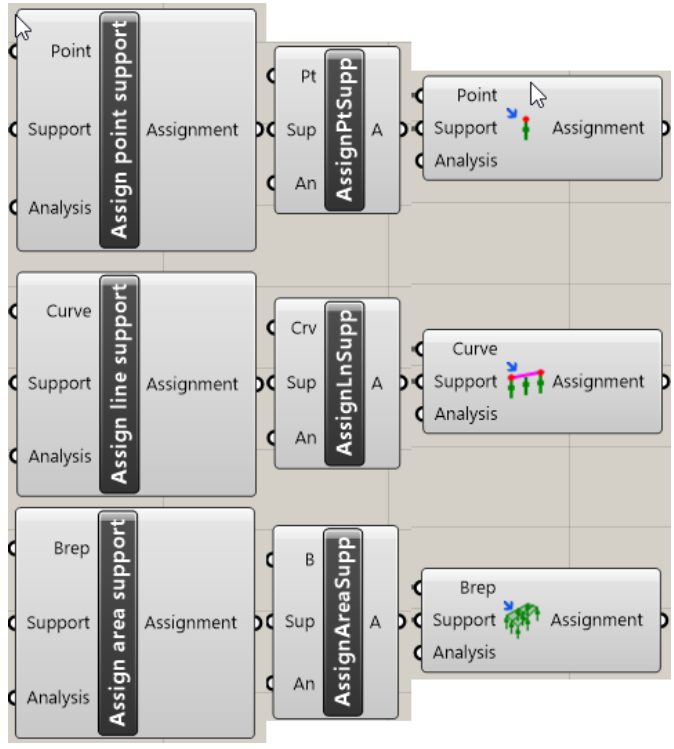
Assign point/line/area support

Supports are assigned by specifying the underlying geometry (points, curves or breps) which must match that of a corresponding line/surface member. This is necessary for point supports in order to know which point of a line member to assign them to, and also for line supports assigned to the edges of surface members. Line supports on line members do not require a curve specified.

Support assignment components also have the ‘Analysis’ input, where you must specify a valid loadcase name, as is defined via an ‘Analysis’ component connected to the ‘Model’ component.

Inputs	Outputs
Point(s)/Curve(s)/Brep(s) – *native Rhino types* Support – *output of the Support component*	Point/Line/Area support assignment

Analysis – *single line text*



All assign support components in all possible display modes.

Model components

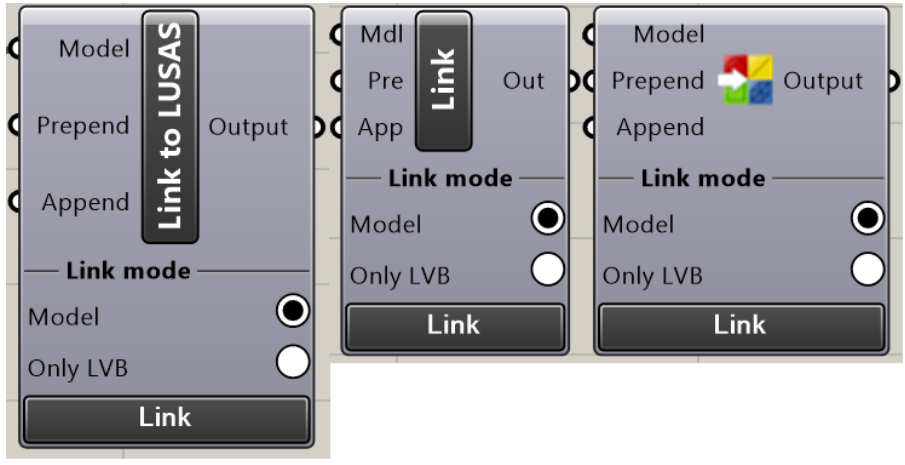
Link to LUSAS

This is the final component which converts the model data and opens the model directly in Modeller (if LUSAS is already installed) or creates a LUSAS LVB script file for loading into LUSAS at a later date. This component has a Link button which carries out the model export in the selected mode to the specified file path.

The ‘Link’ button only works if the component resolves its inputs without errors. You can also optionally prepend or append additional text to the final script by connecting text to the designated inputs.

The ‘Output’ output contains the LVB script which will be created by the ‘Link’ operation.

Inputs	Outputs
Model – *output of the LUSAS model component*	Output
(Prepend) – *multiple line text*	
(Append) – *multiple line text*	



The link component in all possible display modes.

LUSAS Model

The LUSAS model component assembles all of the model information: line/surface members, analyses and their loadcases/loads, load/support assignments and the model units.

The ‘Members’ input accepts both line and surface member inputs. If no members are supplied, the component reports an error as a model without members does not make any sense.

The ‘Analyses’ input accepts the output of one or more analysis components. If no data is supplied, the model is created with 1 analysis (‘Analysis 1’) and 1 loadcase (‘Loadcase 1’) by default. The loadcase names must be unique among all analyses. The base analysis is selected as follows:

- If any analysis component has the ‘Base analysis’ checkbox enabled, it will be marked as base.
- If multiple analysis components have the ‘Base analysis’ checkbox enabled, the first one in order will be assumed with a warning message.

