1 Basic Documentation Kiwi!3d BETA

For Kiwi!3d BETA 0.5.0

1.1 General

Kiwi3d is a plugin for Rhino 6 and Grasshopper3d on Windows, which allows to perform isogeometric-based finite element analysis (IGA) based on NURBS parametrization. Detailed information can be found on kiwi3d.com.

The current version distributed, is a work-in-progress state (BETA status). Therefore, no warranty is provided by the developing team regarding results.

1.2 Setup

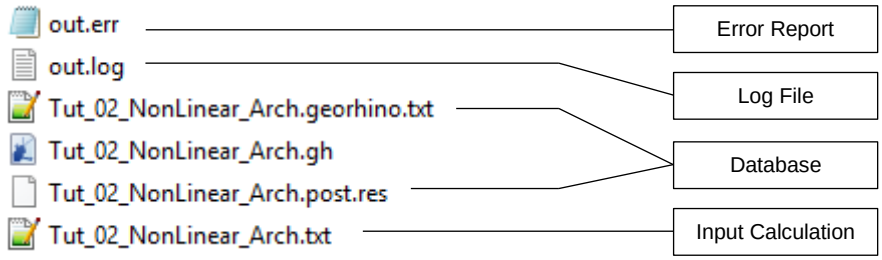
Kiwi3d can be installed by double-clicking the installer file (kiwi3d.rhi).

You can also install the software manually by the following steps:

- Rename installer file from .rhi to .zip
- You may have to unblock the .zip-folder in the file properties.
- Unzip the file
- Move the .gha file to your Grasshopper Libraries folder (in Grasshopper Menu: “File” -> “Special Folders” -> “Components Folder”)
- Install the .rhp file via the Rhino PlugInManager (in Rhino Menu: “Tools” -> “Options” -> “Rhino Options” -> “Plug-ins” -> “Install…”). Important: Do not move the .rhp and the carat.exe to another location after you have installed the plugin in Rhino).

1.3 Files

When working with Kiwi3d a bunch of files are appearing in your working folder. These are the following:



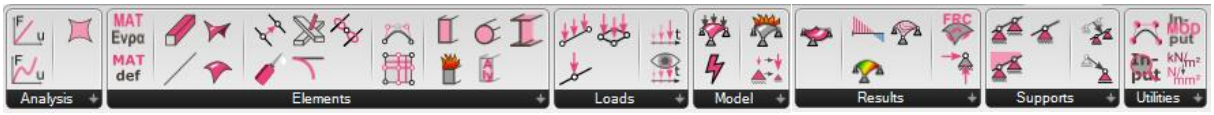
You should keep both database files to store analysis results. The name of the files depends on the name you choose for the analysis (see analysis type components in 1.4.2).

1.4 Components

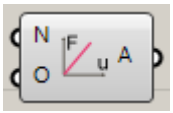
1.4.1 The Kiwi3d Library

The Library contains 7 Categories:

- Analysis
- Elements
- Loads
- Model
- Results
- Supports
- Utilities

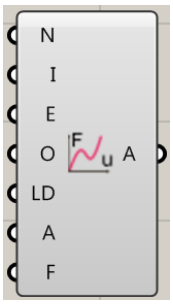


1.4.2 Analysis Components



Linear Analysis

Sets a linear analysis task (I. Order Theory).
N = name of analysis task, ***O*** = defines which results will be outputted for visualization (postprocessing). To set O, connect a „Value List“ component (GH standard).



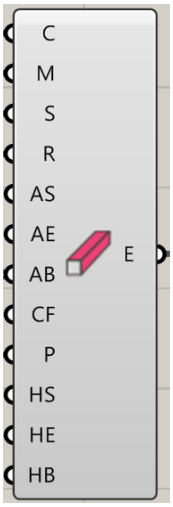
Non-Linear Analysis

Sets a non-linear analysis task and the control parameters for incremental solution process. Currently the application of loads is based on force-controlled solution schemes.
N = name of analysis task. **I** = maximum number of Iterations per time step of LD-curve.
E = accuracy for equilibrium, **O** = defines which results will be outputted for visualization (postprocessing). To set O, connect a „Value List“ component (GH standard). **LD** = add load-displacement curve to control the application of loads via time steps and load values (see *LD-Curve Component*), **A** = maximum number of bisections of the load increment. If the time step can‘t be solved within the defined iterations, the actual load increment is bisected, **F** = output frequency of the results.

