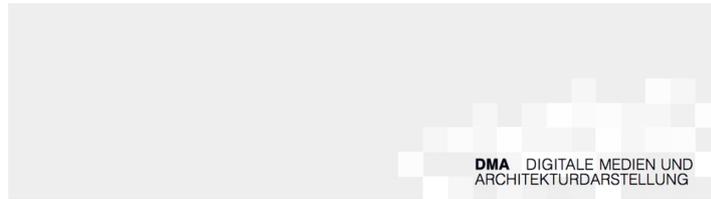

DMA Digitale Medien
und Architekturdarstellung

Prof. Oliver Fritz
HTWG Konstanz
Brauneggerstr 55
Postfach 100543
D - 78462 Konstanz



Beispiele Grasshopper / Karamba



1. einfacher Kragarm

Datei:

01_Kragarm.gh

Variablen:

X coordinates : Länge des Kragarms

load size : Größe der Punktlast

über zwei Punkte wird ein einfacher Kragarm erzeugt, der mit Karamba analysiert wird. Auf dem Kragarm wirkt die Schwerkraft und eine Punktlast am Ende Kragarms. Die Veränderung der Deformation des Trägers kann direkt im Rhino Modell verfolgt werden.



2. einfacher Fachwerkträger 2D

Datei:

02_Fachwerkträger 2D.gh

Variablen:

X coordinates : Länge des Kragarms

load size : Größe der Punktlast

load position : Position der Punktlast auf dem Träger

Teilung : Aufteilung des Trägers

Trägerhöhe : Höhe des Trägers

einfacher Fachwerkträger mit zwei Auflagern der über verschiedene Parameter kontrolliert werden kann.



3. einfaches Flächentragwerk

Datei:

03_Flächentragwerk.gh

Variablen:

shell length / shell width : Länge und Breite der Fläche

oc Y / oc Z / mc Z : Y/Z Koordinaten der Kontrollpunkte der Leitkurven der Fläche

shell height : Dicke der Fläche für die statische Analyse

load size / load position : Größe und Position der Punktlast

durch drei Leitkurven wird eine Fläche erzeugt, die als Schale (z.B. aus Beton) mit Karamba analysiert wird. zusätzlich zur Gravitation wirkt eine Punktlast (dargestellt durch eine schwarze Kugel), deren Position und Größe variabel verändert werden kann. im Rhino-Modell lässt sich direkt die Verformung der Schale verfolgen.



4. einfacher Fachwerkträger 3D

Datei:

04_Fachwerkträger 3D.gh

Variablen:

shell length / shell width : Länge und Breite der Fläche

oc Y / oc Z / mc Z : Y/Z Koordinaten der Kontrollpunkte der Leitkurven der Fläche

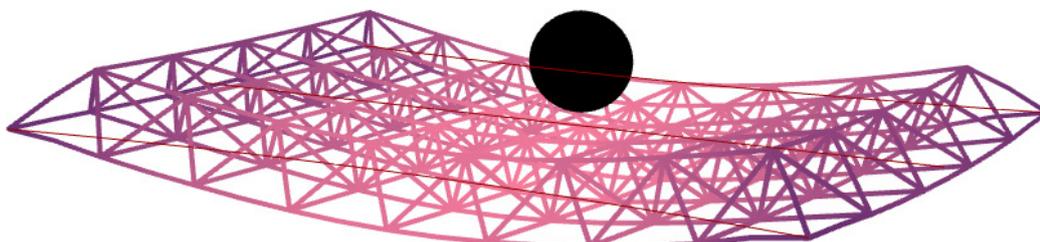
div U / div V : Aufteilung des Fachwerks in beide Richtungen

structure height : Höhe des Fachwerks

diam / wallThick : Durchmesser und Wandstärke der Rundprofile des Fachwerks in cm

load size / load position : Größe und Position der Punktlast

aufbauend auf dem Beispiel 3. *einfaches Flächentragwerk* wird ein 3-dimensionaler Fachwerkträger aus Rundrohrprofilen aufgebaut und mit Karamba analysiert.