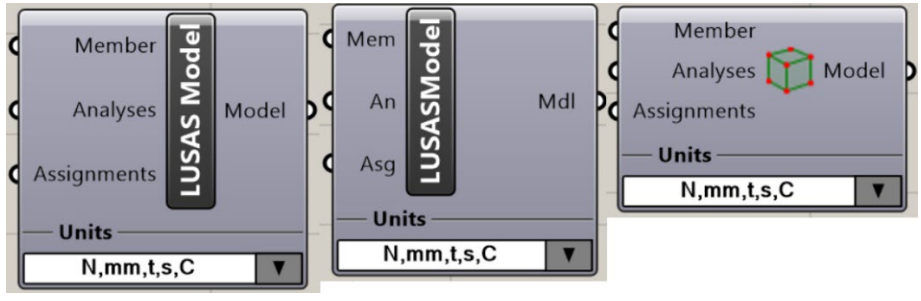
- If no analysis components have the ‘Base analysis’ checkbox enabled, the first one in order will be assumed.

The ‘Assignments’ input accepts the output of the load and support assignment components.

A dropdown list allows for the selection of the desired model units, which must be the same in Rhino and LUSAS. An error message is shown on the component if the units do not match, and no output is produced.

Note that in LUSAS attributes of the same type must have unique names. In case duplicate names are found, a unique name will be created automatically by appending a dash (-) and a number to the original attribute name.

Inputs	Outputs
Member(s) – *output of any Member component* (Analyses) – *output of the Analysis component, default: ‘Analysis 1’* (Assignments) – *output of any Assign load/support component*	LUSAS Model



The LUSAS model component in all possible display modes.

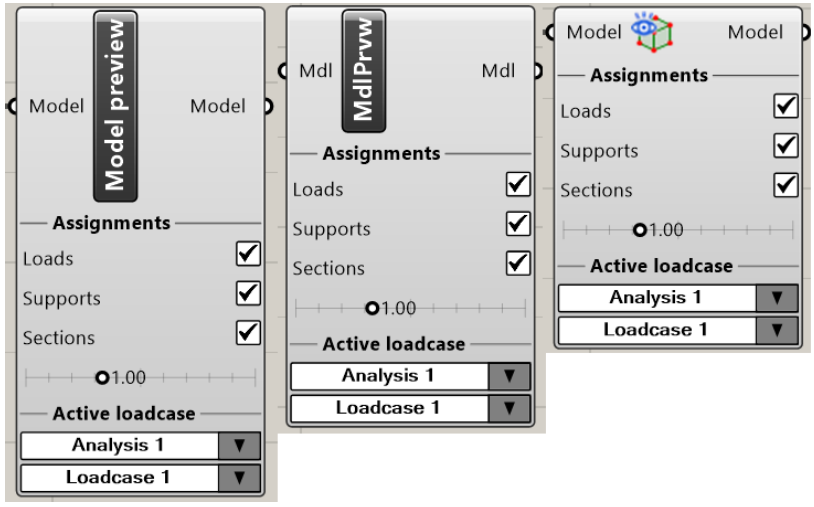
Model preview

The model preview component is responsible for the preview of member sections as well as any load and support assignments in Rhino’s viewport. This component must be used with an existing ‘LUSAS Model’ as input in order for any preview to take place. It outputs the ‘Model’ again so that it can be used between the ‘Model’ and ‘Link to LUSAS’ components, or standalone.

If features settings for the preview in the form of custom controls:

- The ‘Loads’/’Supports’/’Sections’ checkboxes can be used to toggle the respective previews altogether.
- The value slider can be used to change the scale of the load/support preview props in the viewport.
- The active analysis/loadcase dropdown lists can be used to select the analysis/loadcase from which to show sections/loads/supports.

Inputs	Outputs
Model – *output of the LUSAS model component*	Model



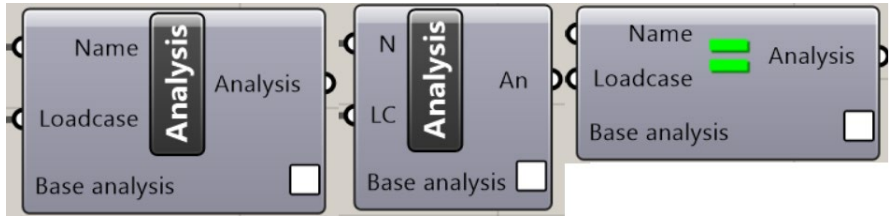
The model preview component in all possible display modes.

Analysis

The analysis component allows the definition of analyses for the LUSAS model. Each analysis has a ‘Name’ and a ‘Loadcase(s)’ input, where you specify the analysis name and any loadcase names to be associated with the analysis. The ‘Loadcase’ input accepts a list of unique loadcase names (either list of texts or multiple lines of text), in the order they will appear in the model. These loadcases must be referred to by name in the load assignment components.

The ‘Base analysis’ checkbox sets this analysis as the base analysis in the model.

Inputs	Outputs
Name – *single line text*	Analysis
Loadcase(s) – *single/multiple line text*	



The analysis component in all possible display modes.