Form Finding

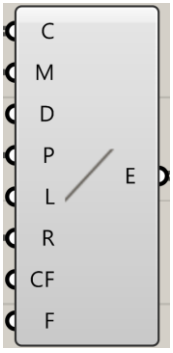
Sets a form finding task based on Updated Reference Strategy (URS).
N = name of analysis task. **S** = maximum number of formfinding steps. A value between 10 and 15 is suggested. **I** = number of Iterations per formfinding step. **O** = defines which results will be outputted for visualization (postprocessing). To set O, connect a Value List component (GH standard).

1.4.3 Element Components



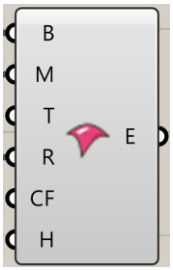
Beam

Defines beam element by assigning structural properties to curve geometry object.
C = curve geometry object, **M** = material, **S** = section, **R** = curve refinement, **AS, AE** = axis of beam section at start (AS) and end point (AE) of curve, **AB** = axis of beam sections between start and end (see Section Axis component) **CF** = coupling features to determine elements to be coupled with,
P = prestress options of the beam (see Prestress Properties Beam component), **HS, HE** = hinge at the start (HS) and end point (HE) of the beam, **HB** = hinge at coupling points between start and end.



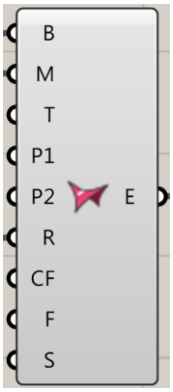
Cable

Defines cable element by assigning structural properties to curve geometry object.
C = curve geometry object, **M** = material, **D** = diameter, **P** = prestress of cable, **L** = set true to let the value of prestress be controlled via the global LD-curve (only for non-linear analysis), **R** = curve refinement, **CF** = coupling features to determine elements to be coupled with.



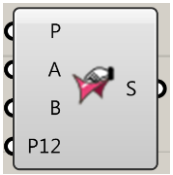
Shell

Defines shell element by assigning structural properties to surface geometry object.
B = brep/surface geometry object, **M** = material, **T** = thickness, **R** = surface refinement, **CF** = coupling features to determine elements to be coupled with, **H** = if shell should only be coupled with hinges (no rotations).



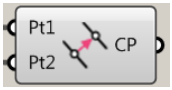
Membrane

Defines membrane element by assigning structural properties to surface geometry object.
B = brep/surface geometry object, **M** = material, **T** = thickness, **P1/P2** = prestress of membrane in u/v direction, **R** = surface refinement, **CF** = coupling features to determine elements to be coupled with, **F** = boolean if part of the form finding, **S** = advanced settings.



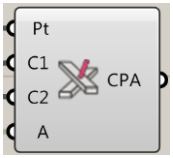
MembraneSettings

Defines advanced settings for the membrane element.
P = projection type of the prestress, **A** = projection direction 1, **B** = projection direction 2, **P12** = shear prestress.



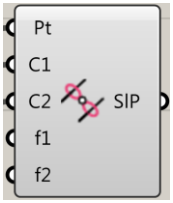
ConnectingPoint

Defines a connection element between two points.
Pt1/Pt2 = points on structural elements which will be connected.



CouplingPointAxis

Defines a point coupling element between two curves, which couples the given local axis (not the rotation around the axis but the axis itself, similar to a scissor).
Pt = location of the point coupling, **C1** = first curve, **C2** = second curve, **A** = axis to be coupled.



SlidingPoint

Defines a point coupling element between two curves, which can slide along the curves.
Pt = location of the point coupling, **C1** = first curve, **C2** = second curve, **f1** = if point is fixed on the first curve, **f2** = if point is fixed on the second curve.

