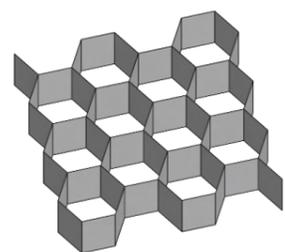




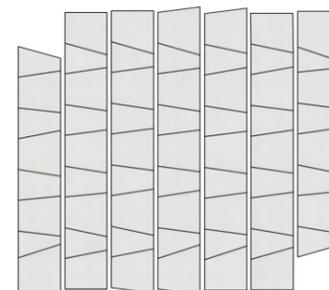
# Foldable Honeycomb

ハニカム構造は、剛性を大きくしたい、費用・材料を少なくしたいといった目的がある中での1つの解だと思われる。さらに先行研究<sup>[1]</sup>において、ハニカム構造をセルの壁が120°で接するねじれのない立体的な構造として研究され、設計方法の紹介と設計できる限界を議論している。そのような機能がある中で、セルの壁が120°で接する条件を緩和し、設計者の負担の軽減、必要部材の削減、折り畳み>運搬>展開のしやすさといった機能をより共有できる形態創生を曲面からの逆解析で行った。

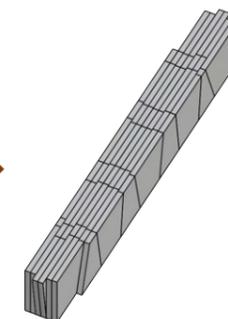
提案するハニカム構造は、設計者が曲面やハニカムの密度を入力することで形態創生を行うため、設計の負担が軽減される。そして、帯状へ折りたたむことができるため、歩留まりがよく、折りたたんで運搬しやすくなる。さらに、展開時にはセルの壁がねじれのない平面であるため、ねじることのできる特殊な部材は必要ない。



機能の共有  
 ・設計者の負担の軽減  
 ・必要部材の削減  
 ・折りたたみ>運搬>展開のしやすさ



必要部材の削減



折りたたみ>運搬>展開のしやすさ



[1] Jiang, Caigui, et al. "Freeform honeycomb structures." *Computer Graphics Forum*. Vol. 33. No. 5. 2014.

# 形態創生の手法

## ハニカム構造と制約条件

今回のハニカムを構成するとき、以下の条件が必要となる。

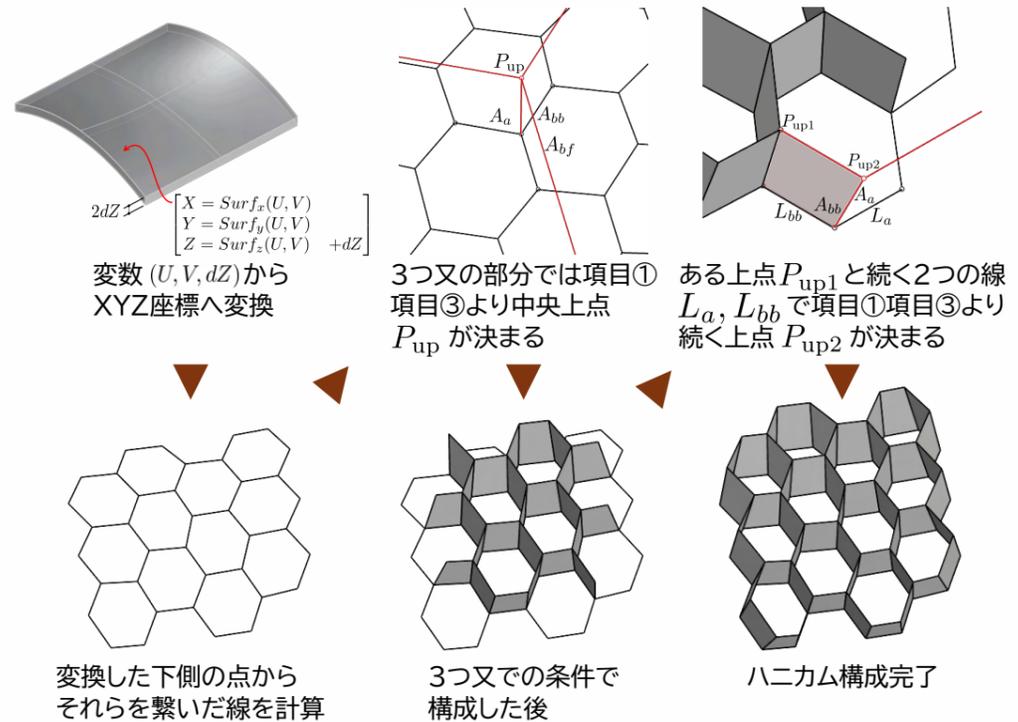
- ・部材幅は一定になる 項目①
  - ・入力した面形状へ展開される 項目②
  - ・ハニカムの $A_a, A_{bf}, A_{bb}$ は $A_{bf} = A_{bb} = 180^\circ - A_a$ とすることで、折り畳むことができる角度となる 項目③制約③
  - ・展開時、セルの壁が平面である 制約④
  - ・ハニカムの六角形の辺のうち、折り畳みで分けられる前辺3つと後辺3つのそれぞれの合計長さが同じ 制約⑤
  - ・折り畳み時に形状が帯状となる 制約⑥
  - ・辺の長さが指定した長さより短い 制約⑦
  - ・入力した面形状の四辺側にある点は四辺上へ固定させる 制約⑧
- 以上でハニカムを構成し、制約関数の値を算出する。

