SOFiSTiK

Getting started with

SOFiSTiK & Grasshopper

Parametric Modeling and Design

This guideline shows you the basics for creating your analytical model for SOFiSTiK using Rhino Grasshopper (a visual programming tool).

BIM neutral format! Grasshopper allows you to 3D model for other platforms like REVIT and TEKLA.

SOFiSTiK

FOREWORD

The goal of this document is to provide a clean and easy way of starting graphical programming to structural engineers. Rhino Grasshopper is a powerful tool to create any kind of geometry for structural analysis in SOFiSTiK. This includes parametrization, load application, cross section generation and much more.

This document will help you to understand the software and to streamline your workflow.

3 examples are included in this document. They will provide you a basic overview of the capabilities of Rhino Grasshopper and the possibilities it gives you by using it with SOFiSTiK

To follow the shown workflows in this document you need this software:

- SOFiSTiK Structural Desktop (via **SOFiSTiK Application Manager**)
- SOFiSTiK Rhinoceros Interface (via **SOFiSTiK Application Manager**)
- Rhinoceros (it includes Grasshopper)

Copyright

SOFiSTiK assures that the script was created to the best of its knowledge and beliefs but does not guarantee that the script is error-free. The user remains responsible for his own application.

Table of Contents

1 General

1.1 SOFiSTiK

SOFiSTiK is Europe’s leading manufacturer of construction software for analysis, design, and detailing. Over 2,000 customers worldwide use the company’s solutions to build large and complex infrastructure and buildings or implement special projects.

1.1.1 Program Structure

The center of all analysis data communication in SOFiSTiK is the Central Data Base (short: CDB). All SOFiSTiK programs can communicate with each other through writing and reading data to and from this database. The typical choice to handle SOFiSTiK´s programs is “SOFiSTiK Structural Desktop”, short: ”SSD“. It not only organizes the programs, but it also allows the programs to interact with each other.

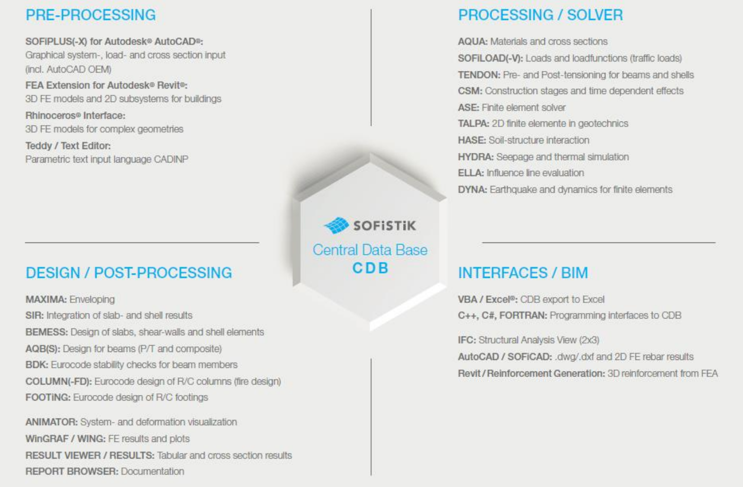
The program structure can be summed up in 4 categories.

Firstly, pre-processing for creating your system. SOFiSTiK has a lot of ways to input your system simply, because it allows you to input numerical data through TEDDY (SOFiSTiK´s text input). That means you can input information through programming languages EXCEL and so on. But, it also allows you to use graphical interfaces like AutoCAD, Revit and of course Rhino/Grasshopper.

Secondly, Processing for adding information to your structure, like loads, construction stages, prestressing, halfspace, earthquake and much more. This also includes various types of analysis like linear, material nonlinear, geometrical nonlinear,…

Thirdly, design and post-processing for retrieving your desired results from your structure and then presenting it nicely in text, lists and graphics.

Lastly, all the interfaces that can be applied, before and after SSD. All modules communicate through the CDB.



PRE-PROCESSING

SOFiPLUS(-X) for Autodesk® AutoCAD®:
Graphical system-, load- and cross section input
(incl. AutoCAD OEM)

FEA Extension for Autodesk® Revit®:
3D FE models and 2D subsystems for buildings

Rhinoceros® Interface:
3D FE models for complex geometries

Teddy / Text Editor:
Parametric text input language CADINP

PROCESSING / SOLVER

AQUA: Materials and cross sections
SOFiLOAD(-V): Loads and loadfunctions (traffic loads)
TENDON: Pre- and Post-tensioning for beams and shells
CSM: Construction stages and time dependent effects
ASE: Finite element solver
TALPA: 2D finite elemente in geotechnics
HASE: Soil-structure interaction
HYDRA: Seepage and thermal simulation
ELLA: Influence line evaluation
DYNA: Earthquake and dynamics for finite elements

SOFiSTiK
Central Data Base
CDB

DESIGN / POST-PROCESSING

MAXIMA: Enveloping
SIR: Integration of slab- and shell results
BEMESS: Design of slabs, shear-walls and shell elements
AQB(S): Design for beams (P/T and composite)
BDK: Eurocode stability checks for beam members
COLUMN(-FD): Eurocode design of R/C columns (fire design)
FOOTiNG: Eurocode design of R/C footings

ANIMATOR: System- and deformation visualization
WinGRAF / WING: FE results and plots
RESULT VIEWER / RESULTS: Tabular and cross section results
REPORT BROWSER: Documentation

INTERFACES / BIM

VBA / Excel®: CDB export to Excel
C++, C#, FORTRAN: Programming interfaces to CDB

IFC: Structural Analysis View (2x3)
AutoCAD / SOFiCAD: .dwg/.dxf and 2D FE rebar results
Revit / Reinforcement Generation: 3D reinforcement from FEA

1.1.2 TEDDY – Text Input Language (CADINP)

The most versatile pre-processing methods for SOFiSTiK is the TEDDY text input language. It is a simple programming language that allows you to apply and modify standard command with variables, loops and formulas. You can combine text tasks with other SOFiSTiK programs and pre-processing tools to generate your project. Generally, the data for pre-processing can be generated by graphical tasks (tasks in SSD, SOFIPLUS, Rhinoceros, Revit), or by TEDDY, but I these two options can also be combined.

The text input data can also be produced by other text editors like EXCEL spreadsheets, scripting languages or graphic scripting software like Grasshopper for Rhinoceros. Teddy is not the only way to create the text input.

1.2 Rhinoceros ®

Rhinoceros (also called Rhino, or Rhino3D) is a 3D computer-aided design (CAD) software developed by Robert McNeel & Associates. Rhinoceros geometry is based on the NURBS mathematical model, which focuses on producing mathematically precise curves and freeform surfaces (in contrary to polygon mesh-based applications). Rhinoceros is developed for Microsoft Windows and MAC OS. Grasshopper is a visual scripting language for Rhino.

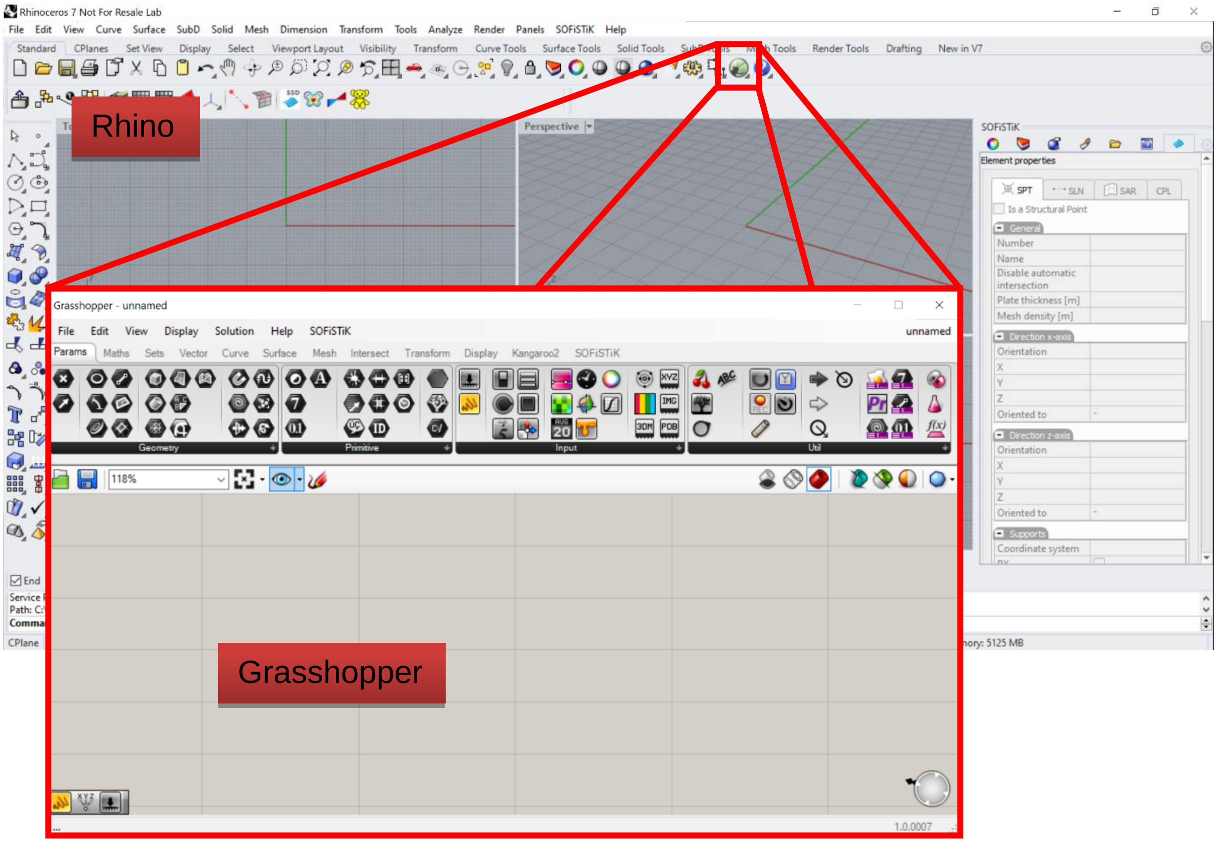
Rhinoceros software requires a license from Robert McNeel & Associates.

We will not take a closer look at Rhino in this document, as we will use it simply as a visual preview tool. There are added on SOFiSTiK functionalities available for Rhino, but the focus of this script is on the visual programming tool “Grasshopper”.

1.2.1 Grasshopper

Grasshopper is a graphical algorithm editor for Rhino’s 3D modeling tools. It is included in the Rhino installation since Rhino.

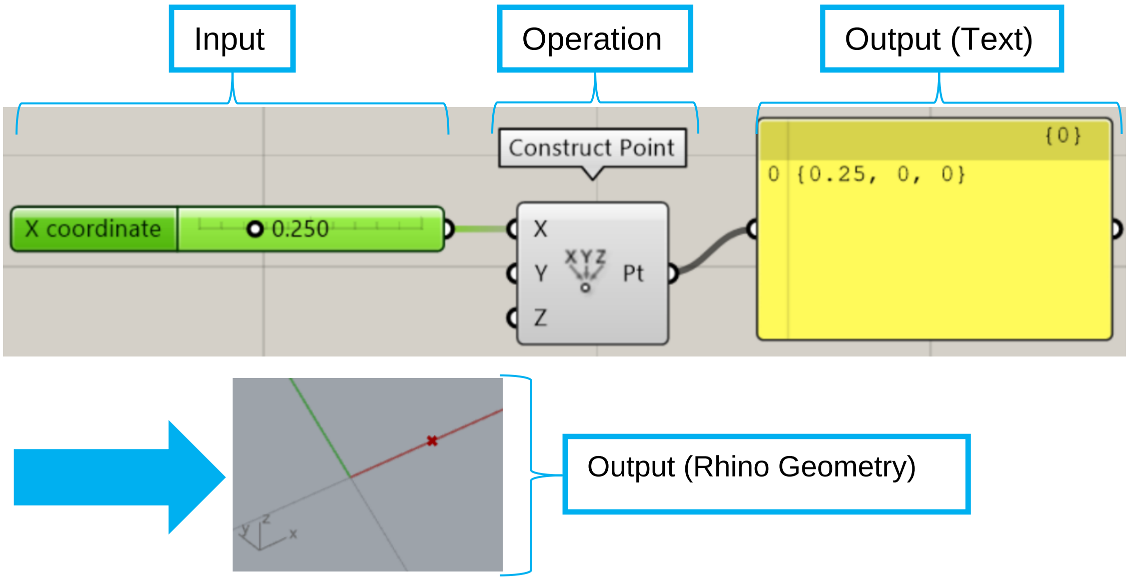
When you start Grasshopper in Rhino, it starts up in a separate window. Grasshopper includes a variety of functionalities to create your geometry via visual scripting. This Grasshopper script then creates or utilizes data in Rhino in real time. SOFiSTiK functionalities allow you to link the created data directly to SOFiSTiK for the analysis process.


Rhinoceros 7 Not For Resale Lab
File Edit View Curve Surface SubD Solid Mesh Dimension Transform Tools Analyze Render Panels SOFiSTiK Help
Rhino
Perspective
SOFiSTiK
Element properties
Is a Structural Point
Grasshopper - unnamed
File Edit View Display Solution Help SOFiSTiK
Params Maths Sets Vector Curve Surface Mesh Intersect Transform Display Kangaroo2 SOFiSTiK
Geometry
Primitive
Input
Util
118%
Grasshopper
1.0.0007

2 Basic Grasshopper Functionality

This chapter explains Grasshopper functions that allow you to create the analytical geometry of your structure. We will not discuss all the possibilities, but some that will lead you in the right way.

The basic principle of Grasshopper is to connect compatable **Input** to **Operations/functions** (connections via drag and drop) to achieve the geometric results you require **(Output)**.