ReferenceGeometry

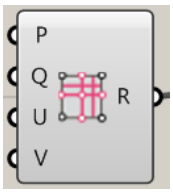
Assigns a reference structural element to a structural Element. The reference element can be referred to as an undeformed reference configuration of the geometry. The element geometry and the reference geometry have to be of same parametrization.
DE = deformed structural element, **RE** = reference structural element.

CouplingFeatures

Defines coupling properties for elements and boundary conditions.
ID = group-ID, **C-IDs** = coupling IDs, which contain all group-IDs, which should be considered for automatic coupling (also for boundary conditions).

CurveRefinement

Defines the refinement for a curve.
P = polynomial degree of the NURBS curve. If the degree of the curve is higher it is not changed, **U** = the nonzero knot spans of the NURBS curve are divided into the defined number of elements if “Knot Subdivision” is chosen in the component menu (default) OR approximate element size if “Approx. Element Size” is chosen.



SurfaceRefinement

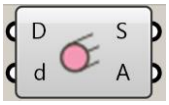
Defines the refinement for a surface.
P/Q = polynomial degree of the NURBS surface in u/v direction. If the degree in one direction of the surface is higher it is not changed, **U/V** = the nonzero knot spans in the respective direction of the NURBS surface are divided into the defined number of elements if “Knot Subdivision” is chosen in the component menu (default) OR approximate element size if “Approx. Element Size” is chosen.

Material

Sets a material.
N = name of material (optional). **T** = type, currently only linear-elastic material types can be defined. **E** = Young’s modulus. **A** = temperature coefficient, **N** = Poisson’s ratio, **D** = density, **fy** = material strength.

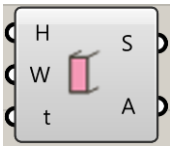
MaterialDefaults

Select from default material types (properties).
T = name of material. To set T, connect a „Value List“ component (GH standard)



Pipe Section

Defines a pipe cross section of a beam.
D = outer diameter of the pipe, **d** = inner diameter of the pipe. Can be set to zero for a full circular cross section.



Box Section

Defines a rectangular cross section of a beam.
H = height of the rectangular cross section, **W** = width of the rectangular cross section, t = thickness, if hollow profile.

Section By Numbers

Defines a custom cross section of a beam.
A = area of the cross section, **Iy** = moment of inertia around the first principal axis of the cross section, **Iz** = moment of inertia around the second principal axis of the cross section, **It** = torsional moment of inertia of the cross section, **Wy** = section modulus corresponding to Iy, **Wz** = section modulus corresponding to Iz, **Az, Ay** = effective shear area in z- and y-direction (only for postprocessing).

Section From Profile

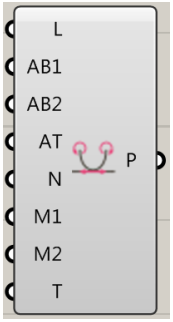
Defines a cross section of a beam by a profile type.
T = type of the profile, **U** = output unit.
S = section, **T** = List of possible profile types.

Deconstruct Section

Returns all cross section values of a beam cross section.
A = area of the cross section, **Iy** = moment of inertia around the first principal axis of the cross section, **Iz** = moment of inertia around the second principal axis of the cross section, **It** = torsional moment of inertia of the cross section, **Wy** = section modulus corresponding to Iy, **Wz** = section modulus corresponding to Iz.

Section Axis

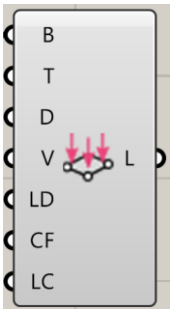
Defines the orientation of a beam cross section at the parametric coordinate.
T = parametric coordinate, **SA** = orientation of the *beam cross section*.



Prestress Properties Beam

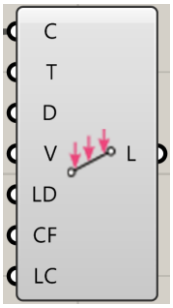
Defines prestress properties for the beam.
L = if prestress is scaled by the global load factor, **AB1, AB2** = derivation of the prestress in bending moment direction 1,2 by the curvature, **AT** = derivation of the torsional prestress by the twist, **N** = absolute value for a prestress in normal force direction, **M1, M2** = absolute value of an additionally applied bending moment 1/2, **T** = absolute value of an applied torsional moment.