ハニカム構成時に考慮する項目

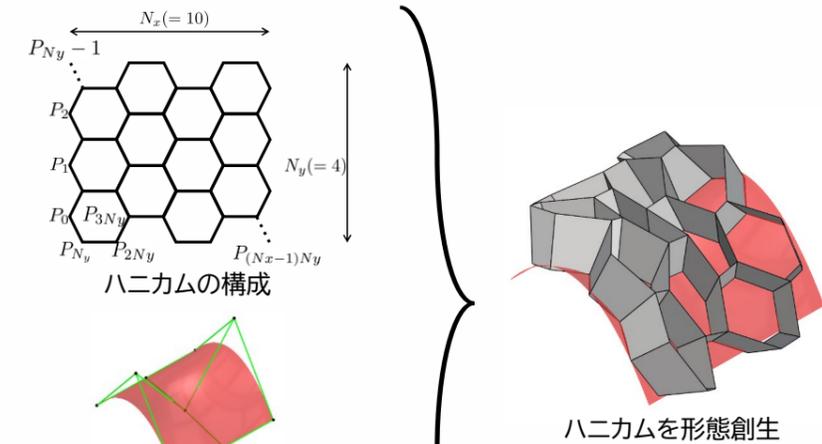
制約

追加の形状制約

## 変数とハニカムの構成



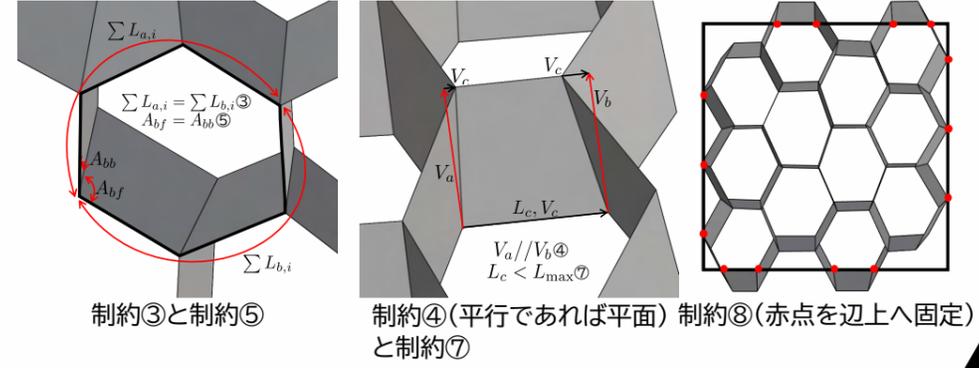
このハニカム構造は制約条件のみを入れて、曲面からの逆解析を行った。入力は、曲面モデル、ハニカムの構成  $(N_x, N_y)$ 、Z方向のずれ幅、ハニカム下辺の長さ上限である。この形態創生ではrhino3dm.py<sup>[2]</sup>で曲面上の点を算出し、pyOpt<sup>[3]</sup>で制約条件を満たすものを探し逆解析を行う。



