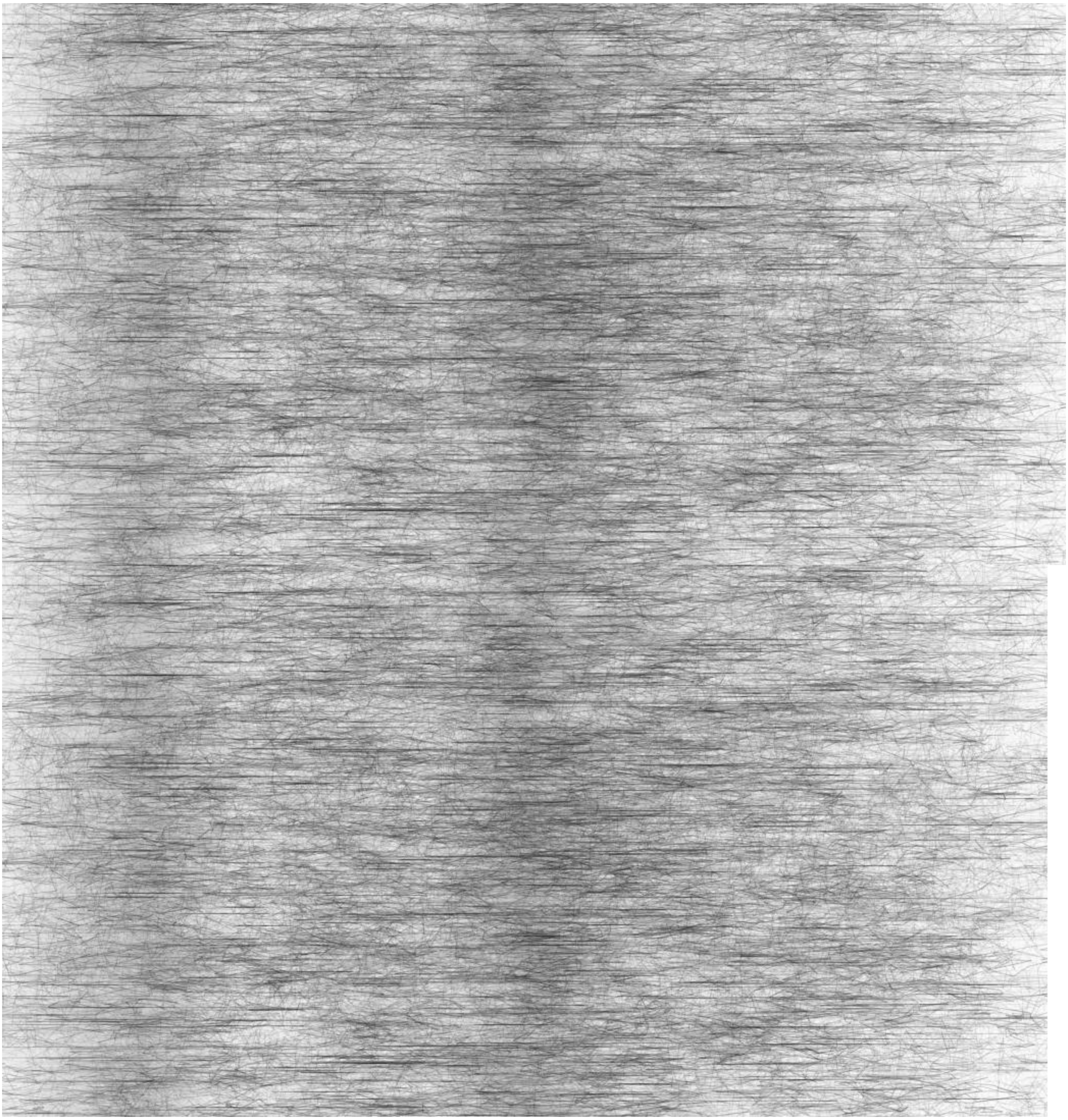
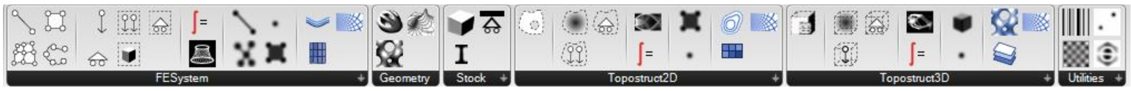


Millipede

Millipede

www.sawapan.eu

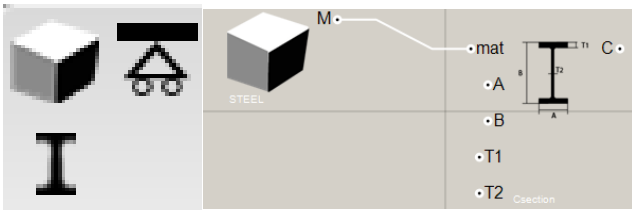


Millipede is a Grasshopper™ component focusing on the analysis and optimization of structures.

α. Overview

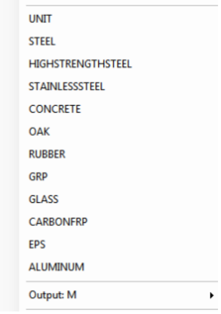
At the core of this component is a library of very fast structural analysis algorithms for linear elastic systems. The library contains its own optimization algorithms based on topology optimization but due to its speed it can be used in combination with Galapagos for solving generic form finding problems [see included examples].

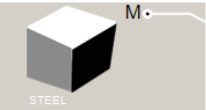
β. Stock Objects



Stock objects are shared by all analysis modules and are used in order to define materials, boundary conditions and cross sections.

1. Material Definition

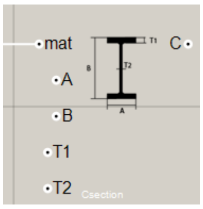




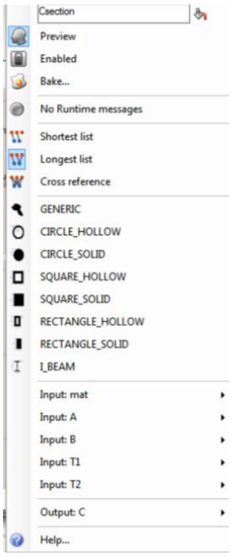
The material component allows you to select a stock material. This can be connected as an input to cross sections and shell elements in order to assign materials.

You can right click on the material component to select one of a pre-set number of materials

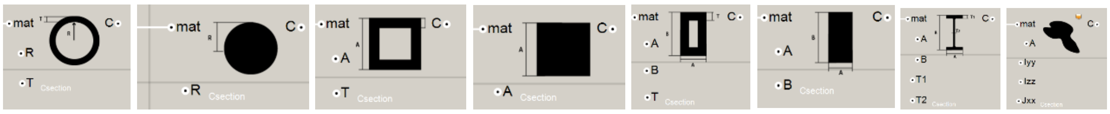
1. Cross Section Definition

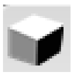


The Cross Section component generates cross section definitions to be assigned to frame elements.



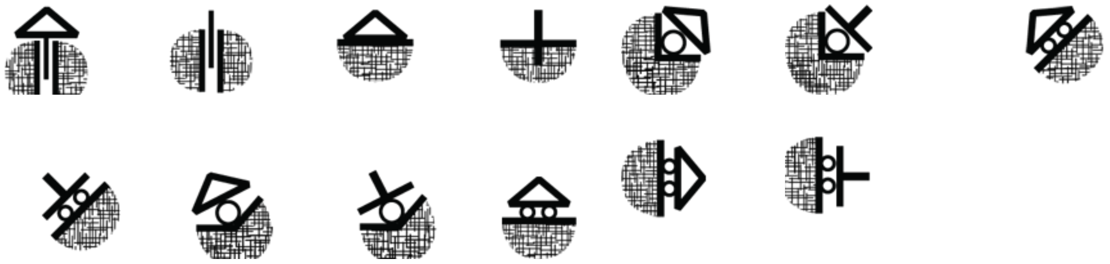
You can right click on the Cross Section component in order to choose a cross section type. Each time you change the cross section type the input parameters of the component are changed so as to reflect the different geometric properties of each section.



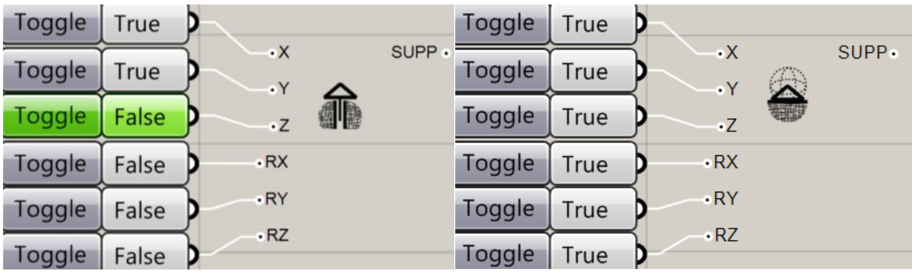
All types of cross sections receive a material  definition as their first input. The rest of the parameters are determine the dimensions according explained in each section type's diagram icon. All dimensions are in SI units [meters].

The special case of a generic cross section allows one to input manually the Area, Moments of Inertia and Torsional constant [m^2 and m^4].

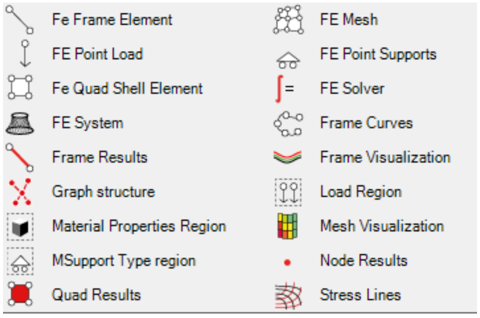
2. Boundary Condition Definition



The support component takes as inputs 6 Boolean values that determine which degrees of freedom are fixed at a point. XYZ represent translations along the corresponding axes and RX RY RZ represent the rotations around the corresponding axes. The component generates and integer which is a bitmask of the 6 on/off states.



γ. Frame and Shell Element Analysis



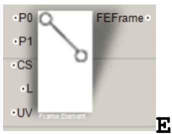
Fe Frame Element
FE Point Load
Fe Quad Shell Element
FE System
Frame Results
Graph structure
Material Properties Region
MSupport Type region
Quad Results

FE Mesh
FE Point Supports
FE Solver
Frame Curves
Frame Visualization
Load Region
Mesh Visualization
Node Results
Stress Lines

This toolbox contains the components necessary to build analyse and optimize structures composed of frame elements [beams, columns, trusses etc…] and shell elements.

1. Geometry constructors

These elements apply structural properties to Rhino geometry in order to be passed as inputs in the structural system builder component.



a. Explicit Frame Element

This component creates a frame element connecting two points with a specified cross section, distributed linear load and cross section orientation.

Inputs:

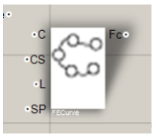
P0 [Point3d]: The start point of the frame element

P1 [Point3d]: The end point of the frame element

CS [Cross Section]: The cross section, input from I *component*

L [Vector3d]: The distributed linear load along the element in global coordinates [N/m]

UV [Vector3d]: This is the vector that determines the orientation of the Y axis of the cross section along this element



b. Curve

This component assigns material properties to a curve object that will be converted into a series of frame elements during the system building phase.

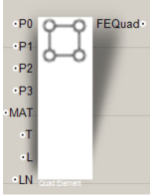
Inputs:

C [Curve]: The Curve geometry

CS [Cross Section]: The cross section, input from I *component*

L [Vector3d]: The distributed linear load along the element in global coordinates [N/m]

SP [Integer support code]: A support type to be applied to all nodes that will be generated along this curve



c. Explicit Shell Quad Element

This component defined a single quad element for shell analysis

Inputs:

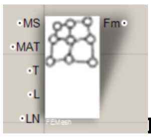
P0, P1, P2, P3 [Point3d]: The four corners given in counter clockwise order

MAT [Material]: A material definition coming from component

T [Thickness]: The thickness of the shell element [m]

L [Vector3d]: Distributed area load in global coordinates [N/m2]

LN [Number]: Normal area load [N/m2]. Useful for defining pressure



d. Mesh to Shell

This component parses a Mesh and generates a quad shell element for each face in the mesh. If the mesh contains any triangular faces then the whole mesh is subdivided into quads before further processing. This may result in a tripling or quadrupling of the number of elements in the mesh.

!For best results make sure that the mesh does not contain elongated triangles or quads that are too twisted or faces that are too small. You can tweak the meshing parameters in rhino [especially the min/max edge lengths] before meshing any brep geometry.

Inputs

MS [Mesh]: The input mesh

MAT [Material]: A material definition coming from component

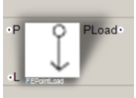
T [Thickness]: The thickness of the shell elements [m]

L [Vector3d]: Distributed area load in global coordinates [N/m2]

LN [Number]: Normal area load [N/m2]. Useful for defining pressure

2. Property modifiers

These components assign properties at specific nodes in the system. Make sure that the points you pass to these components coincide with actual nodes in the system [end point of curves or frame element or vertices of meshes]



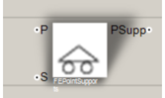
e. Applied point load

Applies a point load at a specific location in the structural system

Inputs:

P [Point3d]: Point of load application

L [Vector3d]: Load [N]

f.  Point support type

This component applies a specific boundary condition at a location in space

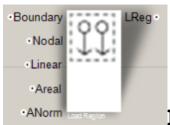
Inputs:

P [Point3d]: Location of support

S [Integer support code]: A support type to be applied at the specified location

3. Region Property modifiers

These modifiers apply a certain property within a region determined enclosed by a solid object [polysurface].

a.  Region of Loads

Apply the specified types of loads to all elements that fall within the specified solid.