1.4.4 Load Components



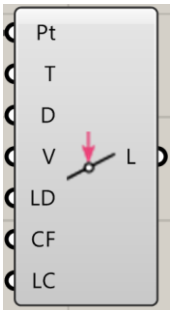
Surface Load

Defines surface-type load.
B = surface/brep geometry where load should act, currently only entire surfaces/breps (which are a structural element at the same time) can be set as loading area,
T = type of loading (DEAD, SNOW, PRES, etc.). To set T, connect a „Value List“ component (GH standard), **D** = direction of load, **V** = value of load,
LD = local LD-Curve to set load control (non-linear analysis only), **CF** = coupling features for defining the elements to be loaded, **LC** = load case id for linear combination with linear analysis.



Curve Load

Defines curve-type load.
C = curve geometry where load should act, **T** = type of loading (DEAD, SNOW, PRES, etc.). To set T, connect a „Value List“ component (GH standard), **D** = direction of load, **V** = value of load, **LD** = local LD-Curve to set load control (non-linear analysis only), **CF** = coupling features for defining the elements to be loaded, **LC** = load case id for linear combination with linear analysis.

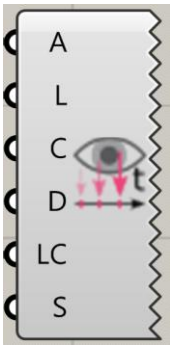


Point Load

Defines point-type load.
Pt = point where load should act, **T** = type of loading (DEAD, SNOW, PRES, MOMENT, etc.). To set T, connect a „Value List“ component (GH standard), **D** = direction of load, **V** = value of load, **LD** = local LD-Curve to set load control (non-linear analysis only), **CF** = coupling features for defining the elements to be loaded, **LC** = load case id for linear combination with linear analysis.

LD-Curve

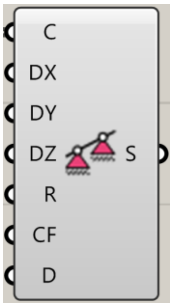
Defines load-displacement curve via a list of time steps and values (= load factor at time step. Note that both lists have to have same length (each value must correspond to a set time step). All LD-Curve components in the model have to have the same last time step value.
T = list of time steps starting with 0, **V** = list of values.



LoadCaseCheck

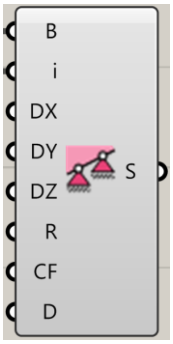
Displays the actually applied load for a respective LC values(time step or load case id)
A = type of analysis, **L** = loads, **C** = cables with prestress, **D** = Support Displacement, **LC** = load case is equal to load case id or time step, **S** = scaling factor

1.4.5 Support Components



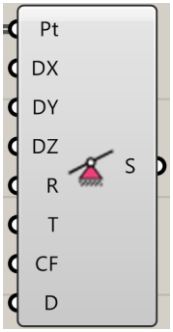
SupportCurve

Defines curve-type support.
C = curve geometry where support should act, **DX/DY/DZ** = set constraint in direction via boolean, **R** = set rotation constraint (only for shell elements), **CF** = coupling features for defining the elements to be supported, **D** = directions, which are blocked by a support (DX = DY = DZ = true).



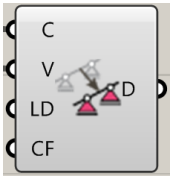
SupportEdge

Defines edge-type support.
B = brep geometry where edge support should be on, **i** = edge index, **DX/DY/DZ** = set constraint in direction via boolean, **R** = set rotation constraint (only for shell elements), **CF** = coupling features for defining the elements to be supported, **D** = directions, which are blocked by a support (DX = DY = DZ = true).



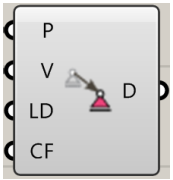
SupportPoint

Defines point-type support.
Pt = point where support should act, **DX/DY/DZ** = set constraint in direction via boolean, **R** = set rotation constraint (only for beam elements at start and end point), **T** = set torsional constraint (only for beam elements at start and end point), **CF** = coupling features for defining the elements to be supported, **D** = directions, which are blocked by a support (DX = DY = DZ = true).



Curve Displacement

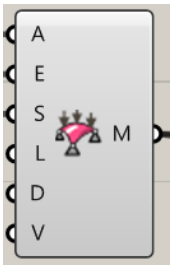
Defines a forced displacement of a curve-type support.
C = curve, **V** = vector which is giving the direction (and length) of the forced displacement. Zero components have to be defined in a separate point-support, **LD** = local LD-curve to control the application of the forced displacement, **CF** = coupling features for defining the elements to be supported.



Point Displacement

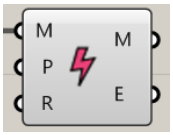
Defines a forced displacement of a point-type support.
P = point, **V** = vector which is giving the direction (and length) of the forced displacement. Zero components have to be defined in a separate point-support, **LD** = local LD-curve to control the application of the forced displacement **CF** = coupling features for defining the elements to be supported.

1.4.6 Model Components



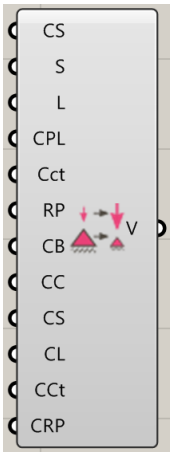
AnalysisModel

Assembles a structural model (including the analysis task). The type of simulation, all elements, supports and loads need to be brought together here.
A = type of analysis, **E** = structural elements, **S** = supports, **L** = loads, **D** = displacement loads, **V** = visualization settings (see Visualize Options).



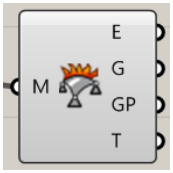
IGASolver

Runs an Isogeometric Analysis using Carat++.
M = connect assembled Structural Model here, **P** = path for the simulation files. (Default: path of the .gh-file), **R** = boolean toggle to push the solver to start.



VisualizeOptions

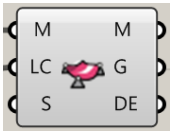
Scale visualization objects of preprocessing here, such as symbols for supports, loads, etc.
CS = scale for the visualization of cross sections (beams and cables), **S** = scale for the visualization of supports, **L** = scale for the visualization of loads, **CPL** = scale for the visualization of coupling, **Cct** = scale for the visualization of connecting points, **RP** = scale for the visualization of the control points of the refined patches, **CB** = colour beam sections, **CC** = colour cables sections, **CS** = colour support symbols, **CL** = colour load symbols, **CCt** = colour connect symbols, **CRP** = colour refined patches symbols.



DeconstructModel

Deconstructs a given structural model into its parts, such as structural elements, geometry objects, Gauss points, etc.
E = structural elements, **G** = geometry, **GP** = Gauss points in parameter space of NURBS patch, **T** = Input text file for Carat++ analysis.

1.4.7 Result Components

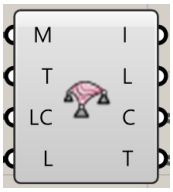


DeformedModel

Visualized the deformed shape of the Structural Model for a given load case. For linear analysis set LC = 0, for Form-finding LC equals a Form-finding step, for non-linear analysis LC equals a TimeStep.
M = structural model, **LC** = load case, **S** = factor for scaling deflections.
M = deformed structural model, **G** = deformed geometries, **DE** = deformed structural elements.

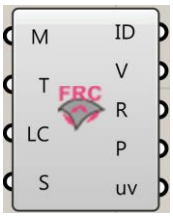
2D Element Result ColourPlot

Provides a colour plot of a chosen result type and load case.
M = structural model, **T** = result type (add Value List component here to choose), **LC** = load case.
M = structural model, **M** = GH mesh for visualization, **D** = minimum and maximum value of result, **C** = legend for colorization, **T** = legend of corresponding values for the colorization.



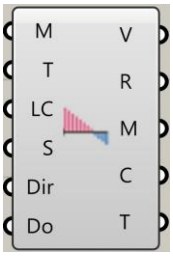
2D Element Result Isolines

Provides an isoline plot of a chosen result type and load case.
M = structural model, **T** = result type (add Value List component here to choose), **LC** = load case, L = values for the isolines (default: result is split in 10 equidistant levels).
I = isolines, **L** = values for the isolines, **C** = legend for colorization, **T** = legend of corresponding values for the colorization.



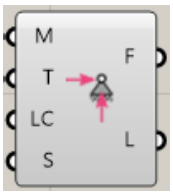
2D Element Result

Provides values and direction (if available) of GP result values as tree.
M = structural model, **T** = result type (add Value List component here to choose), **LC** = load case, **S** = scale vector display length.
ID = tree of GP IDs, **V** = tree of GP result values, **R** = tree of GP result values with direction, **P** = GPs in 3d space, **uv** = GPs in parameter space.



1D ElementResult View

Provides a line style visualization of beam/cable results for chosen result type and load case. The length of each unscaled line is set to result value at the location of the result.
M = structural model, **T** = result type (add Value List component here to choose), **LC** = load case, **S** = scale line length, **Dir** = direction of lines (default: direction of cross section), **Do** = minimum and maximum value of result.
V = results on GPs as values, **R** = results on GPs as lines
M = GH mesh for visualization, **C** = legend for colorization, **T** = legend of corresponding values for the colorization.



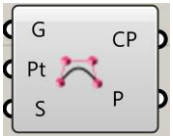
SupportForces

Provides visualization of support forces as vectors.
M = structural model, **T** = result type (add Value List component here to choose), **LC** = load case, **S** = scale vector display length.
F = vector output for support forces, **L** = location of vectors.

1.4.8 Utility Components

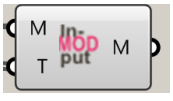
SearchInput

Returns the line numbers of a text fragment in the input file for Carat++. Note that the whole line has to match.
L = input file as text, **V** = text to be searched, **S** = if search for V as substring.
i = indices of the lines, which contain the text.



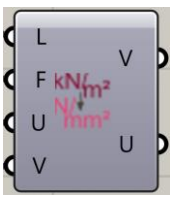
ControlPolygon

Shows the control polygon of the NURBS patches in defined style.
G = NURBS geometry, **Pt** = double(s) for the dashing pattern of the control polygon, **S** = Box size for the visualization of control points.



ModifyInput

Exchanges the input file for Carat++ of a model in order to modify the simulation.
M = structural model, **T** = modified input file for Carat++.



Unit Converter

Converts a value with unit in your global unit.
L = global length unit (connect a „Value List“ component (GH standard)), **F** = global force unit (connect a „Value List“ component (GH standard)), U = unit to be converted (connect a „Value List“ component (GH standard)), **V** = value to be converted.
V = converted value, **U** = converted unit.

DuplicateBrep

Duplicates a Brep geometry in order to apply additional structural elements.
B = Brep.
B = Duplicated Brep.

DuplicateCurve

Duplicates a Curve geometry in order to apply additional structural elements.
C = Curve.
C = Duplicated Curve.

Licence

Licence information including expiry date.

1.5 Units

There are no pre-defined units in Kiwi!3d. You have to choose your units for length, e.g. m, and force, e.g. kN, and set all other values using these units. The results are provided in the same units. The length unit does not necessarily have to match with the units of Rhino, but it is recommended.

Units	Force	Length	Young's Modulus	Area	Stress	Moment	Line Force
N, mm	N	mm	N/mm²	mm²	N/mm²	Nmm	N/mm
kN, m	kN	m	kN/m²	m²	kN/m²	kNm	kN/m
kN, cm	kN	cm	kN/cm²	cm²	kN/cm²	kNcm	kN/cm

1.6 Contact

The plugin is developed in cooperation of str.ucture GmbH (str-ucture.com) and Chair of Structural Analysis, Technical University of Munich (bgu.tum.de/st/).

For questions and any suggestion regarding software performance, layout, interface or bugs please contact us at development@kiwi3d.com