沿わせたい曲面形状

- ・どの程度沿わせるか(dZ)
- ・辺長制約

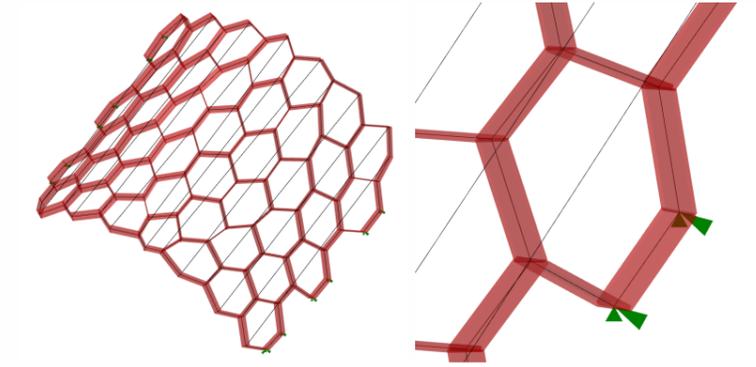
## 制約関数



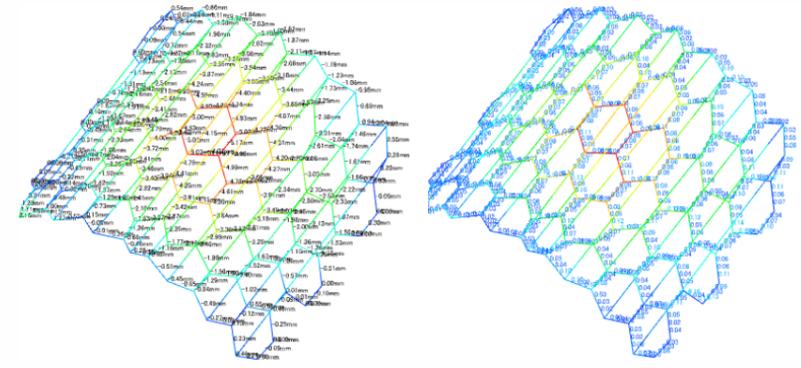
# 構造解析

以下の条件で梁要素の構造解析を行う

- 大きさ 8m x 8m
- 材 2x12木材とワイヤー径5mmのワイヤー
- 荷重 自重のみ
- 節点 剛節合
- 境界条件 Y、Zピン(スラスト方向と鉛直方向に対応)

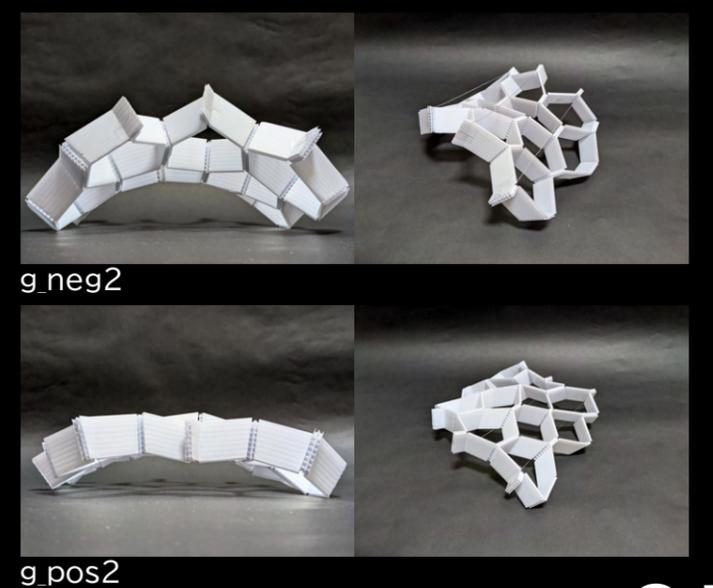


OpenSees for Grasshopper<sup>[4]</sup>で解析を行った。Z方向の最大変位は5.17mm NQMでの検定比の最大値は長期の場合最大で0.18となった。



## モックアップ

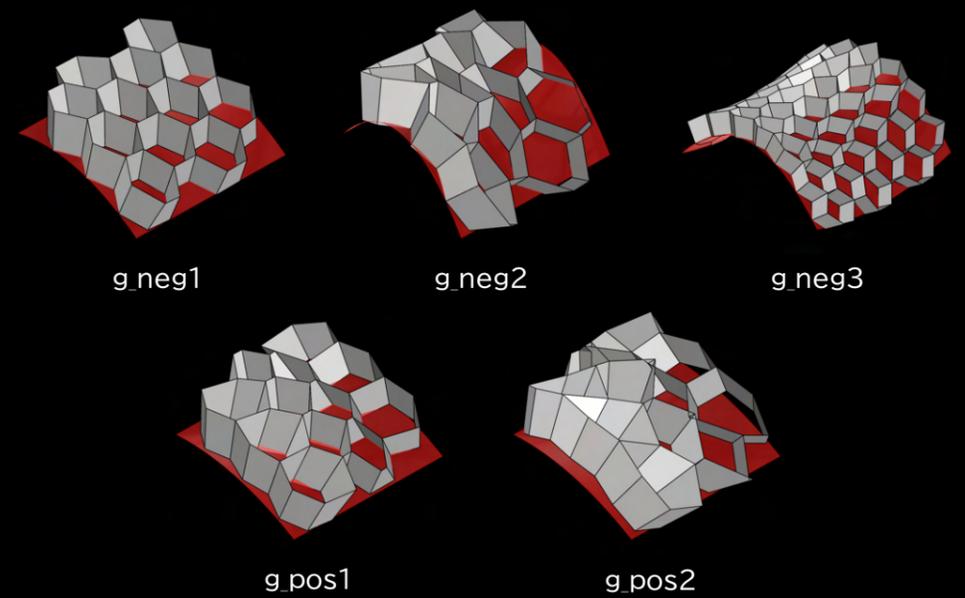
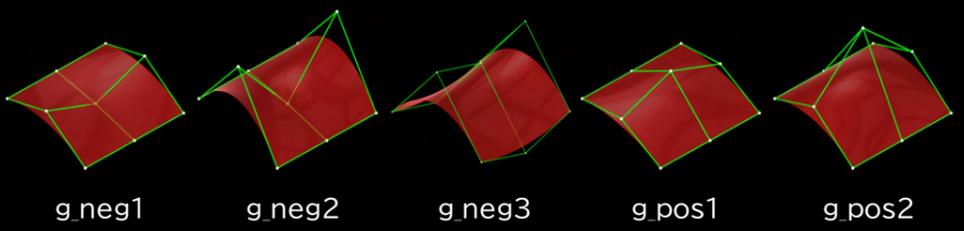
A4サイズのプラダンとナイロン線でのモックアップを作成した。入力する曲面がガウス曲率が負のものと正のもので行っている。折りたたみ>展開は問題なく可動したが、展開時は展開方向への拘束だけでは形状が決まらず、ナイロン線で収縮方向への拘束を行うと形状を決めることができた。



# 逆解析結果

5つのモデルで逆解析を行い、形態創生をしている。以下に入力したデータと結果を示す。

モデル名	ガウス曲率	曲面サイズ	$N_x, N_y$	辺長しきい値	dZ	最適性条件	制約関数最大値
g_neg1	負	4x4	10, 4	0.65	0.1	OK	$4.51 \times 10^{-6}$
g_neg2	負	4x4	10, 4	0.73	0.1	OK	$4.54 \times 10^{-7}$
g_neg3	負	8x8	14, 8	0.81	0.2	NG	$5.76 \times 10^{-6}$
g_pos1	正	4x4	10, 4	0.66	0.1	OK	$4.30 \times 10^{-7}$
g_pos2	正	4x4	10, 4	0.73	0.22	NG	$4.42 \times 10^{-6}$



[2] Scott D. Calling Compute with Python. <https://developer.rhino3d.com/guides/compute/compute-python-getting-started/> GitHub: <https://github.com/mcneel/rhino3dm> [Accessed 22 Sep. 2024]. Robert McNeel & Associates.  
 [3] Perez, Ruben E., Peter W. Jansen, and Joaquim RRA Martins. "pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization." *Structural and Multidisciplinary Optimization* 45 (2012): 101-118.  
 [4] Shinnosuke F. OSforGH. <https://github.com/strdesigner/OSforGH> [Accessed 22 Sep. 2024]