5. Voronoi Fachwerkträger 3D

Datei:

05_Fachwerkträger 3D_voronoi.gh

Variablen:

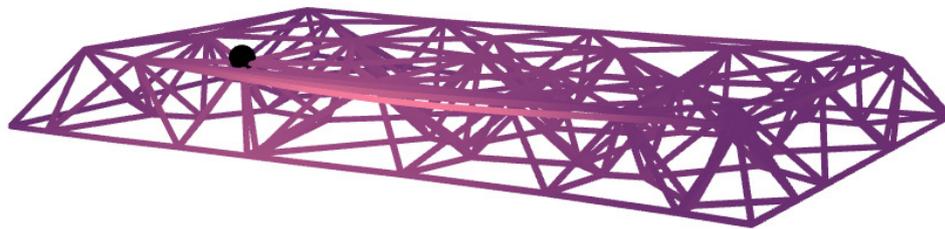
shell length / shell width : Länge und Breite der Fläche

oc Y / oc Z / mc Z : Y/Z Koordinaten der Kontrollpunkte der Leitkurven der Fläche

shell height : Dicke der Fläche für die statische Analyse

load size / load position : Größe und Position der Punktlast

durch drei Leitkurven wird eine Fläche erzeugt, die als Schale (z.B. aus Beton) mit Karamba analysiert wird. zusätzlich zur Gravitation wirkt eine Punktlast (dargestellt durch eine schwarze Kugel), deren Position und Größe variabel verändert werden kann. im Rhino-Modell lässt sich direkt die Verformung der Schale verfolgen.



13. informiertes Flächentragwerk

Datei:

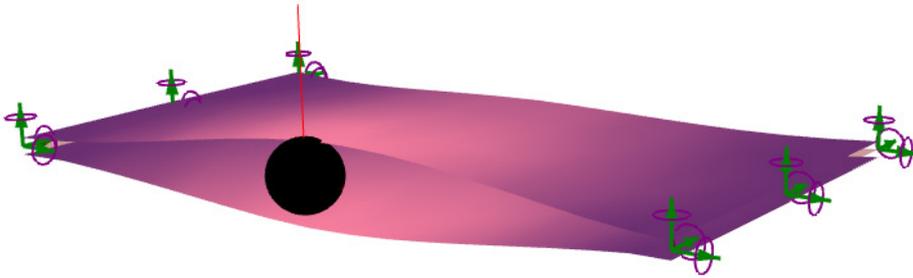
13_informed_Flächentragwerk.gh

Variablen:

analog Beispiel 3

factor : Faktor, wie stark die Deformation aus der statischen Analyse die Höhe der Struktur beeinflusst

die Fläche aus dem Beispiel 3. *einfaches Flächentragwerk* wird mit Karamba analysiert und die daraus gewonnenen Informationen dazu genutzt die Struktur des Tragwerks zu verändern bzw. zu optimieren. Für jede Stelle des Tragwerks wird die Deformation gemessen und die Dicke der Schale neu berechnet. An Stellen mit großer Last und großer Deformation wird dementsprechend die Schale dicker gestaltet.



14. informierter einfacher Fachwerkträger 3D

Datei:

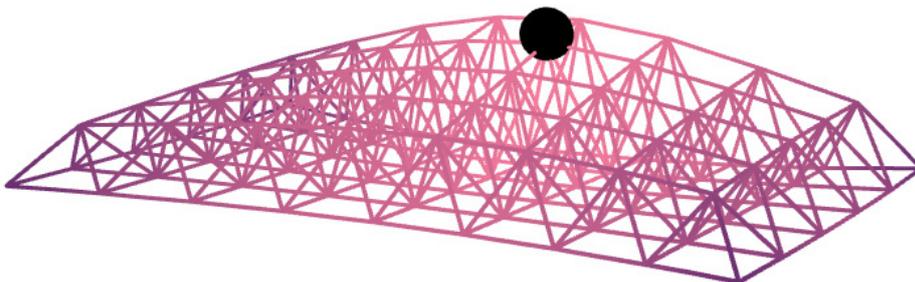
14_informed_Fachwerkträger 3D.gh

Variablen:

analog Beispiel 4

factor : Faktor, wie stark die Deformation aus der statischen Analyse die Höhe der Struktur beeinflusst

analog Beispiel 13. informiertes Flächentragwerk



22. evolutionäre Optimierung 01

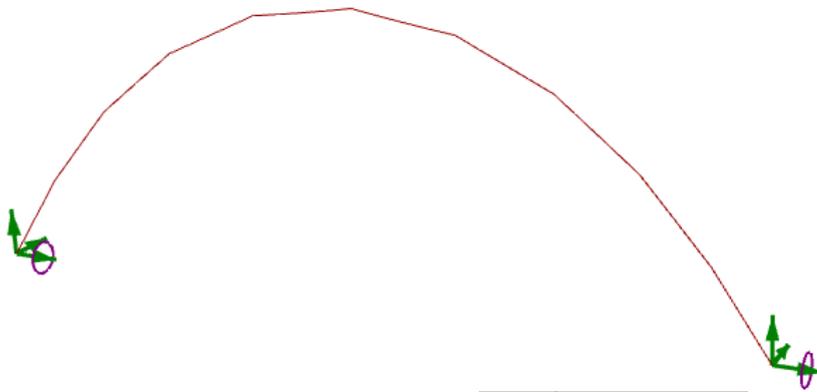
Datei:

22_OptimizationGalapagos_SimpleArc.gh

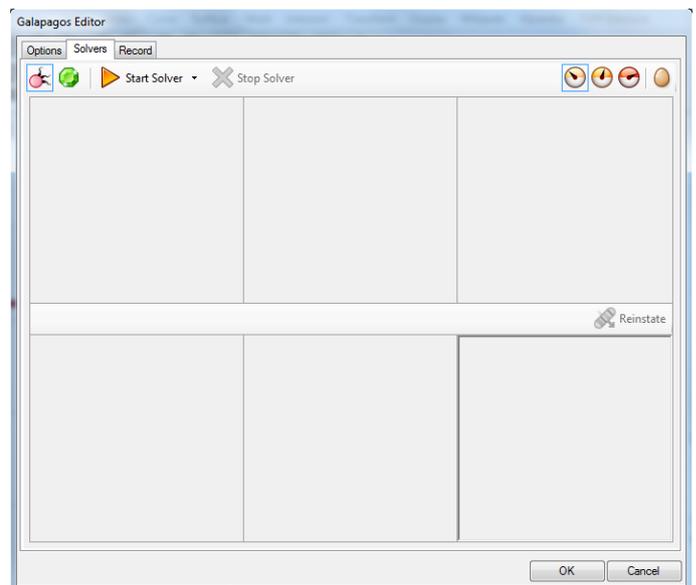
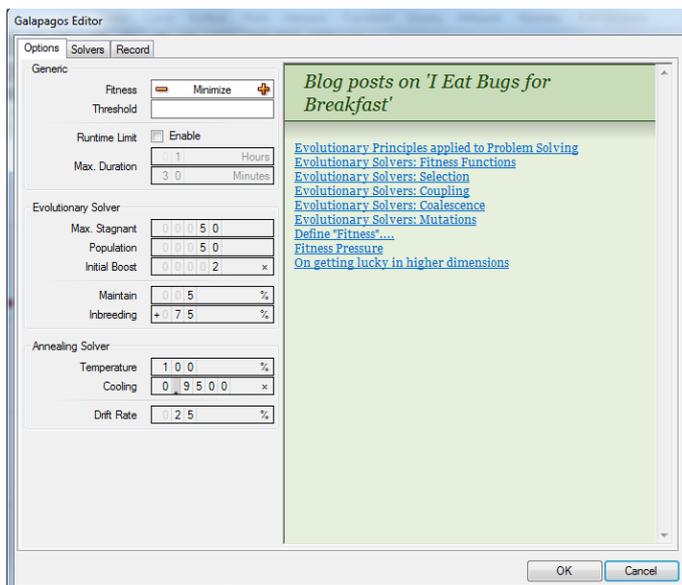
das Grasshopper plugIn *Galapagos* ist ein Werkzeug, mit dem sich verschiedenen Kriterien mittels evolutionärer Algorithmen optimieren lassen.