Inputs:

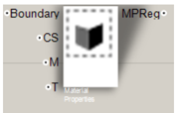
Boundary [Brep]: A closed solid geometry from rhino

Nodal [Vector3d]: Point load to be applied to all nodes within boundary [N]

Linear [Vector3d]: Linear load to be applied to all frame elements within boundary [N/m]

Areal [Vector3d]: Area Load to be applied to all shell elements within boundary [N/m2]

ANorm [Number]: Normal Area Load to be applied to all shell elements within boundary [N/m2]

b.  Region of Material properties

Apply the specified material properties to all elements that fall within the specified solid.

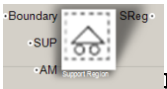
Inputs:

Boundary [Brep]: A closed solid geometry from rhino

CS [Cross Section]: The cross section, input from I *component to be applied to all frame elements within boundary*

M [Material]: A material definition coming from component to be applied to all shell elements within boundary

T [Number]: A thickness to be applied to all shell elements within boundary

c.  Region of Support

Apply the specified support types to all nodes that fall within the specified solid.

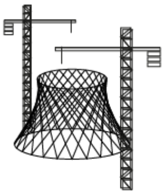
Inputs:

Boundary [Brep]: A closed solid geometry from rhino

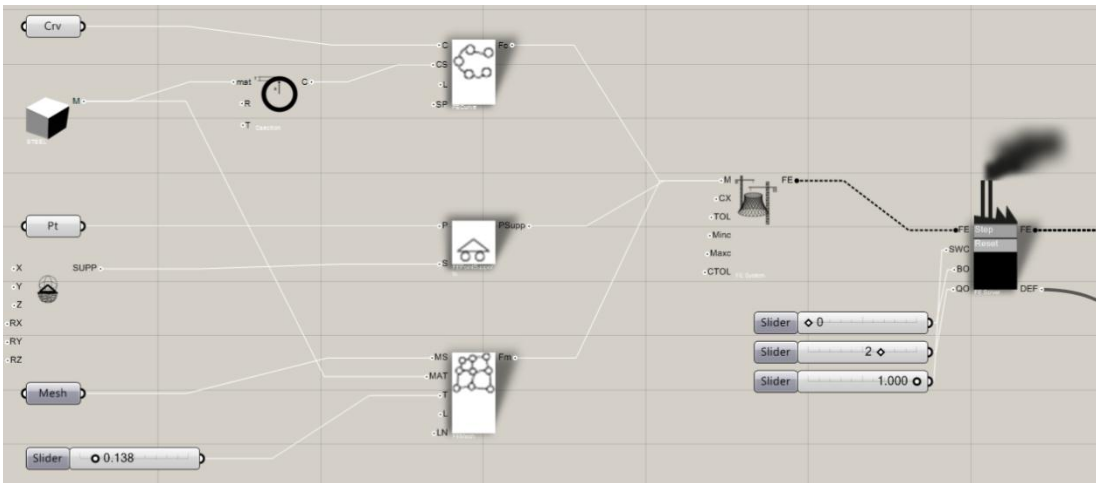
SUP [Integer support code]: A support type to be applied to all nodes within boundary

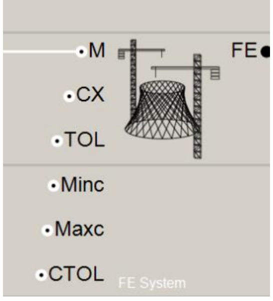
AM [Boolean]: A value indicating whether support types should have an additive behaviour or just replace whatever support is already there.

4. System builder



This is the core component that aggregates all the information [geometry, properties etc…] and builds the actual analysis model but it does not initiate any analysis itself.





Inputs:

M [Generic]: Any number of the outputs of these components are aggregated here.

The rest of the inputs determine how curves are going to be intersected and discretized in order to yield connected frame elements

CX [Boolean]: Intersect curves, if true all curve objects are intersected and split against each other so that nodes are inserted at the intersections

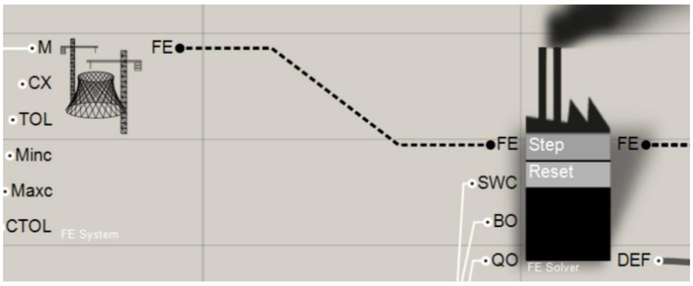
TOL [Number]: Tolerance for curve intersection and merging of points

Minc [Number]: Minimum allowed length for a frame element

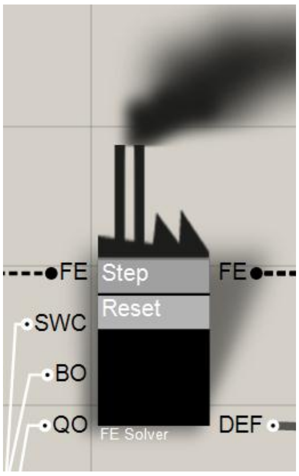
Maxc [Number]: Maximum allowed length for a frame element

CTOL [Number]: Tolerance that controls how close generated polylines approximate the original curves. Smaller values will result in more accurate representations at the expense of more frame elements and slower calculations.

5. Solver



The solver constitutes the slowest step in the process which involves the analysis of the finite element system and recovery of results. It must be connected FE->FE to a system builder component. In addition to its inputs it has two buttons that are used in optimization [when determining optimal sizes and thicknesses for frame and shell elements]



Inputs:

FE [StatFESystem]: The Finite element model coming from a system builder component

SWC [Number]: Self Weight Coefficient. It determines whether the self-weight of the structure is taken into account during the analysis and how much it will affect the final outcome. A value of 1.0 means that the self-weight is applied without modification, while a value of 0 means that it will not be taken into consideration. Values larger than 1.0 are also permitted.

BO [Integer]: A number determining how many iterations of Frame element size optimization should occur each time the user presses the "Step" button

QO [Integer]: A number determining how many iterations of Shell element thickness optimization should occur each time the user presses the "Step" button

Outputs:

FE [StatFESystem]: The analysed Finite Element system to be passed to other components that deal with visualization and recovery of specific results.

DEF [Number]: The maximum deflection in meters of the analysed structure. Useful for quick troubleshooting and getting a quick rough measure of system performance without doing detailed analysis of the results. If this number is too big there may be disconnected elements in the system and if it is zero you may haven't applied loads correctly.

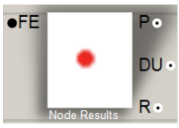
Controls:

Step: When pressed the control carries BO steps of frame element size optimization and QO steps of shell element thickness optimization.

Reset: Reset all sizes and thickness to original [non optimized] state.

6. Results

These components allow for a detailed analysis of the results after the solver has finished the analysis



d. Nodal results

This component returns the locations, displacements and rotations for all nodes.

Inputs:

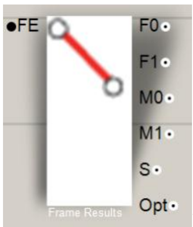
FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

Outputs:

P [Point3d]: The locations of nodes

DU [Vector3d]: The displacements of all the nodes [m]

R [Vector3d]: The rotations around the XYZ axes for each node as a vector.



e. Frame results

This component generates the results for all frame elements [beams, column etc…] in the system.

Inputs:

FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

Outputs:

F0 [Vector3d]: The internal force at the start point in the local coordinate system. The X component is the axial force and the Y and Z components determine the shear force vector [N]

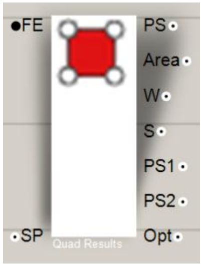
F1 [Vector3d]: The internal force at the end point in the local coordinate system [N]

M0 [Vector3d]: The moment at the start point in local coordinate system. X component is torsion [due to rotation around the axis of the element] while the Y and Z determine the plane and magnitude of the bending moment [N.m]

M1 [Vector3d]: The moment at the end point in local coordinate system [N.m]

S [Number]: An estimate of the maximum normal stress [combined axial force and bending moment contributions] along the element [N/m2]

Opt [Number]: The optimization value. A number from 0.0 to 1.0 that determines the relative importance [or required size] of each frame element.



f. Shell Quad results

This component generates the results for all shell elements in the system.

Inputs:

FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

SP [Number]: the layer at which principal stresses are calculated. This is a number form -1.0 [bottom layer] to +1.0 [top layer]. For shell elements stresses throughout each quad. The component calculates the stresses along a line that passes through the centre of each quad and is parallel to its normal. With this parameter you can choose whether the results returned are at the top or bottom layer of the shell. For example for a slab supported at two sides you should expect to see a lot of tension on the bottom layer and a lot of compression at the top.

Outputs:

PS [Plane]: A plane with its origin at the center of each quad and its axes aligned with the principal stress directions

Area [Number]: the area of each quad [m2]

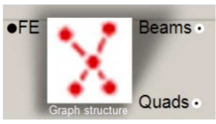
W [Number]: the normal displacement at the center of each quad [m]

S [Number]: Estimated Maximum VonMises stress for all layers of each quad [N/m2]

PS1 [Number]: First principal stress. Positive values designate Tension and negative compression [N/m2].

PS2 [Number]: Second principal stress. Positive values designate Tension and negative compression [N/m2].

Opt [Number]: A number from 0.0 to 1.0 designating the relative strength of thickness multiplier during the optimization process.



g. Connectivity graph

This component generates two trees representing the connectivity graphs for frame elements and shell elements. Nodes are referenced by integer numbers that correspond to the ordering of nodes of the output of the node results component

Inputs:

FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

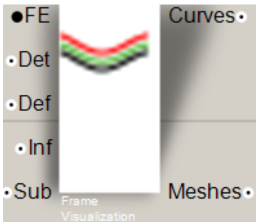
Outputs:

Beams [Integer]: A tree structure with the same number of branches as there are frame elements in the system. Each branch contains two integer numbers designating the nodes connected by the corresponding element

Quads [Integer]: A tree structure with the same number of branches as there are Quad shell elements in the system. Each branch contains four integer numbers designating the nodes connected by the corresponding element

7. Visualization

These components offer a variety of ways to visualize results from the analysis model



h. Frame Elements Visualization

This component offers a quick way to visualize the deformations in the frame elements of the system and see how different forces and stresses vary along the frame elements and around their cross sections.

Inputs:

FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

Det [Boolean]: If true a detailed visualization of the deformations and results along each frame is generated. This involves sampling the element at intermediate points along its length generating more complex meshes.

Def [Number]: The deflection factor. You can use this number to exaggerate the calculated displacements [usually in the order of mm] in the structure so that you can observe the deformation from its original state to its equilibrium state. Usually you can set this number to something like 1.0/maximumdeflection to get a visible effect.

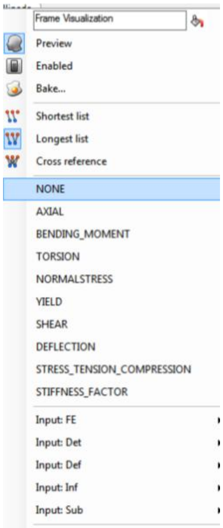
Inf [Number]: This factor determines the inflation of the cross section along each element and along the direction of maximum stress giving an immediate visual feedback about the distribution of stresses throughout each element as well as a suggestion for possible use of variable cross sections.

Sub [Number]: This number determines the length of the subdivisions of each element in detailed visualization mode. The smaller it is the more subdivisions will be generated at the expense of slower calculations and more complex meshes.

Outputs:

Curves [Polyline]: A list of polylines representing the deflected shapes of the centrelines of all the frame elements.

Meshes [Mesh]: A mesh that contains the deflected and coloured shapes for all the frame elements in the system. To determine the colour mode right click on the component and select one of the available colour modes:



NONE: White meshes

AXIAL: Meshes are coloured red or cyan depending on whether the dominant effect is compression or tension

BENDING_MOMENT: Rainbow visualization of the distribution of bending moments

TORSION: Rainbow visualization of the distribution of Torsion

NORMAL_STRESS: Rainbow visualization of the distribution of NORMAL_STRESSES

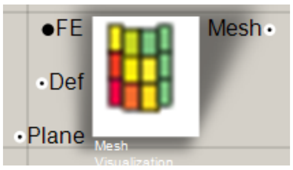
YIELD: Ratio of stress to yield stress for specified material. Red means that the stress is over the yield stress threshold.

SHEAR: Rainbow visualization of the distribution of Shear forces

DEFLECTION: Rainbow visualization of the distribution of deflections

STRESS_TENSION_COMPRESSION: Detailed stress analysis throughout the cross section of each element revealing the distribution of tension and compression [cyan / red] as a result of a combination of axial forces and bending moment

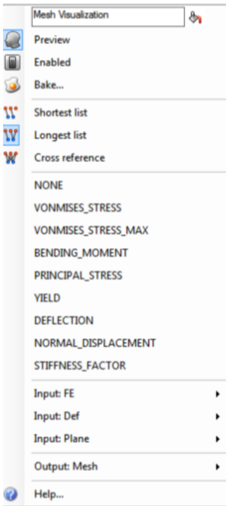
STIFFNESS_FACTOR: This is the optimization calculated factor that determines the relative strength, size or importance of the frame element in the overall system



i. Shell Elements Visualization

This component generates meshes that visualize the distribution of forces and stresses over shell elements

Inputs:



FE [StatFESystem]: The analysed Finite Element system coming from the solver component

Def [Number]: The deflection factor. You can use this number to exaggerate the calculated displacements [usually in the order of mm] in the structure so that you can observe the deformation from its original state to its equilibrium state. Usually you can set this number to something like 1.0/maximumdeflection to get a visible effect.

Plane [Number]: A number form -1.0 [bottom layer] to +1.0 [top layer] determining the layer throughout the thickness of each shell element at which the results are calculated [only affects stress results].

Outputs:

Mesh [Mesh]: A mesh that contains the deflected and coloured mesh for all the shell elements in the system. To determine the colour mode right click on the component and select one of the available colour modes:

NONE: White meshes

VONMISES_STRESS: Rainbow visualization of the vonmises stress at the specified layer

VONMISES_ STRESS_MAX: Maximum estimated vonmises stress [for all layers]

BENDING_MOMENT: Rainbow visualization of the maximum bending moment at the centre of each quad

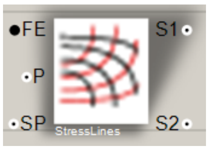
PRINCIPAL_STRESS: Red/Cyan visualization depending on whether the dominant effect locally and at the specified layer is tension or compression

YIELD: Ratio of stress to yield stress for specified material. Red means that the stress is over the yield stress threshold.

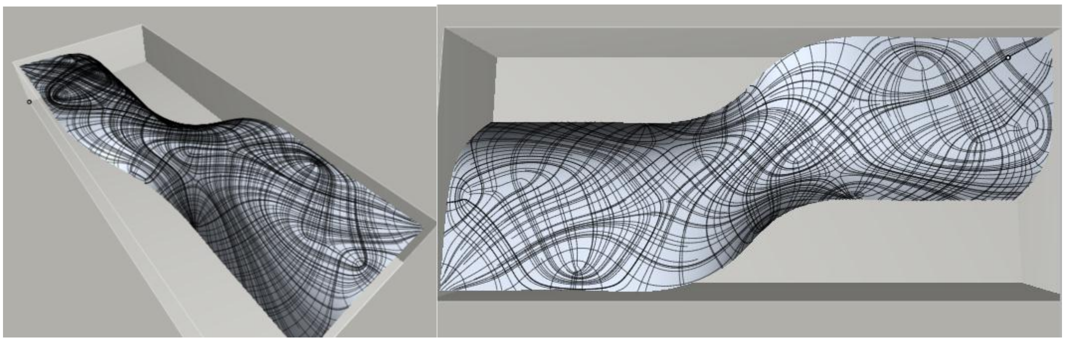
DEFLECTION: Rainbow visualization of the distribution of deflections

NORMAL_DISPLACEMNT: Rainbow visualization of normal displacement [out of plane] at the centre of each quad

STIFFNESS_FACTOR: This is the optimization calculated factor that determines the relative strength, density or thickness of the shell element



j. Stress lines



This component calculates the two principal stress lines emanating from a single point. Stress lines are curves that at each point are tangent to one of the principal stress directions.

Inputs:

FE [StatFESystem]: The analysed Finite Element system coming from the solver component ∫=

P [Point3d]: A point that becomes the starting location for tracing stresslines

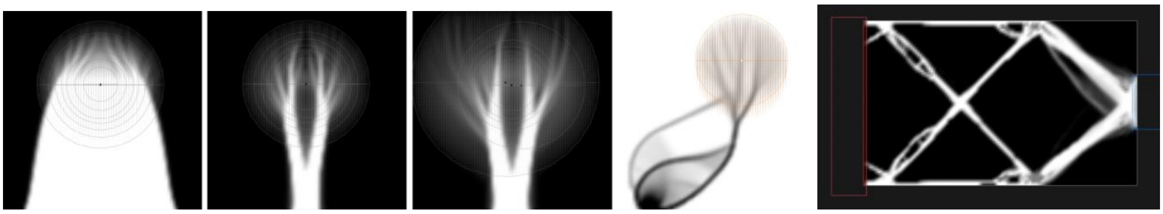
SP [Number]: A number from -1.0 to 1.0 that determines the layer at which the stress lines are traced

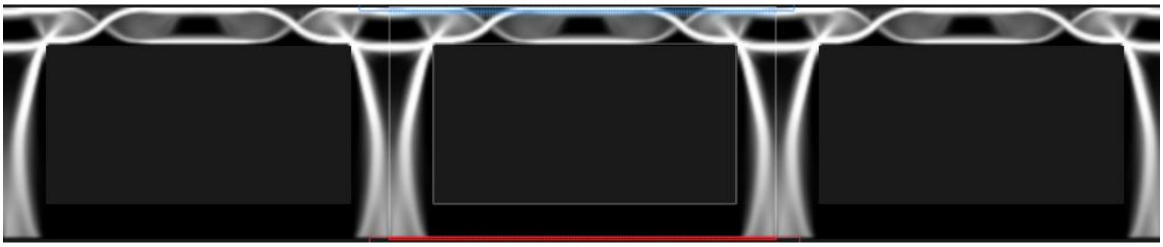
Outputs:

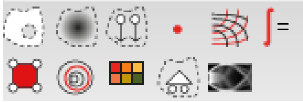
S1 [Polyline]: Principal stress line passing through point P along the 1st principal stress

S2 [Polyline]: Principal stress line passing through point P along the 2nd principal stress

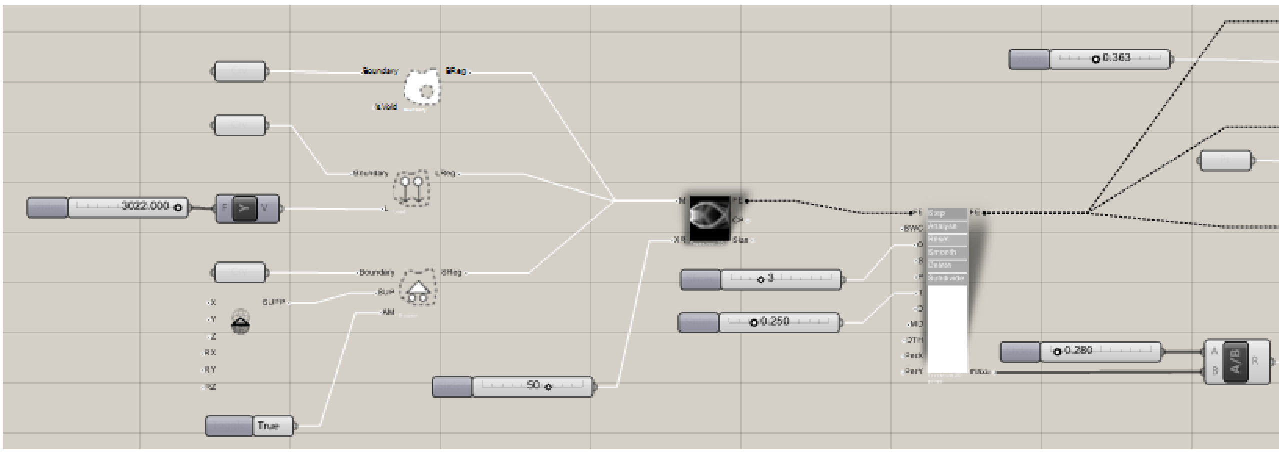
δ. 2D plates with in plane forces analysis and topology optimization







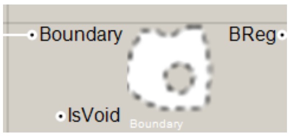
This is a toolbox for analysing and optimizing plates with in plane forces. The analysis is based on a “rasterization” of the problem domain producing a grid of square plate elements. The optimization is done using topology optimization methods.



Similarly to the previous section, a 2d system is constructed in 4 steps:

- Geometry and loads and boundary conditions definition
- System assembly
- Solver
- Results

1. Geometry constructors



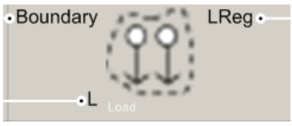
This component converts closed curves to boundaries [or holes] in the system

Inputs:

Boundary [Curve]: Closed planar curves to be converted to boundary elements determining the domain of the problem.

IsVoid [Boolean]: If true the boundary designates a hole otherwise it is part of the domain

2. Region Property modifiers

a. 

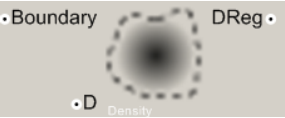
Load Region

Designate a region where area loads are applied

Inputs:

Boundary [Curve]: Closed curve defining the region within which the loads are applied

L [Vector3d]: Load, only the X and Y components are used

b. 

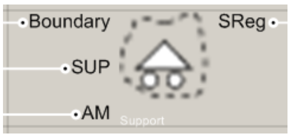
Fixed Density Region

Designate a region where material density is constant [will remain unaffected during optimization]

Inputs:

Boundary [Curve]: Closed curve defining the region within which the density is fixed

D [Number]: Density as a number from 0.0 [no material] to 1.0 [full material]

c. 

Support Region

Designate a region where certain boundary conditions apply

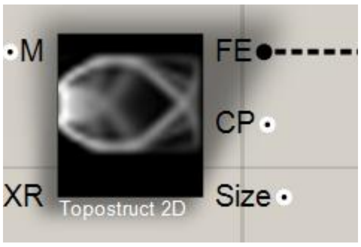
Inputs:

Boundary [Curve]: Closed curve defining the region within which to apply boundary conditions

SUP [Integer support code]: A support type to be applied to all nodes within the boundary

AM [Boolean]: A value indicating whether support types should have an additive behaviour or just replace whatever support is already there.

3. System builder



This is the main model builder tool. It aggregates all the information from the previous components, rasterizes the domain and generates the finite element model.

Inputs:

M [ ]: Any number of regions with properties chosen form the topostruct 2D toolbar can be connected at this input.

XR [Integer]: The resolution along the x axis. This is the number of square plate elements to generate along the x axis of the bounding box of the input geometries. This number can dramatically increase the time and memory *requirements of the solver. Values between 16 and 150 are generally good depending on the trade-off between speed and detail of results.*

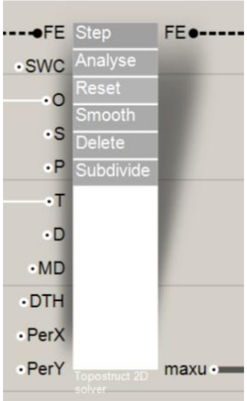
Outputs:

FE [Topostruct 2D model]: The analysis model

CP [list of Points]: A list of the centers of all the generated quad elements.

Size [Number]: The side length of each generated element

4. Solver



The solver is the core analysis component in this toolbox. It needs to be connected to a system builder component from which it acquires the finite element system.

Inputs:

FE [Topostruct 2D model]: A model provided from a system builder component

SWC [Number]: Self Weight Coefficient. A number that multiplies all the loads due to the self weight of the structure. For most optimization problems it should be set to 0 as it can induce a non-convergent feedback behavior in the optimization algorithm [as material gets redistributed the weight distribution changes as well which changes the optimal solution and so on…]. However usually it can generate interesting fine structures.

O [Integer]: Number of optimization iteration steps each time the user presses the "Step" button.

S [Number]: Smoothing factor applied at each iteration to avoid the generation of checker patterns. A number between 0.0 [no smoothing] and 1.0.

P [Number]: Penalization factor. The larger this number the fastest [fewer steps] the optimization will converge to a black and white state. However making it too high it can fail to redistribute material properly. A value from 1.5 to 3.0 is recommended.

T [Number]: Target Density percentage. A number from 0.0 to 1.0 designating the fraction of the overall material that should remain at the end of the optimization process. This is just a hint and not guaranteed to be achieved [you may want to tweak the magnitude of your applied loads relative to the scale of the system etc.…]. It is also a percentage of the bounding box of the system rather than the material enclosed within the bounding regions [that may be curved or irregular].

D [list of Numbers]: A List of numbers, one for each quad element in the model that designates some initial material density distribution. These numbers should be in 1 to 1 correspondence with the points generated at the CP output of the component.

MD [Number]: Minimum allowed density. The system should never have elements with density exactly 0.0 as that would make the analysis step fail. Instead the minimum density is capped at this value. 0.01 or 0.001 are generally good values.

DTH [Number]: Used in combination with the "Delete" button and defines the density value below which all quads are eliminated in order to speed up further steps.

PerX & PerY [Boolean]: These values determine whether the domain is assumed to be periodic in any direction. Useful when generating structural patterns [material design, or facades with periodic supports and loading conditions]. Setting any of these values to true will assume that the system exactly repeats on either side at the specified direction.

Outputs:

FE [Topostruct 2D model]: The analysed Finite Element system to be passed to other components that deal with visualization and recovery of specific results.

DEF [Number]: The maximum deflection in meters of the analysed structure. Useful for quick troubleshooting and getting a quick rough measure of system performance without doing detailed analysis of the results. If this number is too big there may be disconnected elements in the system and if it is zero you may haven't applied loads correctly.

Buttons:

Step: Each time you press this button O number of optimization iterations are executed

Analyze: Analyze the current model with its current density distribution without attempting any optimization

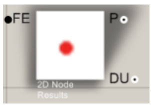
Reset: Reset the densities to either a homogeneous value or to the values provided at the D input discarding any optimization redistribution.

Smooth: Smooth the material distribution

Delete: Delete all elements that fall below the threshold defined by the DTH input. When you optimize you often see large parts of the structure discarded from the first few steps and which are unlikely to get any material assigned to them later. You can remove these element sin order to speed up calculations or increase the resolution of the remaining elements.

Subdivide: subdivide all elements in the system doubling the resolution [allowing you to see finer structures] and quadrupling the number of elements [and the time and memory required]

5. Results



d. Node results

This component returns the locations and displacements for all nodes.

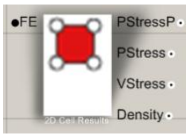
Inputs:

FE [Topostruct 2D model]: The analysed Finite Element system coming from the solver component

Outputs:

P [Point3d]: The locations of nodes

DU [Vector3d]: The displacements of all the nodes [m]



e. Plate element results

This component generates the results for all plate elements in the system.

Inputs:

FE [Topostruct 2D model]: The analysed Finite Element system coming from the solver component ∫=

Outputs:

PStressP [Plane]:   A plane with its origin at the center of each quad and its axes aligned with the principal stress directions

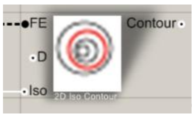
PStress [Vector]: The X and Y components of these vectors hold the 1st and 2nd principal stresses [along the X and Y axes of the corresponding PStressP planes]. Positive values designate tension and negative compression

VStress [Number]: The von Mises stress for each quad. This is a measure of the overall stress at each point combining the values of both principal stresses.

Density [Number]: A number from 0.0 to 1.0 designating the redistributed material density at each quad. 1.0 means solid and 0.0 void.

6. Visualization

Visualization components help clarify the structural behaviour of the system



f. Contours

This component draws contours along lines of constant material density.

Inputs:

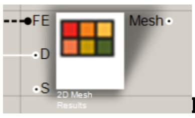
FE [Topostruct 2D model]: The analysed Finite Element system coming from the solver component ∫=

D [Number]: Deflection factor. This is used in order to exaggerate the deformation of the material at its equilibrium state. Useful making sure that all loads and supports are applied correctly and there aren't any disconnected components in the system.

Iso [Number]: contour value. This value is used in order to determine the density levels at which contours are drawn. For example a value of 0.5 means that a contour will be drawn that separates material densities less than 0.5 to material densities greater than 0.5.

Outputs:

Contour: A set of line segments forming a contour



g. Mesh visualization

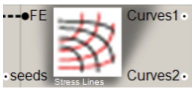
This component visualizes the material distribution and stresses as a mesh

Inputs:

FE [Topostruct 2D model]: the analysed finite elements system coming from the solver component ∫=

D [Number]: Deflection factor. This is used in order to exaggerate the deformation of the material at its equilibrium state. Useful making sure that all loads and supports are applied correctly and there aren't any disconnected components in the system.

S [Boolean]: if this value is true then the mesh will be coloured according to the magnitude of stress, if it is false then the mess will display the material density as a grayscale gradient.

h. 

Stress lines

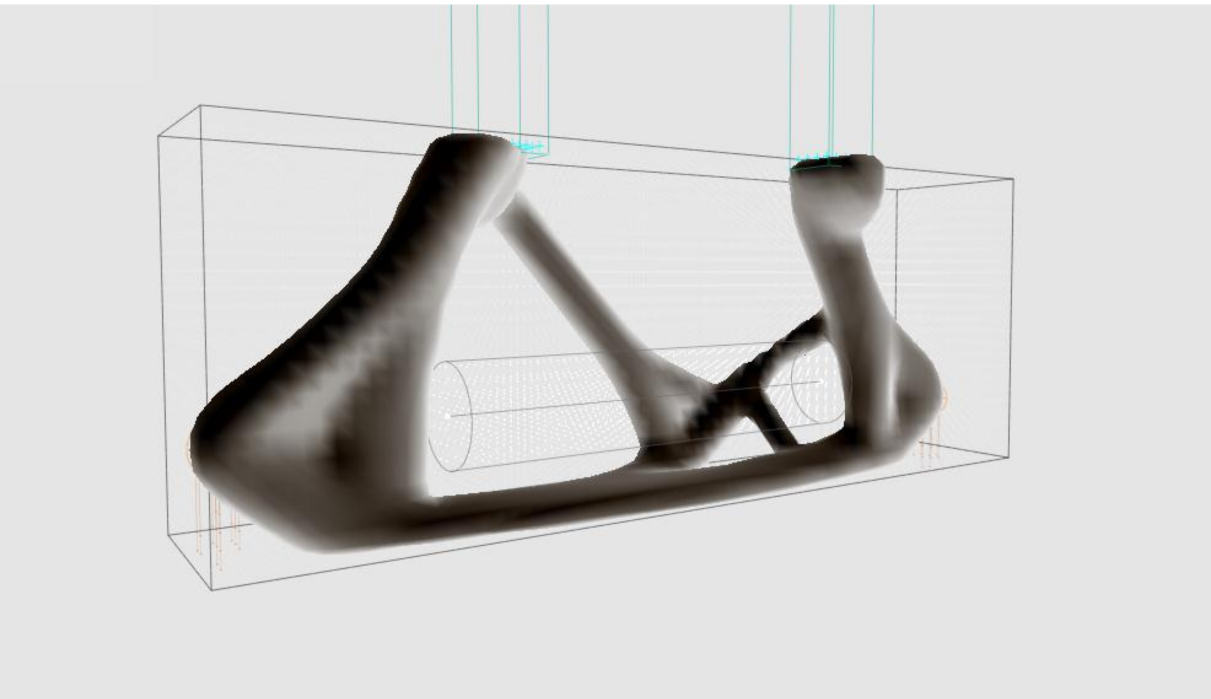
This component visualizes the stress lines in the analysed plate

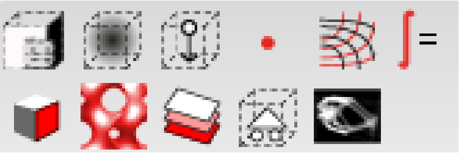
Inputs:

FE [Topostruct 2D model]: the analysed finite elements system coming from the solver component ∫=

Seeds [Point]: Seed points from which stress lines are traced

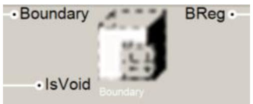
ε. 3D volumetric analysis and topology optimization





This is a toolbox for analysing and optimizing volumetric models of variable material density. The analysis is based on a “voxelization” of the problem domain producing a three dimensional grid of cubic volume elements. The optimization is done using topology optimization methods.

1. Geometry constructors

a. 

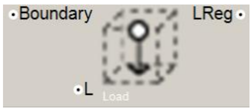
This component converts closed solids to boundaries [or holes] in the system

Inputs:

Boundary [Solid]: Closed solids to be converted to boundary elements determining the domain of the problem.

IsVoid [Boolean]: If true the boundary designates a hole otherwise it is part of the domain

2. Region Property modifiers



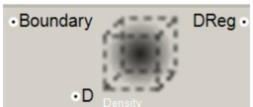
b. Load Region

Designate a region where volume loads are applied

Inputs:

Boundary [Solid]: Closed solid defining the region within which the loads are applied

L [Vector3d]: Load in [N/m3]



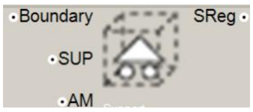
c. Fixed Density Region

Designate a region where material density is constant [will remain unaffected during optimization]

Inputs:

Boundary [Solid]: Closed solid defining the region within which the density is fixed

D [Number]: Density as a number from 0.0 [no material] to 1.0 [full material]



d. Support Region

Designate a region where certain boundary conditions apply

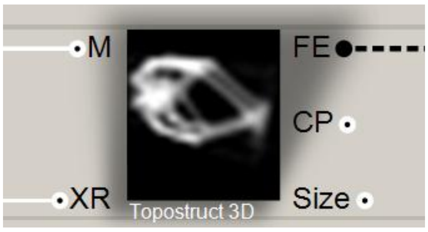
Inputs:

Boundary [Solid]: Closed solid defining the region within which to apply boundary conditions

SUP [Integer support code]: A support type to be applied to all nodes within the boundary

AM [Boolean]: A value indicating whether support types should have an additive behaviour or just replace whatever support is already there.

3. System builder



This is the main model builder tool. It aggregates all the information from the previous components, voxelizes the domain and generates the finite element model.

Inputs:

M [ ]: Any number of regions with properties chosen form the topostruct 3D toolbar can be connected at this input.

XR [Integer]: The resolution along the x axis. This is the number of cubic volume elements to generate along the x axis of the bounding box of the input geometries. This number can dramatically increase the time and memory requirements of the solver. Values between 12 and 40 are generally good depending on the trade-off between speed and detail of results.

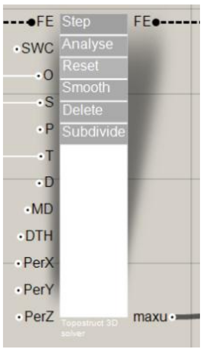
Outputs:

FE [Topostruct 3D model]: The analysis model

CP [list of Points]: A list of the centers of all the generated volume elements.

Size [Number]: The side length of each generated element

4. Solver



The solver is the core analysis component in this toolbox. It needs to be connected to a system builder component from which it acquires the finite element system.

Inputs:

FE [Topostruct 3D model]: A model provided from a system builder component

SWC [Number]: Self Weight Coefficient. A number that multiplies all the loads due to the self weight of the structure. For most optimization problems it should be set to 0 as it can induce a non-convergent feedback behavior in the optimization algorithm [as material gets redistributed the weight distribution changes as well which changes the optimal solution and so on…]. However usually it can generate interesting fine structures.

O [Integer]: Number of optimization iteration steps each time the user presses the "Step" button.

S [Number]: Smoothing factor applied at each iteration to avoid the generation of checker patterns. A number between 0.0 [no smoothing] and 1.0.

P [Number]: Penalization factor. The larger this number the fastest [fewer steps] the optimization will converge to a black and white state. However making it too high it can fail to redistribute material properly. A value from 1.5 to 3.0 is recommended.

T [Number]: Target Density percentage. A number from 0.0 to 1.0 designating the fraction of the overall material that should remain at the end of the optimization process. This is just a hint and not guaranteed to be achieved [you may want to tweak the magnitude of your applied loads relative to the scale of the system etc….]. It is also a percentage of the bounding box of the system rather than the material enclosed within the bounding regions [that may be curved or irregular].

D [list of Numbers]: A List of numbers, one for each volume element in the model that designates some initial material density distribution. These numbers should be in 1 to 1 correspondence with the points generated at the CP output of the component.

MD [Number]: Minimum allowed density. The system should never have elements with density exactly 0.0 as that would make the analysis step fail. Instead the minimum density is capped at this value. 0.01 or 0.001 are generally good values.

DTH [Number]: Used in combination with the "Delete" button and defines the density value below which all quads are eliminated in order to speed up further steps.

PerX & PerY & PerZ [Boolean]: These values determine whether the domain is assumed to be periodic in any direction. Useful when generating structural.

Outputs:

FE [Topostruct 3D model]: The analysed Finite Element system to be passed to other components that deal with visualization and recovery of specific results.

DEF [Number]: The maximum deflection in meters of the analysed structure. Useful for quick troubleshooting and getting a quick rough measure of system performance without doing detailed analysis of the results. If this number is too big there may be disconnected elements in the system and if it is zero you may haven't applied loads correctly.

Buttons:

Step: Each time you press this button O number of optimization iterations are executed

Analyze: Analyze the current model with its current density distribution without attempting any optimization

Reset: Reset the densities to either a homogeneous value or to the values provided at the D input discarding any optimization redistribution.

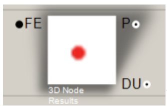
Smooth: Smooth the material distribution

Delete: Delete all elements that fall below the threshold defined by the DTH input. When you optimize you often see large parts of the structure discarded from the first few steps and which are unlikely to get any material assigned to them later. You can remove these elements in order to speed up calculations or increase the resolution of the remaining elements.

Subdivide: subdivide all elements in the system doubling the resolution [allowing you to see finer structures] and increases the number of elements by a factor of 8 [and the time and memory required]

5. Results

e. Node results



This component returns the locations and displacements for all nodes.

Inputs:

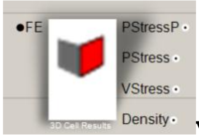
FE [Topostruct 3D model]: The analysed Finite Element system coming from the solver component

Outputs:

P [Point3d]: The locations of nodes

DU [Vector3d]: The displacements of all the nodes [m]

f. Volume element results



This component generates the results for all volume elements in the system.

Inputs:

FE [Topostruct 3D model]: The analysed Finite Element system coming from the solver component ∫=

Outputs:

PStressP [Plane]: A plane with its origin at the center of each voxel and its axes aligned with the three principal stress directions

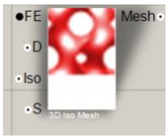
PStress [Vector]: The X, Y and Z components of these vectors hold the 1st , 2nd and 3rd principal stresses [along the X, Y and Z axes of the corresponding PStressP planes]. Positive values designate tension and negative compression

VStress [Number]: The von Mises stress for each quad. This is a measure of the overall stress at each point combining the values of both principal stresses.

Density [Number]: A number from 0.0 to 1.0 designating the redistributed material density at each quad. 1.0 means solid and 0.0 void.

6. Visualization

Visualization components help clarify the structural behaviour of the system



g. Contours

This component draws contours along surfaces of constant material density.

Inputs:

FE [Topostruct 3D model]: The analysed Finite Element system coming from the solver component ∫=

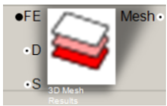
D [Number]: Deflection factor. This is used in order to exaggerate the deformation of the material at its equilibrium state. Useful making sure that all loads and supports are applied correctly and there aren't any disconnected components in the system.

Iso [Number]: contour value. This value is used in order to determine the density levels at which contours are drawn. For example a value of 0.5 means that a contour will be drawn that separates material densities less than 0.5 to material densities greater than 0.5.

S [Boolean]: if this value is true then the mesh will be coloured according to the magnitude of stress.

Outputs:

Contour: A Mesh forming an iso surface through the material distribution



h. Mesh visualization

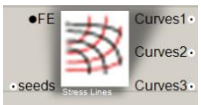
This component visualizes the material distribution and stresses as a series of horizontal meshes cutting through the volume of the domain.

Inputs:

FE [Topostruct 3D model]: the analysed finite elements system coming from the solver component ∫=

D [Number]: Deflection factor. This is used in order to exaggerate the deformation of the material at its equilibrium state. Useful making sure that all loads and supports are applied correctly and there aren't any disconnected components in the system.

S [Boolean]: if this value is true then the mesh will be coloured according to the magnitude of stress, if it is false then the mess will display the material density as a grayscale gradient.



i. Stress lines

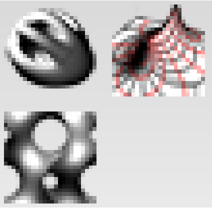
This component visualizes the stress lines in the analysed plate

Inputs:

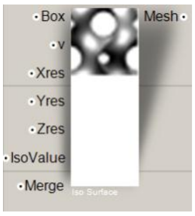
FE [Topostruct 3D model]: the analysed finite elements system coming from the solver component

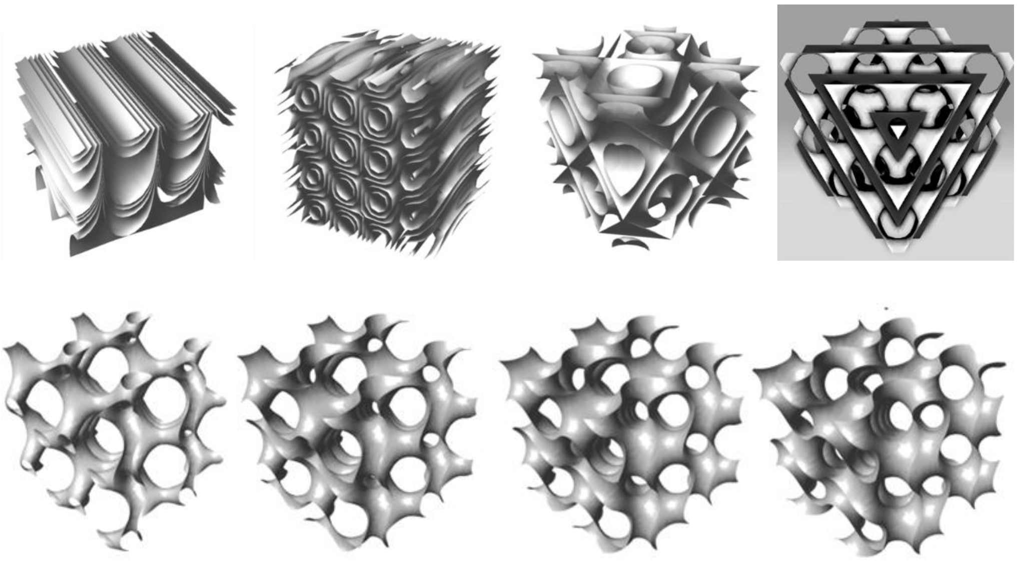
Seeds [Point]: Seed points from which stress lines are traced

στ. Geometry Utilities



1. Marching Cubes Iso surfaces





This utility uses the marching cubes algorithms in order to extract iso-surfaces from a field of density values. These values must be defined on a three dimensional grid and can be generated through simple function [F(x,y,z)] or using the Geometry wrapper component below.

Inputs:

Box [Bounding Box]: The bounding box of the geometry

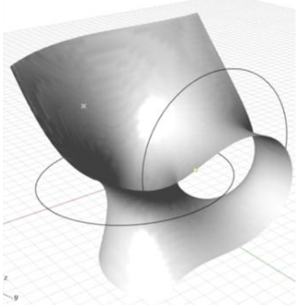
V [List of Numbers]: This list must contain Xres x Yres x Zres numbers designating the field values at each point of the notional grid inside Box. The values should be ordered in such a way so that x direction is traversed first then y and then the z layers.

Xres, Yres, Zres [Integer]: The resolution of the three dimensional grid.

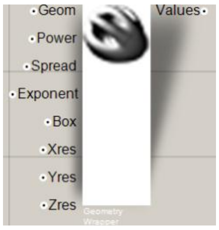
IsoValue [Number]: The threshold at which to cut an iso surface through the density field.

Merge [Boolean]: If true the resulting mesh will have its coinciding vertices fused and will look smoother [it does not alter the locations of vertices it just fuses coincident vertices so that normal and hence lighting calculations are continuous and not faceted]

2. Geometry Wrapper



This component creates grids of values that can be used in conjunction with the iso surface component above in order to generate smooth meshes that wrap around groups of any type of geometry [points, curves, solids]



Inputs:

Geom [Geometry]: Any number of points, curves and surfaces to compute a decaying density field around

Power [Number]: The strength of the field of each object. This is either a list of numbers [one for each geometric object] or a single value applied to all objects

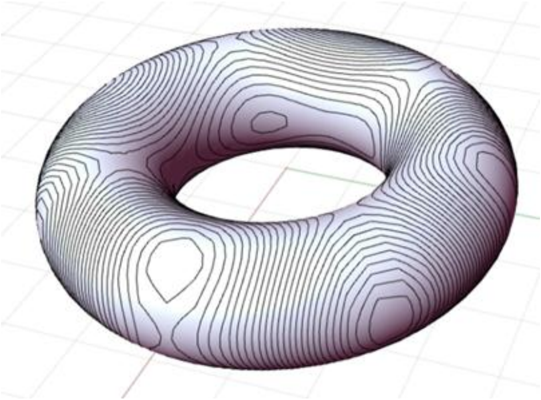
Spread [Number]: The spread of the field of each object. This is either a list of numbers [one for each geometric object] or a single value applied to all objects

Exponent [Number]: Not used

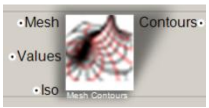
Box [Bounding box]: The bounding box within which the grid of values will be generated

Xres, Yres, Zres [Integer]: The resolution of the three dimensional grid.

3. Mesh contours



Given a mesh and a set of values defined at each vertex of the mesh, this component can extract contours from the density field of values.



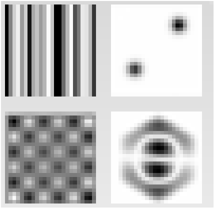
Inputs:

Mesh [Mesh]: A mesh over which the contours are distributed

Values [Number]: A list of numbers of equal length to the number of vertices in the mesh.

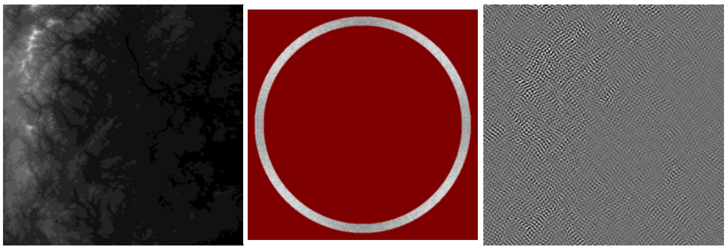
Iso [Number]: the threshold at which a contour is generated

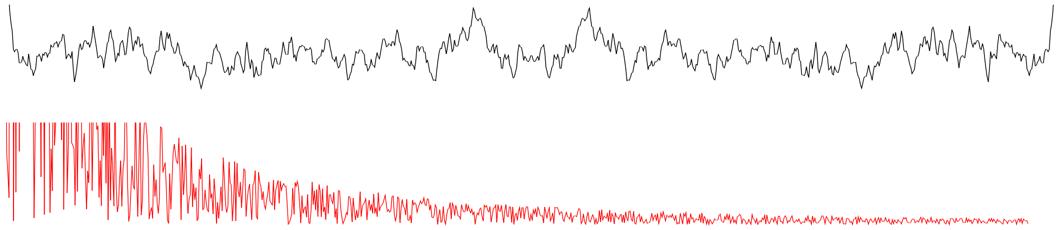
ζ. Numerical Utilities



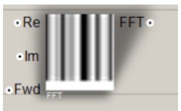
These utilities are interfaces to the Intel Math Kernel Library exposing some standard analytical tools.

1. FFT





These two components can calculate the Fast fourier transform of an input signal over 1 and two dimensions. For example in sound analysis the fourier transform of the wave form yields the spectrum of frequencies. You can also apply the 2d transform on images do some filtering and invert the transform to isolate certain scales in the image.



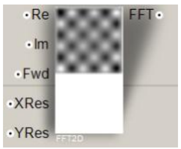
a. 1D FFT

Inputs:

Re [Numbers] : A list of numbers forming the real part of the signal

Im [Numbers]: A list of numbers forming the imaginary part of the signal [if this parameter is not linked to anything then the signal is assumed to have only real values].

Fwd [Boolean]: If true compute the forward Fourier transform otherwise compute the inverse Fourier transform



b. 2D FFT

Inputs:

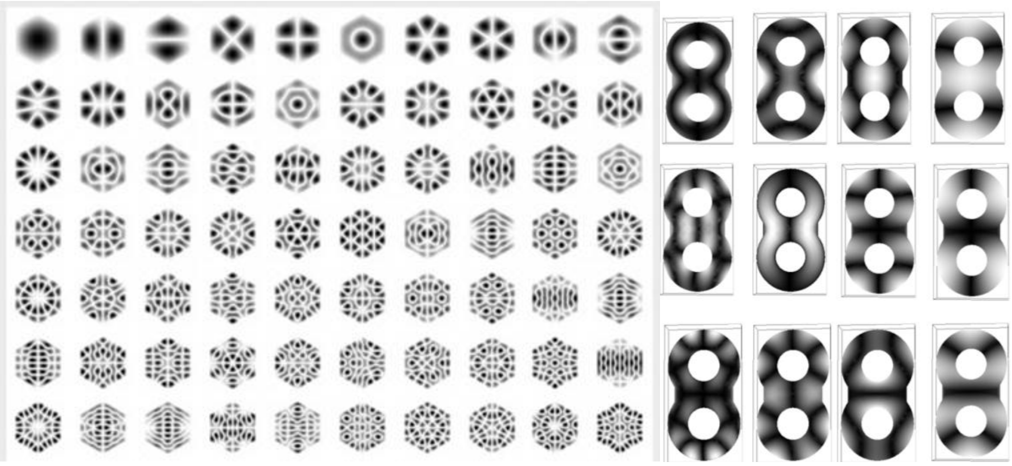
Re [Numbers] : A list of numbers forming the real part of the signal

Im [Numbers]: A list of numbers forming the imaginary part of the signal [if this parameter is not linked to anything then the signal is assumed to have only real values].

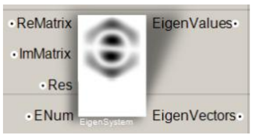
Fwd [Boolean]: If true compute the forward Fourier transform otherwise compute the inverse Fourier transform

XRes & YRes [Integer]: the dimensions of the two dimensional matrix that constitutes the input signal. The number of values in "Re" should be equal to XRes x YRes and the values should be in row major order.

2. EigenSystems



This component computes the eigenvalues and eigenvectors of a real symmetric or Hermitian matrix



Inputs:

ReMatrix [Numbers]: A Res x Res list of numbers representing the real part of the matrix [it must represent a symmetric real matrix]

ImMatrix [Numbers]: If this component is defined then it should contain a Res x Res list of numbers representing the imaginary part of a matrix [assuming a Hermitian matrix]. If this parameter is left undefined then the system is assumed to be real.

Res [Integer]: The dimension of the matrix [number of rows]

ENum [Integer]: Number of eigenvalues and eigenvectors to compute. It can be any integer number between 0 and Res. 0 means compute all [Res] eigenvectors.

Outputs:

EigenValues [Numbers]: A list of numbers representing the eigenvalues of the system

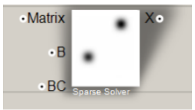
EigenVectors [Complex Numbers]: A tree with ENum branches and where each branch contains Res values corresponding to one eigenvector.

3. Sparse Linear solver

This component is an interface to the PARDISO solver in the intel Math Kernel Library. This is a sparse linear system solver. A linear system of equations can be represented as:

A.X=B

Where A is a n x n symmetric matrix of coefficients, B is n long vector of values and X is an n long vector of unknowns. In many problems the matrix A can easily become very big. For example for a structural analysis problem with just 10 nodes assuming 6 degrees of freedom per node the matrix would be 60x60. However in many engineering applications the matrix A is sparse, which means that many of its entries are 0 and we don't have to waste a lot of memory [often more than your average desktop system can handle] storing all those entries. Here's where sparse solvers are useful as they operate on a representation of the matrix that only stores non zero values.



Inputs:

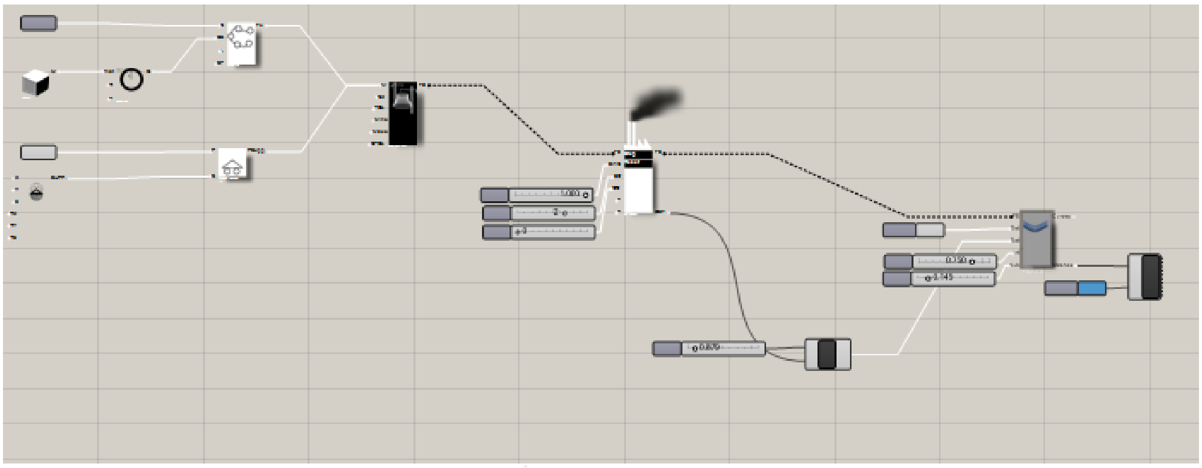
Matrix [Points]: A list of points representing entries to be added to the matrix. The X and Y coordinates represent the Row and column indices of the entry and the z coordinate is the actual value to be added to that entry.

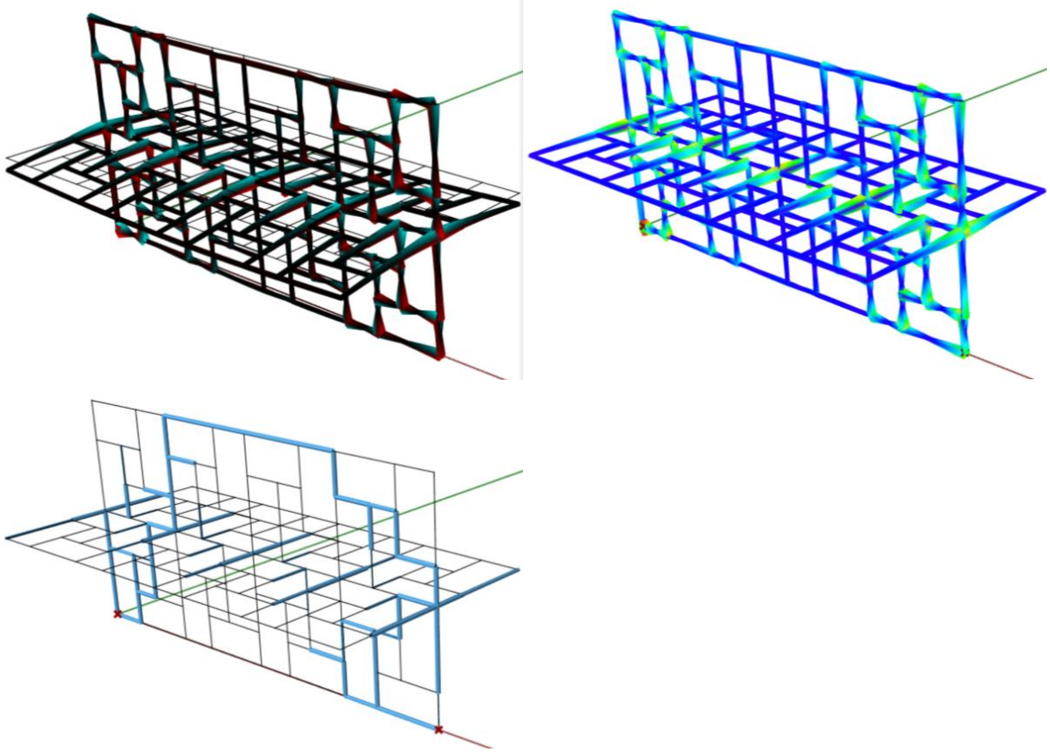
B [Numbers]: A List of numbers holding the right hand side of the system of equations. From the length of this list the algorithm determines the dimensions of the matrix.

BC [Points]: A list of points representing boundary conditions [usually known X values that can be replaced and fixed]. The x coordinate of each point is the index of the value to fix and the y coordinate is the value to be fixed to.

η. Samples

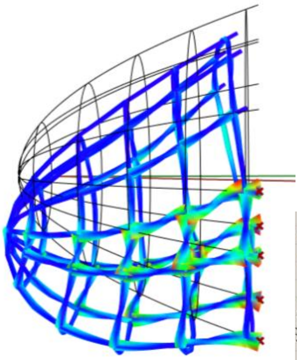
4. 000_simpleframes.gh

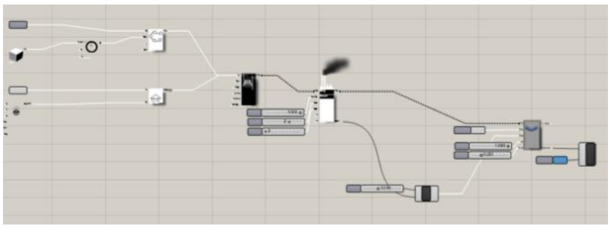




Using frame elements generated form a set of lines. Different visualizations of results and beam size optimization.

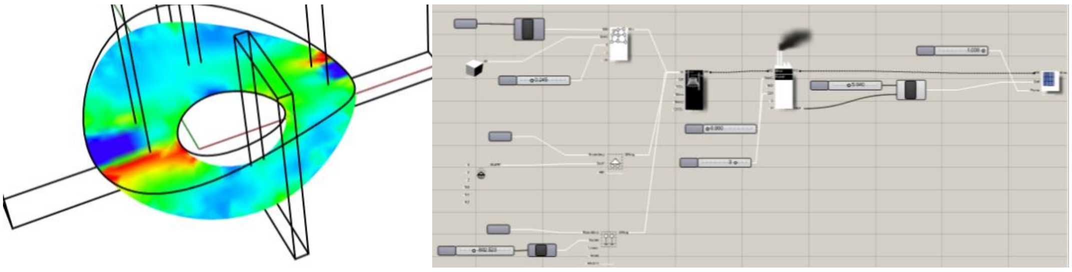
5. 001_Curvedframes.gh





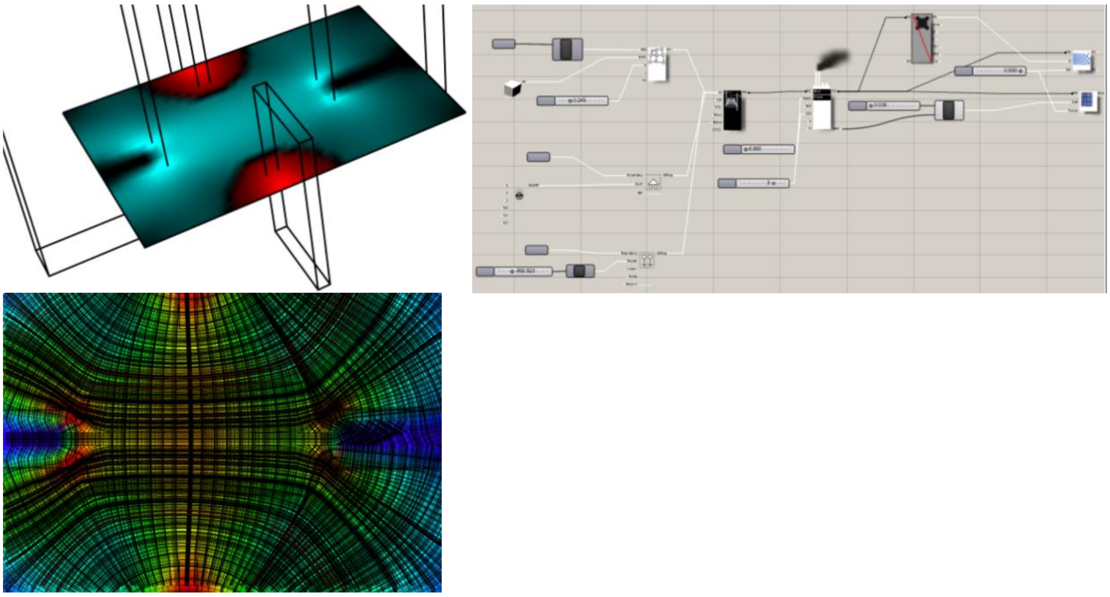
Using frame elements generated from a network of curves. Curves are automatically intersected and discretized by the FE system builder.

6. 002_ShellSurface.gh



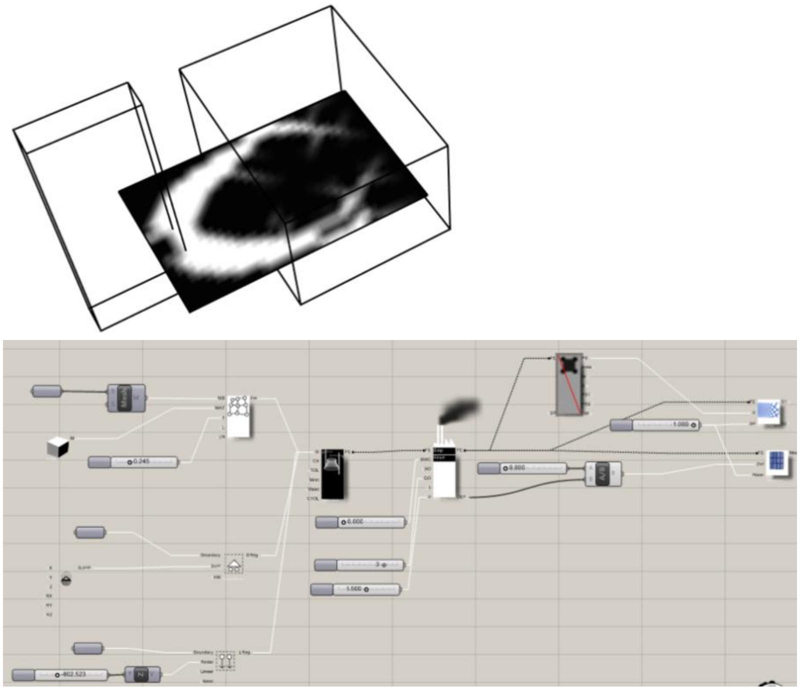
Using shell elements to analyse a surface.

7. 003_slab.3dm



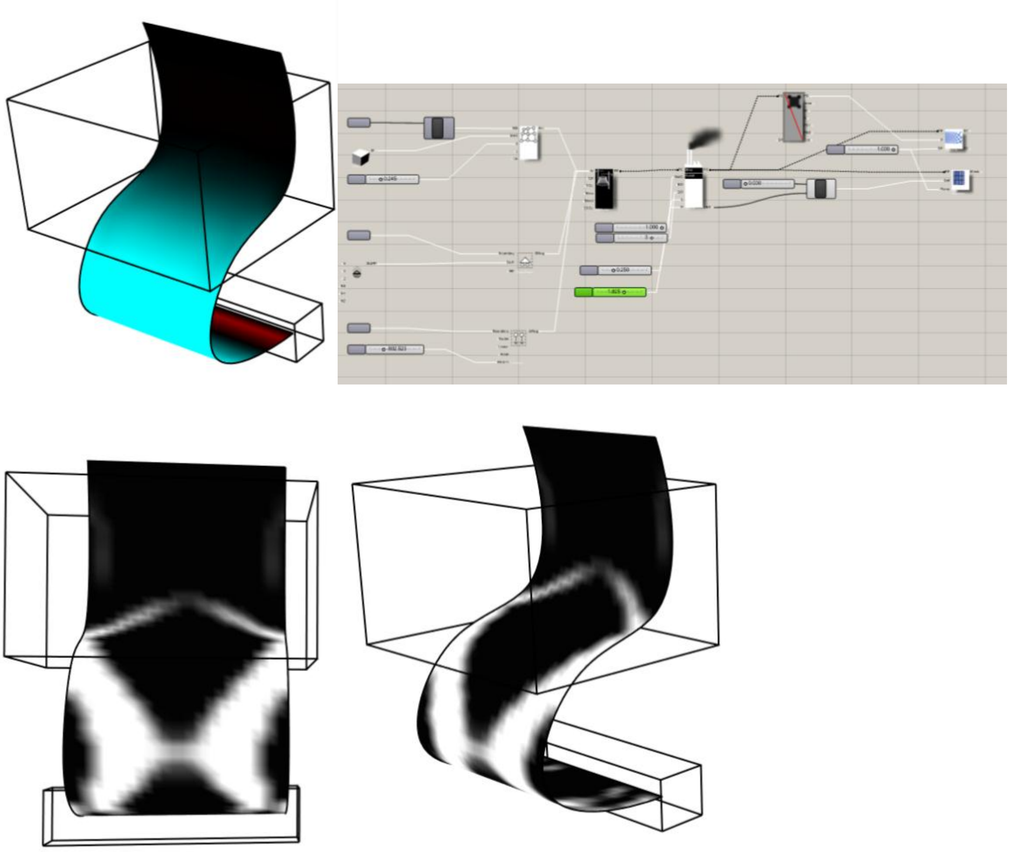
Shell element based analysis of slab and extraction of stress lines.

8. 004_leaf.gh



Cantilevering slab with thickness optimization. Branching structures appear.

9. 005_chair.gh

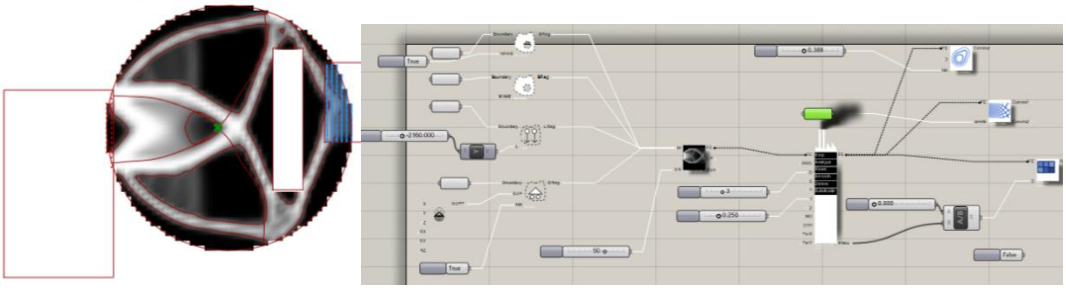


An s shaped chair and its thickness optimization diagram. Most material is concentrated on the bulging part that tends to fold on itself.

10. 006_topostruct2d.gh

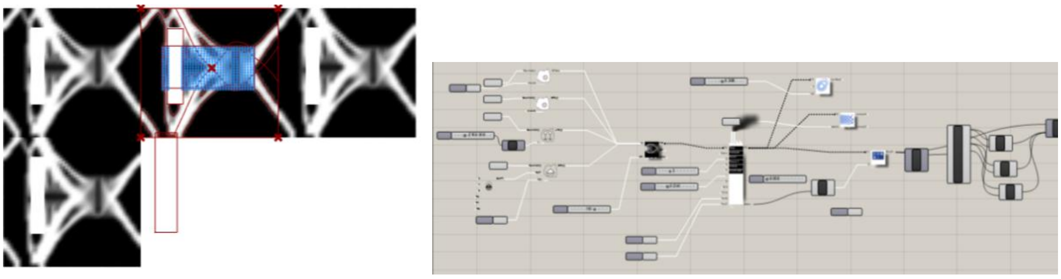
Estimated Time Remaini

```
Complete in 00:00:00.0620035
Complete in 00:00:00.0490028
Complete in 00:00:00.0490028
Complete in 00:00:00.0500028
OptTime :00:00:00.2200125
```



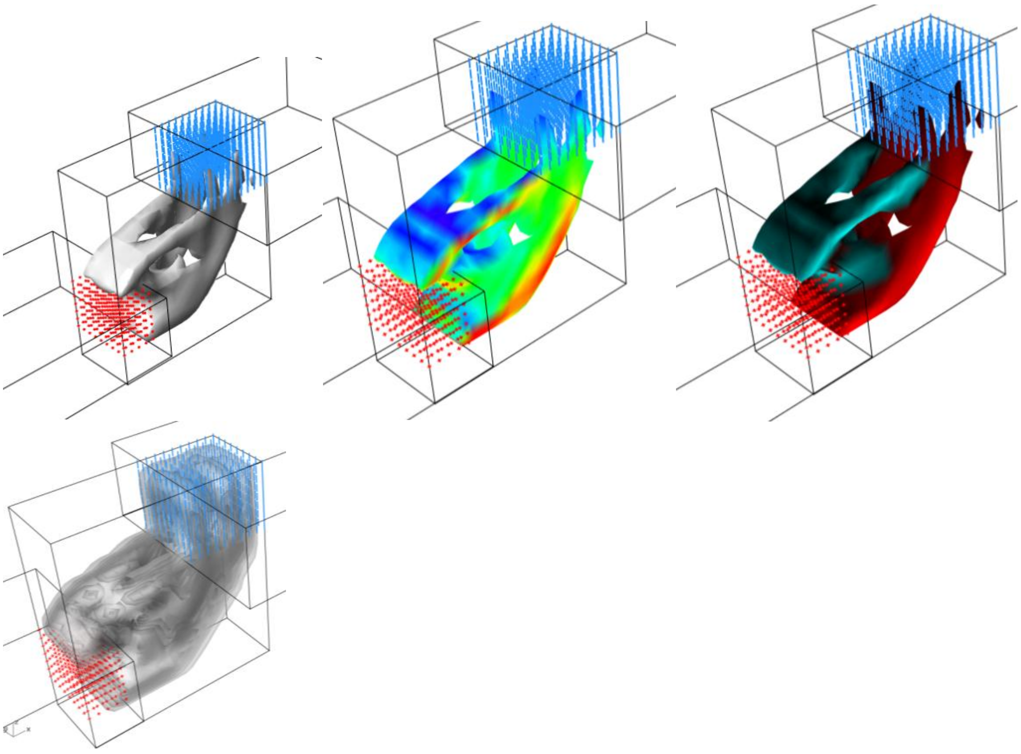
Simple two dimensional topology optimization

11. 007_topostruct2dPeriodic.gh



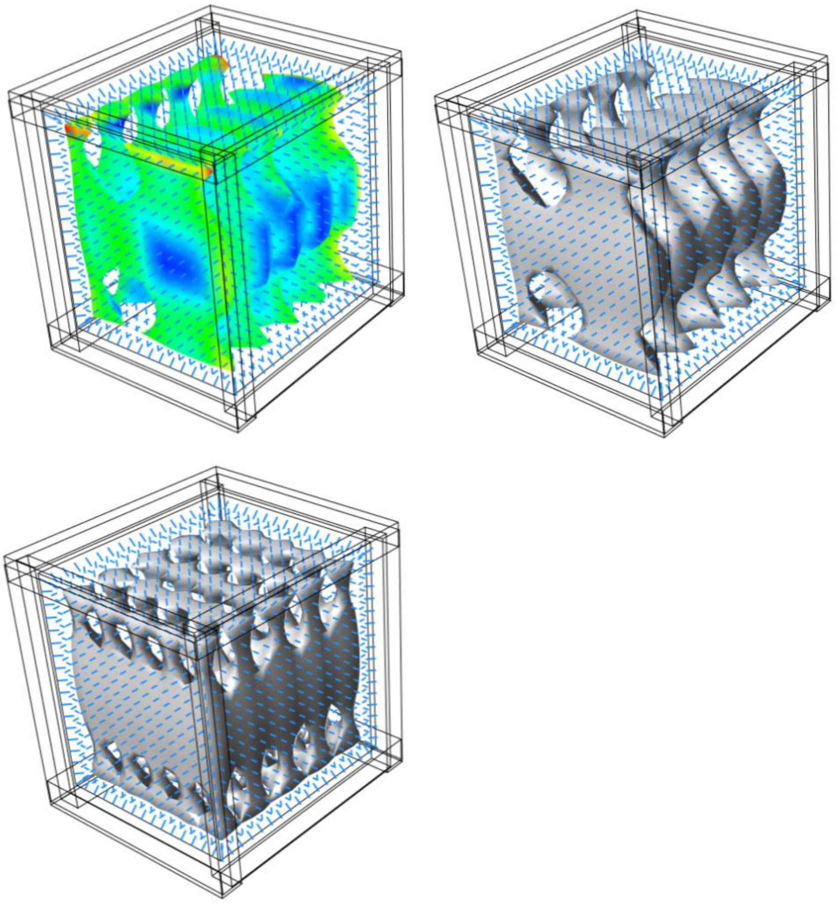
Periodic two dimensional topology optimization

12. 008_topostruct3d.gh



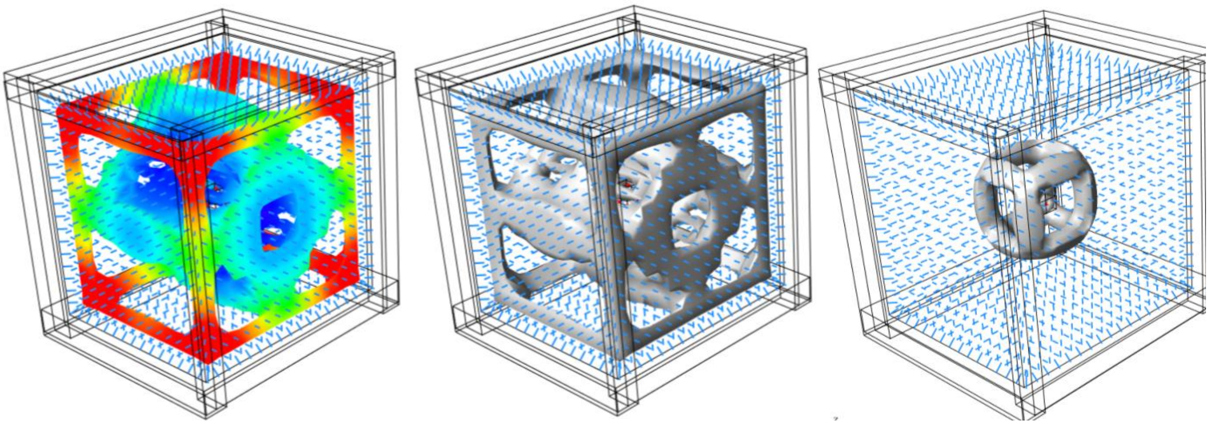
Basic three dimensional topology optimization.

13. 009_CustomDensityFieldAnalysis.gh



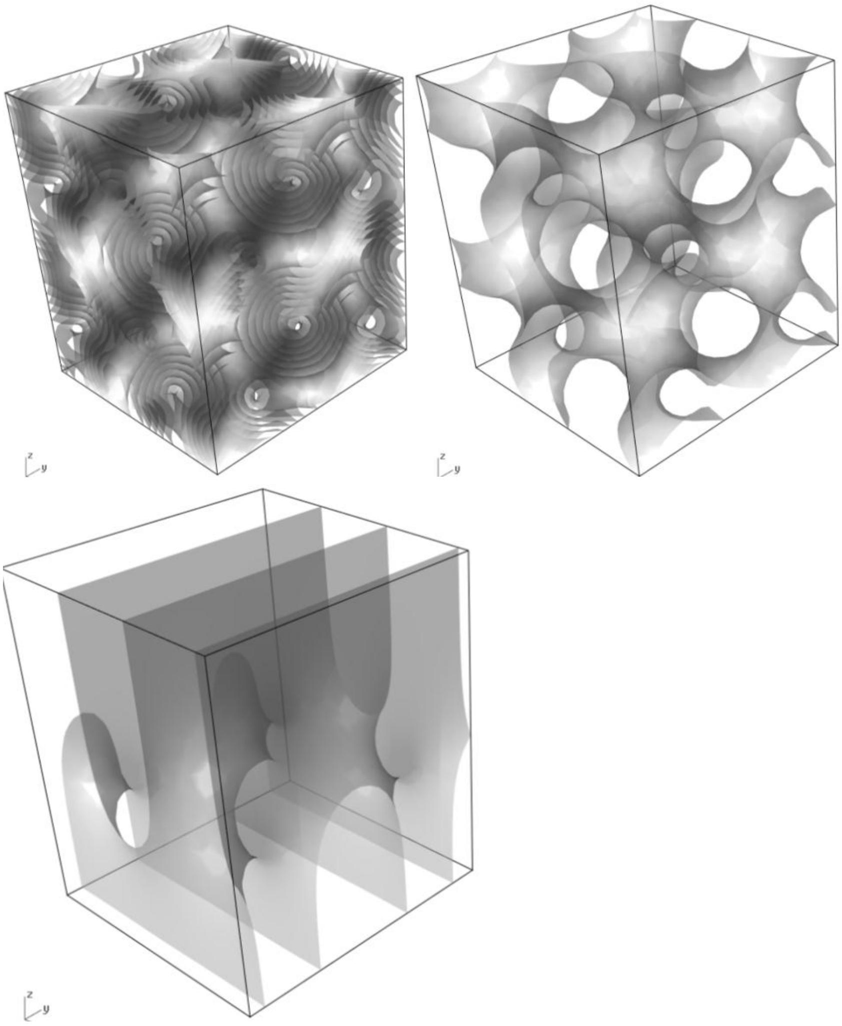
Optimization of inhomogeneous material distribution determined by a periodic density function.

14. 010_PressureCell.gh



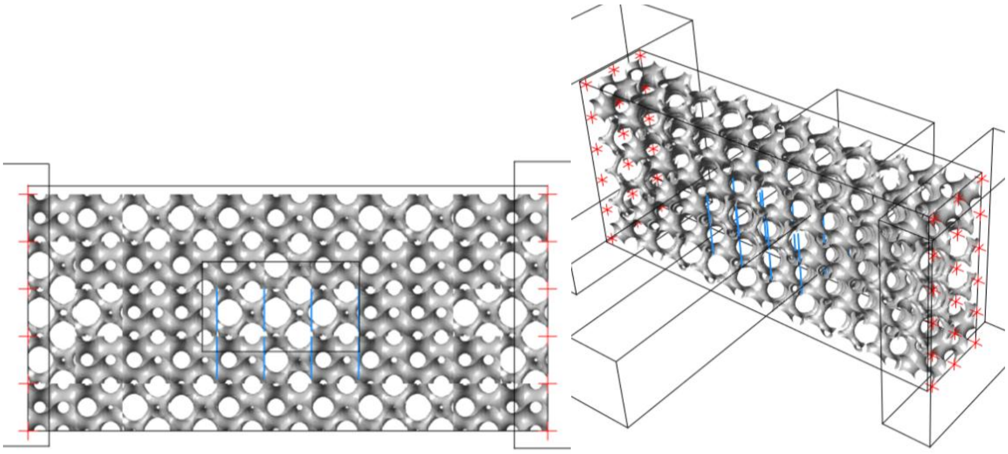
Optimization of cell under omnidirectional pressure

15. 011_isoSurfaces.gh



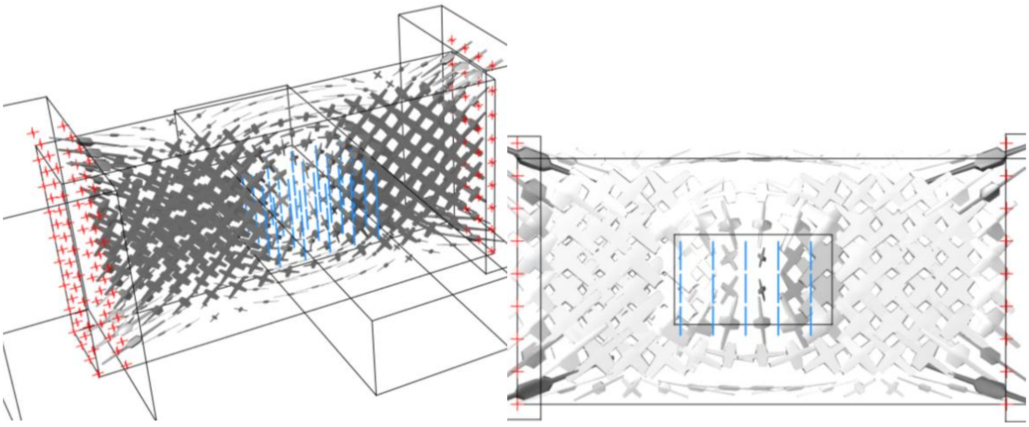
Generation of iso surfaces defined in implicit form [F[x,y,z]=const].

16. 012_TopostructPorosity.gh



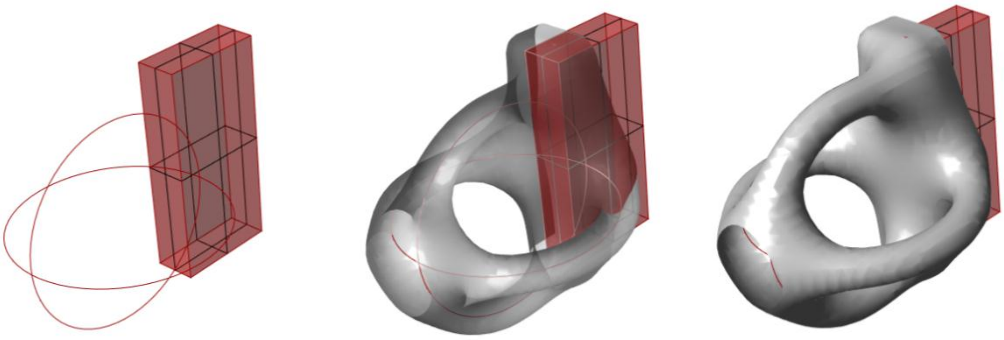
Use of iso surfaces of varying threshold to generate cells for variable porosity interpretation of topology optimization results.

17. 013_stressAlign.gh



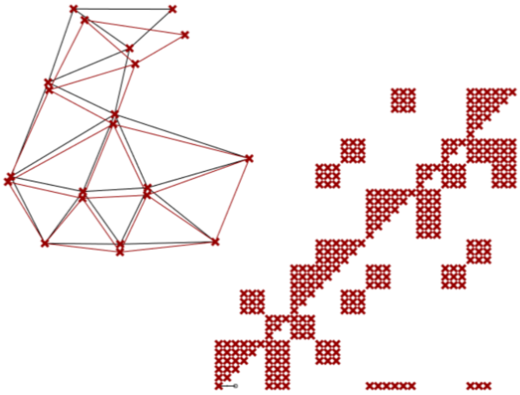
Visualization of principal stresses within a volume.

18. 014_object_contours.gh



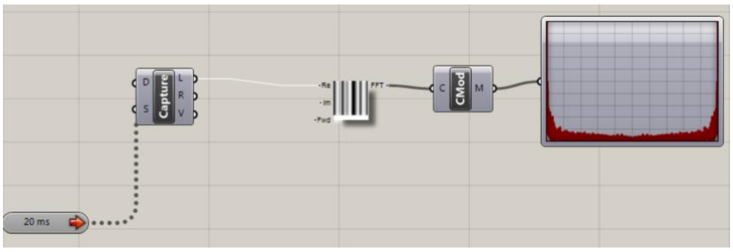
Using the object wrapper component to generate meshes that smoothly wrap around different types of geometry.

19. 015_SparseSolver.gh



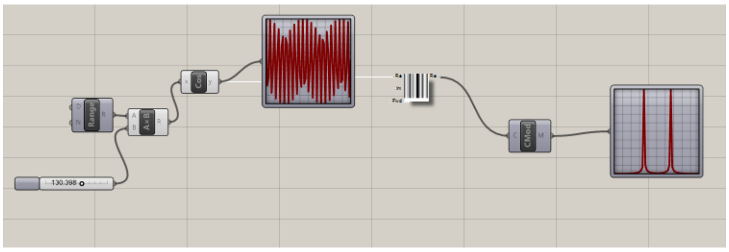
Use of the sparse linear system solver to simulate a basic truss [building the stiffness matrix etc…]

20. 016_FFTsound.gh



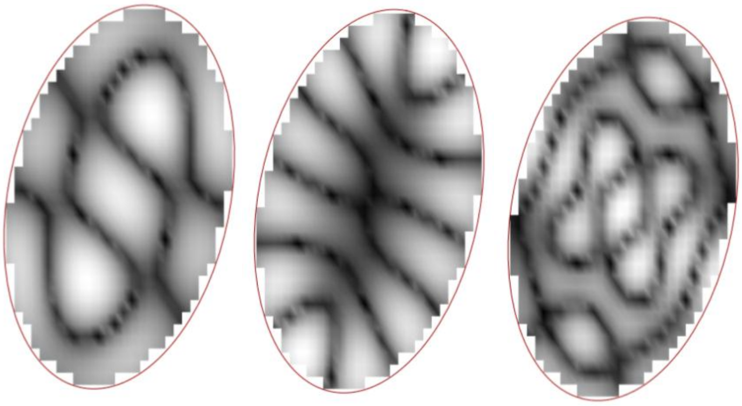
Use of the FFT component for the generation of the frequency spectrum of live sound [requires firefly].

21. 017_FFTSimple.gh



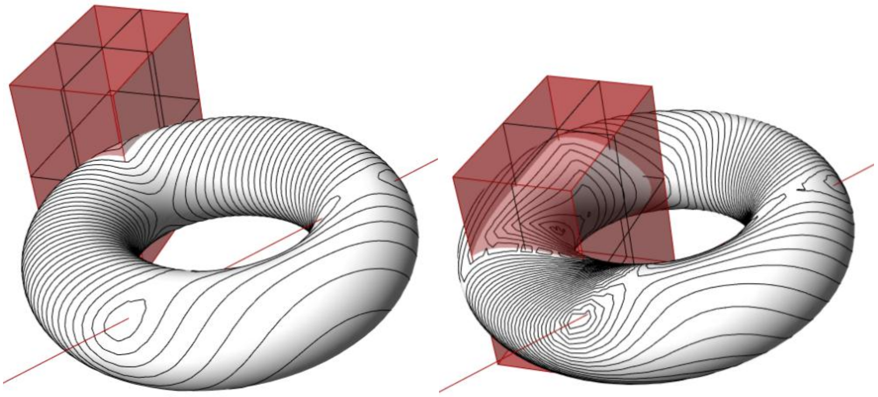
Use of FFT to analyse a sinusoidal wave form [notice the twin peaks at the frequency spectrum representing the pure tone]

22. 018_eigenfunctions.gh



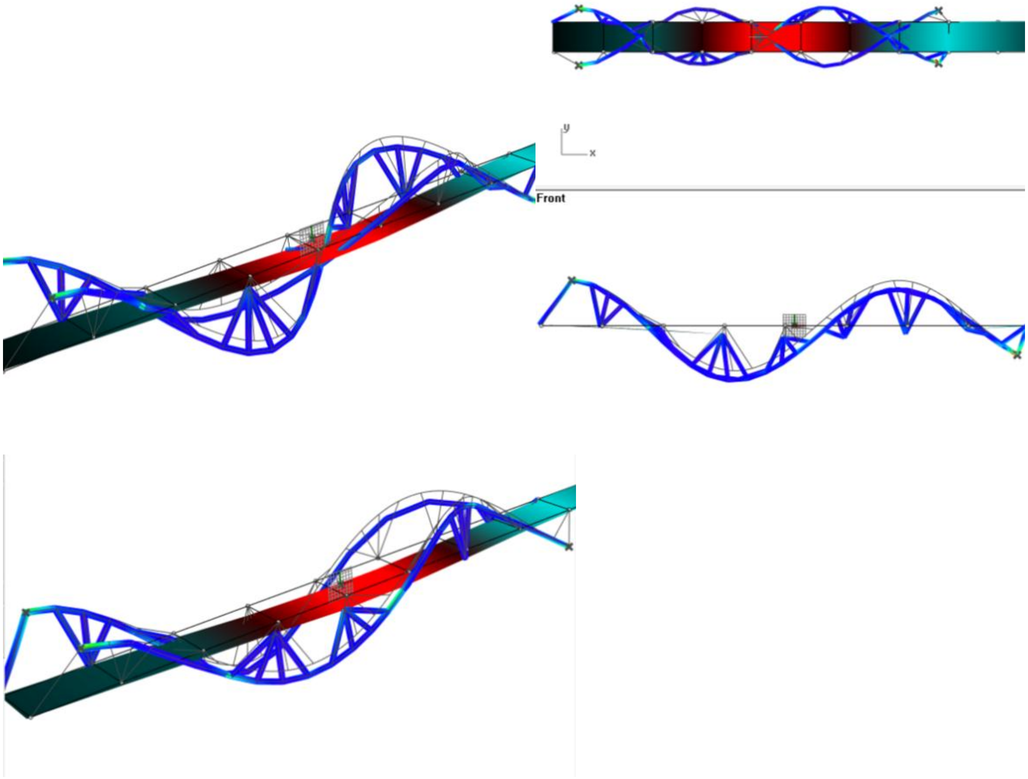
Eigenfunctions of the graph Laplacian of a mesh.

23. 019_contour_dist.gh



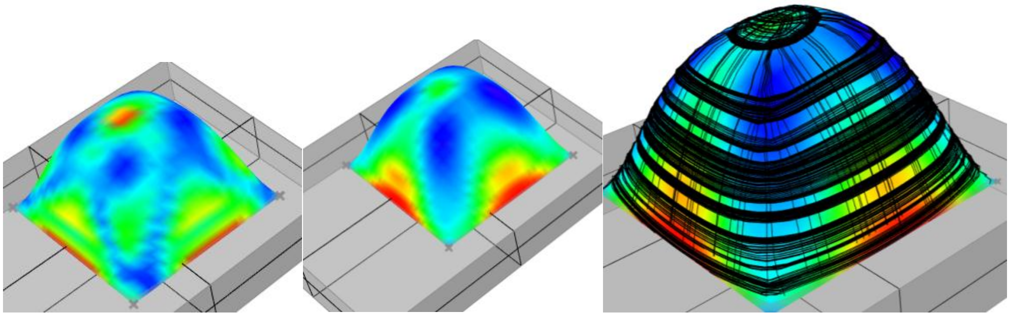
Use of the mesh contouring component for the generation of contours on a torus related to the distance from two object.

24. 020_bridge_02.gh



Use of the millipede structural analysis in conjunction with Galapagos for form finding.

25. 025_readingresults.gh



Shell analysis and stress lines.

0.Disclaimer

LICENSE AGREEMENT FOR Millipede

(Educational Use ONLY)

By downloading or using the software, You (either an individual or an entity), the End User, acknowledge that you have read and accepted the terms and conditions of this agreement. If you do not agree to the terms and conditions, do not download, install, or attempt to use the software.

The software is available only to academic, and other non-profit institutions for non-commercial, non-profit internal research purposes. Please note that the license terms specifically limit its use to such purposes. If you are interested in obtaining a commercial use license for this or other software tools from Panagiotis Michalatos and Sawako Kaijima, please send your name, address, fax and telephone numbers and email address, along with the name of the tool, to: sawapandesign@gmail.com.

YOU MAY:

(i) use the software for educational purposes in academic, and other non-profit institutions for non-commercial, non-profit internal research purposes.

YOU MAY NOT:

(i) sublicense, rent, sell, or lease any portion of the software; (ii) reverse engineer, de-compile, disassemble, modify, translate, make any attempt to discover the source code of the software

Ownership of the SOFTWARE.

Panagiotis Michalatos and Sawako Kaijima retains all right, title, and interest in the software (including all copies), and all worldwide intellectual property rights therein. Panagiotis Michalatos and Sawako Kaijima reserves all rights not expressly granted to Licensee. This License is not a sale of the original software or of any copy.

Termination.

Panagiotis Michalatos and Sawako Kaijima may terminate this Agreement immediately if Licensee breaches any provision. Upon notice of termination by Panagiotis Michalatos and Sawako Kaijima, all rights granted to Licensee under this Agreement will immediately terminate, and Licensee shall cease using the SOFTWARE and return or destroy all copies (and partial copies) of the SOFTWARE and documentation.

Disclaimer of warranty.

Licensee acknowledges that the Software is experimental. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS, Panagiotis Michalatos and Sawako Kaijima, AND CONTRIBUTORS [AS IS] AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Technical Support.

Panagiotis Michalatos and Sawako Kaijima have no obligation to furnish You with technical support.

No Other Obligations.

This Agreement creates no obligations on the part of Panagiotis Michalatos and Sawako Kaijima other than as specifically set forth herein.