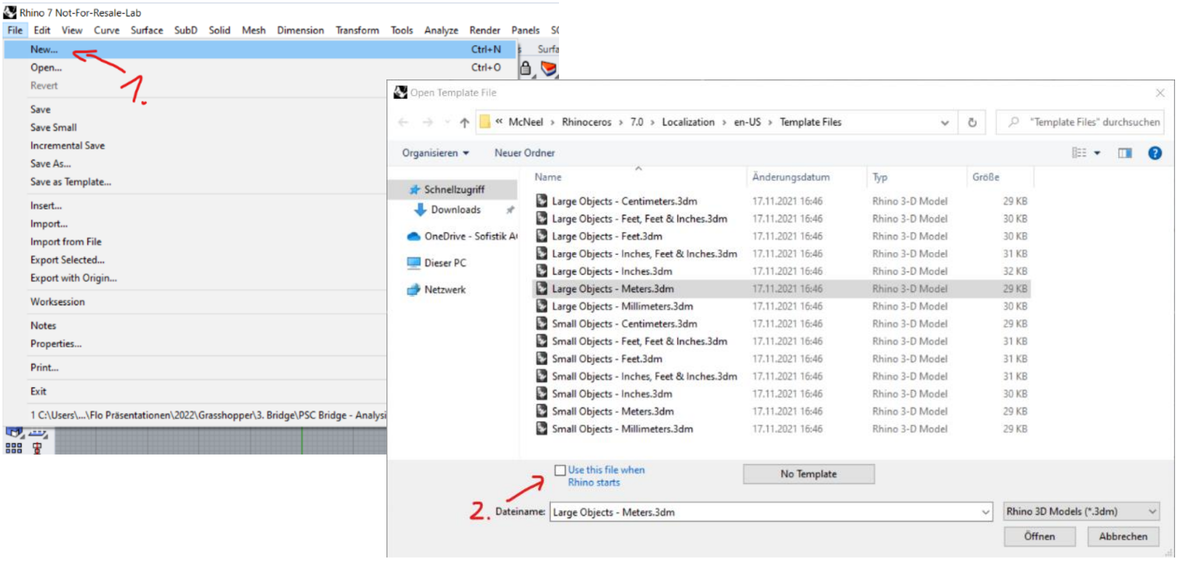


2.1 Before you start

To get the same representation as in the examples in this document, follow these next steps.

2.1.1 Template

The Rhino template used in this document is “Large Objects – Meters”. You can select it when you start a new project in Rhino (not in Grasshopper!).



Name	Änderungsdatum	Typ	Größe
Large Objects - Centimeters.3dm	17.11.2021 16:46	Rhino 3-D Model	29 KB
Large Objects - Feet, Feet & Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	30 KB
Large Objects - Feet.3dm	17.11.2021 16:46	Rhino 3-D Model	30 KB
Large Objects - Inches, Feet & Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	31 KB
Large Objects - Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	32 KB
Large Objects - Meters.3dm	17.11.2021 16:46	Rhino 3-D Model	29 KB
Large Objects - Millimeters.3dm	17.11.2021 16:46	Rhino 3-D Model	30 KB
Small Objects - Centimeters.3dm	17.11.2021 16:46	Rhino 3-D Model	29 KB
Small Objects - Feet, Feet & Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	31 KB
Small Objects - Feet.3dm	17.11.2021 16:46	Rhino 3-D Model	31 KB
Small Objects - Inches, Feet & Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	31 KB
Small Objects - Inches.3dm	17.11.2021 16:46	Rhino 3-D Model	30 KB
Small Objects - Meters.3dm	17.11.2021 16:46	Rhino 3-D Model	29 KB
Small Objects - Millimeters.3dm	17.11.2021 16:46	Rhino 3-D Model	29 KB

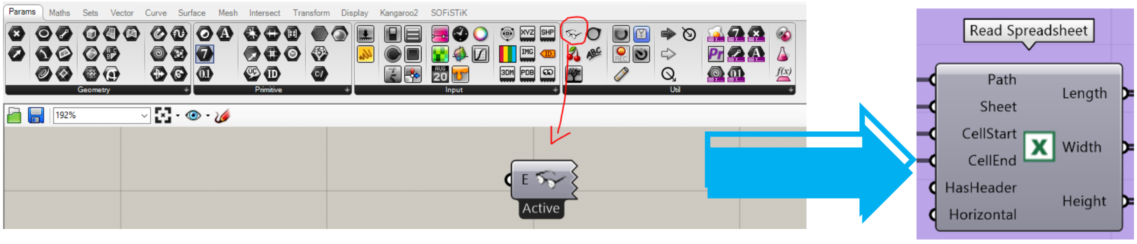
Dateiname: Large Objects - Meters.3dm

2.1.2 Bifocals

The best and fasts way to learn what you are handling in Grasshopper is to visualize, as well as read repeatedly the functions that you are using. To have the advantages of both worlds (icons+names) we recommend using the free Bifocals add-on for Grasshopper.

You can download it here:
https://www.food4rhino.com/en/app/bifocals

Once installed you can activate this add-on by clicking on these glasses and placing them in your project.

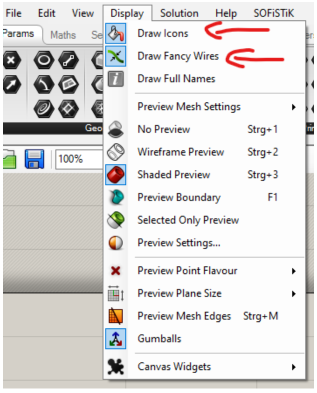


All your Icons now have little speech windows on top, that show their name.

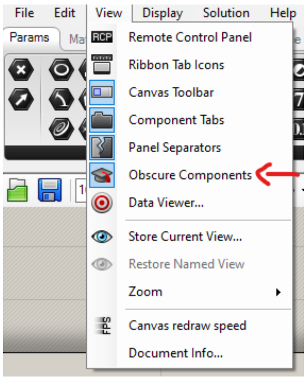
2.1.3 Display and View Settings

If you want to work with the same representation shown in this article, you have to select the following settings:

- **Draw Icons & Draw Fancy Wires in "Display"**

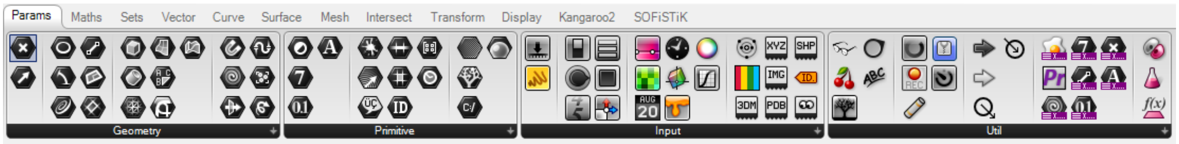


- **Obscure Components in "View"**

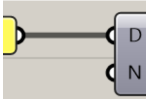


2.2 Parameters and Input

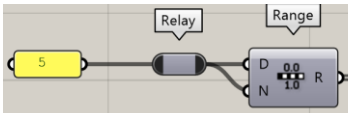
The first step in your Grasshopper journey always starts right here. The parameter tab. It contains simplified inputs.



Single data inputs are displayed as a full line between the operators:



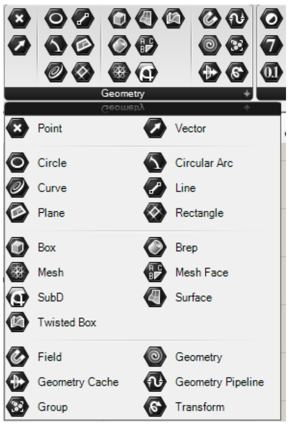
If you double click on this line, a relay is created, that can be used to clean up your project representation. It allows you to connect multiple inputs to a single output operator without creating long lines:

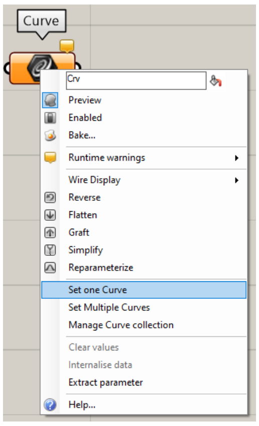


2.2.1 Geometry

Geometry allows you to input geometries from Rhino to Grasshopper. (from existing Rhino projects, or geometry that you drew in Rhino) Meaning, you can freely draw lines in Rhino and then use those lines in your Grasshopper coding.

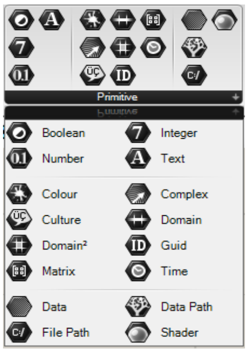
To do so, you have to place a function like "Curve" in your Grasshopper project. With a right click you can select the "set one Curve" option to select an existing curve from Rhino to use it in Grasshopper.

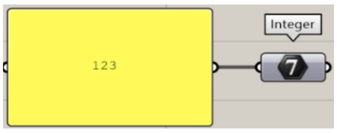


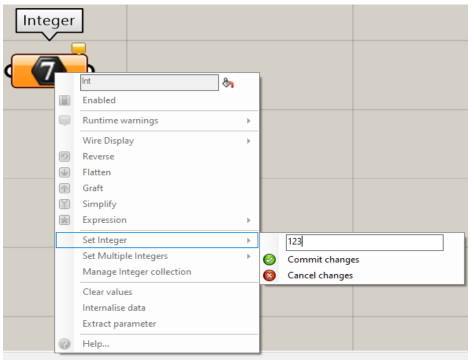


2.2.2 Primitive

The next input form is **Primitive**. It allows you to convert data into a specific form, or manually input data via right click on the function.

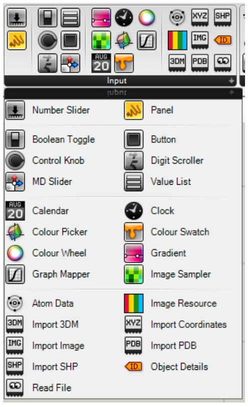






2.2.3 Input

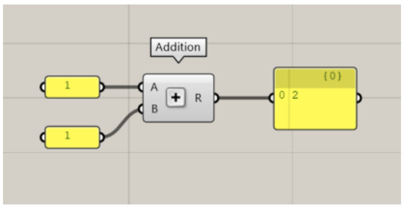
The most versatile methods to create data for your Grasshopper projects are found in **Input**.



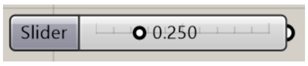
Input contains a wide variety to create/ import data. For this document you will require Panels, Number Sliders, Boolean Toggles and Value Lists.

- **Panel**

With a double click you can write you want to in it. If the information is easily assignable to a format (Text, Integers, Numbers,..) then the program will do so. Otherwise you can force the program to assign a format to it (see Primitive). It can not only be used for input, but also to display the output of an operation:

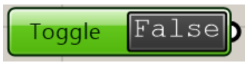


- **Number Slider**



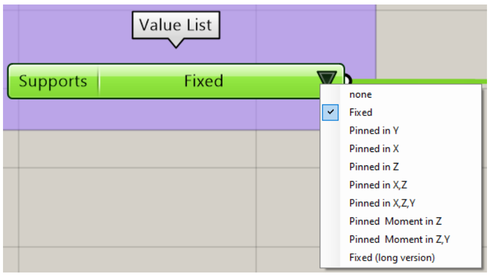
Number sliders can be used as a numerical input. You can adjust step intervals and minimum/maximum values in its right click menu.

- **Boolean Toggle**

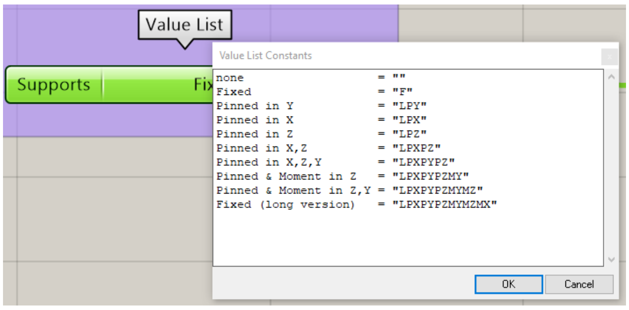


Is used for True/False inputs.

- **Value List**



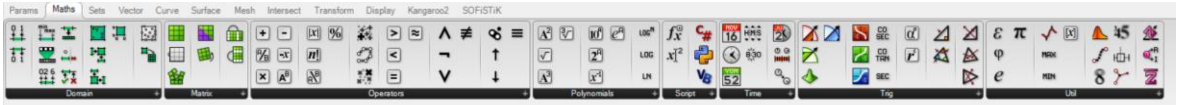
A value list allows you to create a list of automated inputs. You can personalize value lists with a double click on them:



```
none                    = ""
Fixed                   = "F"
Pinned in Y             = "LPY"
Pinned in X             = "LPX"
Pinned in Z             = "LPZ"
Pinned in X,Z           = "LPXPZ"
Pinned in X,Z,Y         = "LPXPYPZ"
Pinned & Moment in Z    = "LPXPYPZMY"
Pinned & Moment in Z,Y  = "LPXPYPZMYMZ"
Fixed (long version)    = "LPXPYPZMYMZMX"
```

2.2.4 Mathematics

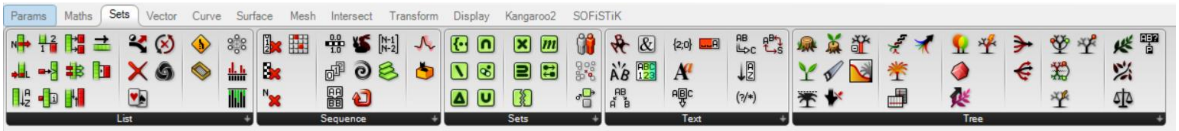
You will find all the operators to do complex mathematical operations in this tab.



You can even create whole formulas via Script – Expression.

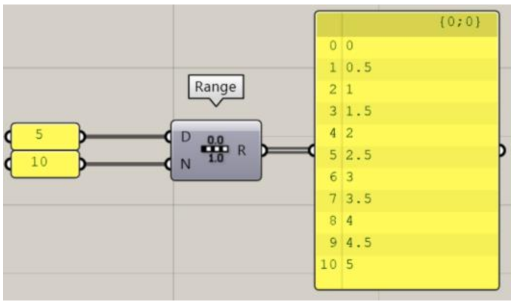
2.3 Sets

Lists and especially trees really enable complex geometries in Grasshopper.



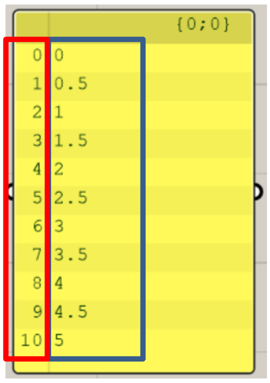
2.3.1 Lists

Lists are exactly what you expect, a list of information. One information after another.



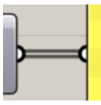
```
{0;0}
 0 0
 1 0.5
 2 1
 3 1.5
 4 2
 5 2.5
 6 3
 7 3.5
 8 4
 9 4.5
10 5
```

There are lots of ways to create lists (like the Range command), as well as to manipulate lists in the “List” and “Sequence” tabs. They all depend on the numbering scheme in Grasshopper. The index numbering starts at 0 and increases in increments of 1 (red). Then the assigned value of the list to this index is shown (blue)



```
{0;0}
 0 0
 1 0.5
 2 1
 3 1.5
 4 2
 5 2.5
 6 3
 7 3.5
 8 4
 9 4.5
10 5
```

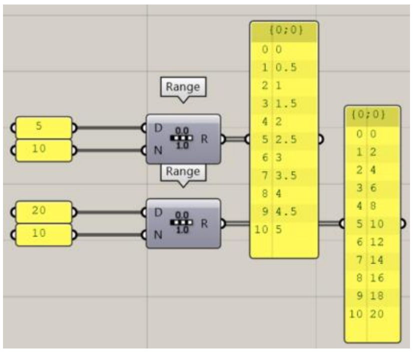
An indication, that you are working with a list can also be found in the wire style of the connection (double line):



2.3.2 Trees

A tree is a list of lists. Meaning a tree contains multiple lists.

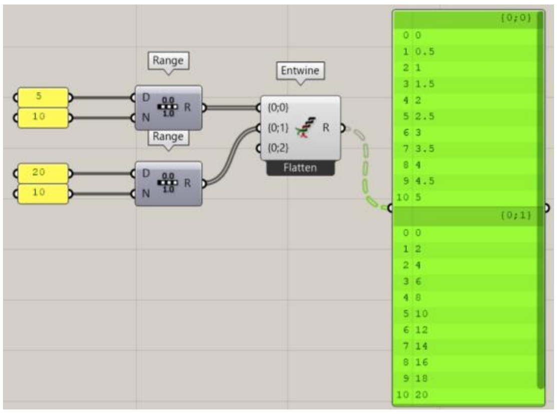
Let’s take a look at these two lists:



```
{0;0}
0 0
1 0.5
2 1
3 1.5
4 2
5 2.5
6 3
7 3.5
8 4
9 4.5
10 5
```

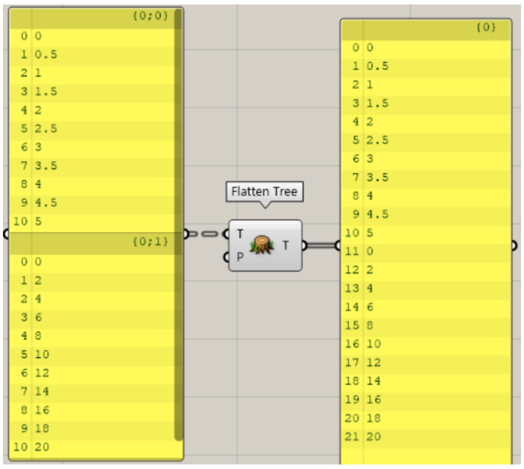
```
{0;0}
0 0
1 2
2 4
3 6
4 8
5 10
6 12
7 14
8 16
9 18
10 20
```

You can use the tree-command “Entwine” to create a tree containing both lists. Note, that every list still has its own indexes, as well as a tree index on top:



```
{0;0}
0 0
1 0.5
2 1
3 1.5
4 2
5 2.5
6 3
7 3.5
8 4
9 4.5
10 5
{0;1}
0 0
1 2
2 4
3 6
4 8
5 10
6 12
7 14
8 16
9 18
10 20
```

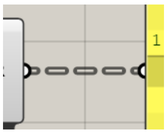
Tree-commands allow you to create trees, manipulate them and even create lists based on tree information. Tree information is indicated by a dashed double line. The tree-command “Flatten Tree” was used to transform the tree into a list:



```
{0;0}
0 0
1 0.5
2 1
3 1.5
4 2
5 2.5
6 3
7 3.5
8 4
9 4.5
10 5
{0;1}
0 0
1 2
2 4
3 6
4 8
5 10
6 12
7 14
8 16
9 18
10 20
```

```
{0}
0 0
1 0.5
2 1
3 1.5
4 2
5 2.5
6 3
7 3.5
8 4
9 4.5
10 5
11 0
12 2
13 4
14 6
15 8
16 10
17 12
18 14
19 16
20 18
21 20
```

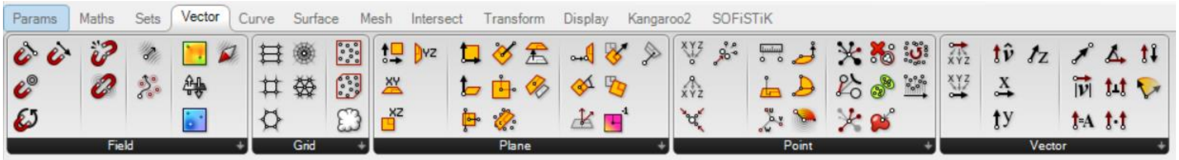
You can recognize tree information by its dashed double line:



2.4 Geometry Creation

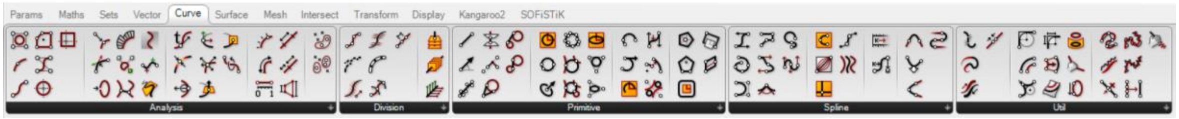
The aim of this document is to show you how to use Rhino Grasshopper for your structural analysis in SOFiSTiK. That means your main tools to create geometry will be Points, Lines and Areas.

2.4.1 Points (Vector)



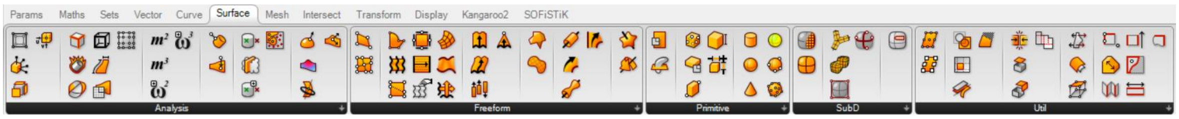
You can find operators to create and disassemble points in the Point-tab. There are also functions to create point-based operations like vectors, that can also be used to give directions to loads.

2.4.2 Lines (Curve)



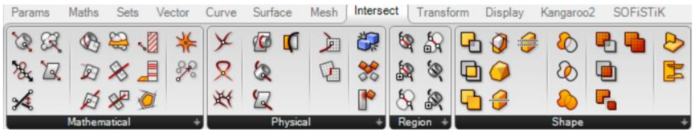
You can find lot of ways to create and manipulate lines in Curve. You will find the first couple of operators in the Primitive and Spline tabs most useful to create even very complex geometries.

2.4.3 Areas (Surface)



Surfaces can be created based on lots of different data input. They range from points and lines to trees and extrusions. There are also many transformations and interactions available in Surfaces. The most practical operators can be found in Freeform.

2.4.4 Intersect



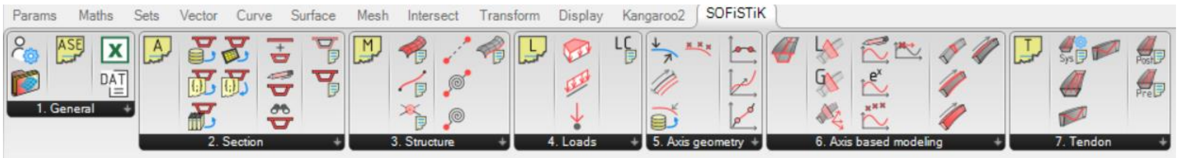
A special notice for the Intersect-tab, as it allows you to analyze intersections between your geometries. You can find these intersection operators in “Shape”.

3 Rhino Interface – SOFiSTiK Workflow

With the Rhino Interface you can directly use your points, Lines and Surfaces as Structural Points, Structural Lines and Structural Areas in your analysis, as well as many more functions.

3.1 Overview

This chapter gives you a rough overview of SOFiSTiK functionality for Grasshopper. (Version 0.0.95 for Rhino 7 and SOFiSTiK FEA 2022)



Most SOFiSTiK components are based on their respective modules in the SOFiSTiK-World. These **modules** are implemented directly in Grasshopper, you can recognize them by their **yellow sticker**-style. All **SOFiSTiK functions must be connected to a module**, before you can use them in your SOFiSTiK Analysis.

Here a short summary of the modules and what you can connect to them:

- **Advanced Solving Engine**

Allows you to do define and initiate a linear analysis, based on existing loadcases in your system. The required input is a list with loadcase-numbers.

- **AQUA**

Standard, Materials and Cross Sections. The functionalities in “2.Section” give you a multitude of options to input your Cross Section, as well as adjusting them. You can also input material and code information as a TEDDY-ASCII text.

- **SOFiMSHC**

This module generates the finite element mesh. You can connect all structural information like points, lines and areas to it. This includes axis-based geometry, as well as couplings, springs and boundary conditions.

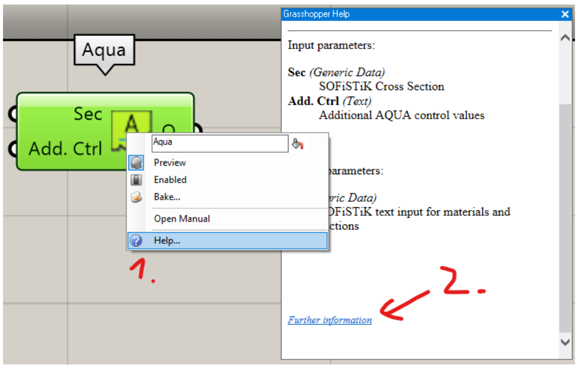
- **SOFiLOAD**

Load application. This module takes the information from your load definition and applies it to the meshed system. The input can be loads, as well as load attributes.

- **TENDON**

  Prestress. It lets you define the prestressing system, pre- and posttensioning attributes, as well as geometry definition, based on an axis geometry.

If you need additional information for the individual operators, you can get it from SOFiSTiK´s online documentation. Right click the operator in Grasshopper, “Help” and click on “Further Information” to access it directly:



Input parameters:

Sec *(Generic Data)*
SOFiSTiK Cross Section
Add. Ctrl *(Text)*
Additional AQUA control values

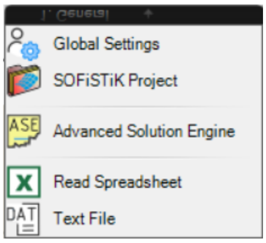
parameters:

ric Data)
DFiSTiK text input for materials and
ctions

Further information

3.1.1 General

Here a short overview of the operators in the “General”-tab, as not all of them are used for the same workflows. These functions are used for input/output, settings as well as calculation.



- **Global Settings**

Let’s you adjust colors, densities and scales for the analytical representation.

- **SOFiSTiK Project**

One way to link Grasshopper to your SOFiSTiK Project. You can manually initiate the export, stream to a .dat-file that can be read by SOFiSTiK and set it to silent calculation (manual, but without opening an additional calculation window). The buttons below are useful shortcuts to e.g. open the current file folder, the database visualization,…

- **Advanced Solving Engine**

Allows you to do define and initiate a linear analysis, based on existing loadcases in your system. The required input is a list with loadcase-numbers.

- **Read Spreadsheet**

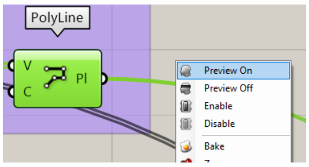
Read data from a spreadsheet and use it directly in Grasshopper. This will be always up to date (as soon as you save changes in your spreadsheet).

- **Text File**

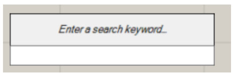
Another way to link Grasshopper to your SOFiSTiK Project. Text File creates .dat files for every SOFiSTiK-Module you assign to it.

3.2 Examples

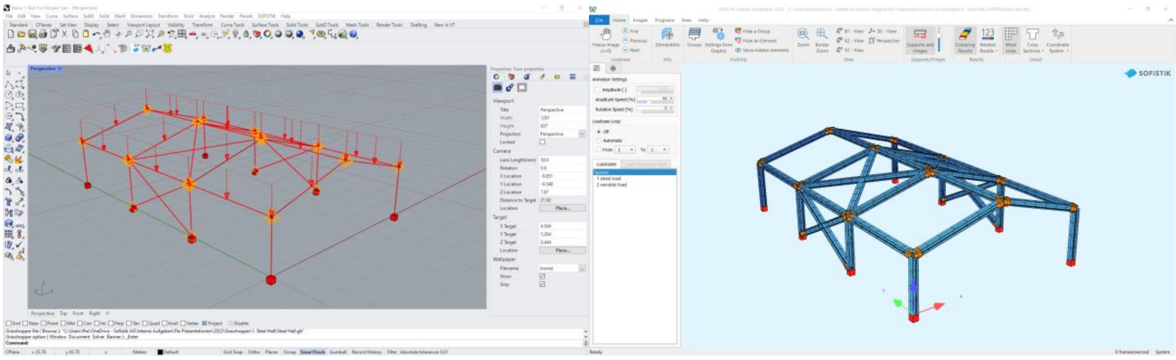
Make sure to visualize the information you want to see in Rhino. Right click the operator, that you want Rhino to show, in Grasshopper and turn its preview on and off.

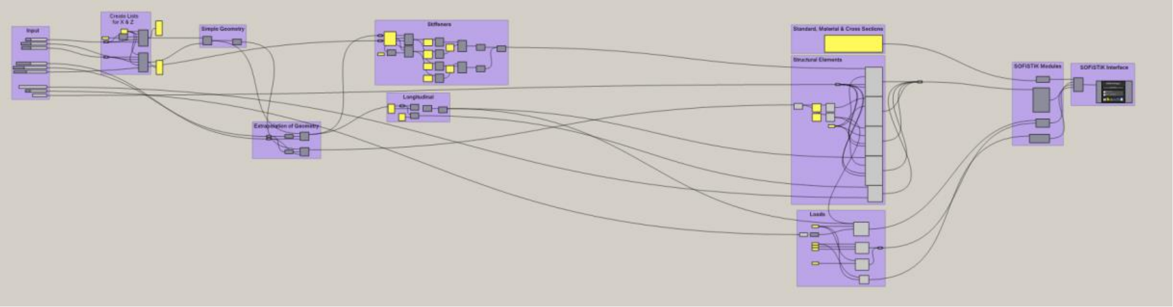


Useful Trick: Double left clock in an empty space in Grasshopper to access the text input. Insert the name of a function to quick access it.



3.3 Steel Hall





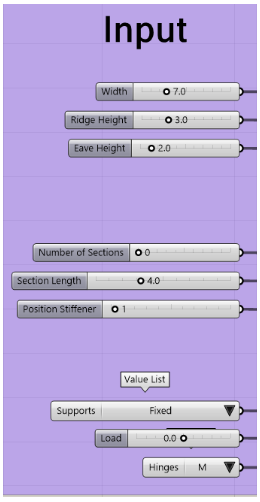
3.3.1 Grasshopper Operations

- **The Input**

The input for your geometry definition is usually based on Grasshopper operators (see 2.2).

In this project the first three inputs define a section of the steel hall. The second cluster is used as an input for the extrapolation of this section along the Y axis of the project. The last cluster can then be used to define supports, loads and hinges.

You should always think about which inputs you want to parametrize and set aside as a variable input.


Input

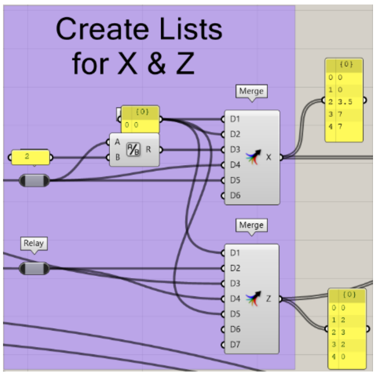
Width 7.0
Ridge Height 3.0
Eave Height 2.0

Number of Sections 0
Section Length 4.0
Position Stiffener 1

Value List
Supports Fixed
Load 0.0
Hinges M

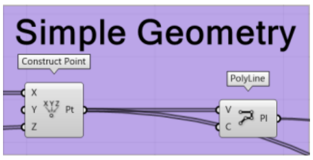
- **Creating Lists**

The first three input lines are now used to create 5 entries in a list for X and Y coordinates to later create points.



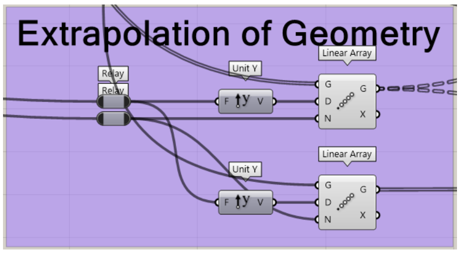
- **Section Geometry**

The X and Z lists can now be input to a construction point (see 2.4.1). Then a polyline (see 2.4.2) is created, based on the point information.



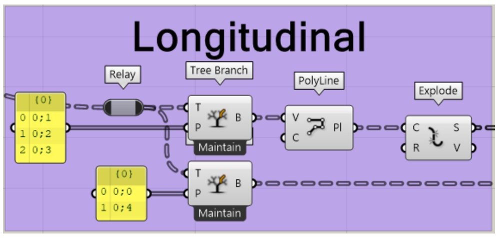
- **Extrapolating Points and Lines**

The points and Lines from the simple geometry can be copy pasted in a direction using “Linear Array”. In this example the direction is preset as an offset in Y-direction via a Y-Vector input.

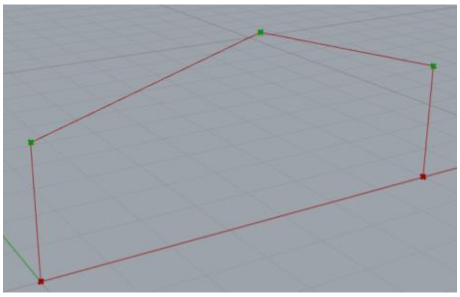


- **Connecting the Sections**

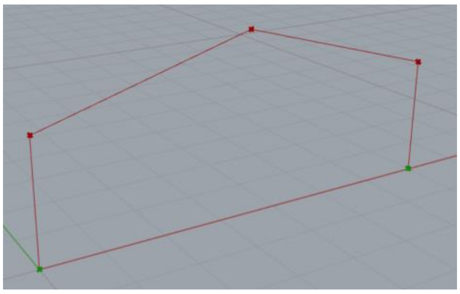
The longitudinal sections are created based on the extrapolated points.



The “Tree Branch” operator lets you select and split up the point tree to select only the points on top for your longitudinal connections.



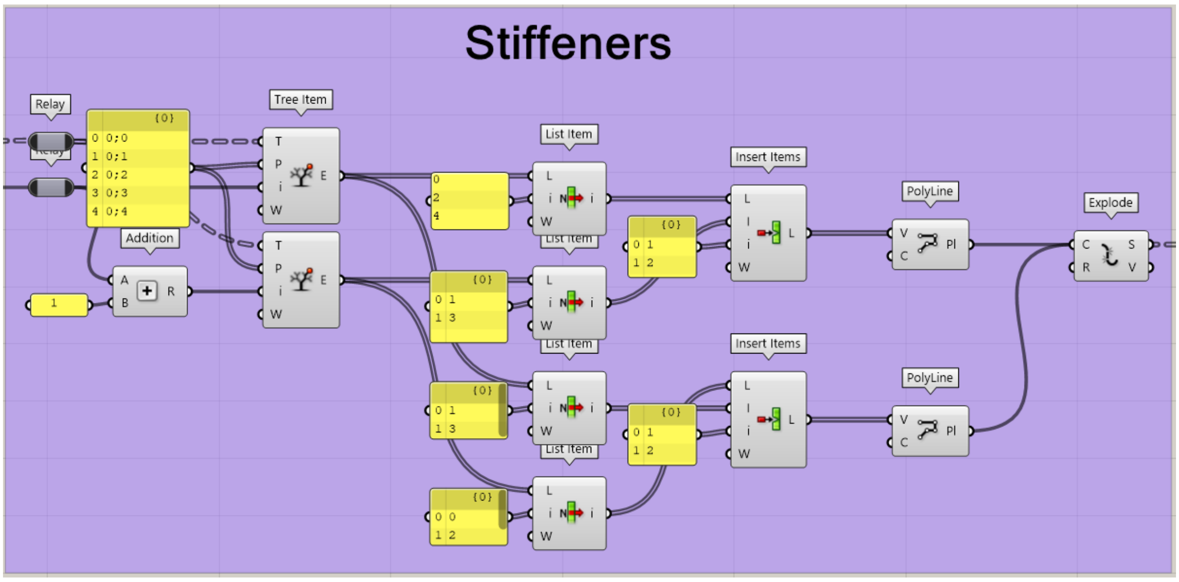
As well as the bottom points for your supports.



- **Stiffeners**

The operations used in Stiffeners firstly select alternating points. Secondly, these points are used to create alternating lines fitting the rest of the geometry. All of this is based on Grasshopper list manipulation.

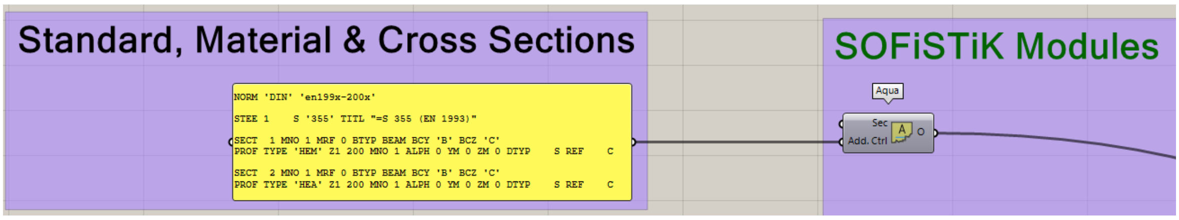
This is a complex operation to parametrize. But you can control it via “Position Stiffener” in your input.



3.3.2 SOFiSTiK Operations

- **Standard, Material & Cross Sections (AQUA)**

The method used in this example to input this information is via SOFiSTiK-TEDDY text input. This can be input via Panel to the “Additional Control” of AQUA. TEDDY input can be used for all SOFiSTiK modules in Grasshopper.



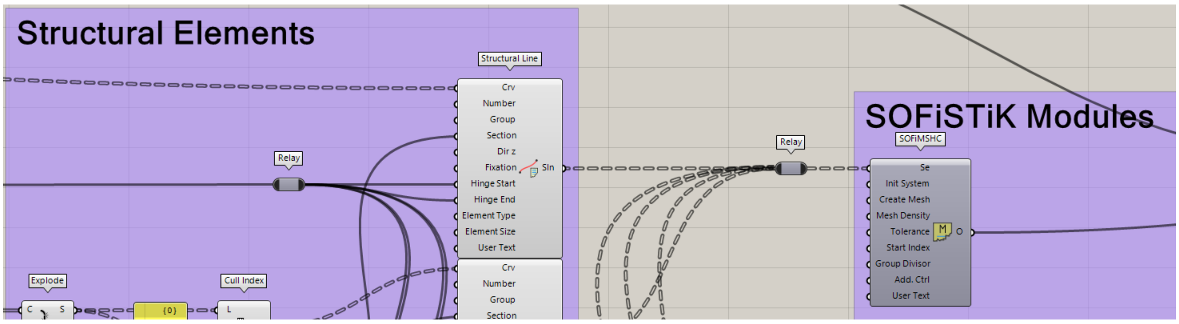
```
NORM 'DIN' 'en199x-200x'

STEE 1    S '355' TITL "=S 355 (EN 1993)"

SECT  1 MNO 1 MRF 0 BTYP BEAM BCY 'B' BCZ 'C'
PROF TYPE 'HEM' Z1 200 MNO 1 ALPH 0 YM 0 ZM 0 DTYP    S REF    C

SECT  2 MNO 1 MRF 0 BTYP BEAM BCY 'B' BCZ 'C'
PROF TYPE 'HEA' Z1 200 MNO 1 ALPH 0 YM 0 ZM 0 DTYP    S REF    C
```

Another method to set Standard, Material & Cross Sections is to not define them in Grasshopper, but later in the SOFiSTiK Analysis (SSD). You would still reference materials and cross sections by their numbers in Grasshopper, but you define them in SSD. The information comes together once you export the system from Grasshopper.

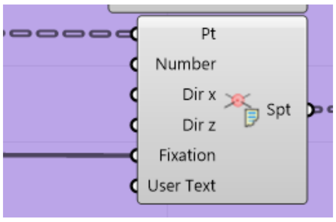
- **Structural Elements**

In this example we only have structural point and line elements. SOFiSTiK provides operators for grasshopper to define them. The input you provide what the structural element will do.

The geometry of the steel hall is solely defined by structural lines. You can input which Grasshopper lines shall be considered via “Crv”. You can assign beam information to the structural line by inputting it via “Section”. In this case this is simply done by a numerical input “1”, as Section “1“ is already defined in AQUA. As this is a steel hall, we want a couple of these beams to not transfer a momentum. This can be assigned via “Hinge Start” and “Hinge End”.



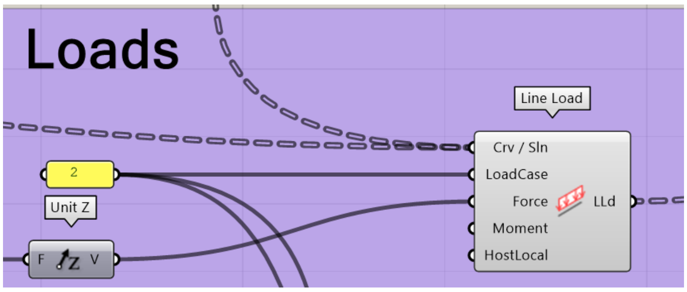
The structural points in this project are used to create support conditions. Simply select the relevant points and set a fixation degree.



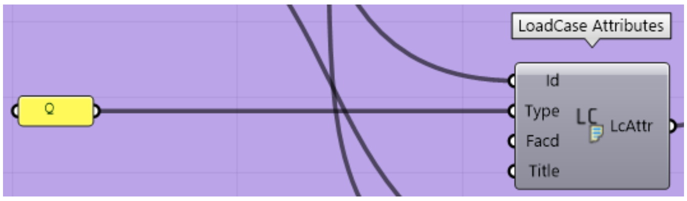
Once you finished your structural elements, make sure to connect it to the SOFiMSHC-module. Connected elements will be meshed and finite elements are generated for your analysis.

- **Loads and Analysis**

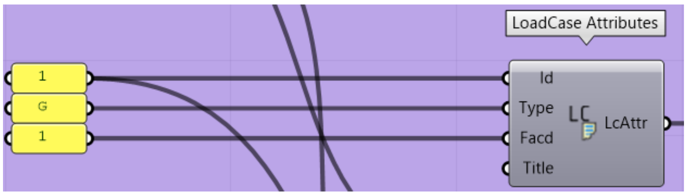
One way of applying loads to your system is via reference to a Grasshopper line. Use the SOFiSTiK "Line Load" operator and select the lines via "Crv/Sln" that shall be loaded. Assign a loadcase via "LoadCase". Forces and moments are input via vector. Meaning, that you can use a number slider input and convert it to a vector (see 2.4.1).



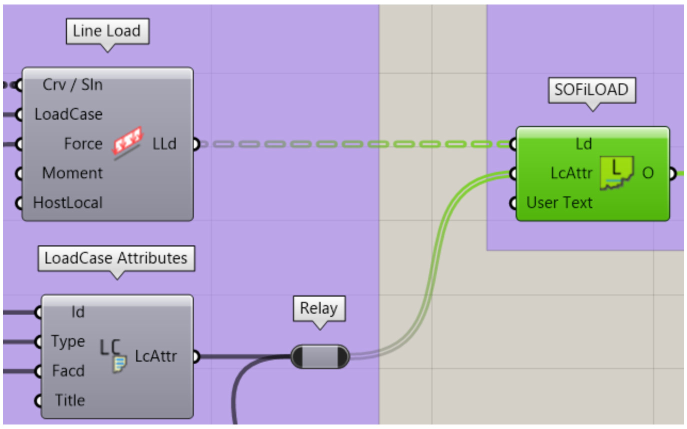
The loadcase then still needs some more attributes for the analysis. You can add them via "LoadCase Attributes". This allows you to assign a type to your loadcase.



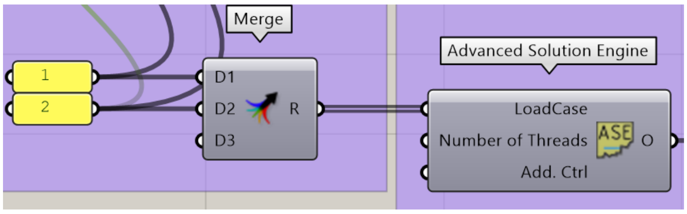
“Facd” lets you automatically calculate the selfweight of a structure and assign it to a loadcase. By setting Facd to 1 the selfweight for every structural element is calculated and will be available in loadcase 1. There is no need to additionally apply this loadcase as a load to the structure.



Applied loads and attributes are then assigned to the SOFiLOAD-module:

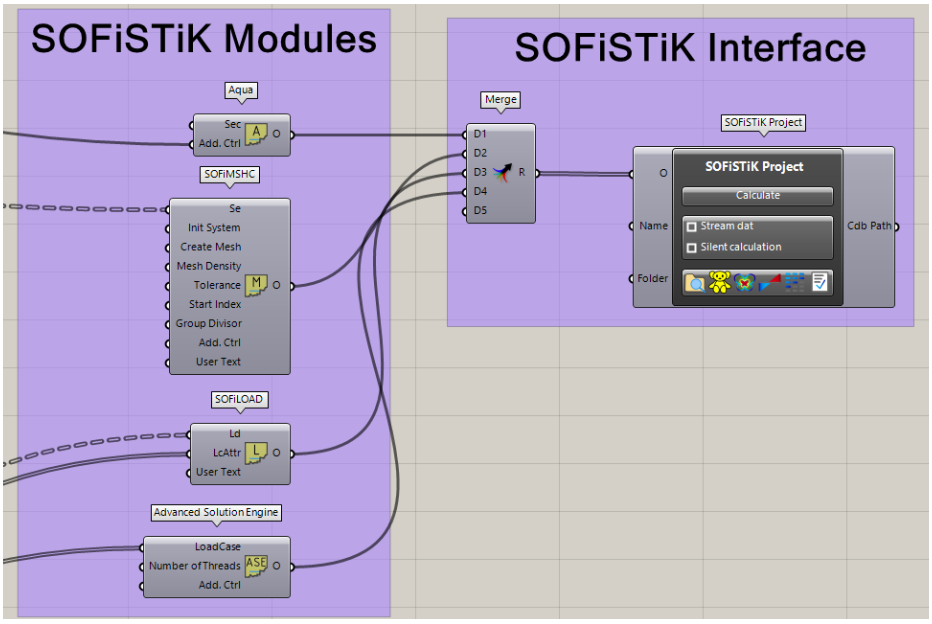


To initiate a linear analysis based on your loadcases, you can simply create a list of your loadcases and connect them to the ASE-module.

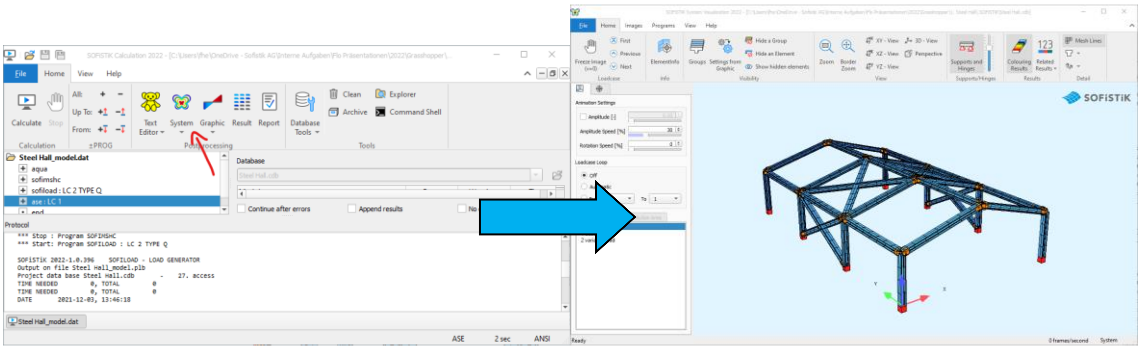


- **Initiating your SOFiSTiK Export**

Once you fed all necessary information about your project to the SOFiSTiK modules, you can create a list based on them with the Merge-operator. This way you can tell the export which module shall be run after another in your SOFiSTiK Project. Once merged, it is time to connect it to the SOFiSTiK Project operator.



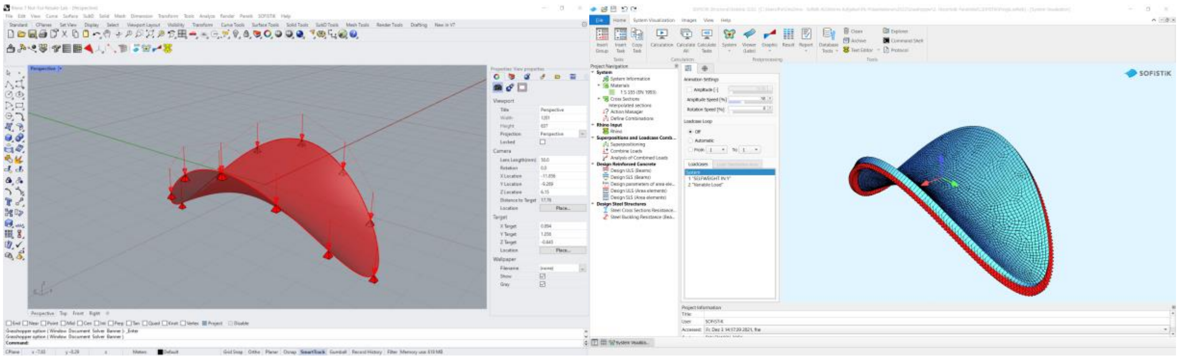
Now you can simply click the calculate button and see the magic happen. SOFiSTiK´s calculation opens up, the modules are calculated and the results are written in the SOFiSTiK Database. You can visualize the content of your database via “System”:

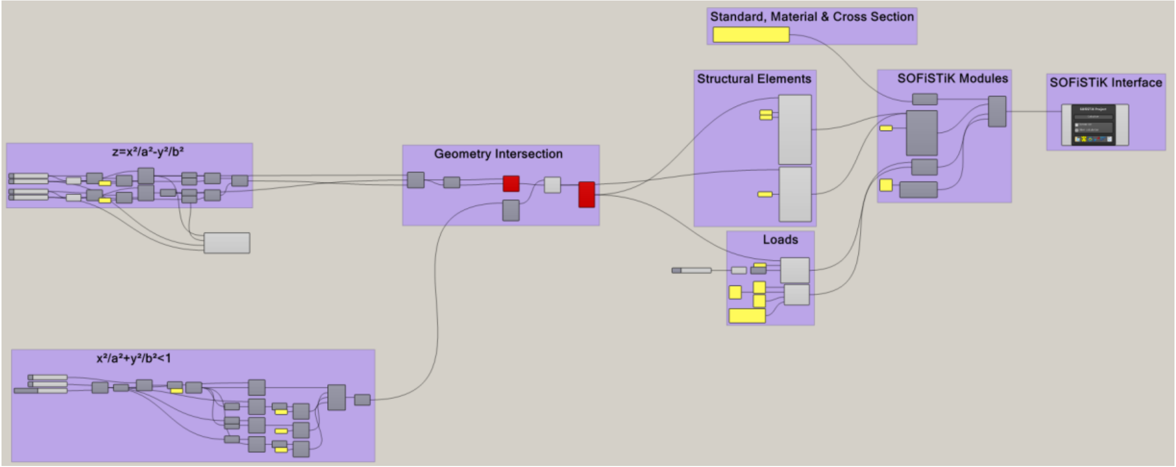


```
*** Stop : Program SOFIMSHC
*** Start: Program SOFILOAD : LC 2 TYPE Q

SOFiSTiK 2022-1.0.396    SOFILOAD - LOAD GENERATOR
Output on file Steel Hall_model.plb
Project data base Steel Hall.cdb      -    27. access
TIME NEEDED         0, TOTAL         0
TIME NEEDED         0, TOTAL         0
DATE       2021-12-03, 13:46:18
```

3.4 Hyperbolic Paraboloid

This example is about showing you freeform capabilities of Grasshopper, creating an area-based analytical model and a slightly different export method to the steel hall.





3.4.1 Grasshopper Operations

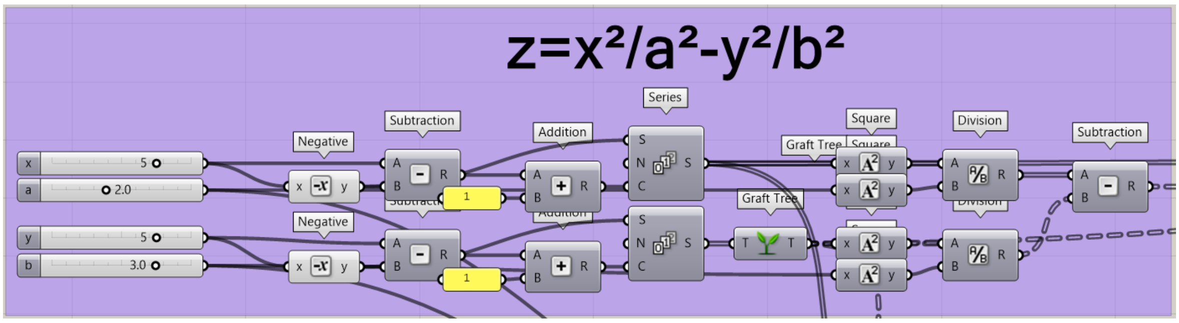
- **Mathematical Operations**

One of Grasshoppers biggest strengths is that you have access to all the tools you need to master even the most complex geometries rather simple.

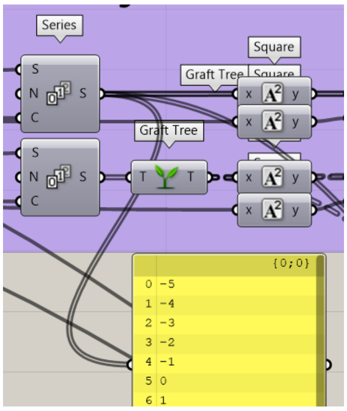
The geometry of this project is based on two math-based geometries. The first one is a hyperbolic paraboloid which can be describes with this formula:

$$z = \frac{x^2}{a^2} - \frac{y^2}{b^2}$$

You can use mathematical operations (see 2.2.4) to describe this formula in Grasshopper:

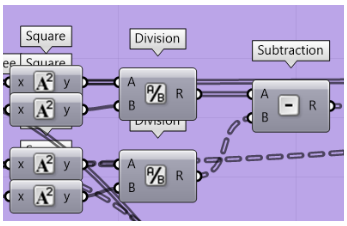


Grasshopper needs a range of input values to determine the boundaries within the geometry will be created. In this example the preset of x&y is set to 5. This is what is happening up to “Series”. If you connect a panel to the output of Series, you can see that a list of input values is created (range -5 to 5, step interval =1):

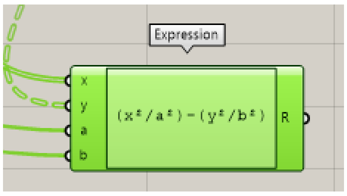


```
{0;0}
0 -5
1 -4
2 -3
3 -2
4 -1
5 0
6 1
```

You can now either use the distinct mathematical operators to describe your formula:

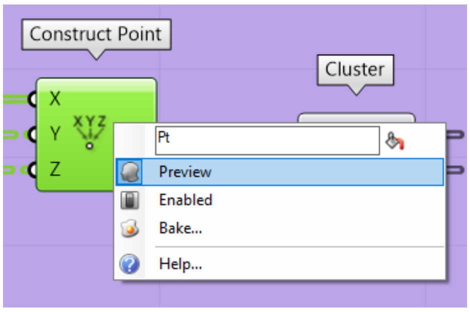


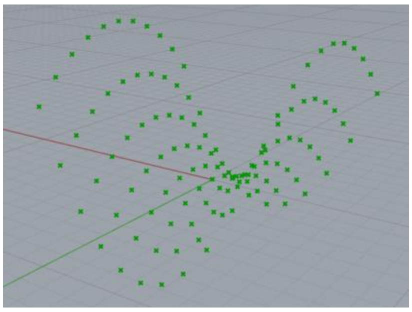
Or use an “Expression” to input a whole formula directly. (find it at Math – Script – Expression)



```
(x²/a²)-(y²/b²)
```

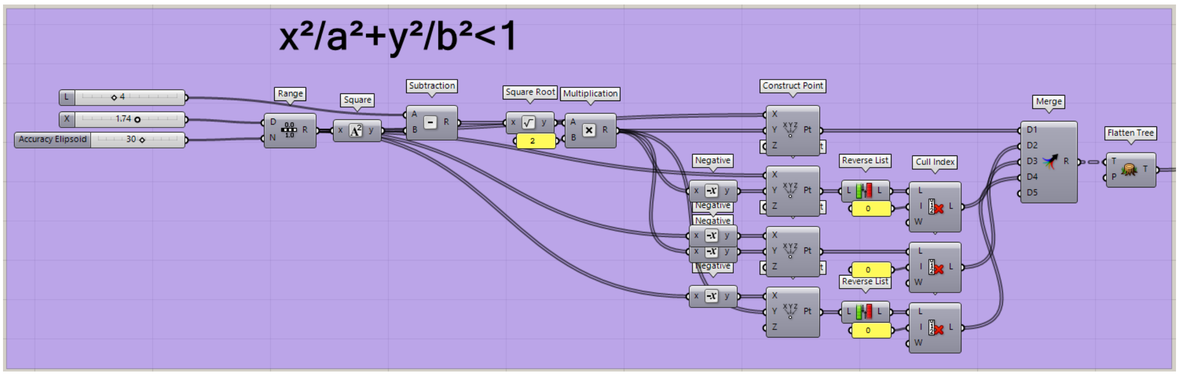
The result is the same. You can visualize it by previewing the construction points that the output connects to:

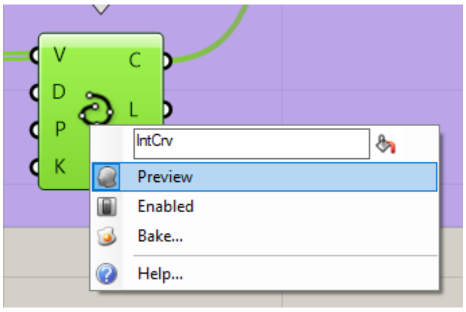


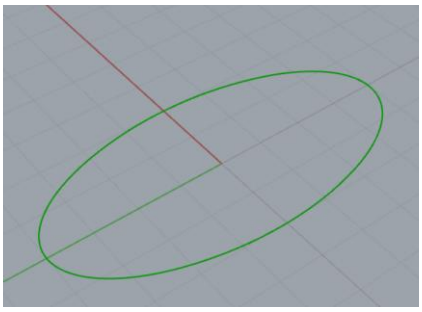


The second formula in the project creates an ellipse that will be used as an outer boundary for the hyperbolic paraboloid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} < 1$$

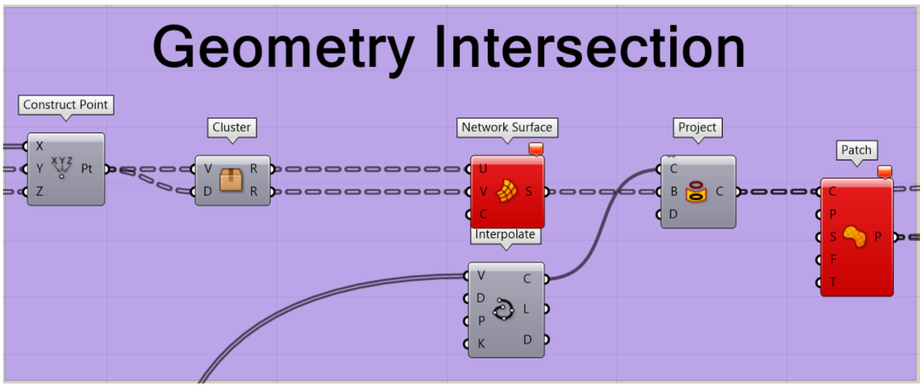




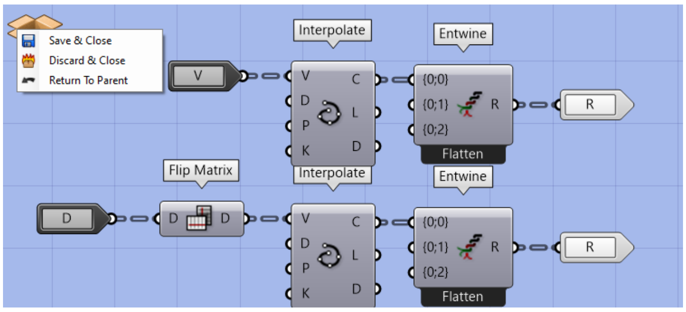


- **Creating the geometry**

The two base conditions are now created. The next step is to create a surface based on these boundaries.



You can create clusters to compress complex operations into a single operation. Double click onto a cluster to open it and visualize the hidden functions. You can leave via right click in the top left box:



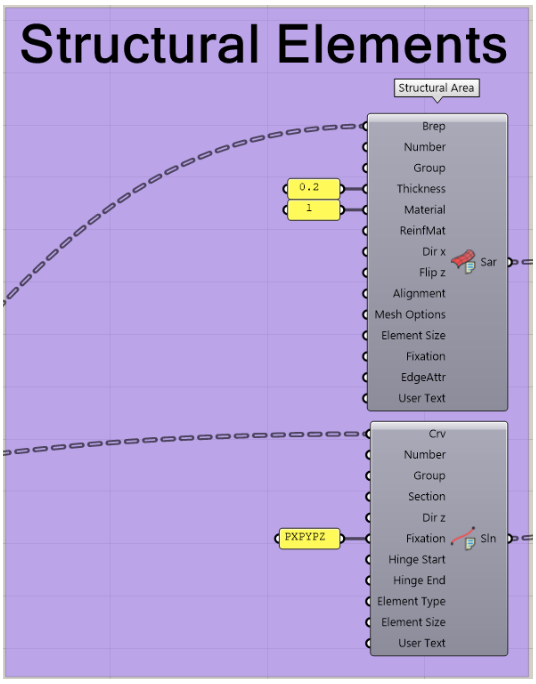
This cluster is used to create a mesh of lines, based on the before created points. Network surface then creates a surface from these lines. Afterwards Project is used to image the ellipse onto that surface geometry. Patch is a handy command that creates a surface in between the boundary of a closed line.

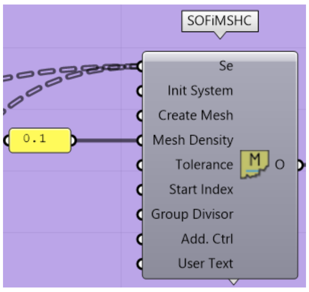
3.4.2 SOFiSTiK Operations

Already shown features will not be explained in detail.

- **Structural Elements**

The three-dimensional curved line, as well as the hyperbolic paraboloid-surface can be directly used to generate structural elements for SOFiSTiK. Simply connect them to the respective SOFiSTiK operators and modules:

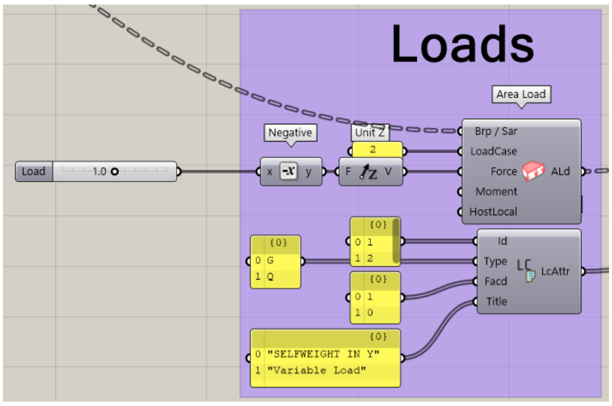




A thickness and material is applied to the structural area. The only information that was added to the structural line is a support fixation, which means that it only carries support information. The finite element meshing in SOFiMSHC can also be adjusted. In this case with a max mesh density of 0,1m.

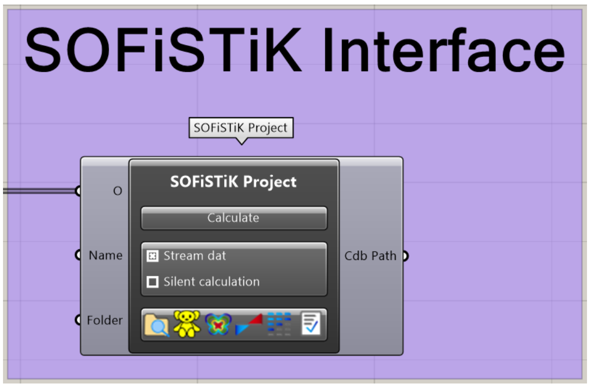
- **Loads**

Area loads are used in this example. They are applied directly onto an area you select. This time lists are used to input the Loadcase Attributes. Another difference to the Steel Hall example is that a “Title” was added to the loadcases that will be shown in the analysis later.



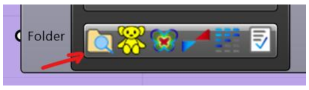
- **Initiating your SOFiSTiK Export**

The “Stream dat” checkbox is checked. SOFiSTiK will automatically generate a .dat file in the folder, where your project file is saved. This file is always updated with your Rhino Grasshopper geometry.

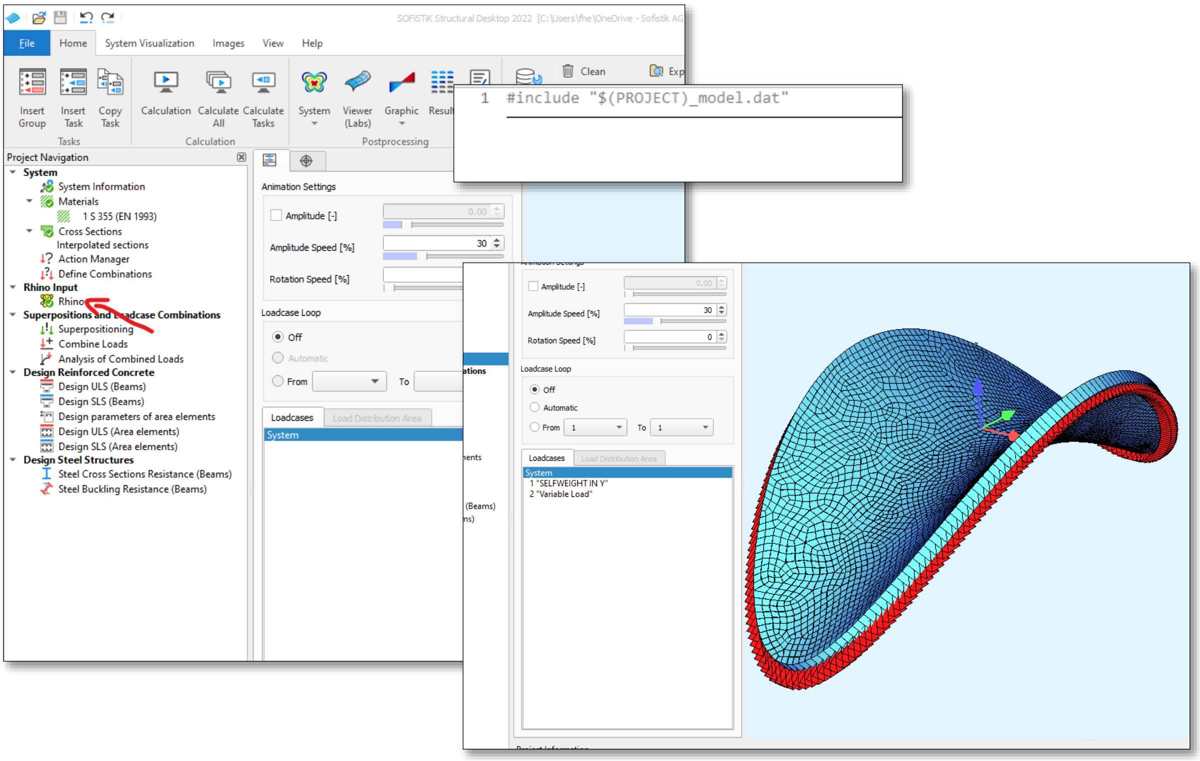


☒ Stream dat
☐ Silent calculation

Sidenote: You can access your current file folder by clicking this icon:



This .dat file can then be imported to SOFiSTiK Structural Desktop and be used for your analysis. A single line of code is necessary to do so (a .sofistik-file with this input is attached to this example).

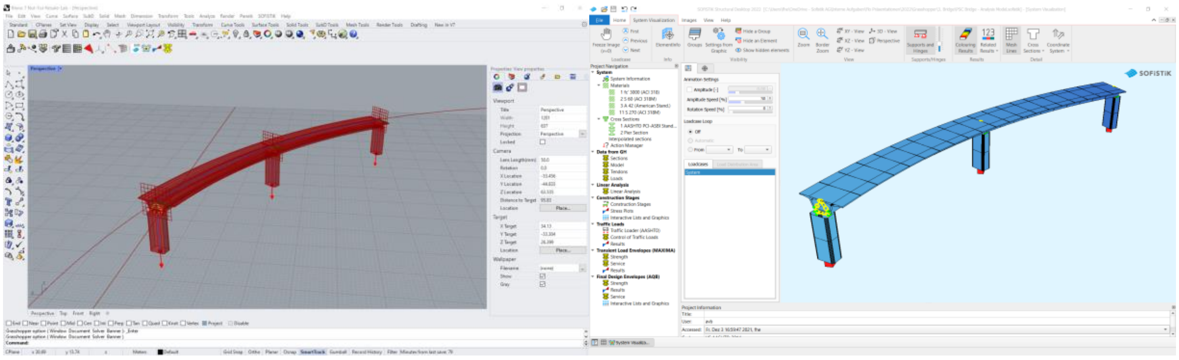


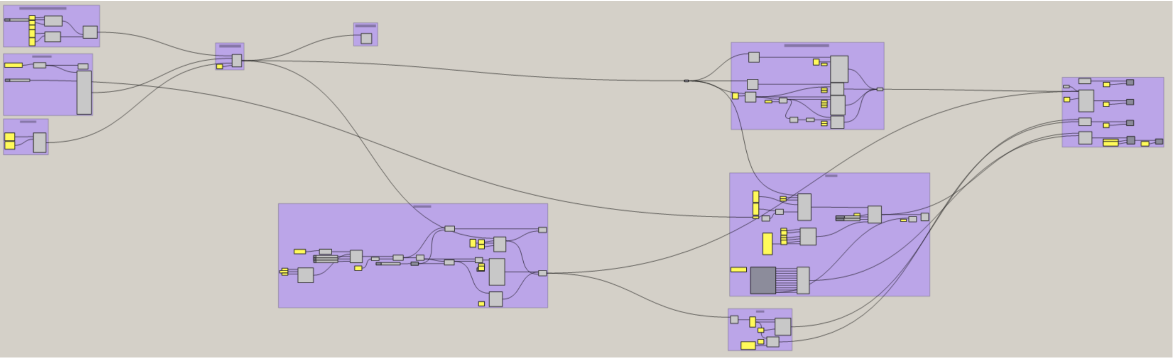
```
#include "$(PROJECT)_model.dat"
```

Right click and calculate this task to import your up-to-date geometry from Rhino.

3.5 Bridge Example

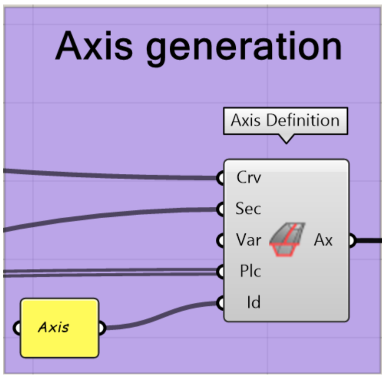
This example shows SOFiSTiK´s Computer Aided Bridge Design Workflow (CABD) in Grasshopper. Axis based input, the possibility to parametrize your cross section along the axis, placement (point of interest along the axis) based substructures and much more. There will be no clean boundary between Grasshopper and SOFiSTiK functions in this example, as most of this workflow is based on the axis-based functions from SOFiSTiK.





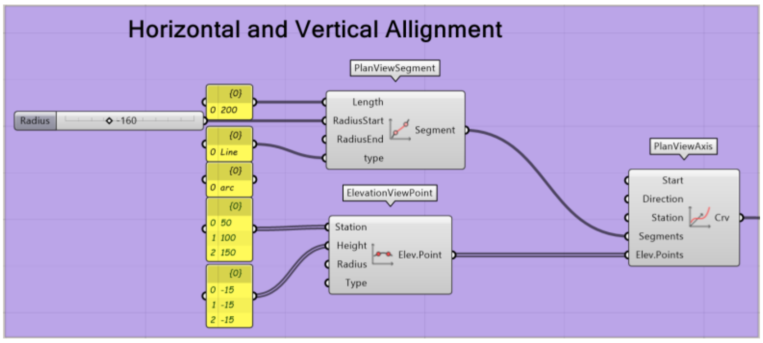
- **Axis Geometry**

You can decide how much information you want to add to your Axis. It all comes together in “Axis Definition”. Horizontal and vertical alignment can be added via Crv. Sec allows you to add cross section information. Var lets you add variables along your axis, based on stationing information. Placements are points along your axis, where you want specific things to happen (in this example supports). Lastly, with Id you can assign a name to your Axis.



Crv

In this project we chose to create the curve, that will define our Axis geometry, via PlanViewSegment and ElevationViewPoint. PlanViewAxis then converts the input data into a curve geometry. This way you can create your axis-curve in a professional, verifiable way.



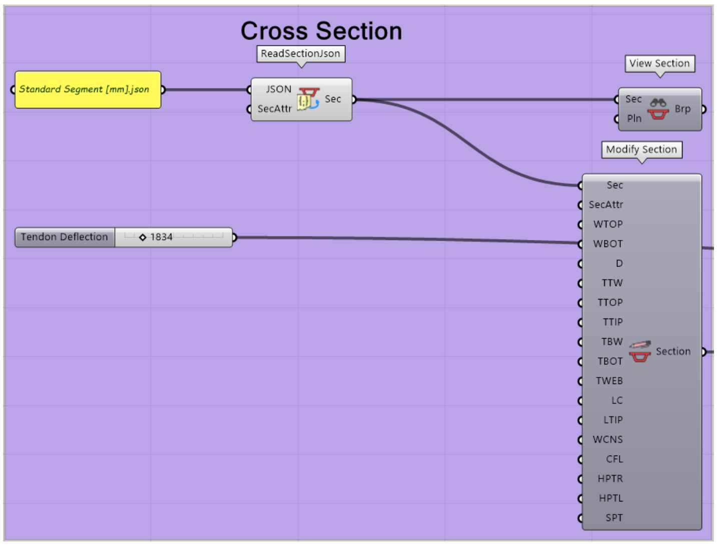
PlanViewSegment… Horizontal information for your axis. Length and radius information, as well as type (Line, Arc, Bloss, Clothoid, Sinusoid, Cosinusoid). Use a list input for multiples.

ElevationViwPoint… Vertical information for your axis. Input via Elevation Points along your axis length. You can also add radiuses and transition (interpolation) types (Parabola, Circular, Asymmetric).

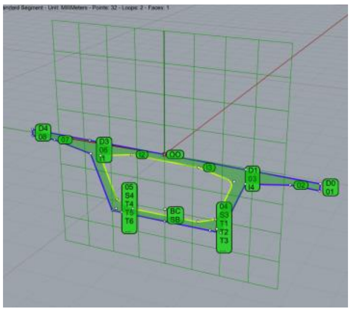
PlanViewAxis… Brings horizontal and vertical information together to create a curve. Additionally, to the shown input, you can also define a specific start point, direction and stationing.

Sec

SOFiSTiK gives you multiple options to input your cross-section information. The method used in this project is to import it from a .json file. This simply requires the file path and name to import it.



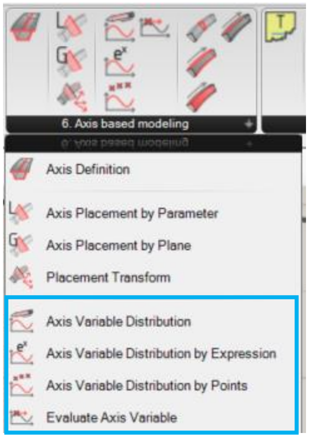
View Section… creates a preview of the section at the origin point (0,0,0). Shows the point names when selected.



Modify Section… showcases all adjustable parameters of the cross section.

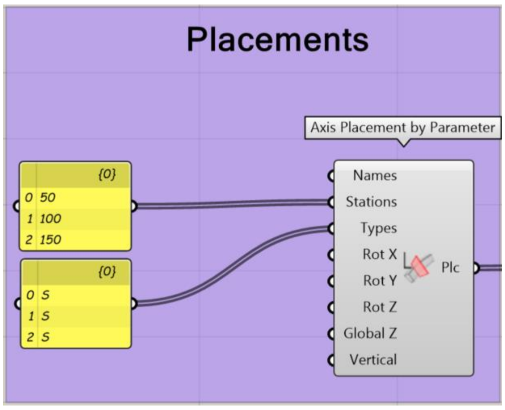
Var

To define Variables along your axis, use one of the following commands:


6. Axis based modeling
Axis Definition
Axis Placement by Parameter
Axis Placement by Plane
Placement Transform
Axis Variable Distribution
Axis Variable Distribution by Expression
Axis Variable Distribution by Points
Evaluate Axis Variable

Plc

Placements are points of interest, that are displayed as reference planes along the axis. In this case Supports (S at Types). You can assign names, as well as rotate these reference planes, based on the reference of the axis.



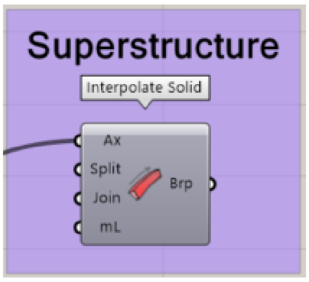
Id

Id simply assigns a four-character name to the axis as an identifier.

- **Superstructure**

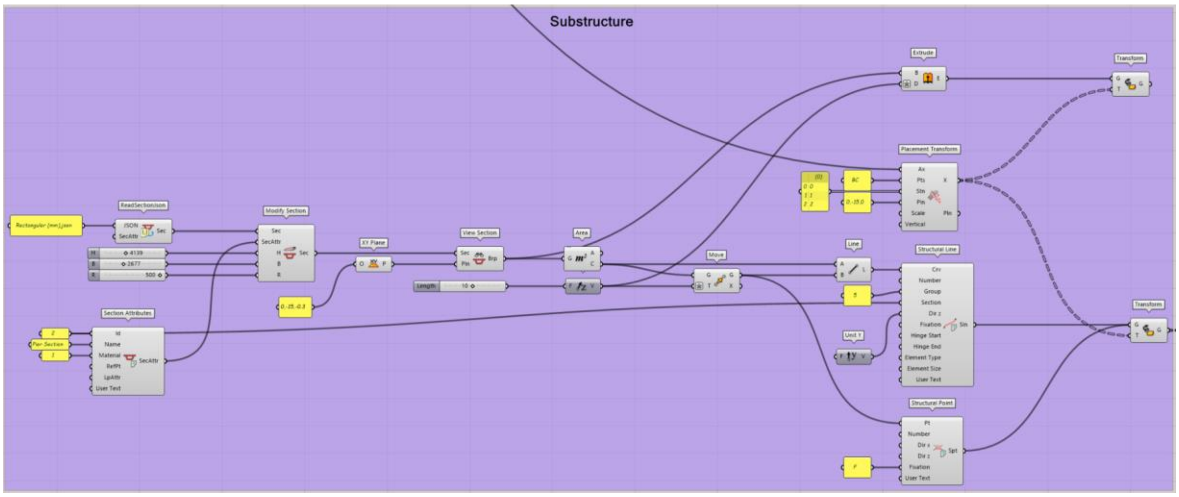
At this point the axis carries all the information. There is no volume geometry generated yet.

“Interpolate Solid” reads the information from the axis and generate the model.

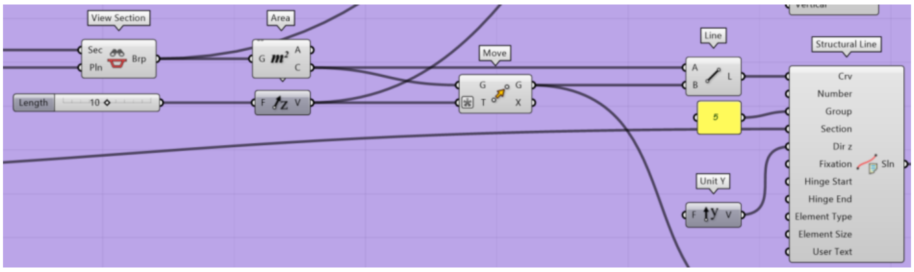


- **Substructure**

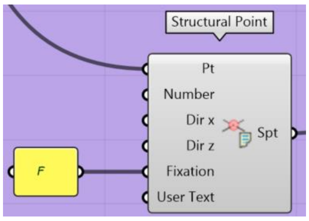
Up to “View Section” the workflow is identical to the cross section assigned to the superstructure. With one little difference, the reference plane of the substructure was rotated (Pln of View Section).



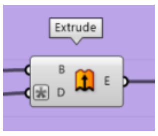
With the midpoint of the cross section (“Area”) as a reference a structural line with a length of 10 is created. The structural line already contains the cross-section information, as the Id of the section (2) is accessed.



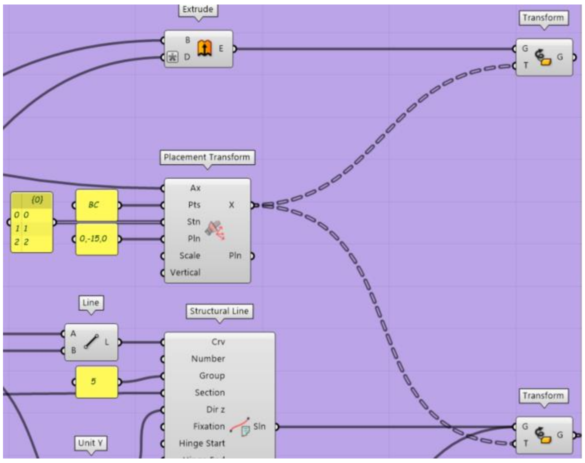
A structural point is generated at the end of this structural line and is defined as a fixed support (Fixation = F)



To receive a volume representation in Rhino, you still need to extrude a volume model based on the cross-section information via “Extrude”.

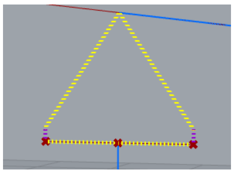


Lastly, to assign the substructure to their respective supports, you can use “Transform” in combination with “Placement Transform”. As only the transformed structural lines are connected to the SOFiMSHC-module, only they will be considered for the structural model (not the original representation at the origin).

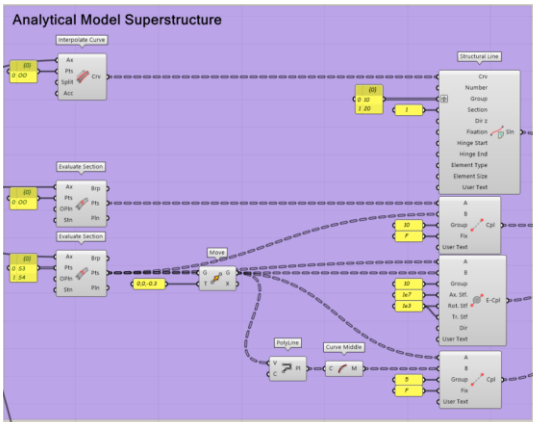


- **Analytical Model of the Superstructure**

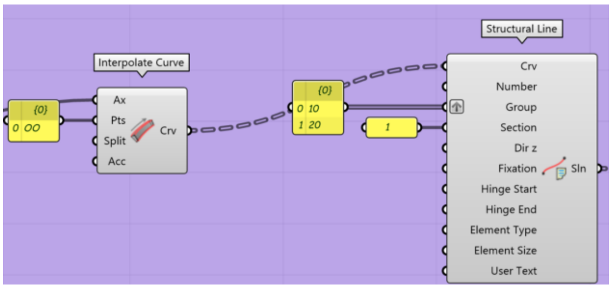
The analytical model of the superstructure consists of a Structural Line as a reference, couplings that connect to vertical elastic links (that represent the bearings). They are then connected to the structural line of the substructure.



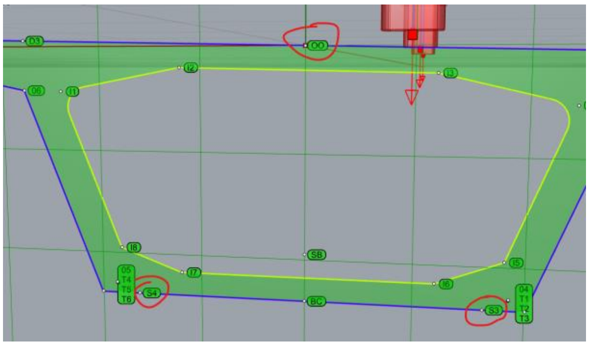
All these SOFiSTiK operations are based on information coming from the axis.



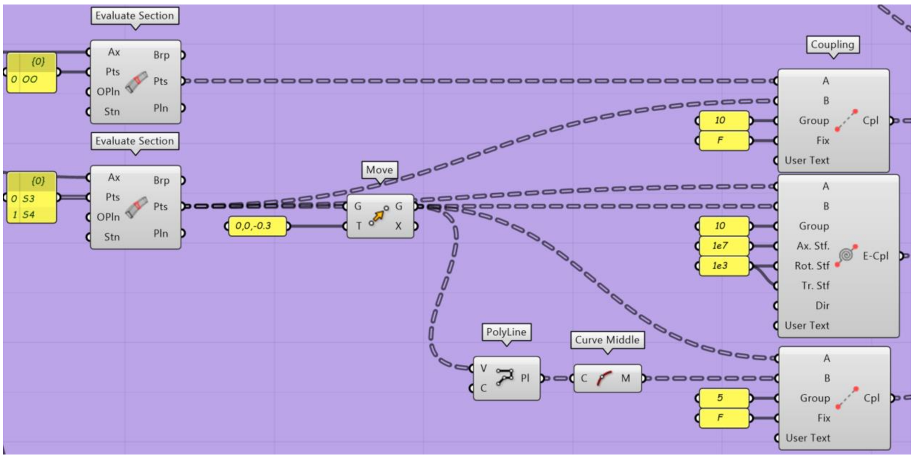
“Interpolate Curve” extracts a curve from the cross-section information along the axis. “OO” is the index of the origin point of the cross-section. You can select any point of the cross section to generate a curve this way (e.g. “D3”). The curve is automatically split at placements. Assign the curve to “Structural Line” to make it analytical.



Select “View Section” in “Cross Section” to review the cross-section points.

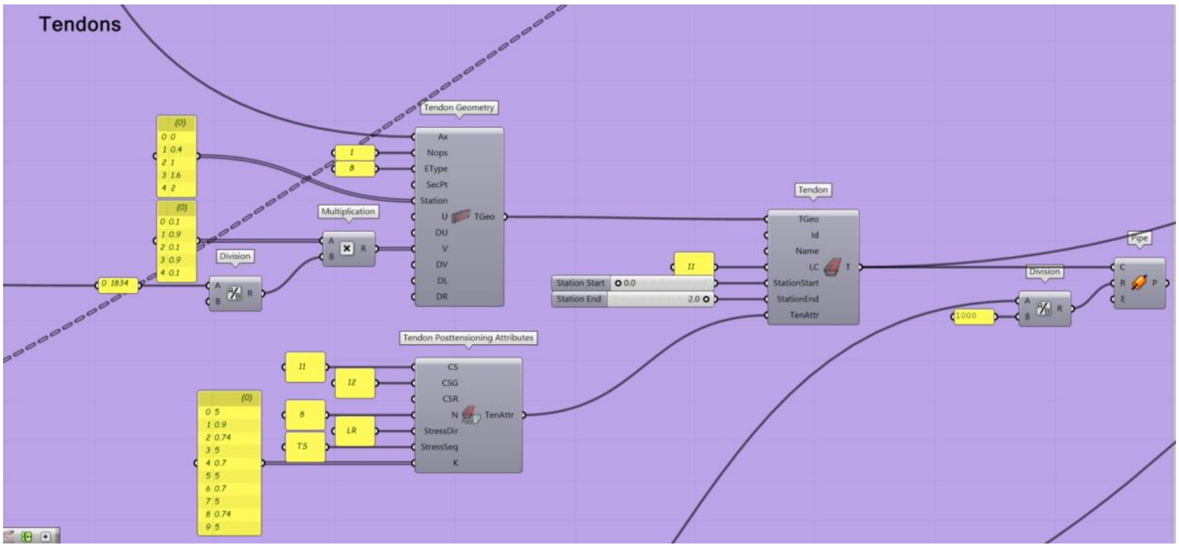


“Evaluate Section” outputs exact point coordinates of the cross-section at placements. This way it is just a matter of assigning offsets to these points to use them for references of our couplings and elastic links. (for “OO”, “S3” and “S4” see last page)

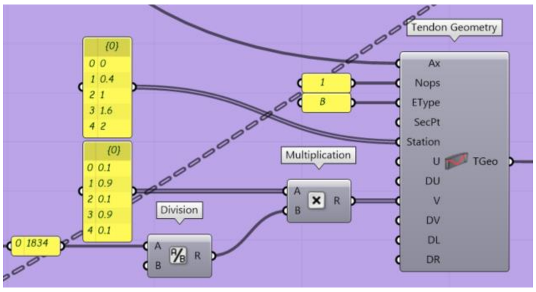


- **Tendons**

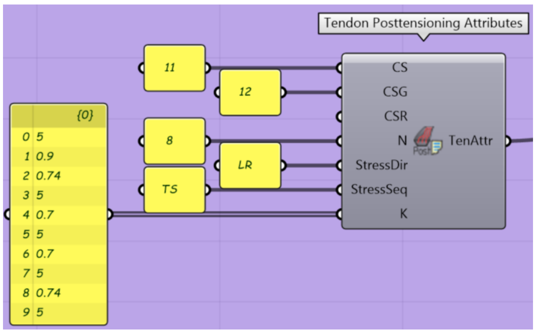
Tendons follow the same logic of following the axis with an offset, as the cross section.



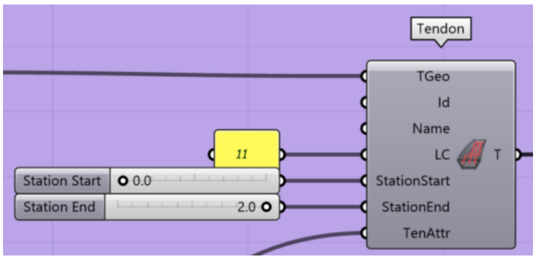
The geometry definition can be summed up to vertical and horizontal offset definition, based on the stationing of the axis.



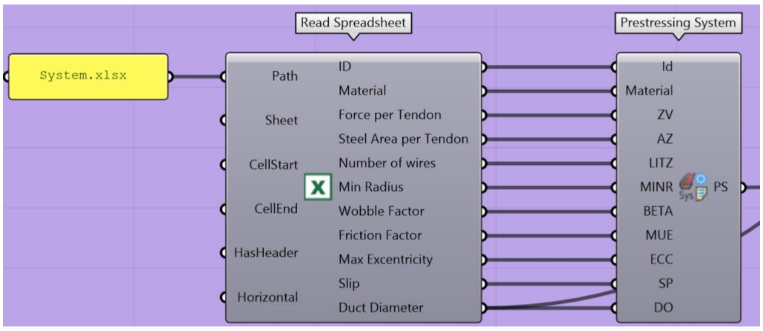
The attributes of the tendon must be differentiated between pre- and post-tensioning. A construction stage input for stressing and grouting were applied. Additionally, the number of tendons, stressing direction and sequence, as well as the K-values can be input directly.



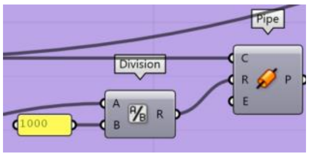
All this information comes together in “Tendon”. It requires a loadcase, with which the tendon stresses are applied to the structure. The tendons can also be shortened along the axis, based on stationing.



The Prestressing system is imported from an EXCEL-sheet. SOFiSTiK´s “Read Spreadsheet” operator allows you to directly access spreadsheets from a directory. As soon as you save your EXCEL, the information will be updated in Grasshopper.

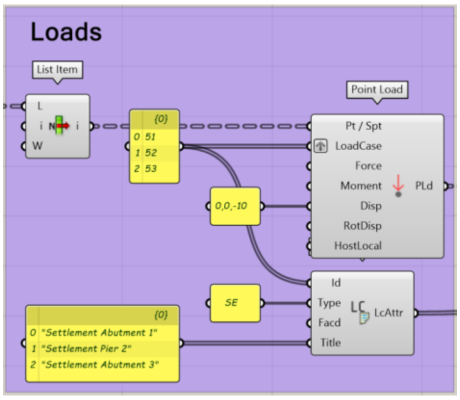


The SOFiSTiK input is used for analytical properties. You can extract the necessary information from it to create a 3D geometry with Grasshopper components.



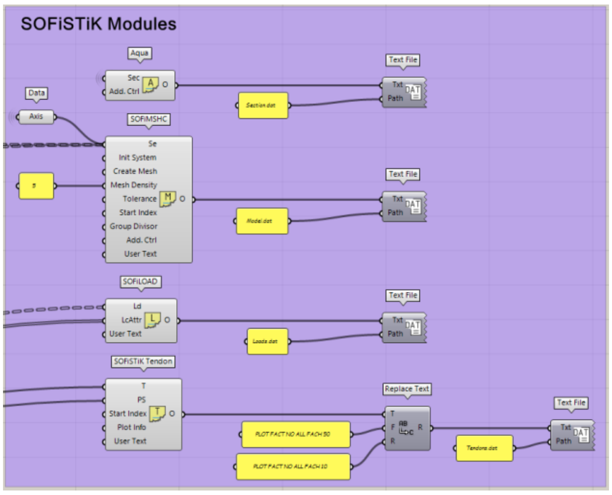
- **Loads**

This project shows how to apply settlement to structural points. A displacement at the bottom of the structural points, that act as supports. A simple list-based input and a Type-assignment as “SE” is all it needs.

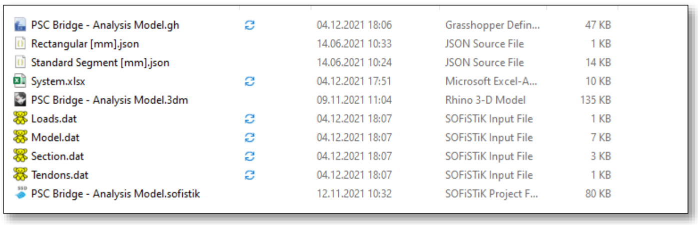


- **SOFiSTiK Output**

The respective SOFiSTiK modules are fed with the information from Grasshopper, that we want to use for analysis. For this project we used another method of exporting the modules information. A direct TEDDY-text export to .dat (Text File).

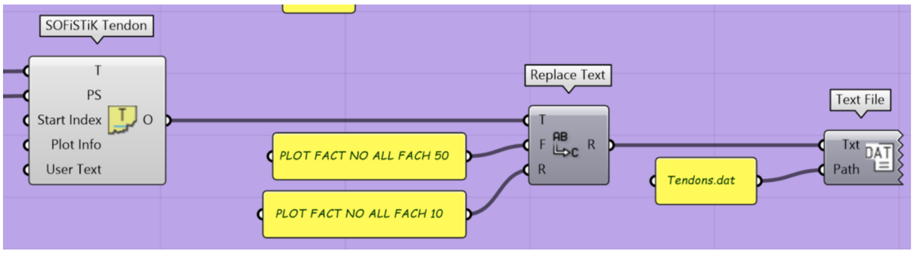


These .dat-files are always up to date. If you don’t give them a specific file path, they are exported in your project file directory.



PSC Bridge - Analysis Model.gh	04.12.2021 18:06	Grasshopper Defin...	47 KB
Rectangular [mm].json	14.06.2021 10:33	JSON Source File	1 KB
Standard Segment [mm].json	14.06.2021 10:24	JSON Source File	14 KB
System.xlsx	04.12.2021 17:51	Microsoft Excel-A...	10 KB
PSC Bridge - Analysis Model.3dm	09.11.2021 11:04	Rhino 3-D Model	135 KB
Loads.dat	04.12.2021 18:07	SOFiSTiK Input File	1 KB
Model.dat	04.12.2021 18:07	SOFiSTiK Input File	7 KB
Section.dat	04.12.2021 18:07	SOFiSTiK Input File	3 KB
Tendons.dat	04.12.2021 18:07	SOFiSTiK Input File	1 KB
PSC Bridge - Analysis Model.sofistik	12.11.2021 10:32	SOFiSTiK Project F...	80 KB

Changes to these texts can be done directly in Grasshopper. As we are handling simple text files, we can use Grasshopper operators like “Replace Text” to adjust it. In this case for example the superelevation factor for the plot from 50 to 10:

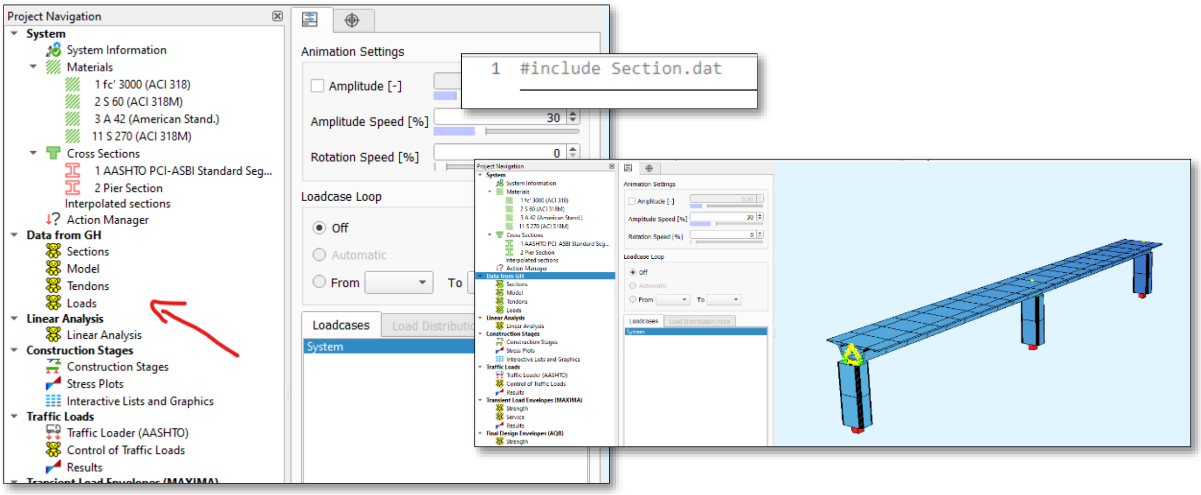


PLOT FACT NO ALL FACH 50

PLOT FACT NO ALL FACH 10

Tendons.dat

To use these .dat-files in your analysis, it is necessary to include them in your SOFiSTiK-project (see included .sofistik-file)



Amplitude Speed [%] 30

Rotation Speed [%] 0

```
#include Section.dat
```

Every time you calculate these tasks, your Grasshopper output is imported to SOFiSTiK and you are all set to do your analysis.

Thank you for reading!
The SOFiSTiK team wishes you a lot of fun and creativity with this versatile, parametric and entertaining modelling platform.

If you have any questions, please feel free to send an email to info@sofistik.de.