In diesem sehr einfachen Beispiel wird ein Bogen zwischen zwei Auflagern erzeugt, dessen Höhe variabel ist. Der Bogen wird mit Karamaba analysiert und dessen maximale Durchbiegung gemessen. Vereinfacht gesagt testet Galapagos verschiedene Varianten des Bogens, erzeugt weitere Variationen der besten Varianten und testet diese wieder usw. usw. bis eine zufriedenstellende Lösung gefunden ist.

Galapagos benötigt als "input" auf der linken Seite den / die Slider, die während des Tests verändert werden (Genom // in unserem Fall die Bogenlänge) und auf der anderen Seite einen Wert, anhand dessen entschieden werden kann, welche Varianten besser oder schlechter sind (Fitness // in unserem Fall die maximale Durchbiegung):



Durch Doppelklick auf das Symbol öffnet sich das Menu des PlugIns:



Es lassen sich viele Werte für das finetuning des evolutionären Algorithmus einstellen, die an dieser Stelle nicht weiter erörtert werden.

Auf dem zweiten Reiter "Solvers" lässt sich durch klicken des Buttons "Start Solver" die Optimierung starten. Bei den drei Zeigern rechts oben ist standardmässig das mittlere eingestellt. Das bedeutet, dass Rhino nur das aktuell beste Ergebnis anzeigt. Wenn man das linke Symbol aktiviert, wird jede von Galapagos getestete Variante angezeigt. So lässt sich im Rhino-Fenster direkt die Optimierung beobachten.

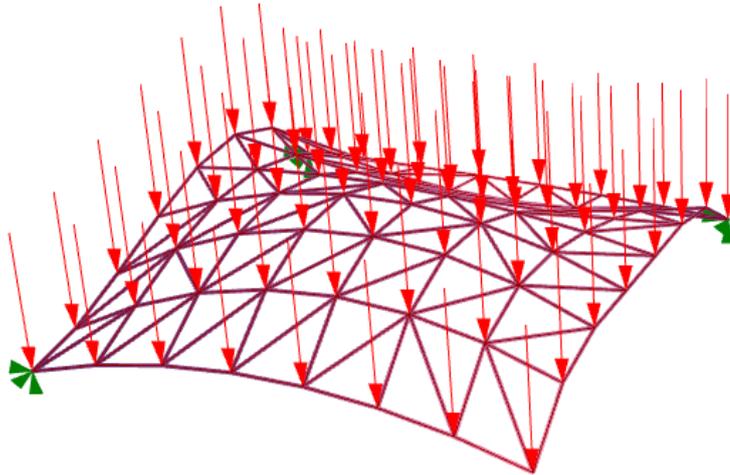
23. evolutionäre Optimierung 02

Datei:

23_OptimizationGalapagos_Surface.gh

in diesem Beispiel versucht Galapagos die Geometrie einer Fläche, die über drei Auflager gespannt ist, so zu optimieren, dass eine geringstmögliche Durchbiegung auftritt.

Diese Optimierung ist bereits wesentlich komplexer als das vorherige Beispiel und benötigt mehr Zeit bis ein Ergebnis gefunden wurde.

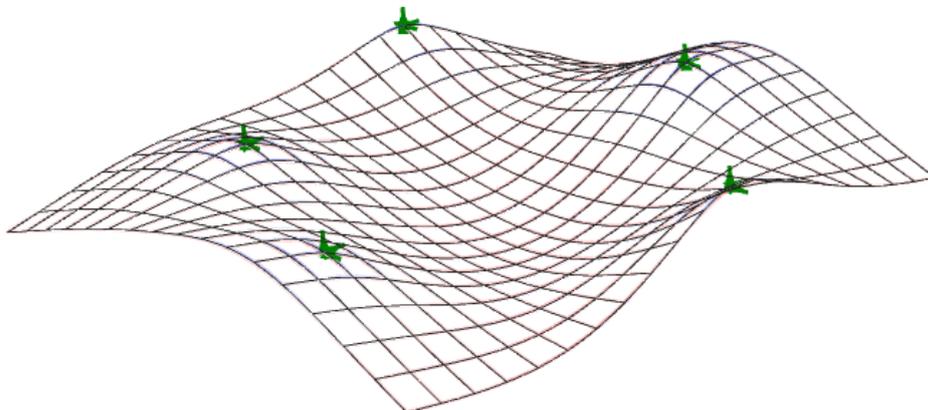


24. evolutionäre Optimierung 03

Datei:

24_OptimizationGalapagos_FindBestSupportPositions.gh

in diesem Beispiel versucht Galapagos für eine vorgegebene Fläche die bestmöglichen Positionen für fünf Auflager anhand der minimalen Durchbiegung zu finden.

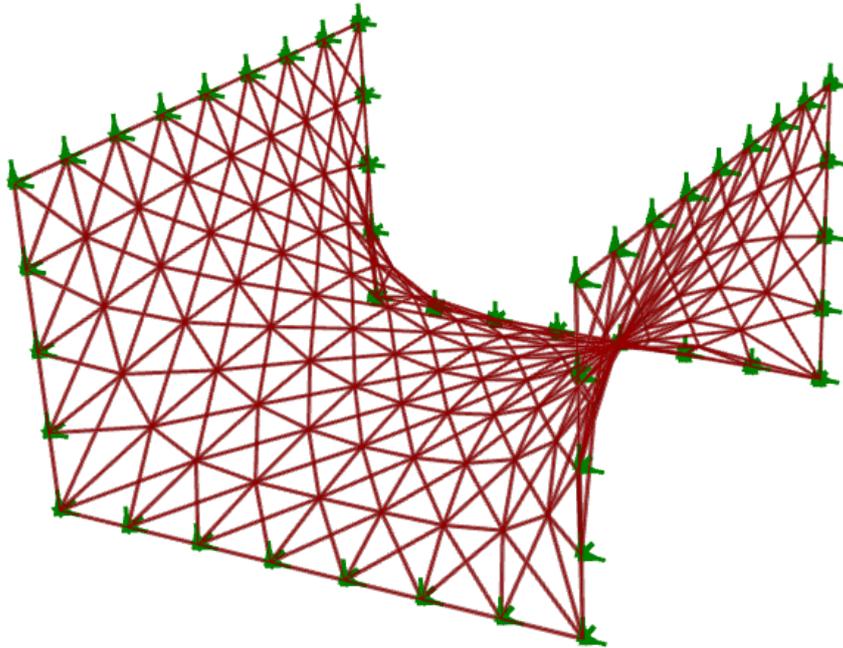


31. Formfindung mit Karamba 01

Datei:

31_FormFinding_MinimalSurface.gh

Karamba lässt sich auch für Formfindungsprozesse einsetzen. Die Vorgehensweise ist eine digitale Variante der Hängemodelle Gaudi's: Eine Grundstruktur wird einer Last ausgesetzt (z.B. Schwerkraft), wodurch sie sich verformt. Unter Angabe einer Vorspannung lässt sich so z.B. die minimale Fläche (z.B. Seifenblasen) zwischen einem Rahmen aufspannen:



32. Formfindung mit Karamba 02

Datei:

32_FormFinding_largeDeformationBubble.gh

auch komplexere Geometrien lassen sich so berechnen:

