Computational Assemblies: Analysis, Design, and Fabrication

Peng Song

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Ziqi Wang

ETH zürich

Marco Livesu

imati

Timetable

		Peng	Ziqi	Marco
Introduction	~20 mins	X		
Computational analysis of assemblies	~ 50 mins	X		
Computational design of assemblies	~50 mins		X	
Computational fabrication of assemblies	~ 50 mins			X
Q & A	~ 10 mins	X	X	X

Computational Design of Assemblies

- Our goal is to design assemblies to achieve required objectives with the help of computational methods.

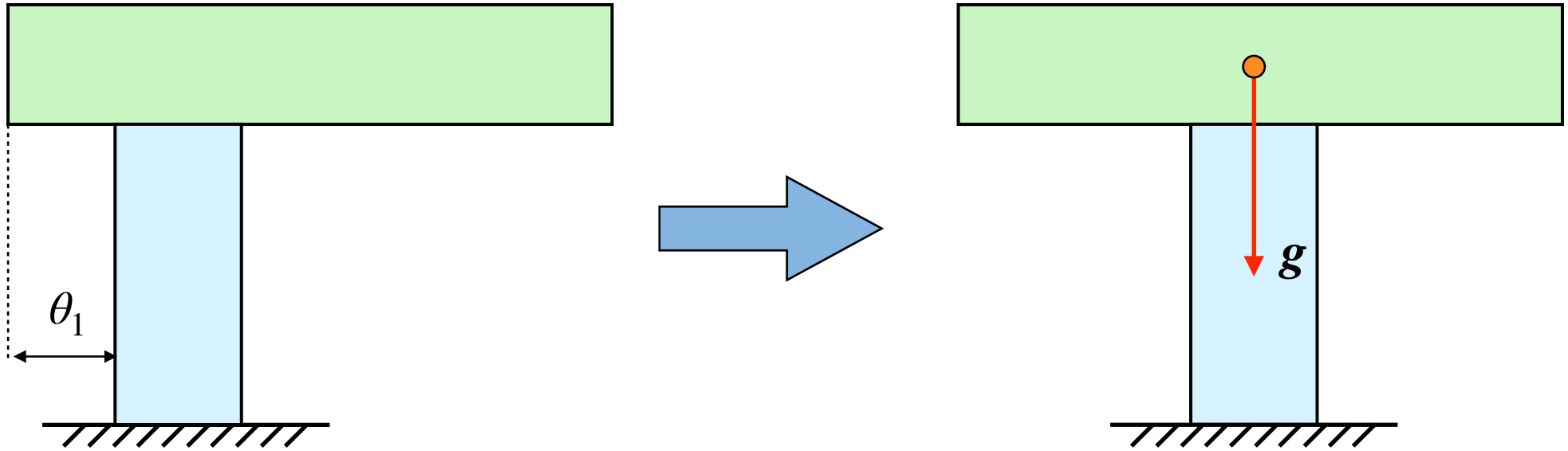
Puzzle

Furniture

Architecture

Case Study

- Two main components: **parts**' **geometry** and **design objectives**.

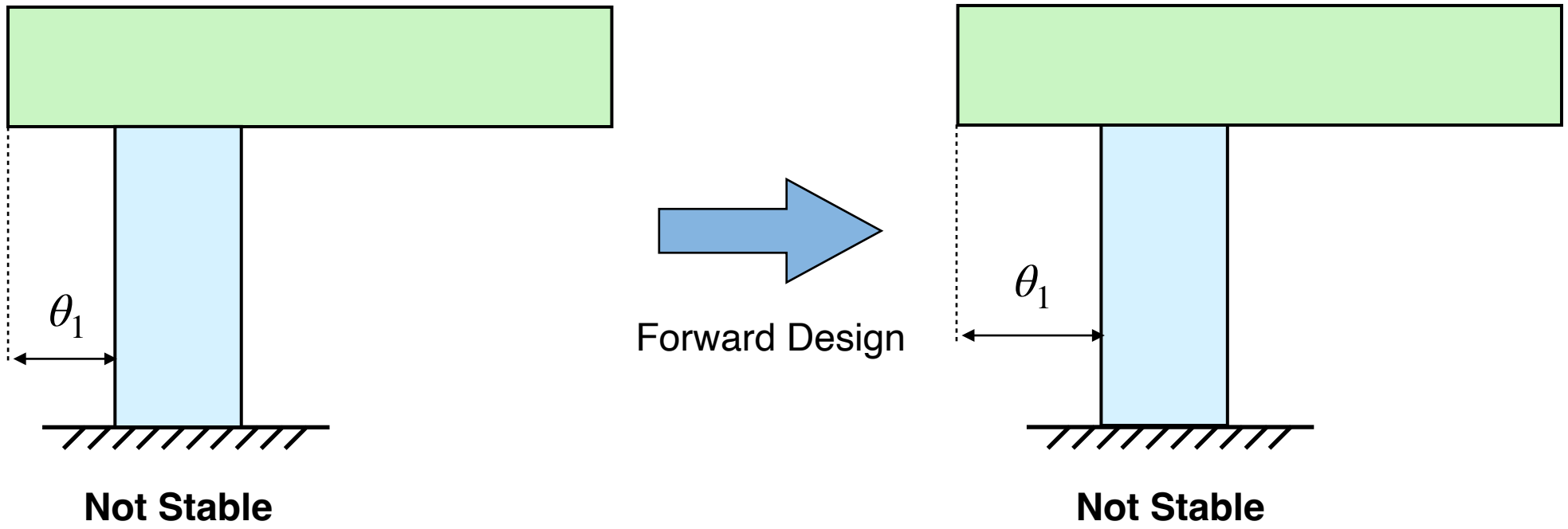


Parametric Model (t)

Equilibrium under Gravity

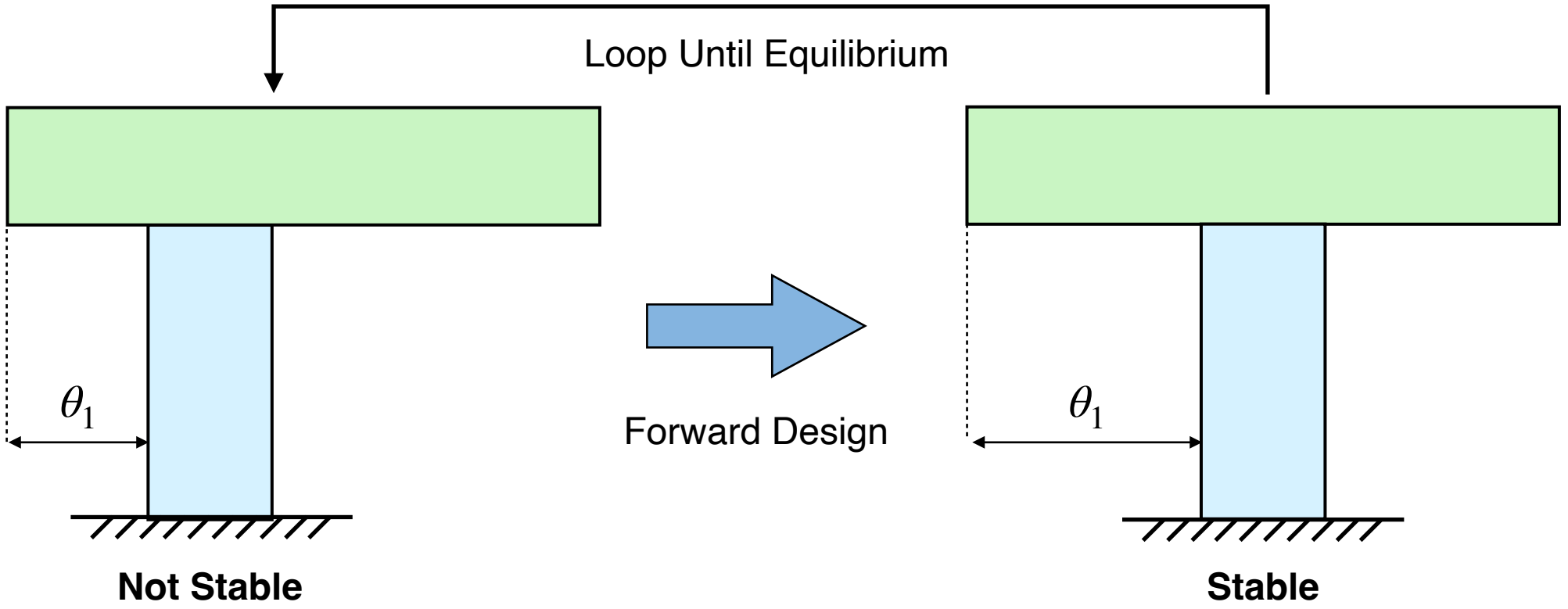
Forward Design Framework

- Manually tune the design parameters until the assembly can stay in equilibrium under gravity.



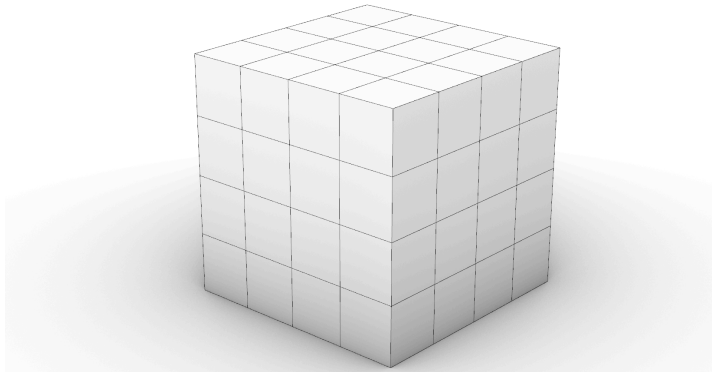
Forward Design Framework

- More design iterations are required if the current design is not stable.



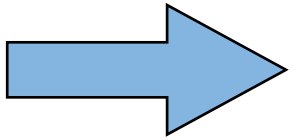
Challenges of Forward Design

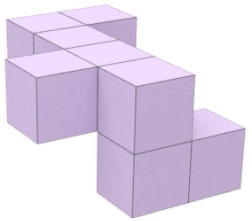
- The problem can have an enormous design space but finding one feasible solution is non-trivial.



Design Space

6^{64}



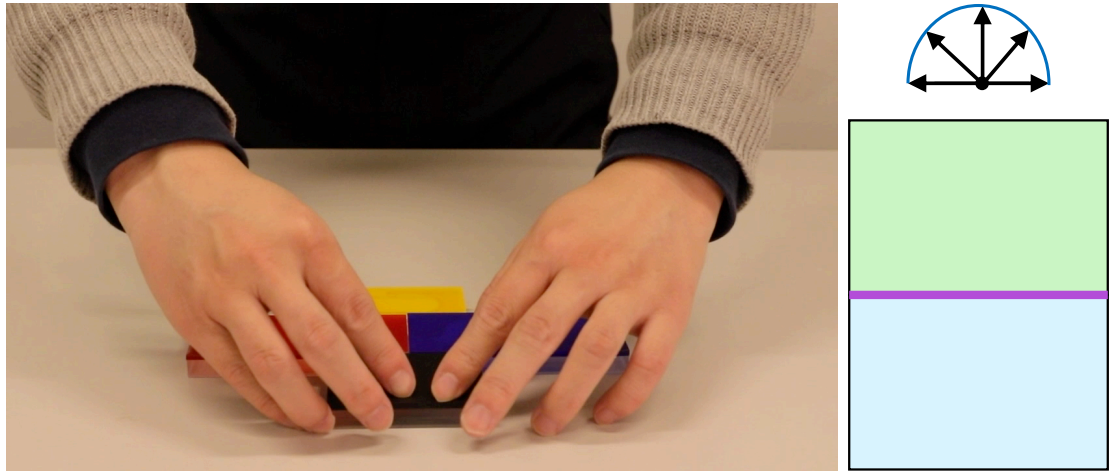


Objectives

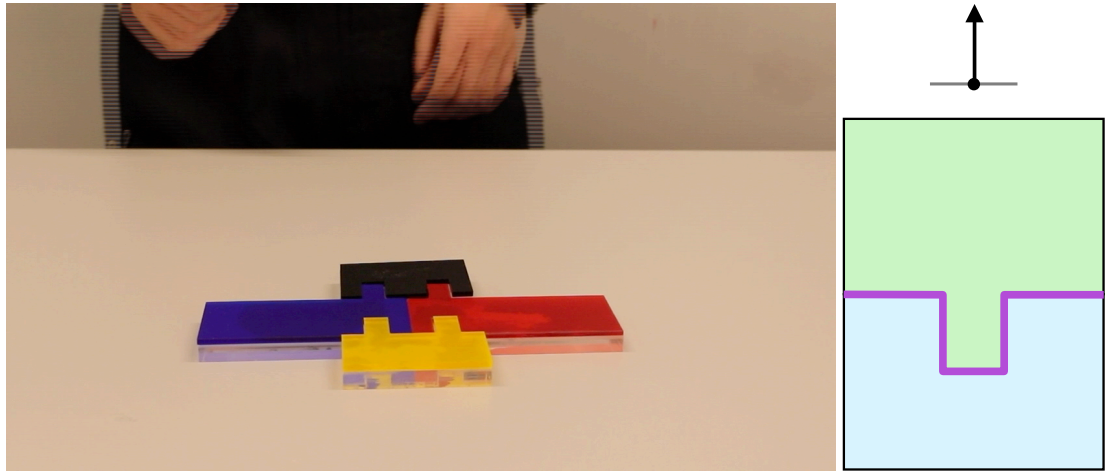
Stable under arbitrary forces

Challenges of Forward Design

- The problem can have multiple design objectives.
- They might be contradicting.



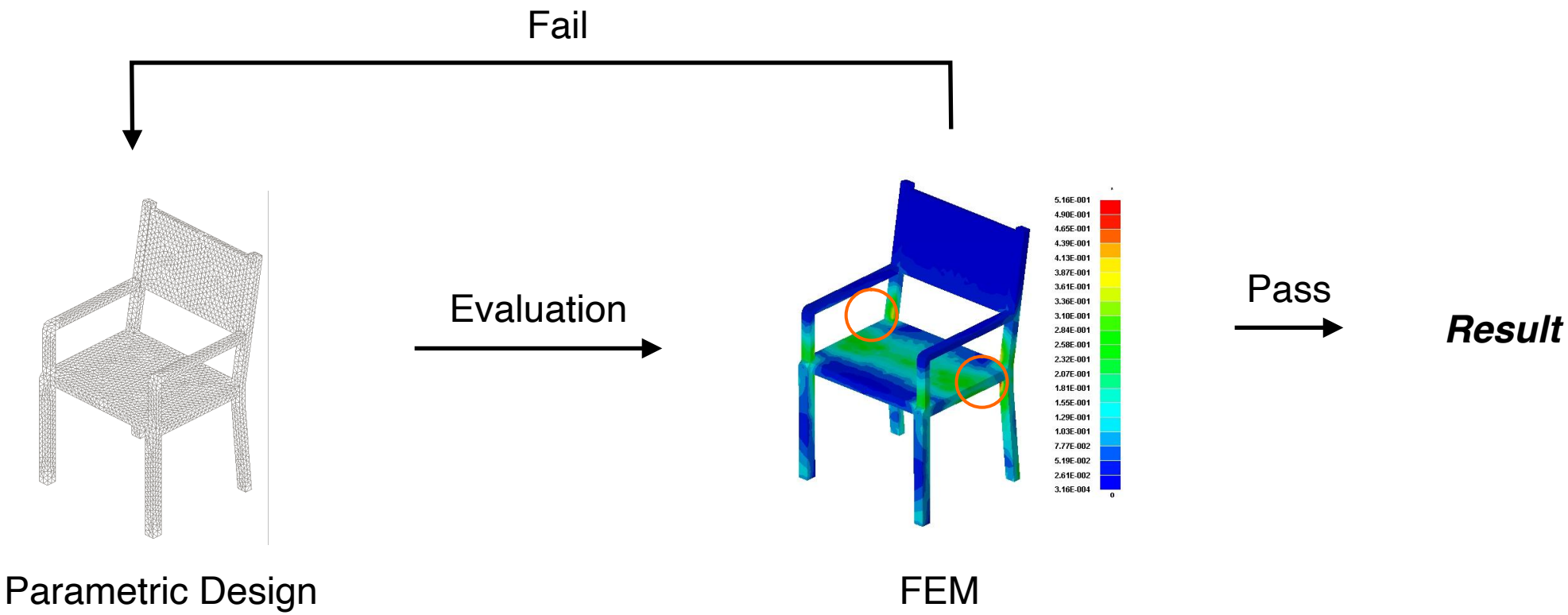
Easy-to-Assemble
Non-stable



Hard-to-Assemble
Stable

Challenges of Forward Design

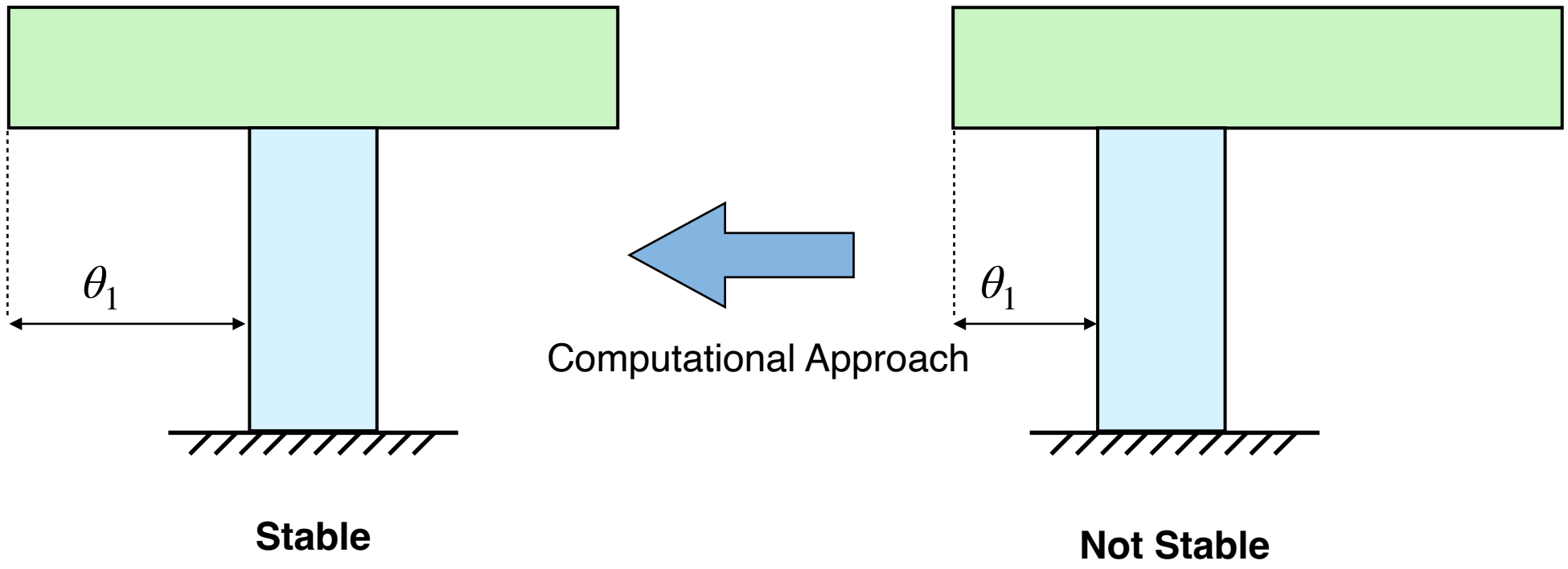
- Some evaluation processes are not time-efficient.



[Laemlaksakul et al. 2008]

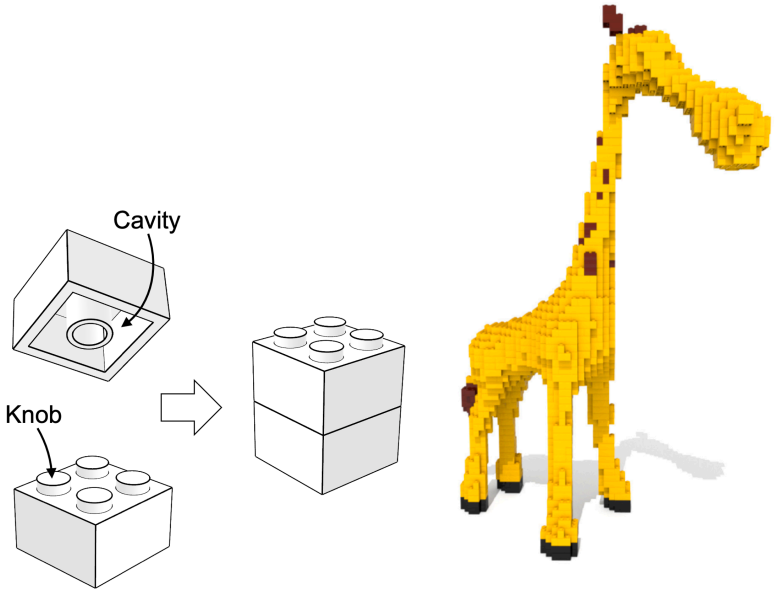
Inverse Design Framework

- Automatically generate assemblies that satisfy design objectives.



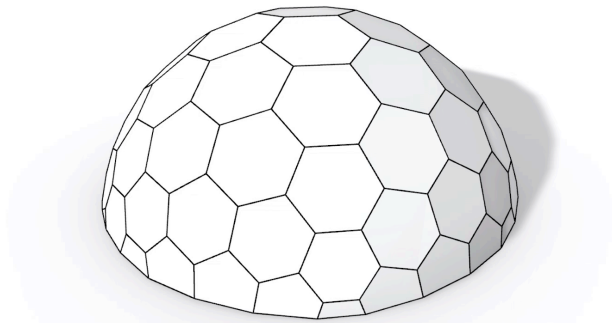
Part Geometry

- Discrete Geometry: searching algorithm.
- Continuous Geometry: gradient-based algorithm.



Discrete Geometry

[Luo et al. 2015]

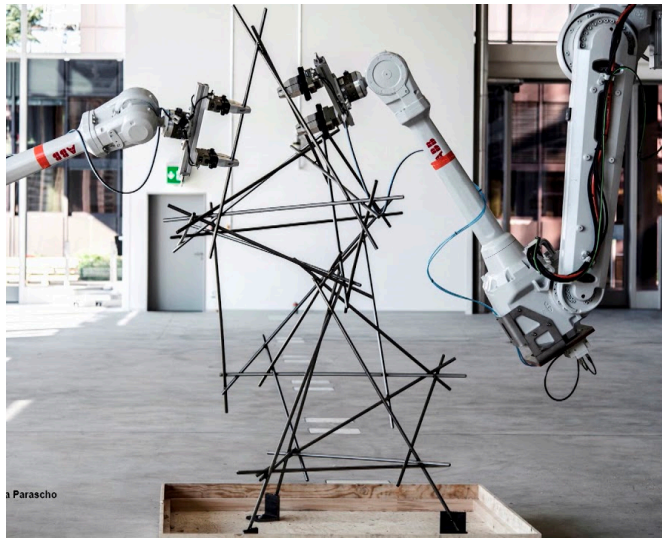


Continous Geometry

[Wang et al. 2019]

Objectives

Assemblability



[Parascho et al. 2017]

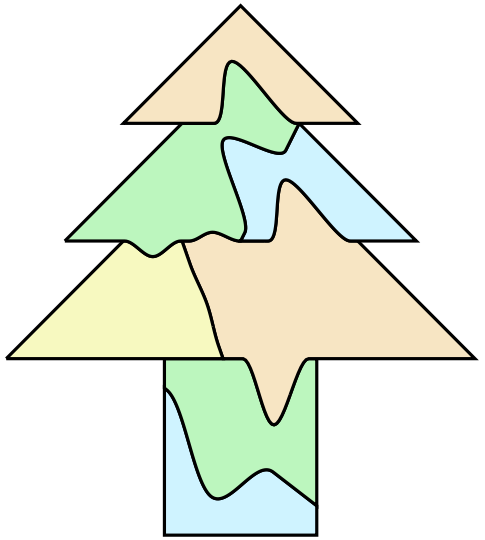
Fabricatility

[Cignoni et al. 2014]

Functionality

[Song et al. 2017]

Stability



[Wang et al. 2021]

Stability Optimization

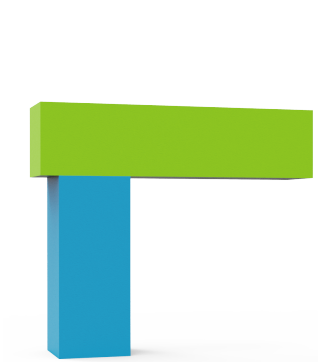
- Stability is the most fundamental requirement.
- Designing stable structures without glue/mortar is non-trivial.

[Nara Todaiji]

[MIT Sean Collier Memorial]

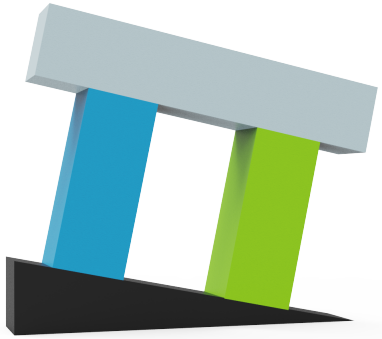
Stability Spectrum

- Besides gravitational equilibrium, we will also cover other types of structural stability.

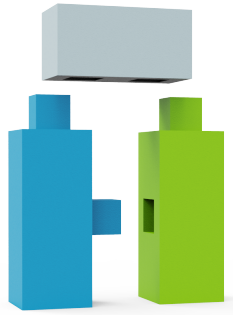


Non-equilibrium

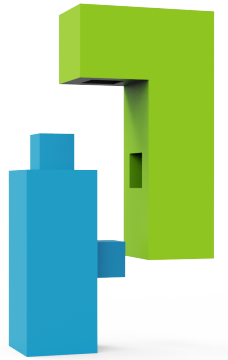
Equilibrium under gravity



Lateral stability



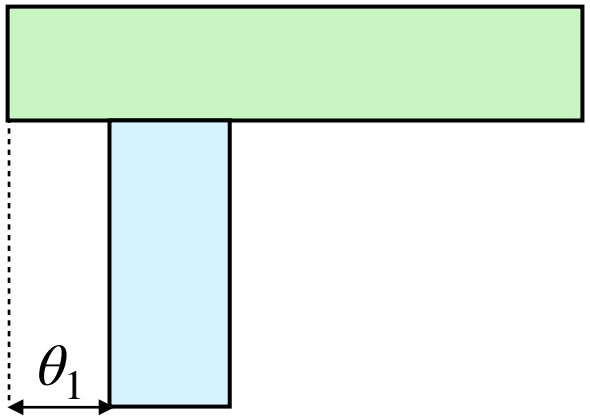
Globally Interlocking



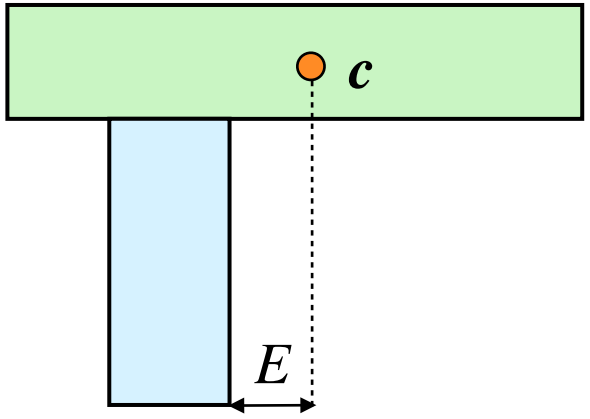
Deadlock

Overview

- Part 1: General stability optimization framework using the gradient information.



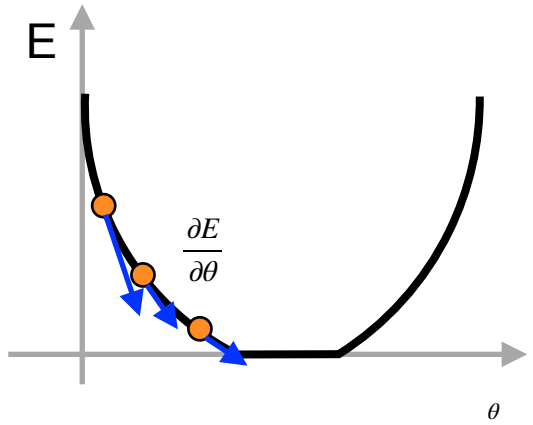
Step 1
Geometrical Property



Step 2
Infeasibility Energy

$$\boldsymbol{\theta} \xrightarrow{\partial} \boldsymbol{c} \xrightarrow{\partial} E$$

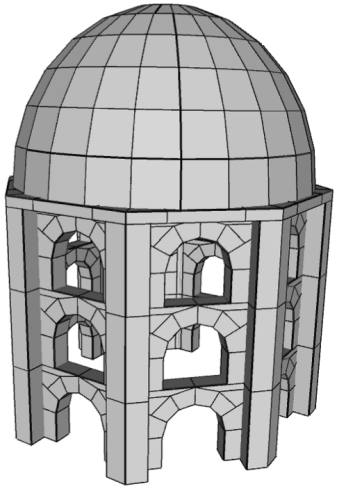
Step 3
Sensitivity Analysis



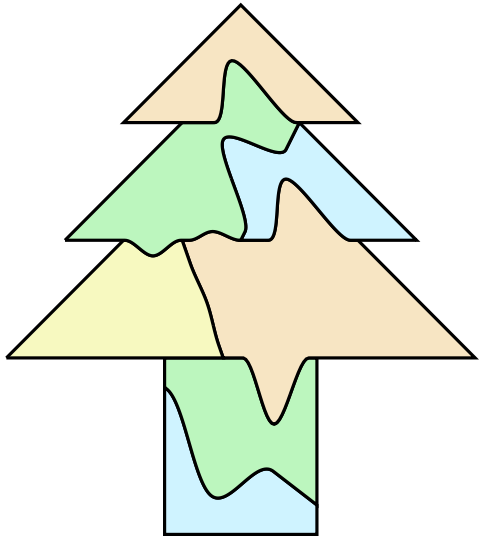
Step 4
Numerical Optimization

Overview

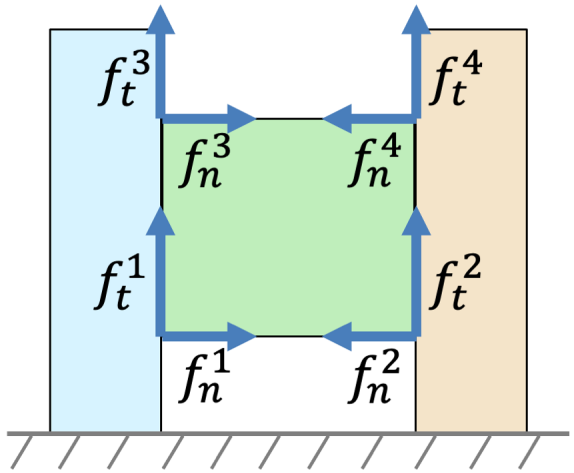
- Part 2: Stability optimization for gravitational equilibrium.
  - Force-based equilibrium method
  - Kinematic-based equilibrium method
  - Friction



[Whiting et al 2009, 2012]



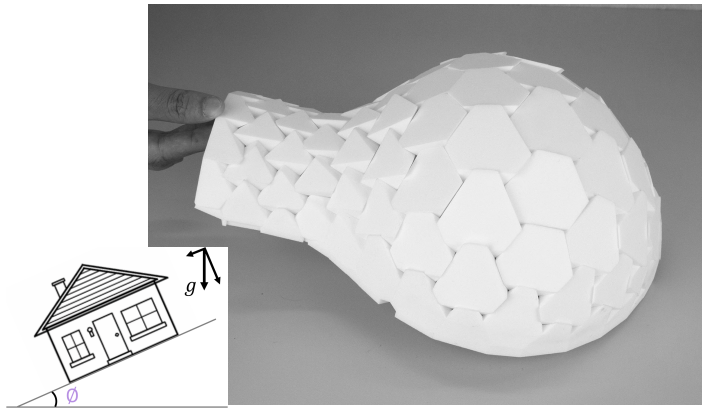
[Wang et al. 2021]



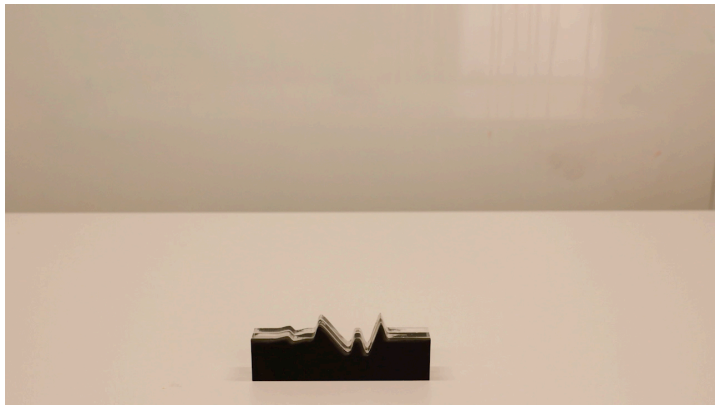
[Yao et al. 2017]

Overview

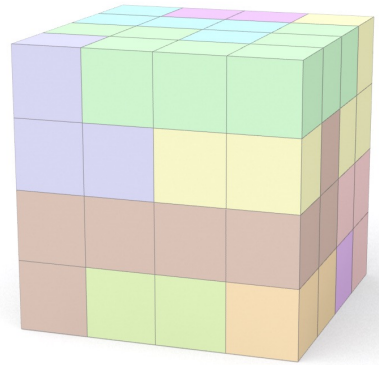
- Part 3: Design for stability under other types of forces
  - Lateral stability
  - Scaffolding-free assembly
  - Globally interlocking



[Wang et al. 2019]



[Wang et al. 2021]

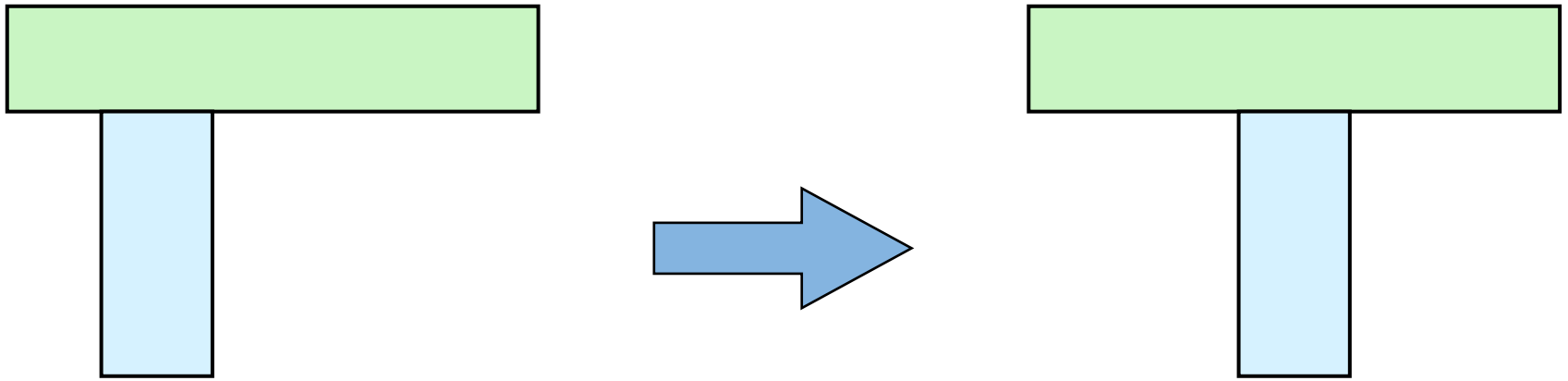


[Wang et al. 2018]

Part 1: General Stability Optimization Framework

Stability Optimization

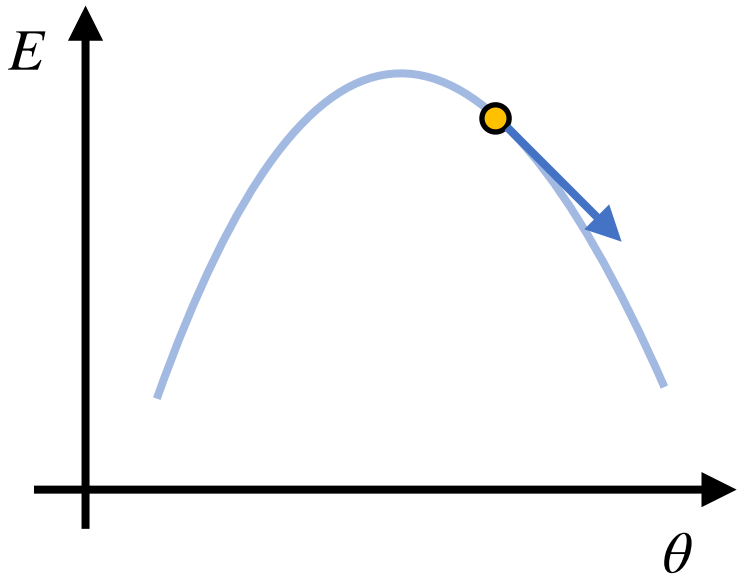
- By alternating the parts' geometry, making the assembly stable under certain loading conditions.



Unstable **Stable**

Gradient-based Optimization

- Gradient-based optimization is the most common approach to solve the inverse design problem.

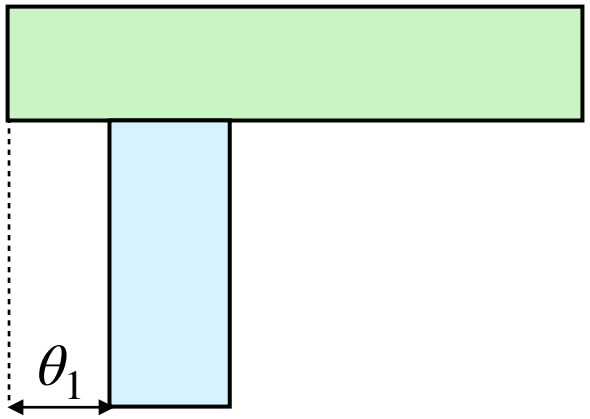


Strategy #1: Take a random downhill slope. **Slow**

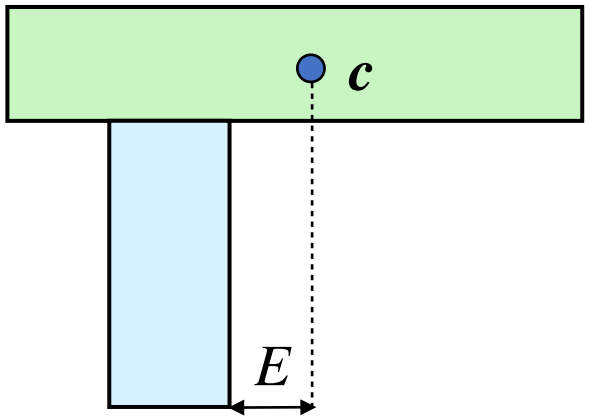
Strategy #2: Take the *steepest slope*! **Fast**

Gradient-based Stability Optimization

- Gradient-based stability optimization has four main components:



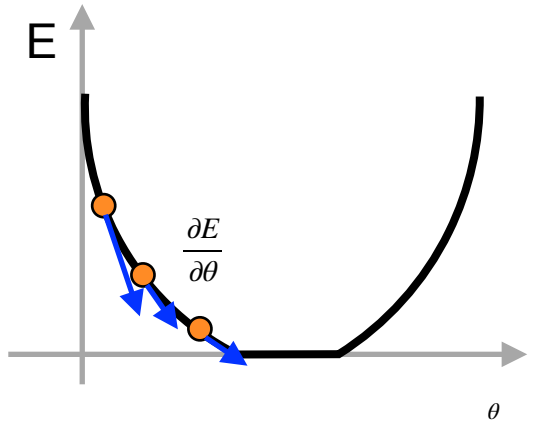
Step 1
Geometrical Property



Step 2
Infeasibility Energy

$$\boldsymbol{\theta} \xrightarrow{\partial} \boldsymbol{c} \xrightarrow{\partial} E$$

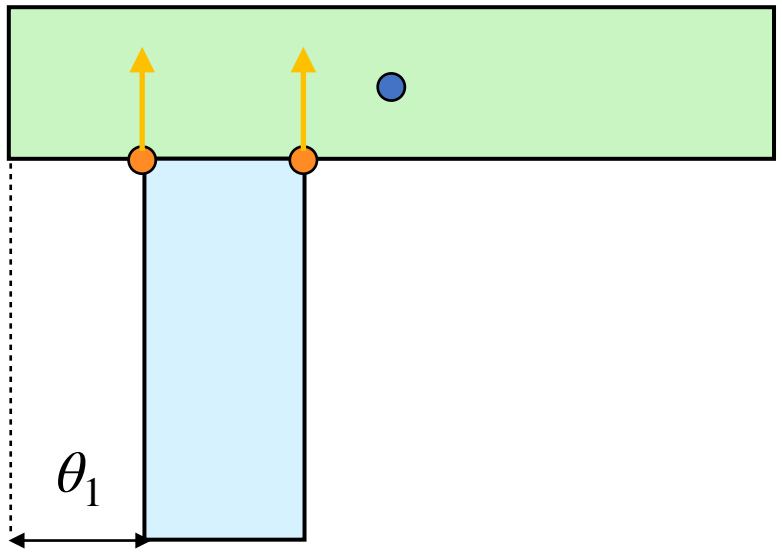
Step 3
Sensitivity Analysis



Step 4
Numerical Optimization

Step #1 Geometrical Property

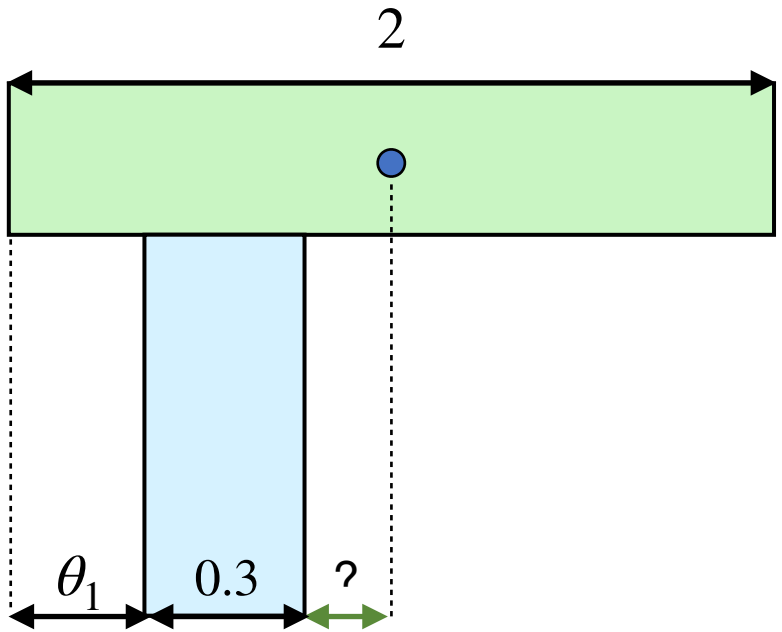
- Compute necessary geometrical properties for stability analysis.



1. Contact Points
2. Contact Normals
3. Parts' Centroids
4. Parts' Volumes

Step #2 Infeasibility Measurement

- Compute the infeasibility energy, which measures how unstable the structure is.



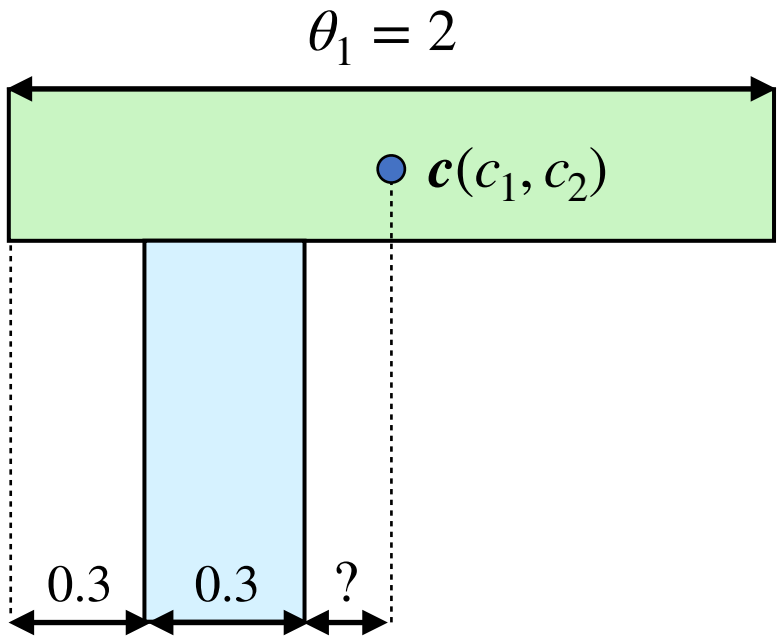
Infeasbility Energy

$$E = \left\| \text{—} \right\|^2$$

$$= (0.7 - \theta)^2$$

Step #3 Sensitivity Analysis

- Compute the infeasibility energy's gradient/hessian with respect to the design parameters.



Gradient: $\dfrac{\partial E}{\partial \theta_1}$

Chain Rule: $\dfrac{\partial E}{\partial \theta_1} = \dfrac{\partial E}{\partial c_1}\dfrac{\partial c_1}{\partial \theta_1}$

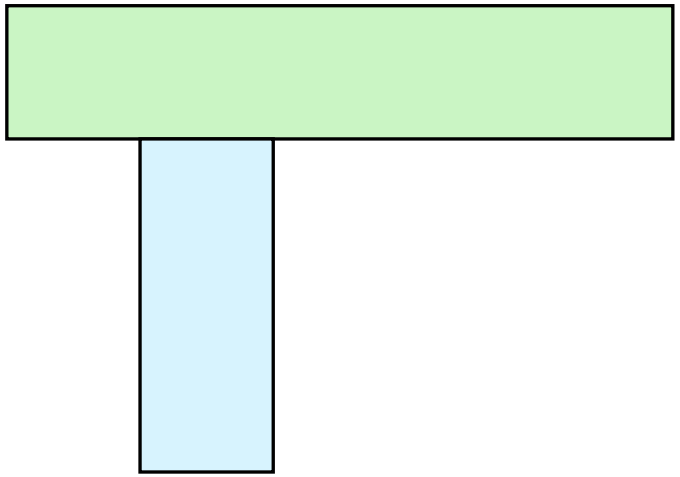
Step #4 Numerical Optimization

- Various numerical optimization tools can be used to solve the inverse design problem.

	Gradient Descent	**Netwon Method**	**Quasi-Netwon Method**
Data	Gradient	Hessian	Gradient
Speed	Slow	Fast	Medium
Code	Easy-to-implement	Hard-to-implement	Easy-to-implement

Part 2: Stability Optimization For Gravitational Equilibrium

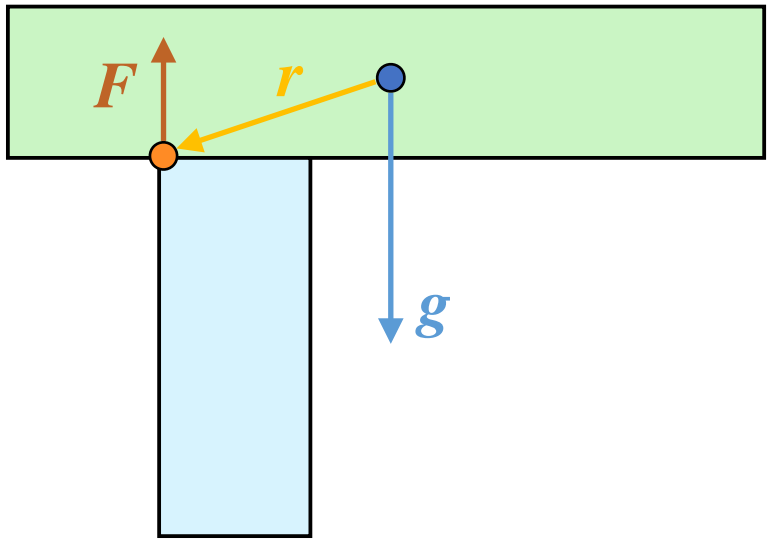
Assumptions



1. Parts are rigid body.
2. Friction is ignored.
3. The bottom part (blue) is fixed.

Recap: Rigid Body Equilibrium

- Rigid body equilibrium can check whether the internal and external forces/torque of a given structure are balanced.



Force Balance: $\sum F + g = 0$

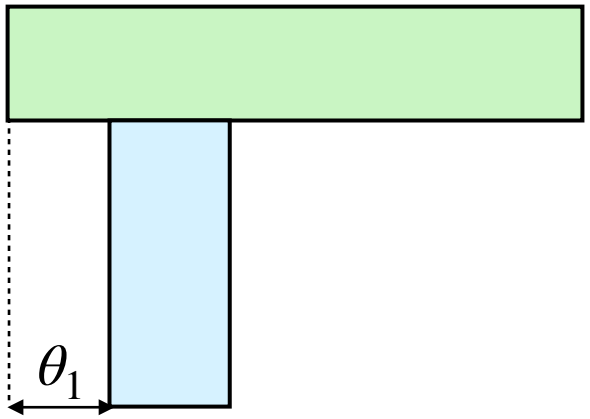
Torque Balance: $\sum r \times F = 0$

Non-negative: $F \geq 0$

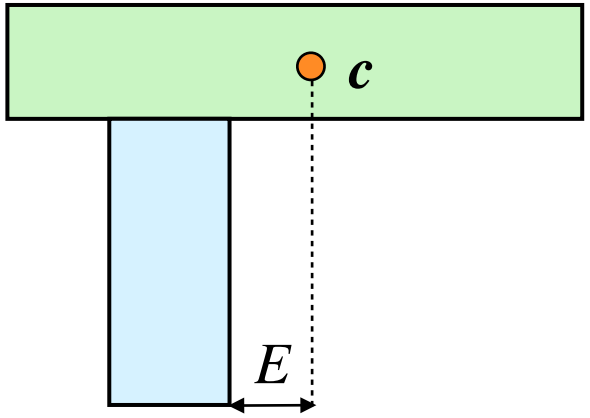
Problems: only provides a binary result (yes/no).

Recap: Gradient-based Stability Optimization

- Compute faithful infeasibility energy.



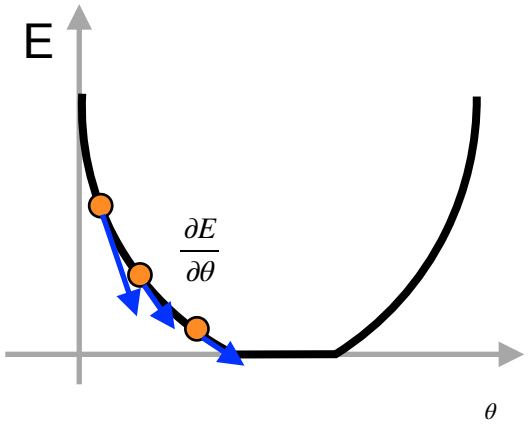
Step 1
Geometrical Property



Step 2
Infeasibility Energy

$$\boldsymbol{\theta} \xrightarrow{\partial} \boldsymbol{c} \xrightarrow{\partial} E$$

Step 3
Sensitivity Analysis

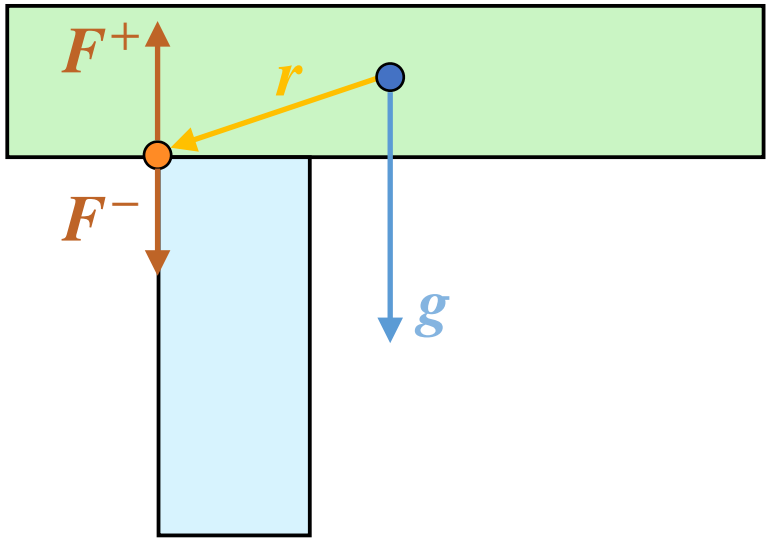


Step 4
Numerical Optimization

Equilibrium Infeasibility Energy

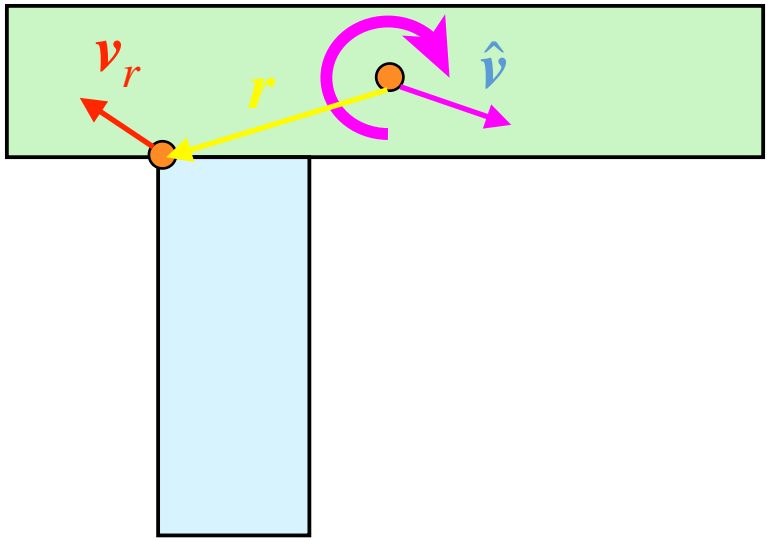
- Two ways of computing infeasibility energy for equilibrium problems.

Force-based Equilibrium Method



[Whiting et al 2009, 2012]

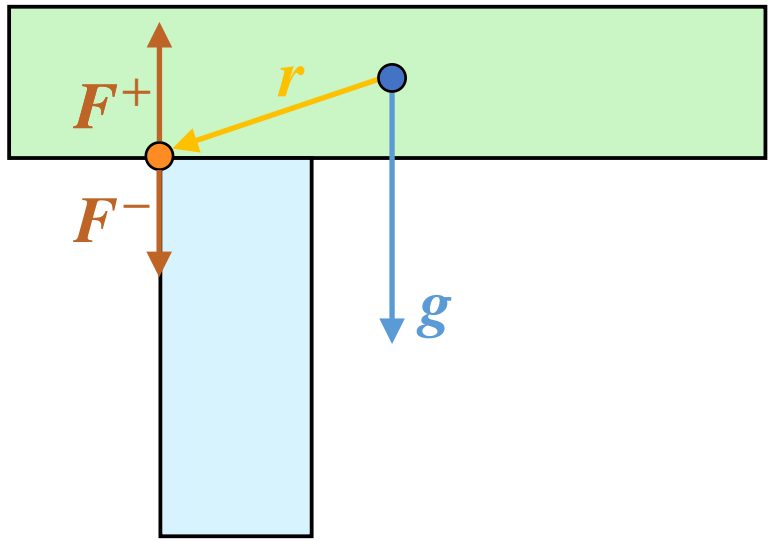
Kinematic-based Equilibrium Method



[Wang et al 2021]

Force-based Infeasibility Measurement

- Split each contact force $\boldsymbol{F}$ into the positive and negative parts $\boldsymbol{F}^+, \boldsymbol{F}^-$.
- The norm of the negative contact force is used to compute the infeasibility energy.



[Whiting et al 2009, 2012]

Minimizing tension: $\min \sum ||F^-||^2$

Force Balance: $\sum F + g = 0$

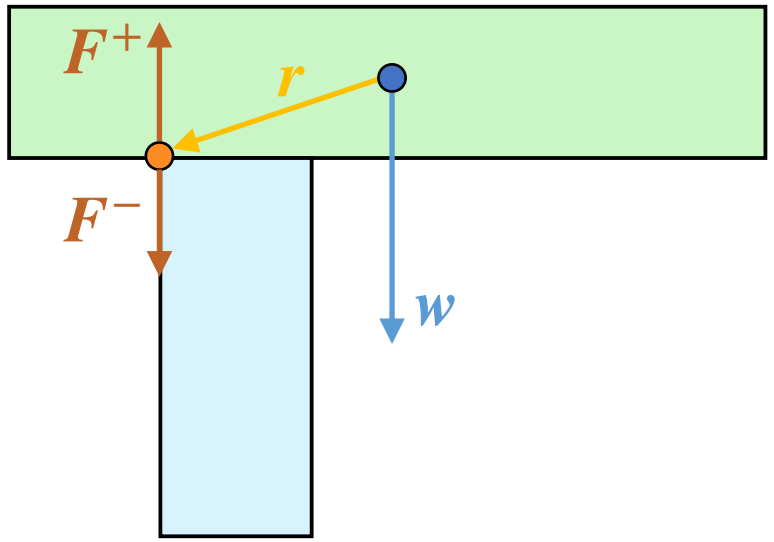
Torque Balance: $\sum r \times F = 0$

Non-negative: $F^+, F^- \geq 0$

$F = F^+ - F^-$

Quadratic Programming

- The infeasibility energy can be computed by a quadratic programming solver.



[Whiting et al 2009, 2012]

Minimizing tension: $\min \sum ||F^-||^2$

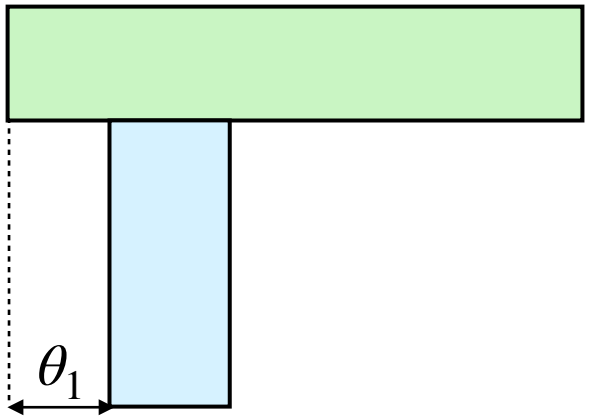
Force/Torque Balance: $A_{eq}F + w = 0$

Non-negative: $F^+, F^- \geq 0$

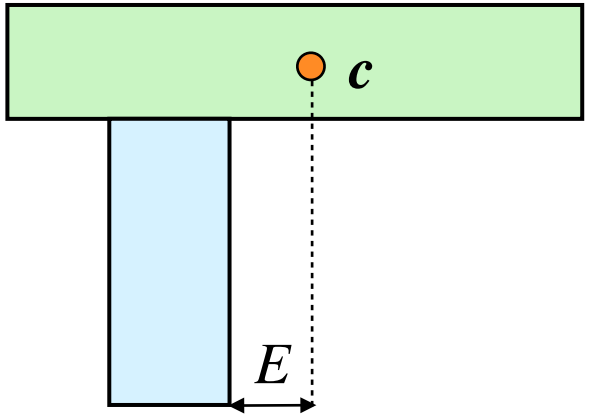
$F = F^+ - F^-$

Gradient-based Stability Optimization

- The next challenging step is to compute gradient using sensitivity analysis.



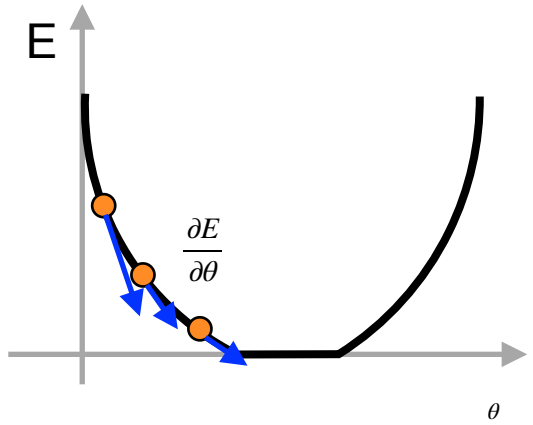
Step 1
Geometry Computing



Step 2
Infeasibility Energy

$$\boldsymbol{\theta} \xrightarrow{\partial} \boldsymbol{c} \xrightarrow{\partial} E$$

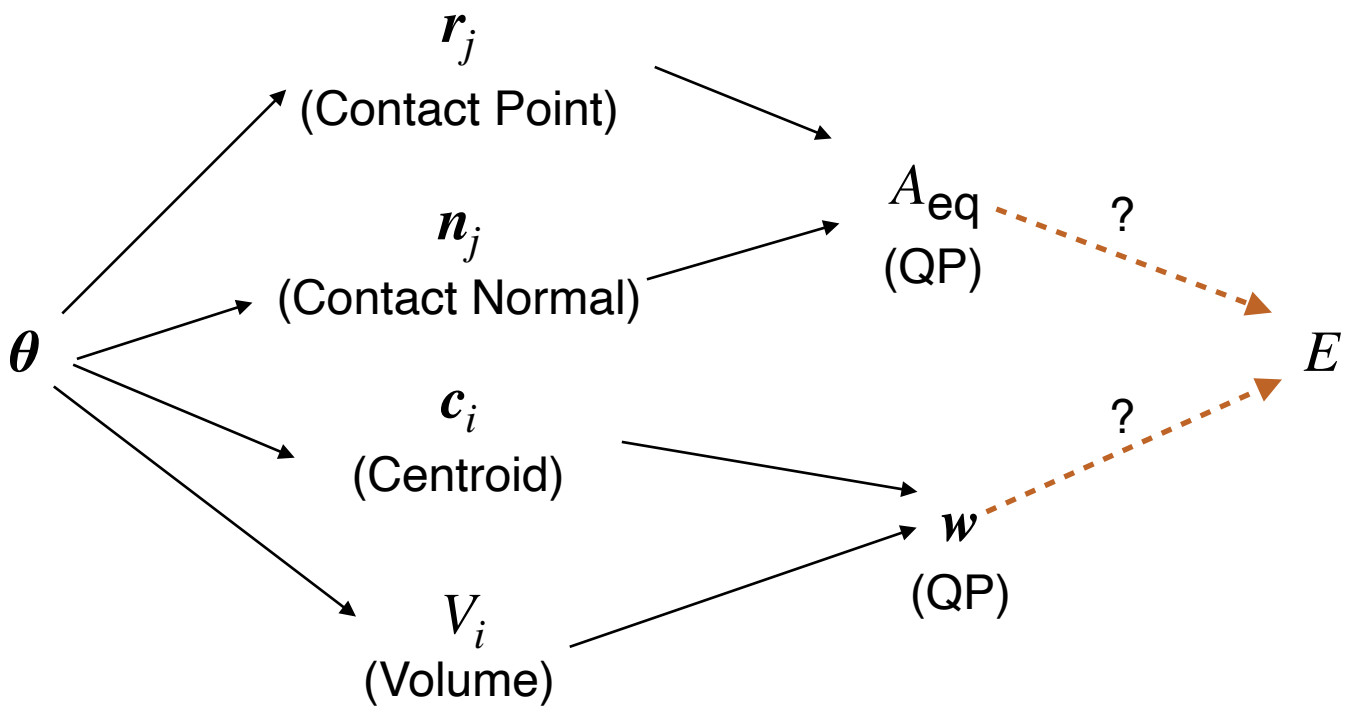
Step 3
Sensitivity Analysis



Step 4
Numerical Optimization

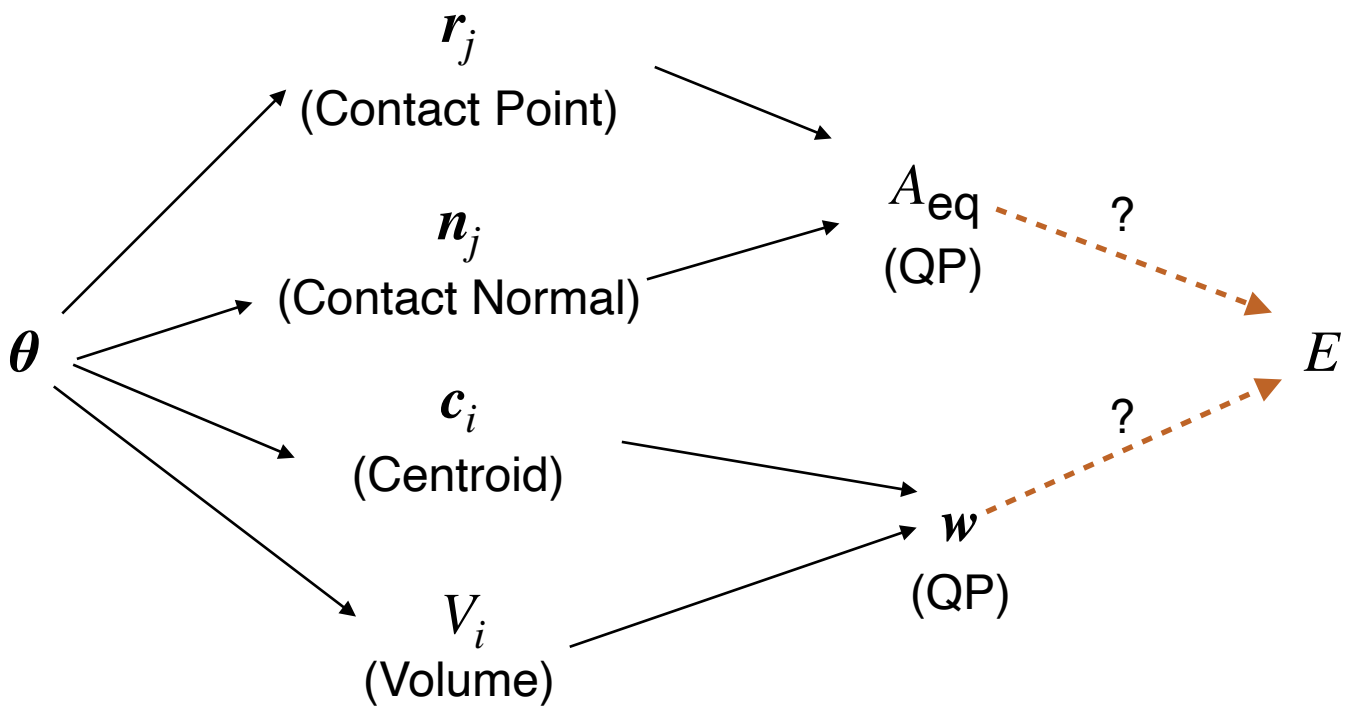
Chain Rule

- The chain rule help compute the gradient.
- However, the infeasibility energy's gradient with respect to the QP's coefficients are missing.



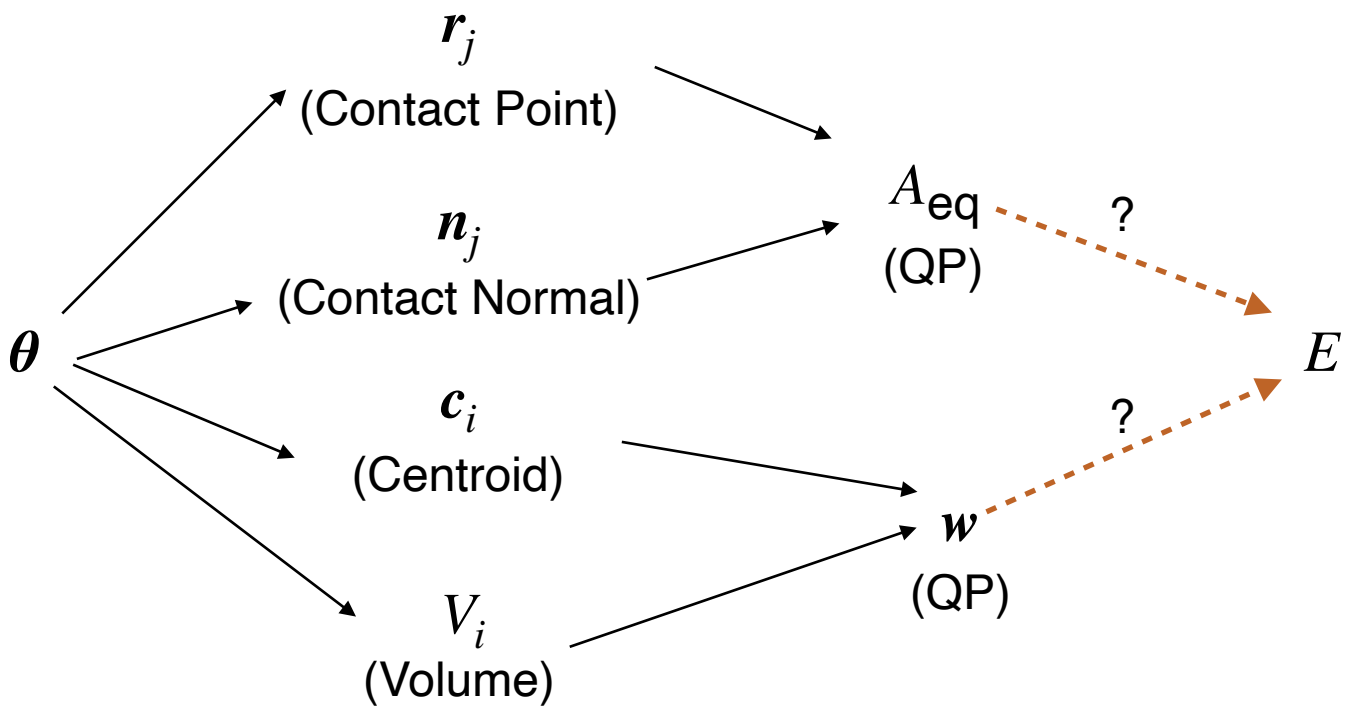
Chain Rule

- The chain rule help compute the gradient.
- However, the infeasibility energy's gradient with respect to the QP's coefficients are missing.



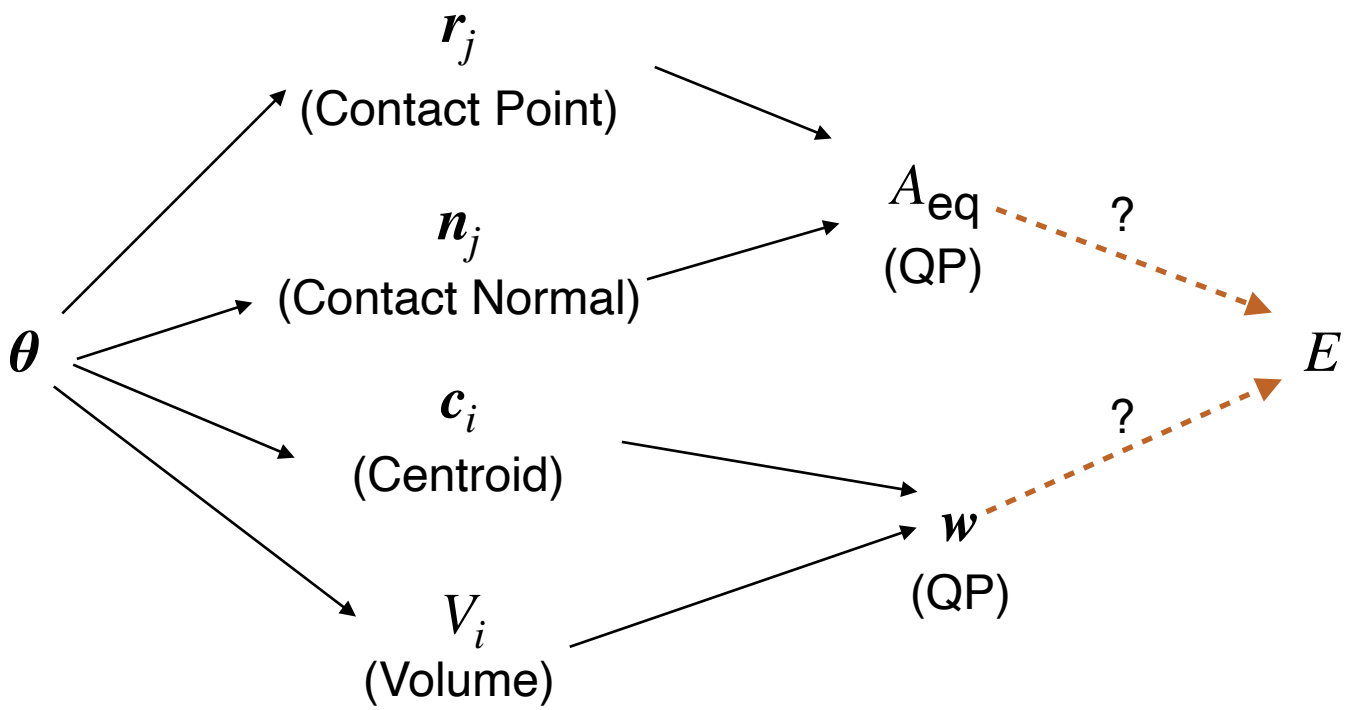
Chain Rule

- The chain rule help compute the gradient.
- However, the infeasibility energy’s gradient with respect to the QP's coefficients are missing.



Chain Rule

- The chain rule help compute the gradient.
- However, the infeasibility energy's gradient with respect to the QP's coefficients are missing.



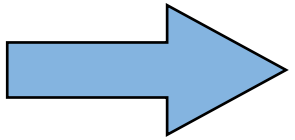
Sensitivtiy Analysis of QP

- A closed-form solution is available for the QP problem with only equality constraints.

$$E(A_{\text{eq}}, w) = \min \sum ||\boldsymbol{F}^{-}||^2$$

$$A_{\text{eq}}\boldsymbol{F} + \boldsymbol{w} = 0$$

$$\boldsymbol{F}^{+}, \boldsymbol{F}^{-} \geq 0$$

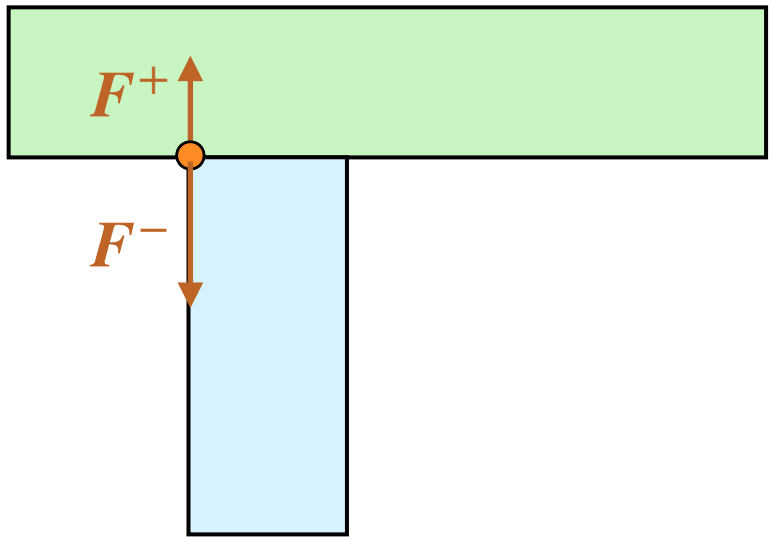


Closed-Form Solution

[Whiting et al 2009, 2012]

Sensitivtiy Analysis of QP

- Local perturbation of the geometry will only change the resulting force slightly.



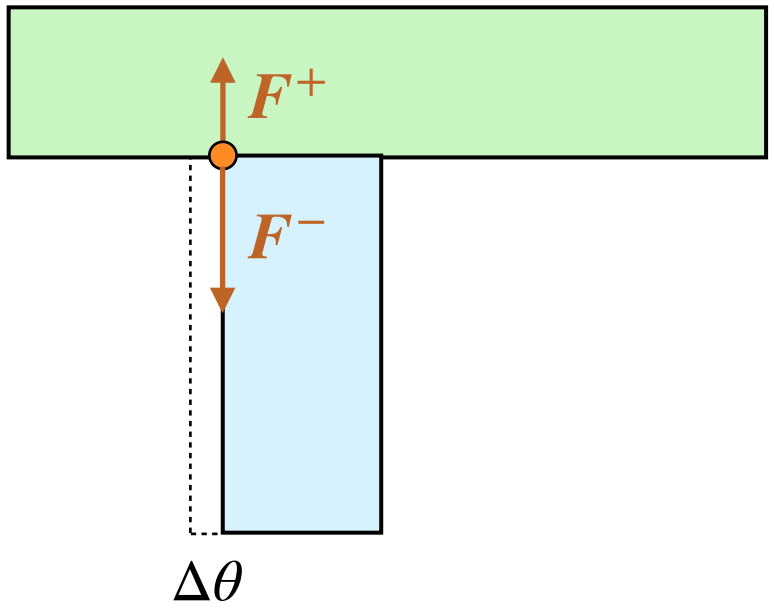
$$F^+ = 1.0$$

$$F^- = 1.5$$

[Whiting et al 2009, 2012]

Sensitivtiy Analysis of QP

- Local perturbation of the geometry will only change the resulting force slightly.



$$F^+ = 1.01$$

$$F^- = 1.49$$

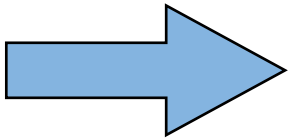
[Whiting et al 2009, 2012]

Sensitivtiy Analysis of QP

- Applying region trust algorithm to replace inequalities with equalities.

$$E(A_{\text{eq}}, w) = \min \sum ||F^-||^2$$

$$A_{\text{eq}}F + w = 0$$



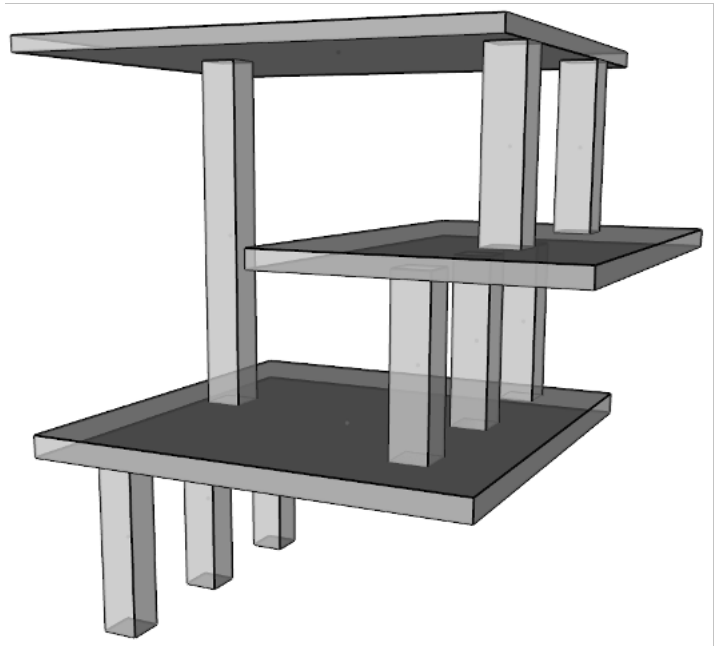
Closed-Form Solution

$$F^+, F^- \geq 0$$

$$F_i^+ = 0, \quad F_j^- = 0$$

[Whiting et al 2009, 2012]

Result



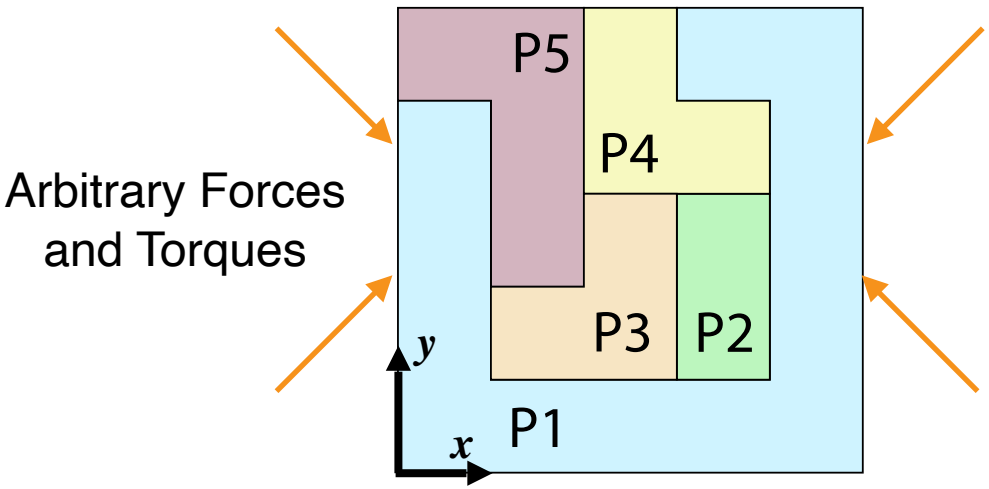
Unstable output

Stable output

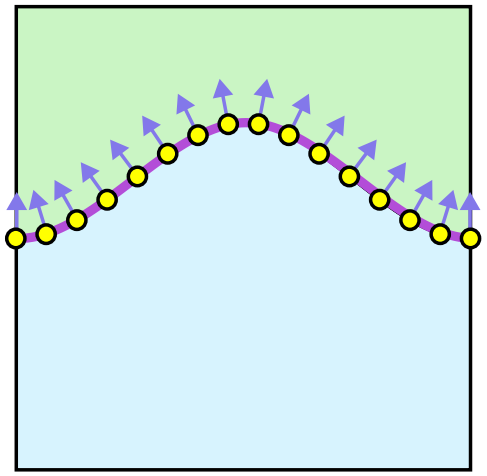
[Whiting et al 2012]

Limitations of Force-based Method

- Hard to test for some stability types (i.e., globally interlocking)
- Less efficient when handling parts with non-planar contacts.



Globally Interlocking Assemblies

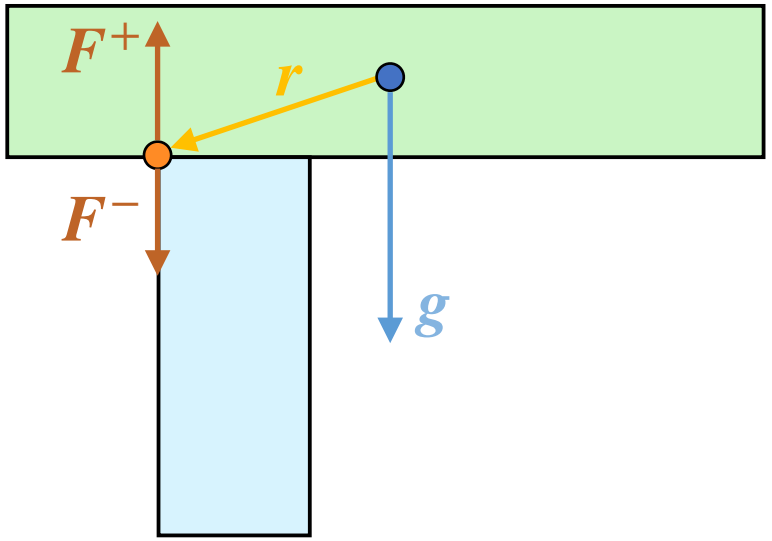


Curve Contacts

Kinematic-based Equilibrium Method

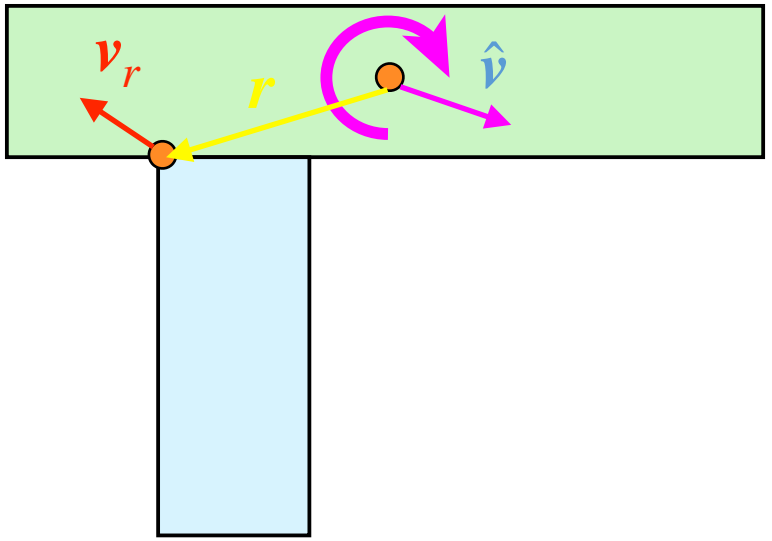
- Kinematic-based method measures infeasibility in the motion space.

Force-based Equilibrium Method



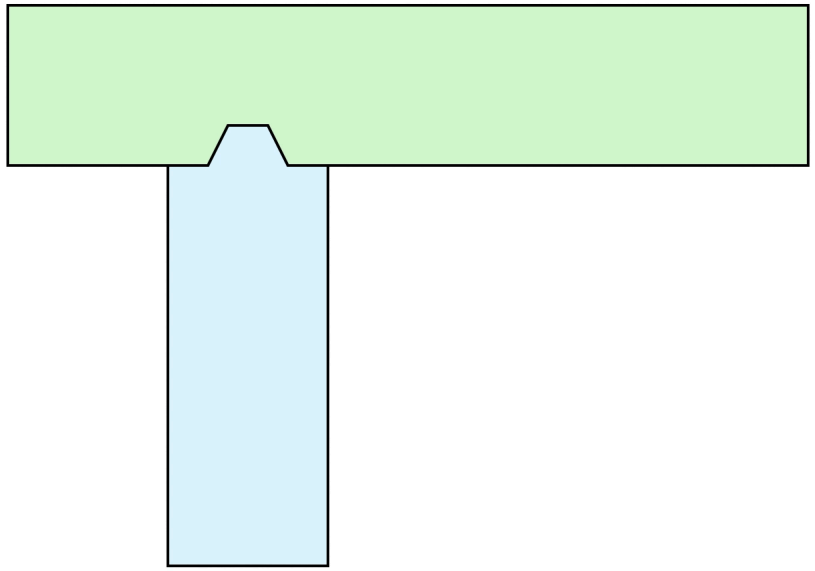
[Whiting et al 2009, 2012]

Kinematic-based Equilibrium Method



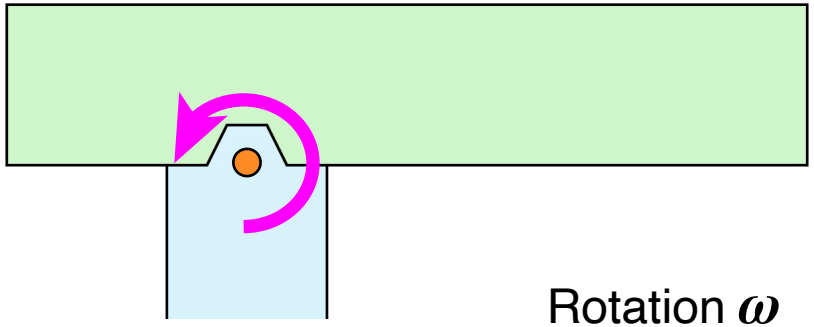
[Wang et al 2021]

Infinitesimal Rigid Motion



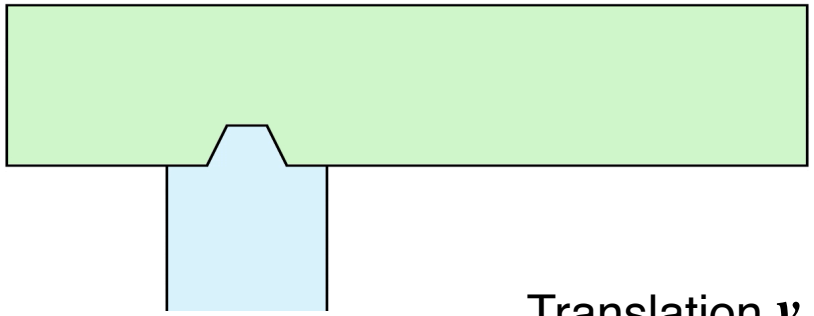
Infinitesimal rigid motion $\hat{\boldsymbol{v}} = (\boldsymbol{v}, \boldsymbol{\omega})$

=



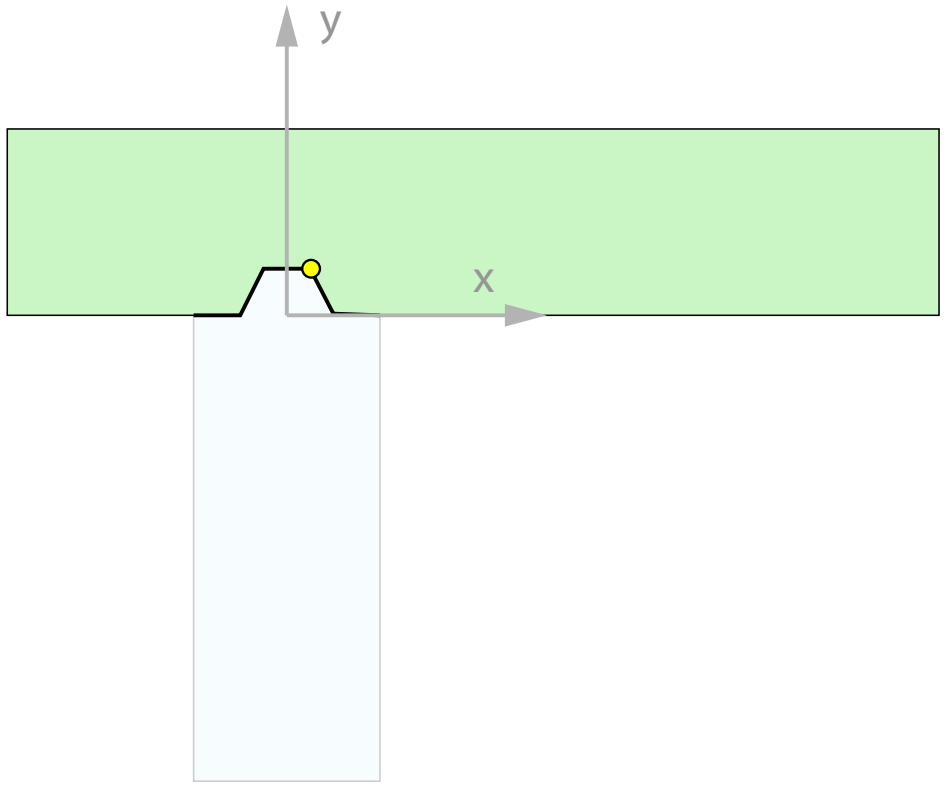
Rotation $\boldsymbol{\omega}$

+



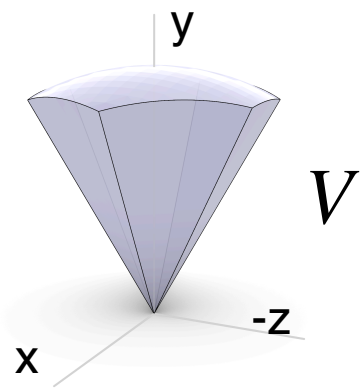
Translation $\boldsymbol{v}$

Motion Space



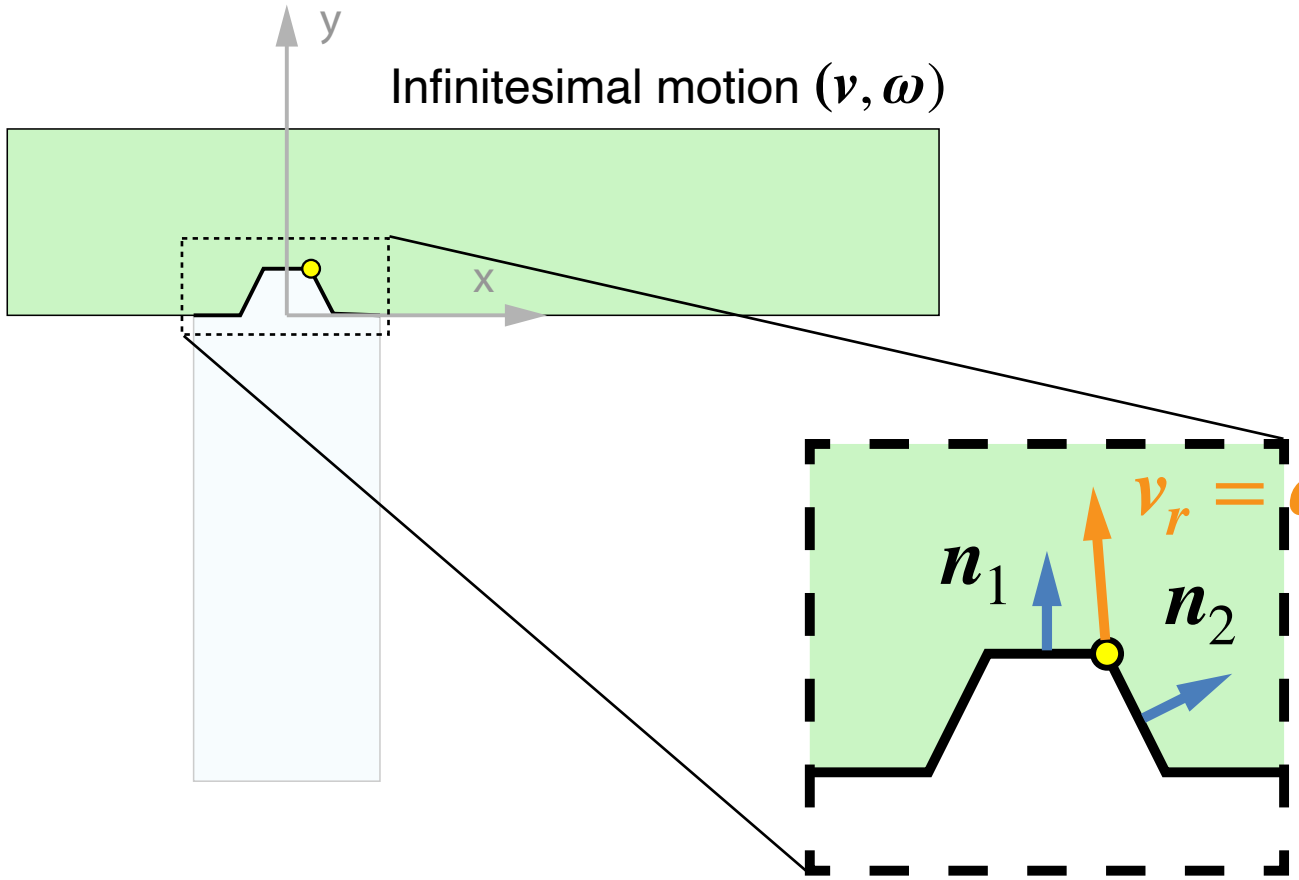
The motion space V of green part

=



{collision-free infinitesimal rigid motions $\hat{v}$}

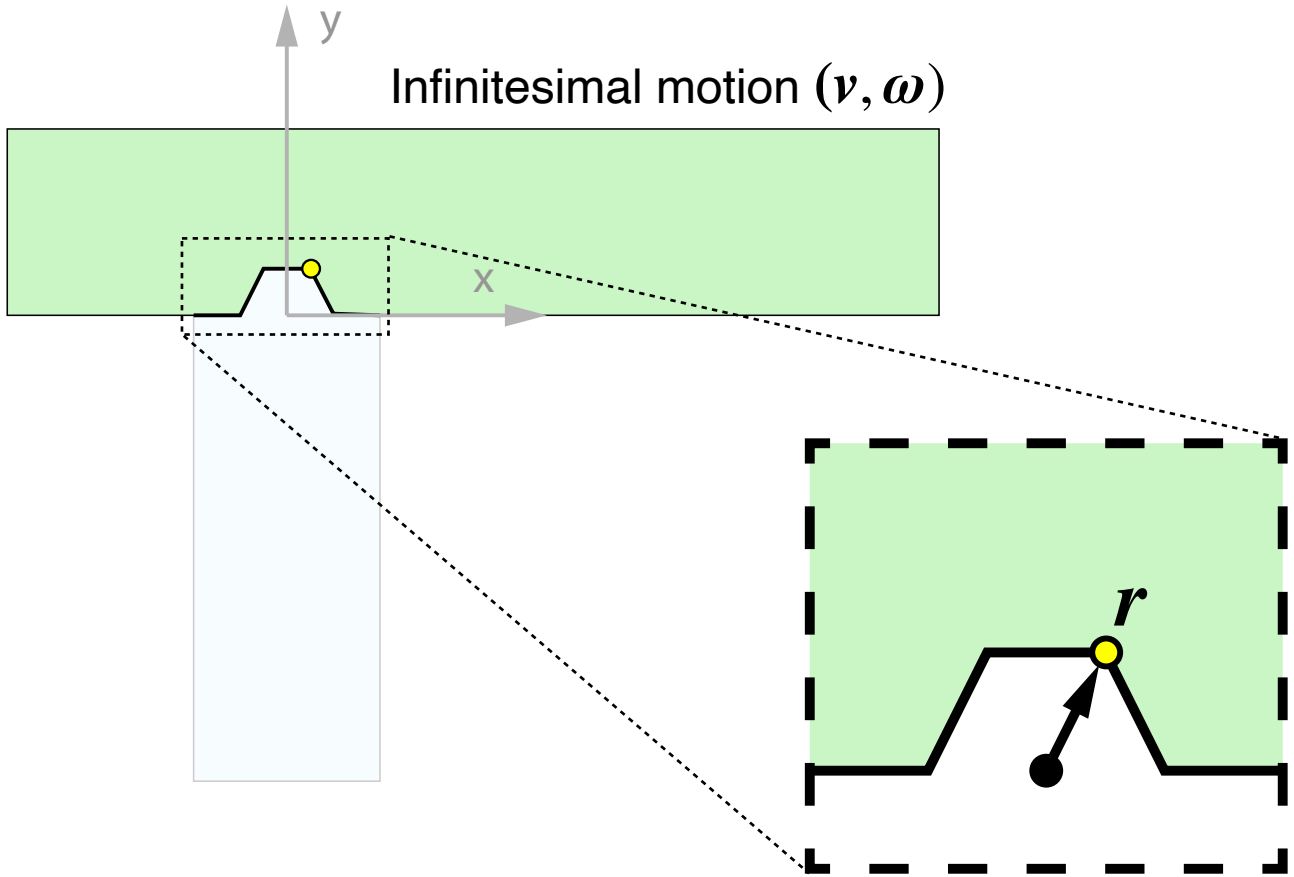
Non-collision constraints



$$v_r \cdot n \geq 0$$

Non collision constraints

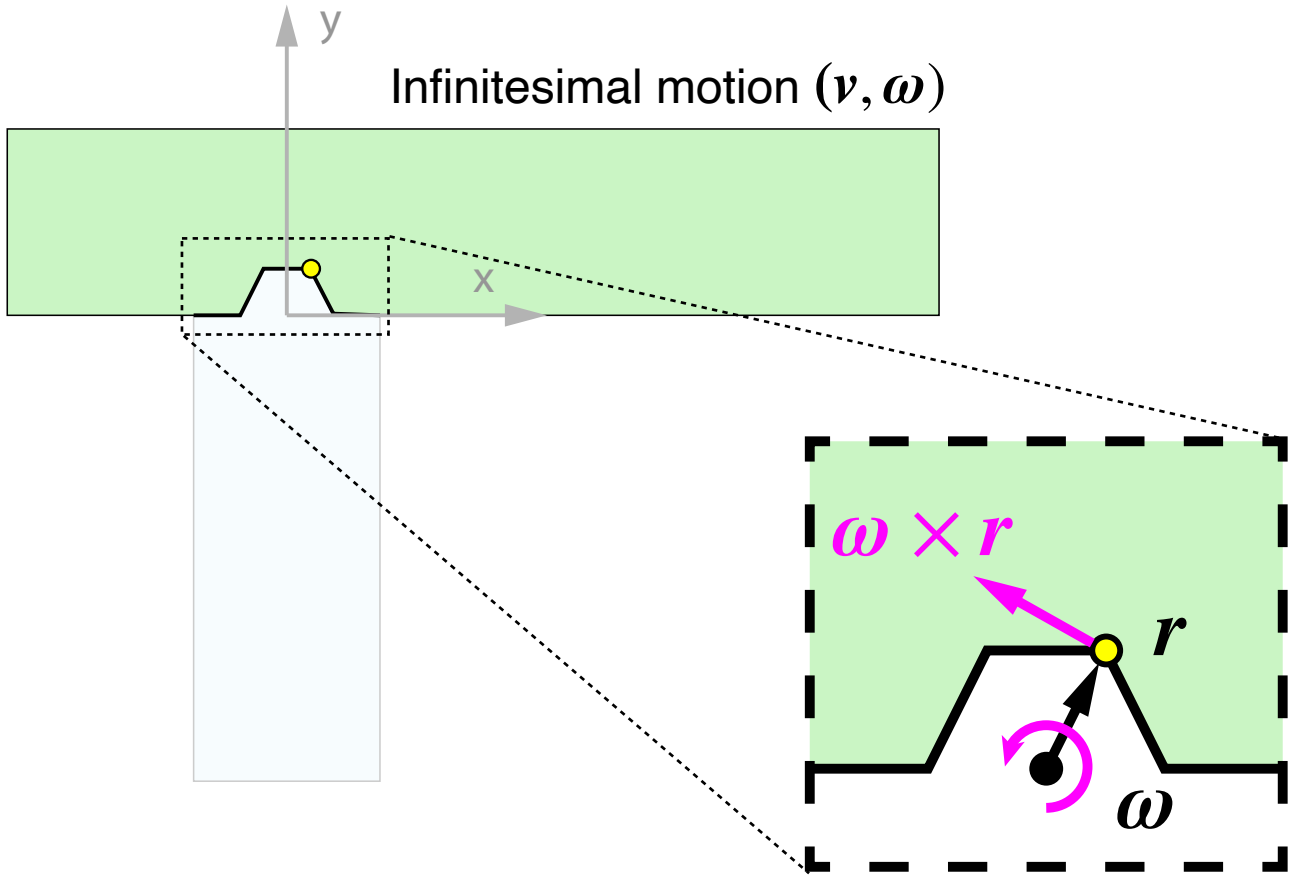
Contact Points' Velocities



velocity v_r

=

Contact Points' Velocities



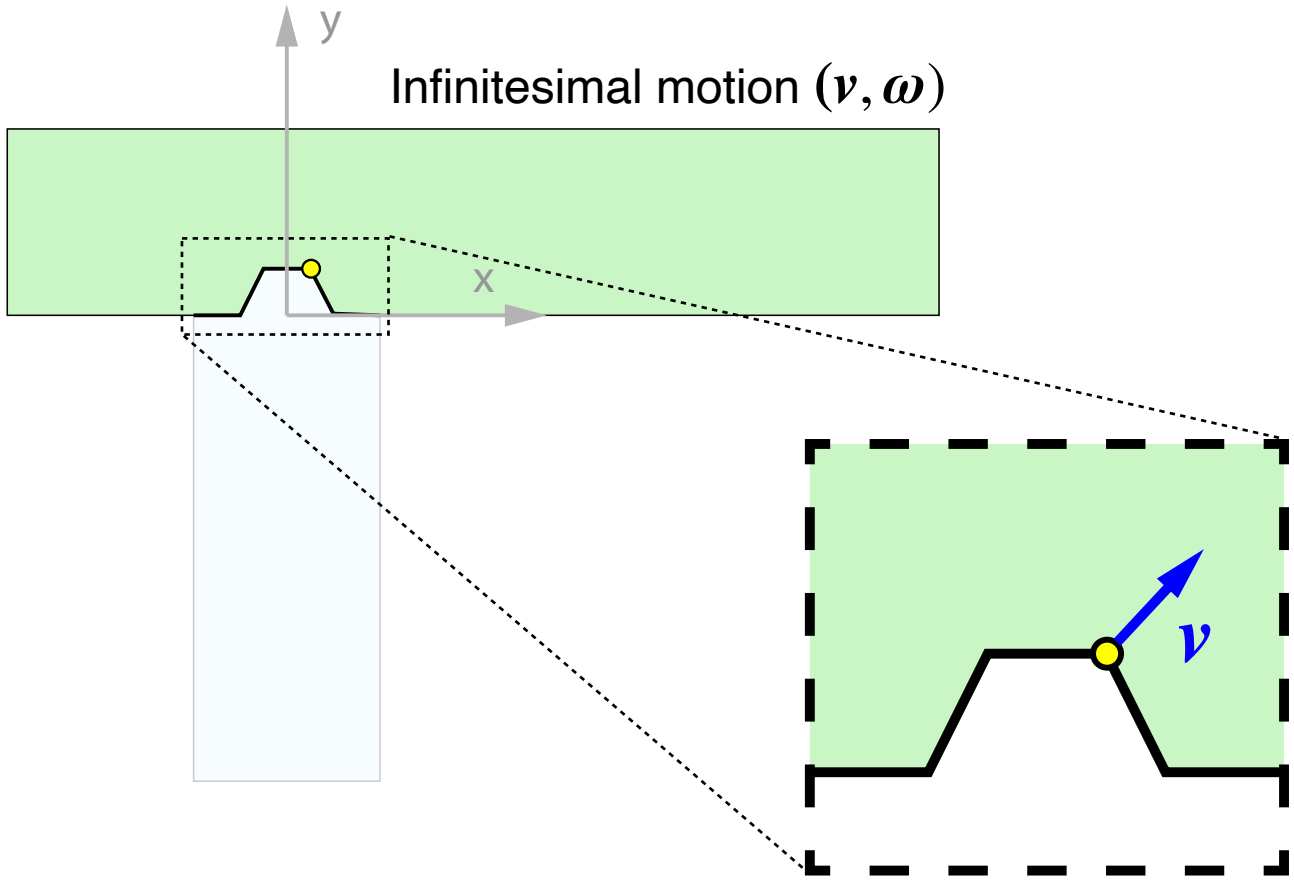
velocity v_r

=

rotation $\omega \times r$

+

Contact Points' Velocities



velocity v_r

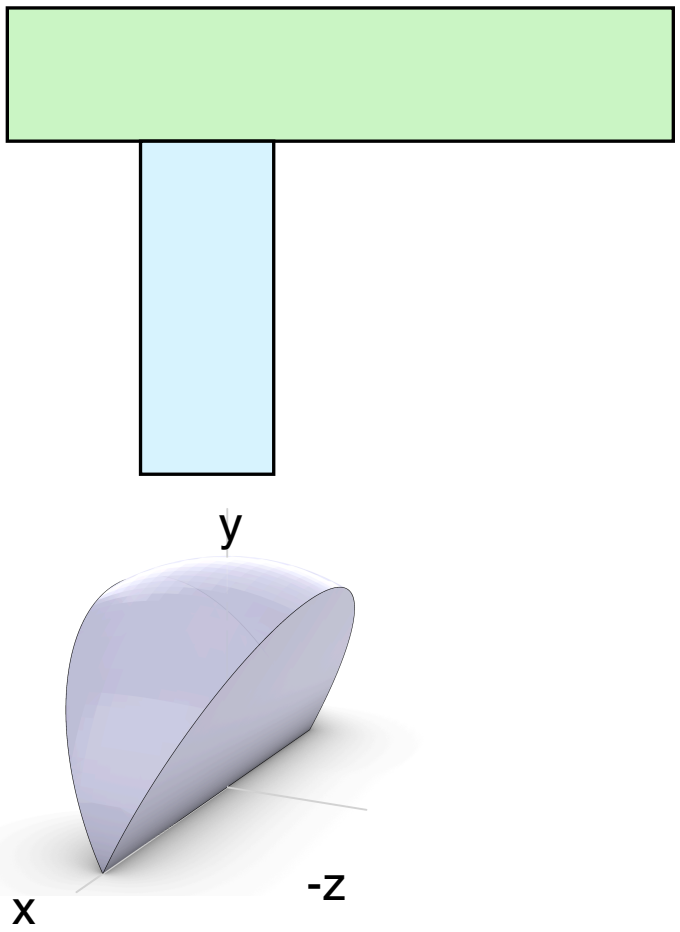
=

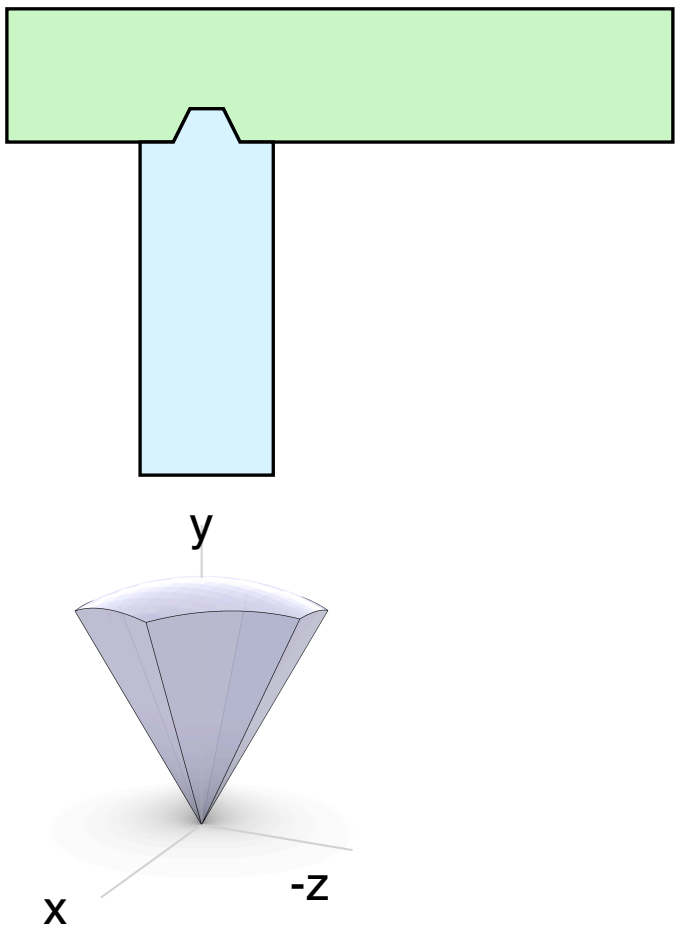
rotation $\omega \times r$

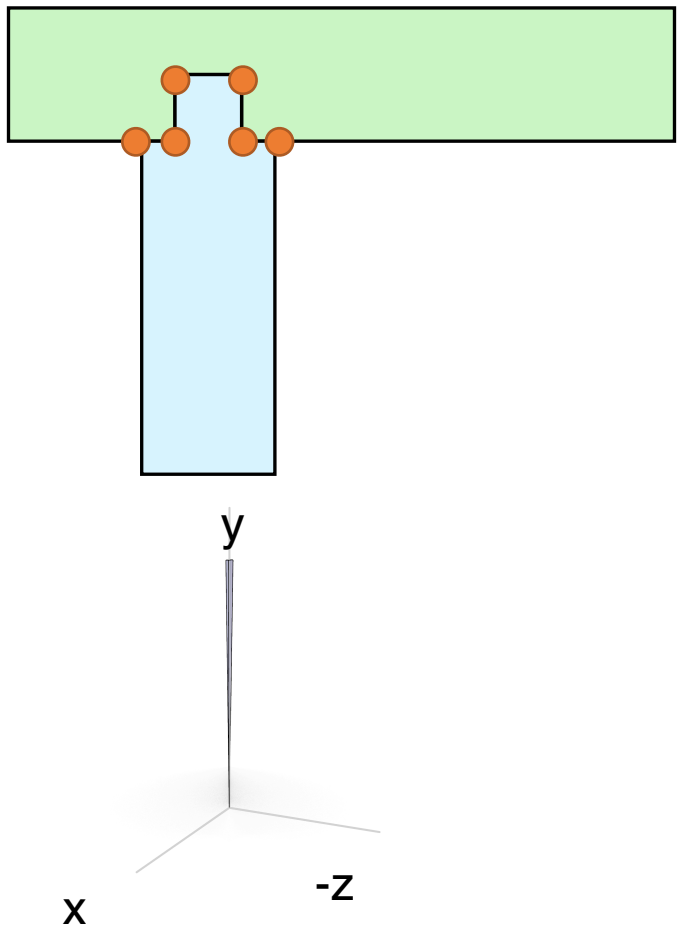
+

Translation v

Motion Cone of Contacts

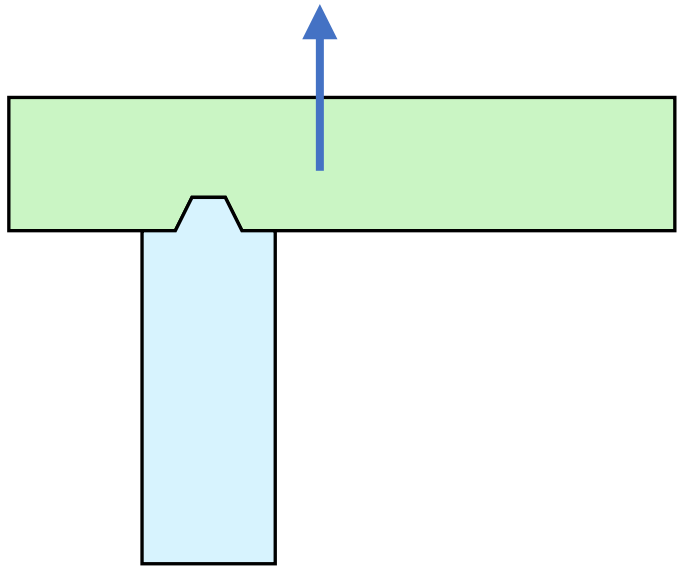


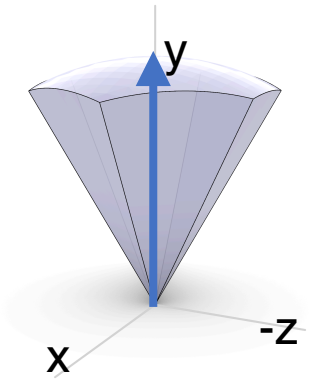




Physically Feasible Motion

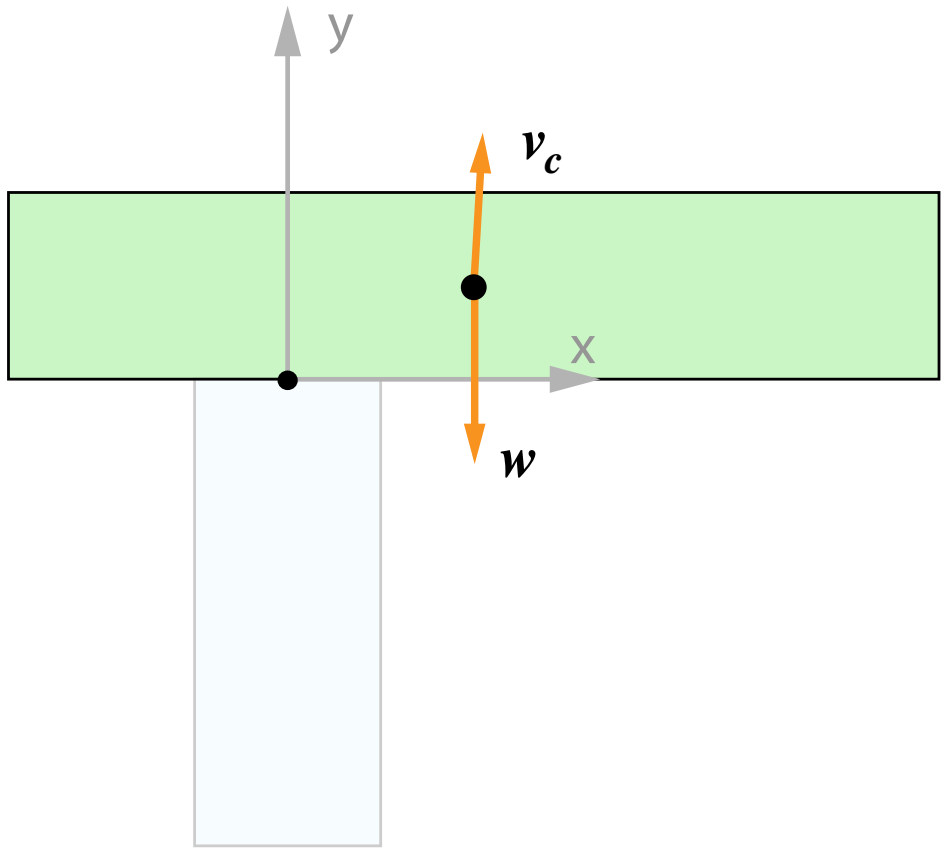
- Not every motion in the motion cone is physically plausible.





The translation along +y direction is not physically achievable.

Feasible Motion Space



Infinitesimal motion $\hat{v} = (v, \omega)$

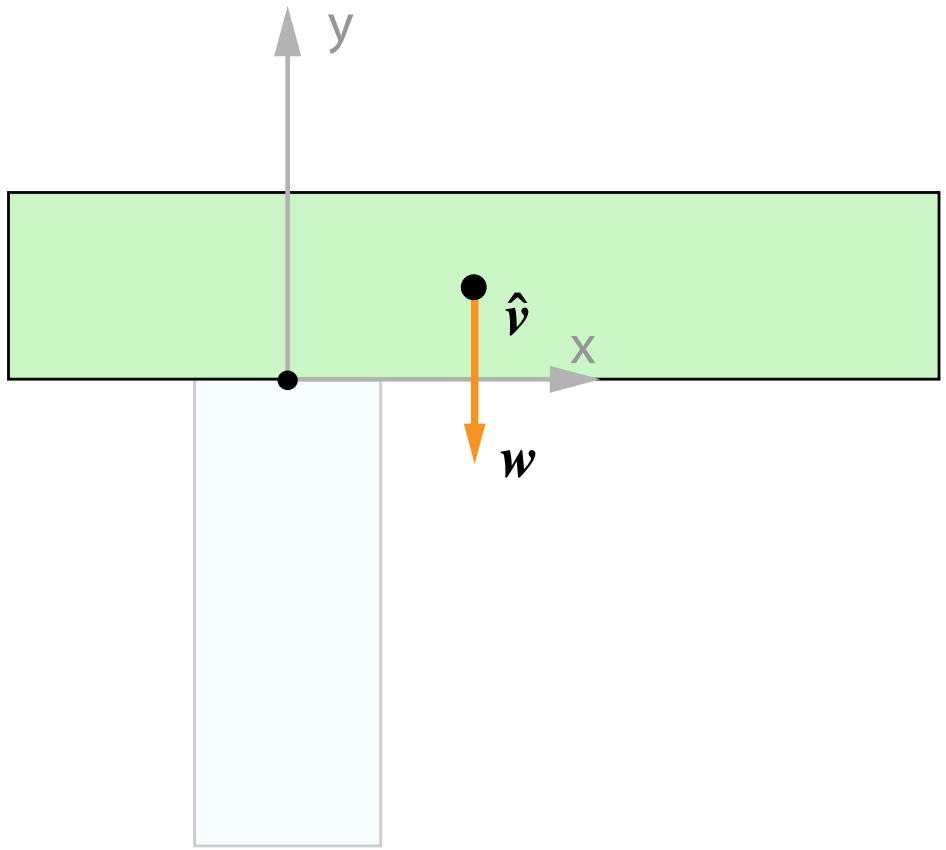
Velocity v_c at part's centre of mass

decreases its gravitational potential

$$v_c \cdot g > 0$$

$$\hat{v} \cdot w > 0$$

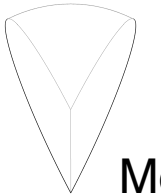
Feasible Motions



Infinitesimal motion $\hat{v} = (v, \omega)$

Assembly is in equilibrium when

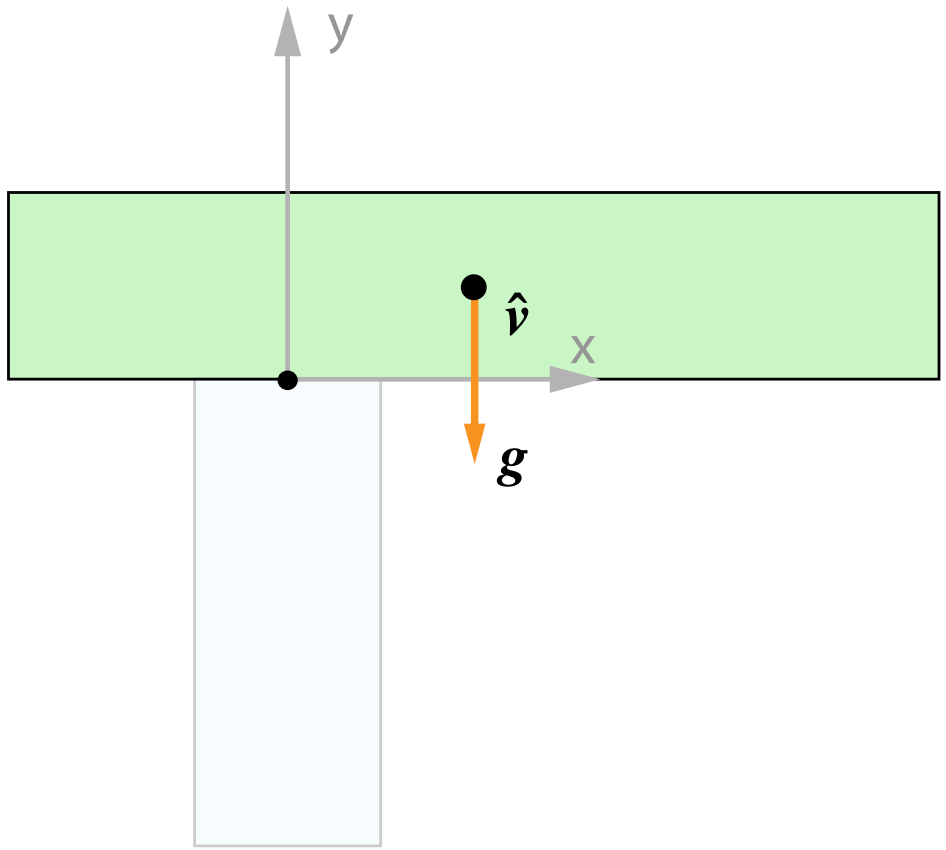
$$\begin{cases} \hat{v} \cdot w > 0 \\ \hat{v} \in \end{cases}$$

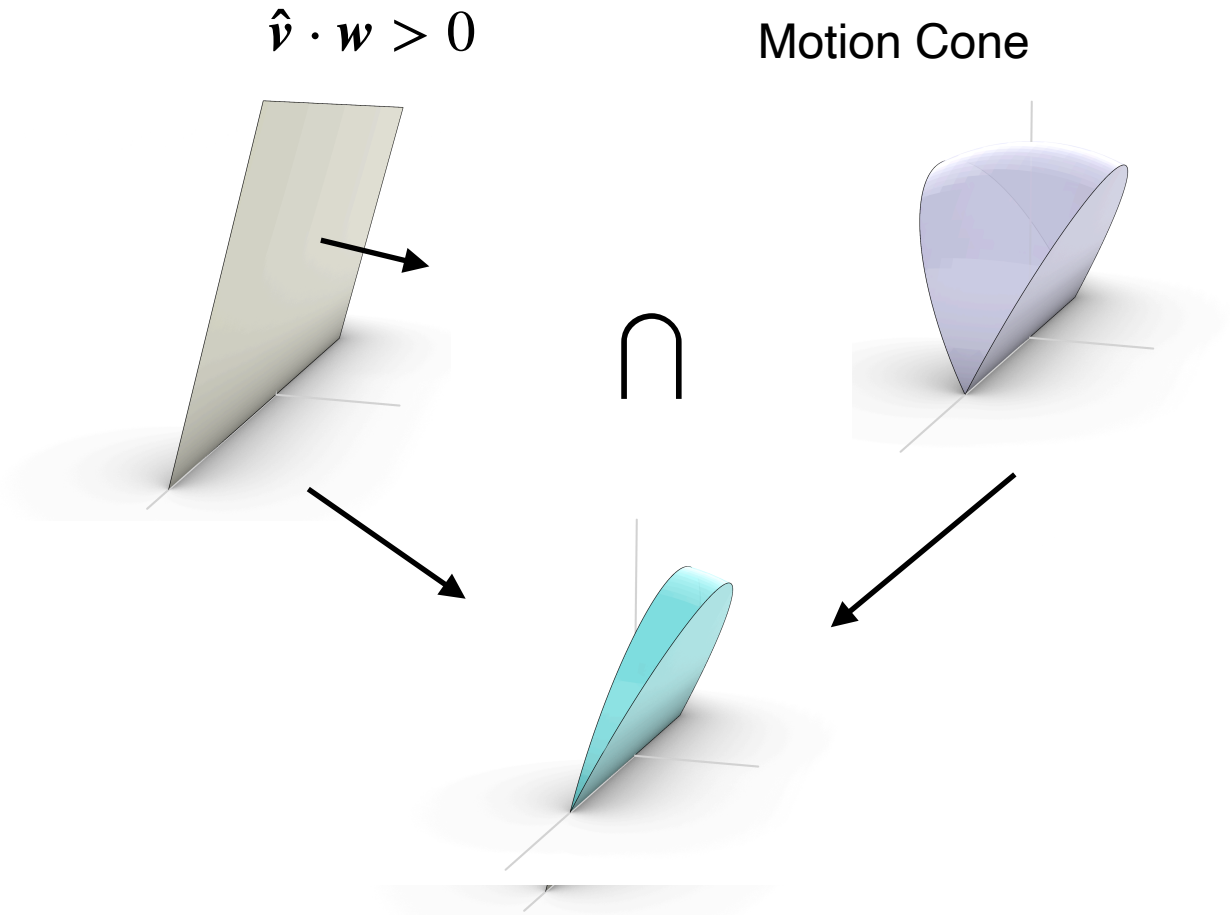


Motion Cone

does not have solutions.

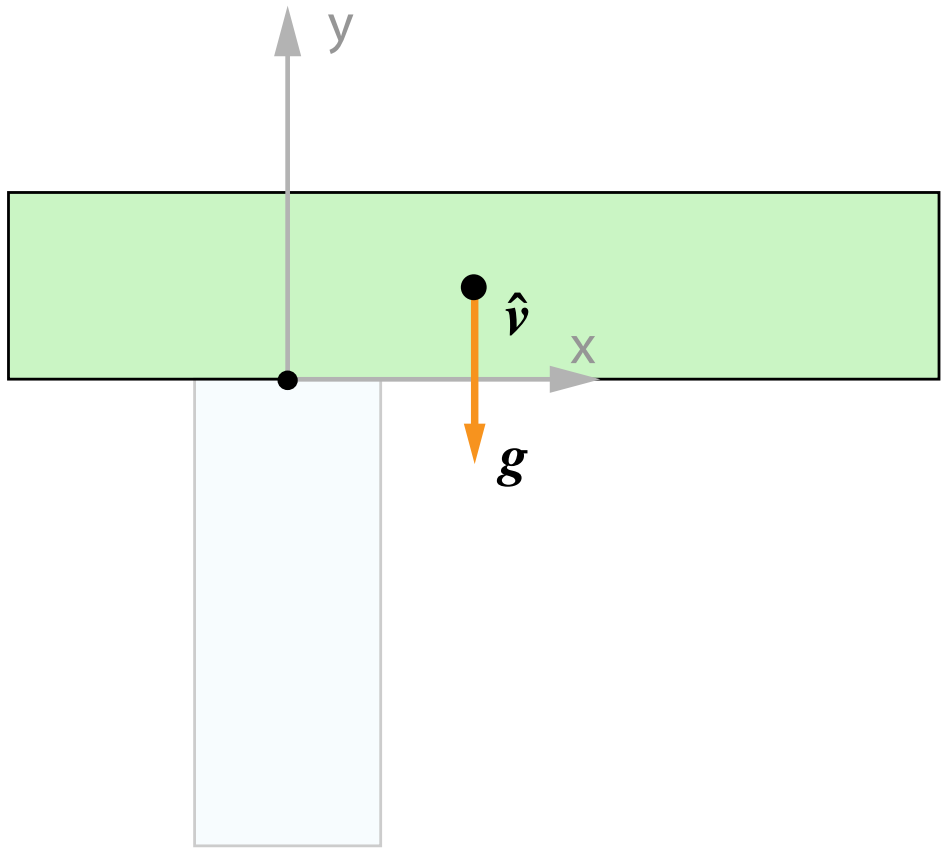
Feasible Motion Space





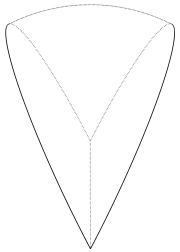
Infinitesimal motion $\hat{v} = (v, \omega)$

Infeasibility Measurement



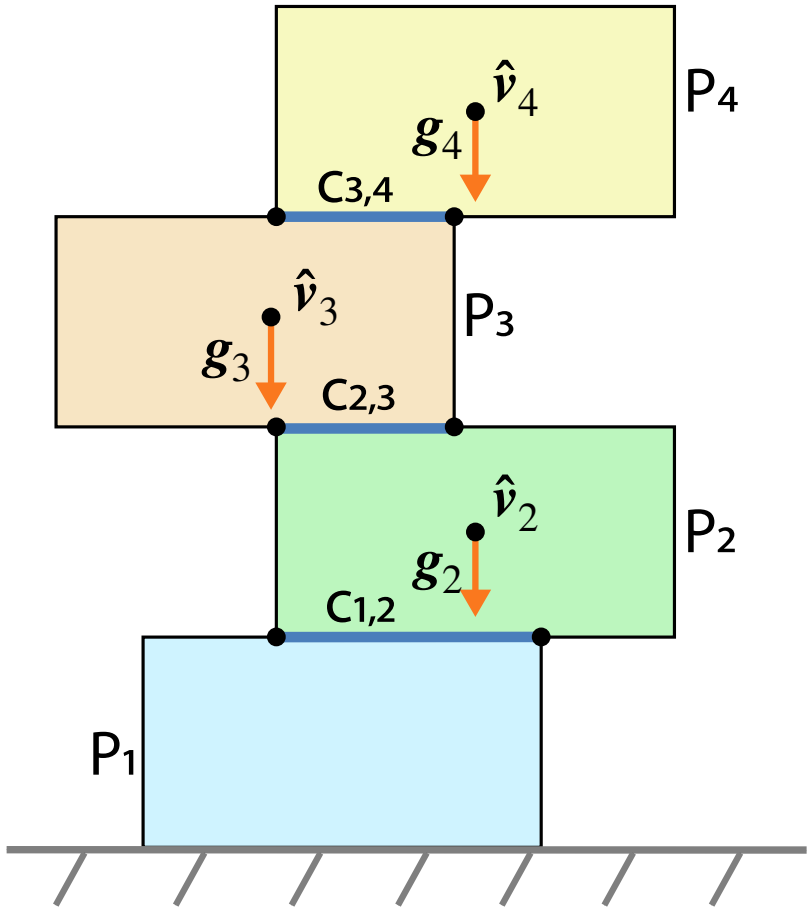
$$\max \quad w \cdot \hat{v} - \frac{1}{2}\hat{v} \cdot \hat{v}$$

$\hat{v} \in$



Motion Cone

Infeasibility Measurement for Assembly



$$\hat{v} = \begin{bmatrix} \hat{v}_2 \\ \hat{v}_3 \\ \hat{v}_4 \end{bmatrix}$$

$$\hat{g} = \begin{bmatrix} \hat{g}_2 \\ \hat{g}_3 \\ \hat{g}_4 \end{bmatrix}$$

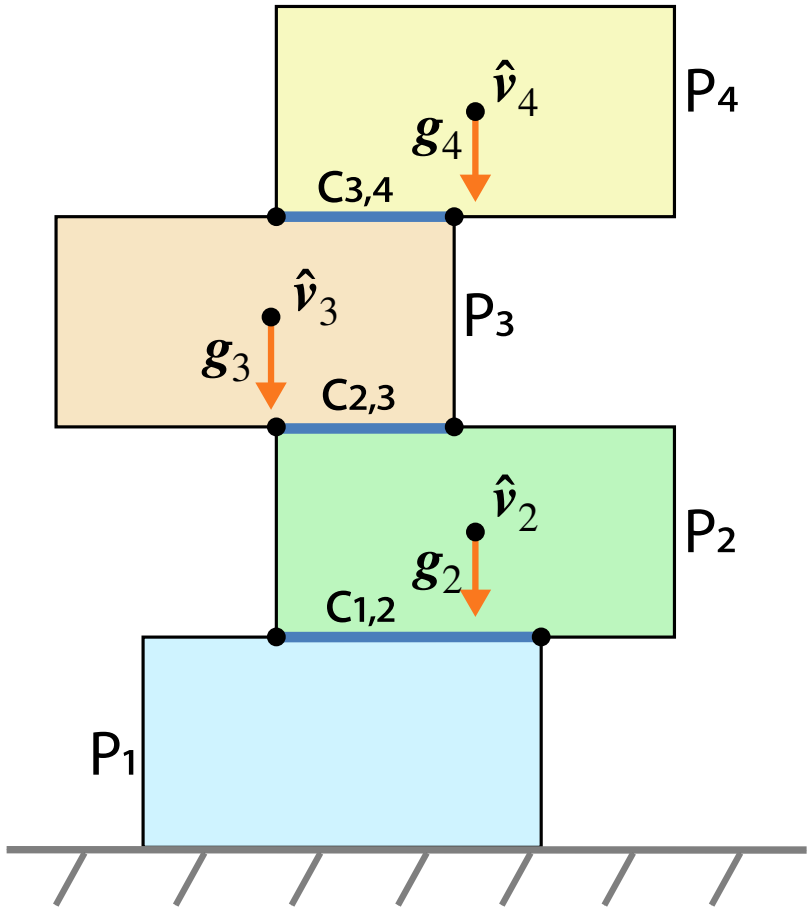
$$\max \boldsymbol{w} \cdot \hat{\boldsymbol{v}} - \frac{1}{2}\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}$$

$$\hat{v}_2 \in V(C_{1,2})$$

$$\hat{v}_3 - \hat{v}_2 \in V(C_{2,3})$$

$$\hat{v}_4 - \hat{v}_3 \in V(C_{3,4})$$

Infeasibility Measurement for Assembly



$$\hat{v} = \begin{bmatrix} \hat{v}_2 \\ \hat{v}_3 \\ \hat{v}_4 \end{bmatrix}$$

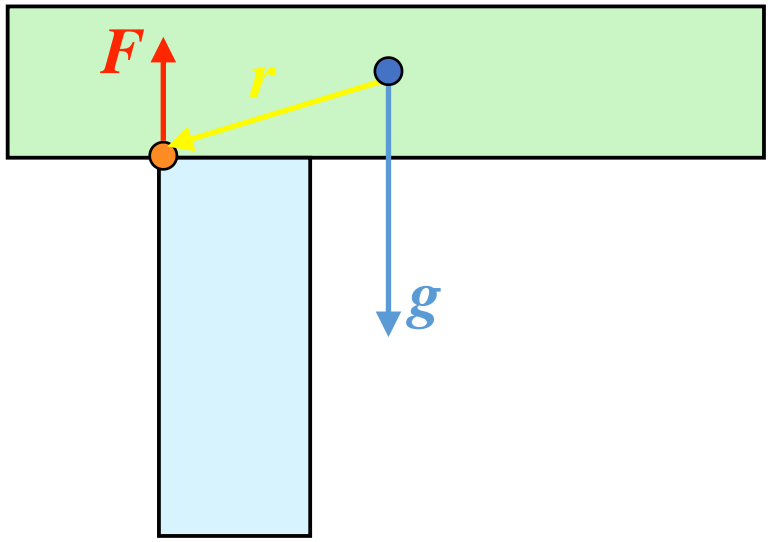
$$\hat{g} = \begin{bmatrix} \hat{g}_2 \\ \hat{g}_3 \\ \hat{g}_4 \end{bmatrix}$$

$$\max \boldsymbol{w} \cdot \hat{\boldsymbol{v}} - \frac{1}{2}\hat{\boldsymbol{v}} \cdot \hat{\boldsymbol{v}}$$

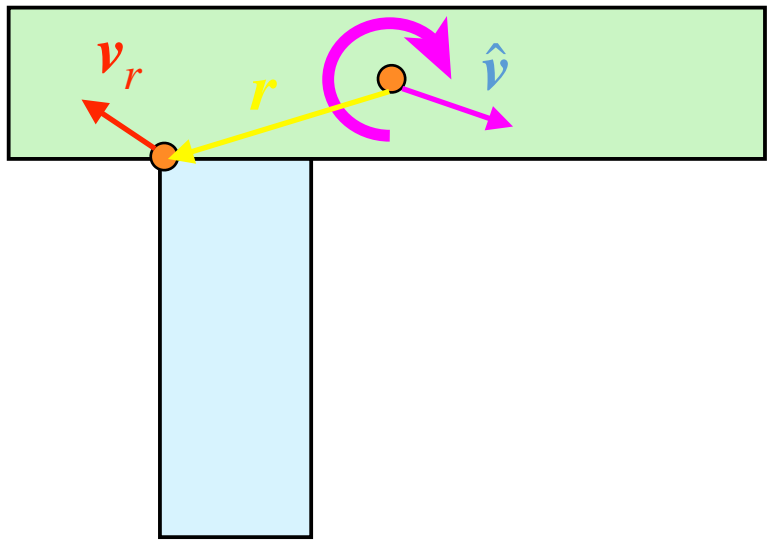
$$\text{s.t.} \qquad B_{\text{in}}\hat{\boldsymbol{v}} \geq 0$$

Static-Kinematic Duality

- The correctness of the kinematic-based method is due to the static-kinematic duality.



Statics

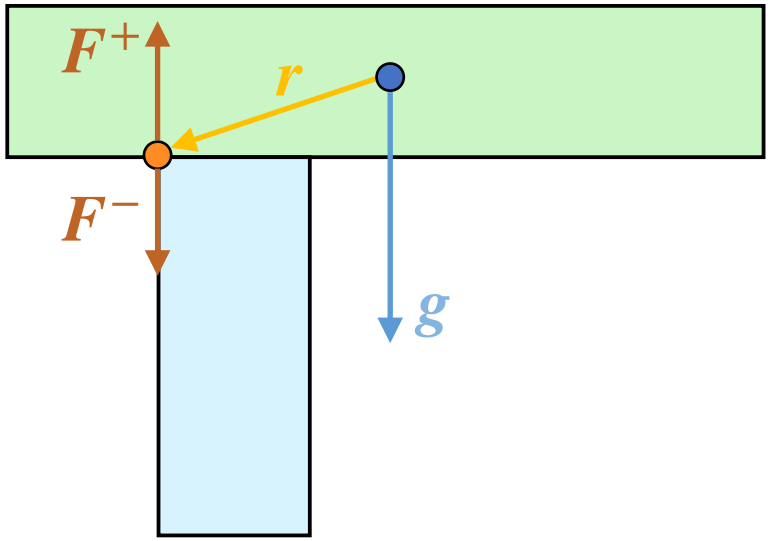


Kinematics

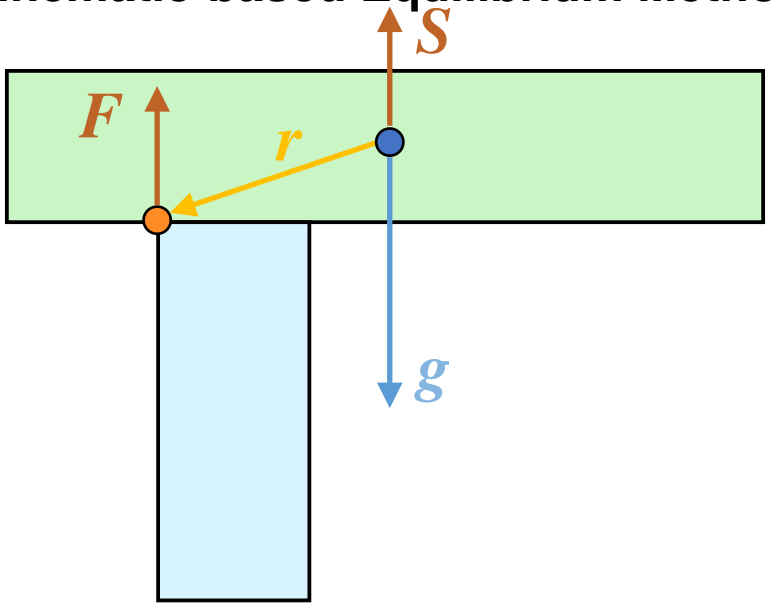
Static-Kinematic Duality

- The kinematic-based method can be reformulated using forces.

Force-based Equilibrium Method



Kinematic-based Equilibrium Method

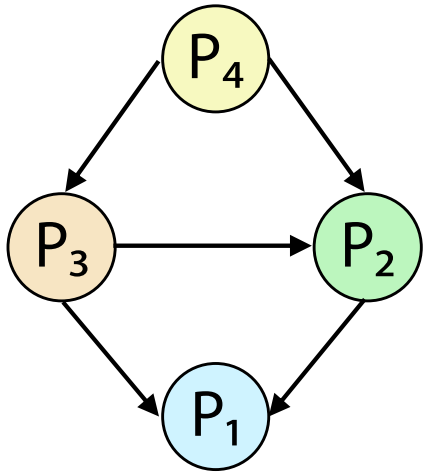


Reformulate: Non-negative Condition

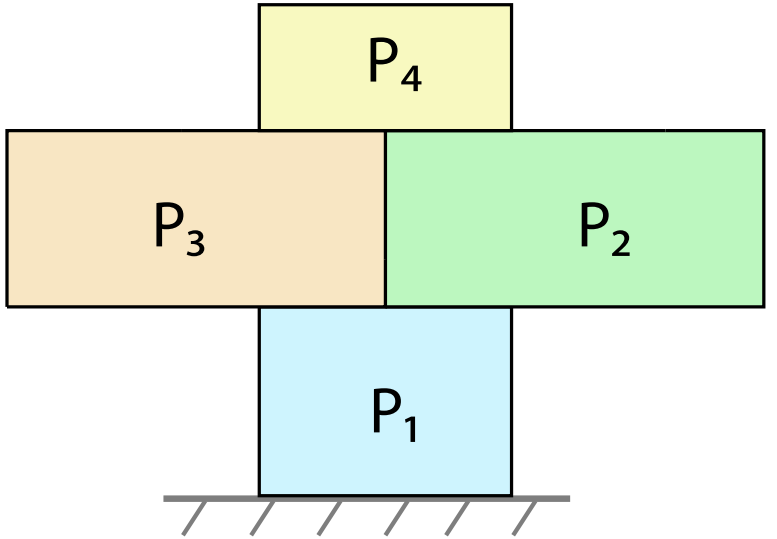
Force/Torque Balance Condition

Representation for Stability Analysis

- Both representations have their own drawbacks.



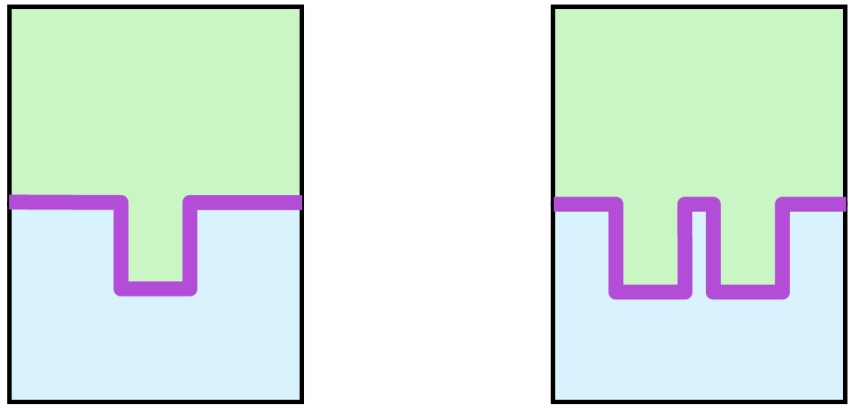
Part Graph



Part Geometry

Geometric-based Representation

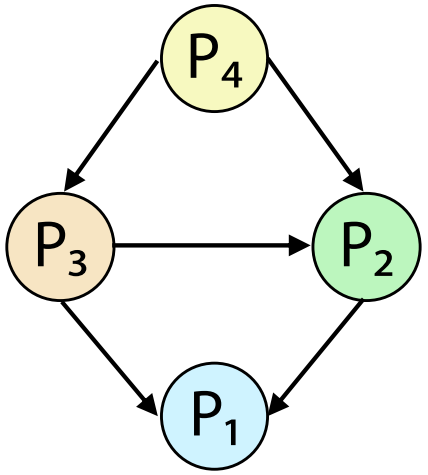
- Geometric-based Representation may have redundancy.



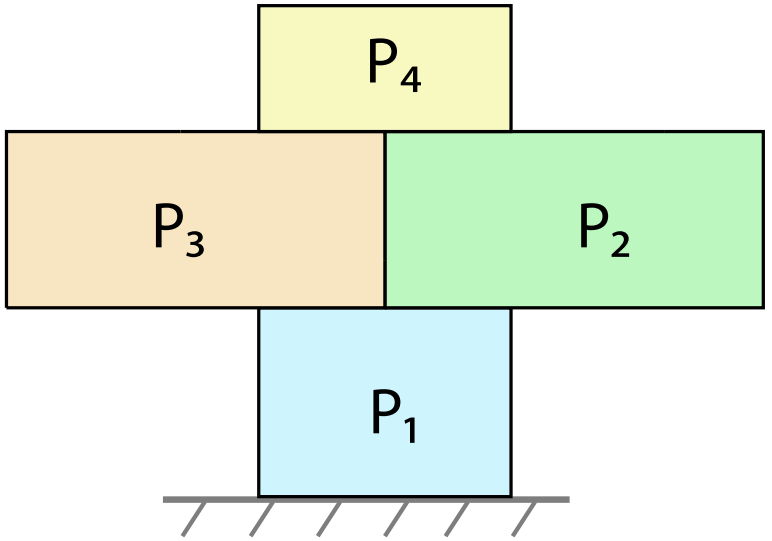
The two assemblies have the same structural stability.

Graph-based Representation

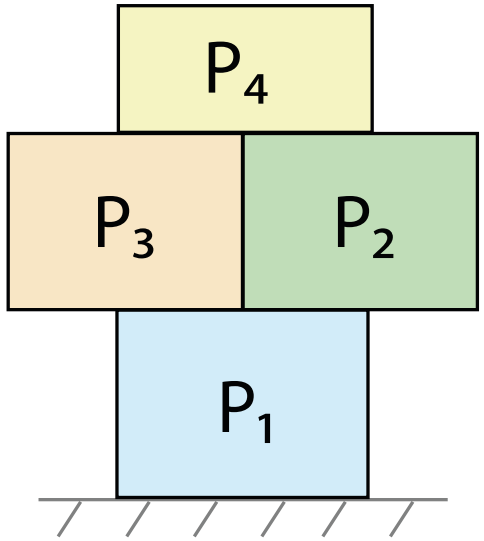
- Graph-based representation is not adequate for structural stability analysis.



Part Graph



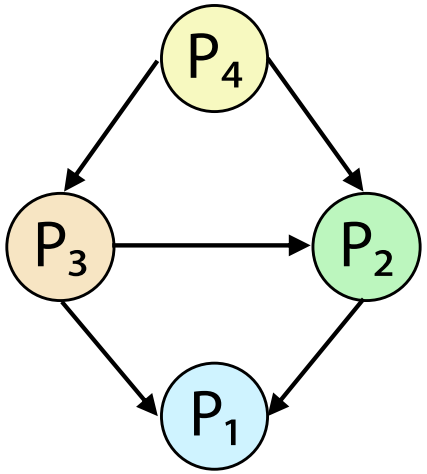
Unstable Assembly



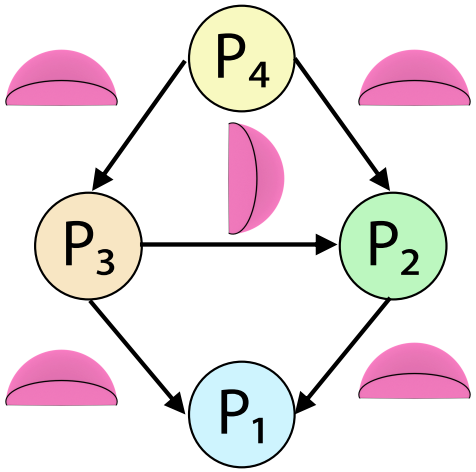
Stable Assembly

Motion-based Representation

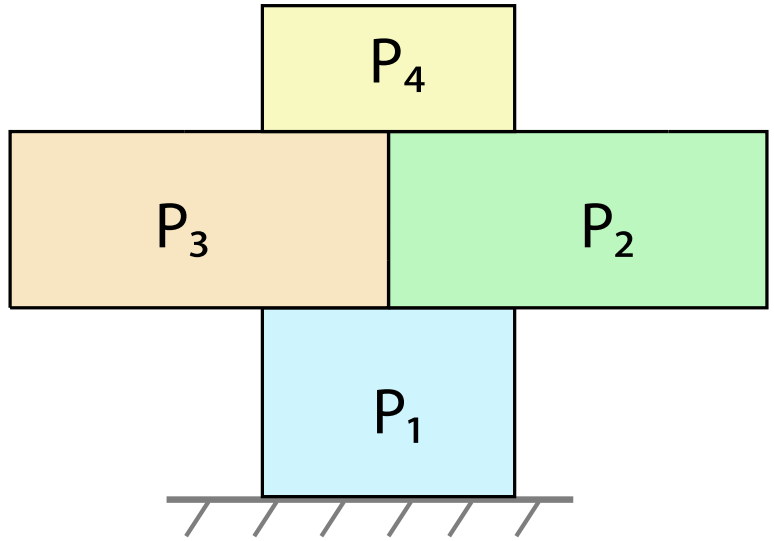
- We propose a motion-based representation which is a condensed representation for measuring structural stability of assemblies.



Part Graph



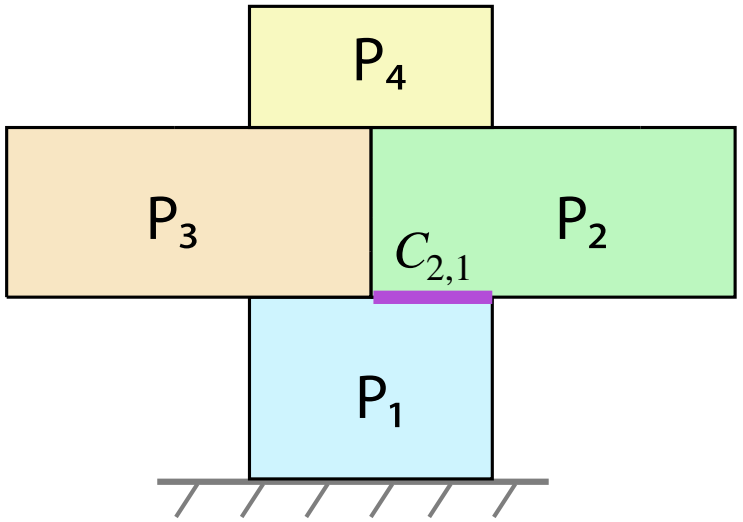
Graph-based Representation

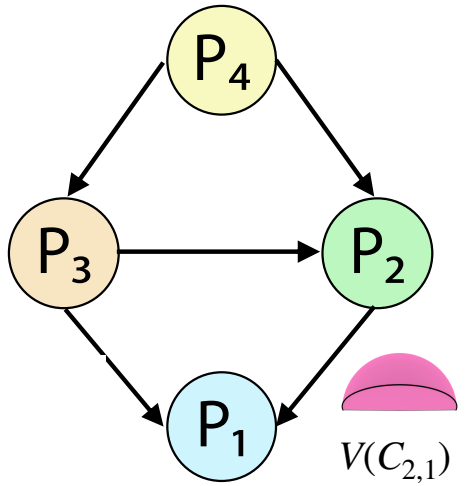


Part Geometry

Motion-based Representation

- Our motion-based representation is an augmented part graph with motion cones at its edges.

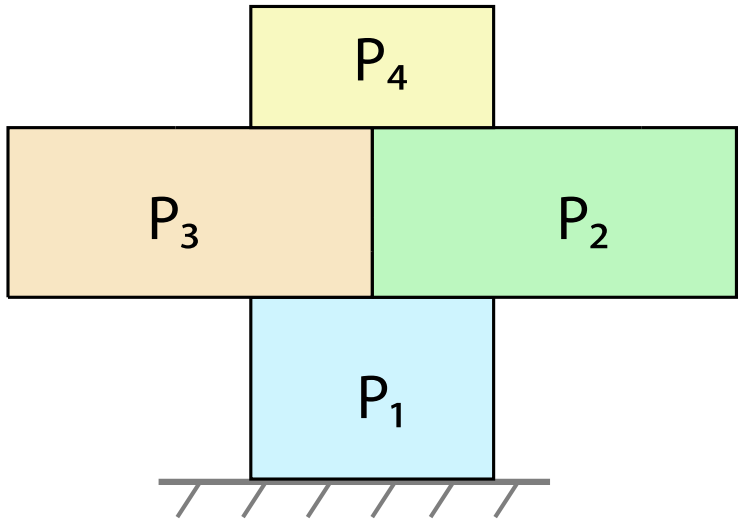


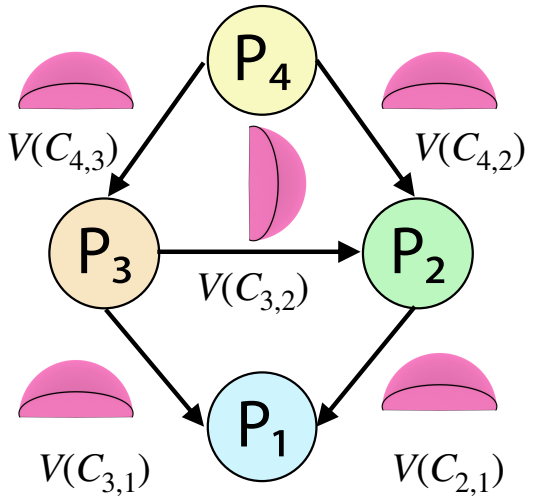


*Arrow means P_2 is installed after P_1

Motion-based Representation

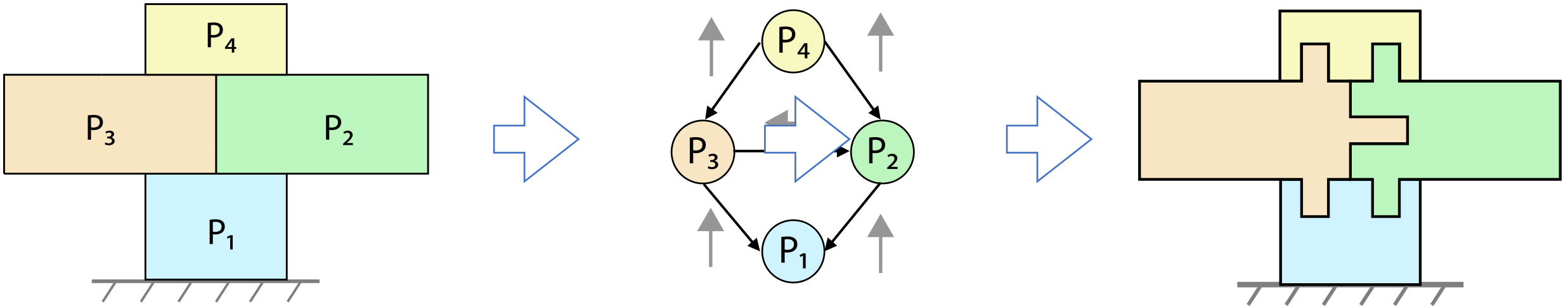
- Because of the duality between statics and kinematics, our motion-based representation can test for equilibrium.





Kinematic-Geometric Design Framework

- Decoupling motion and geometry.



Unstable Input Motion-based Representation Stable Output

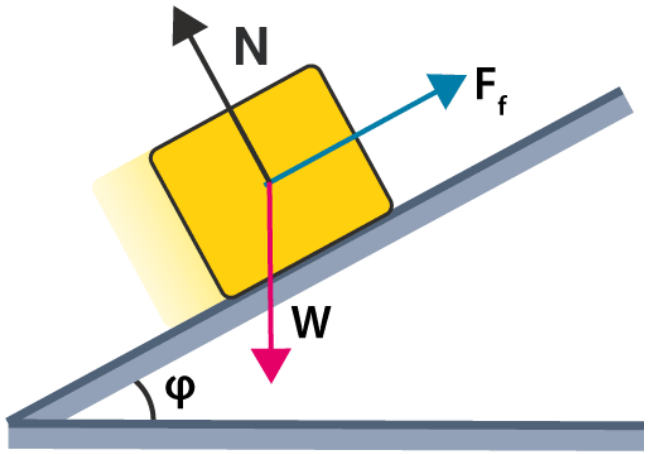
Result

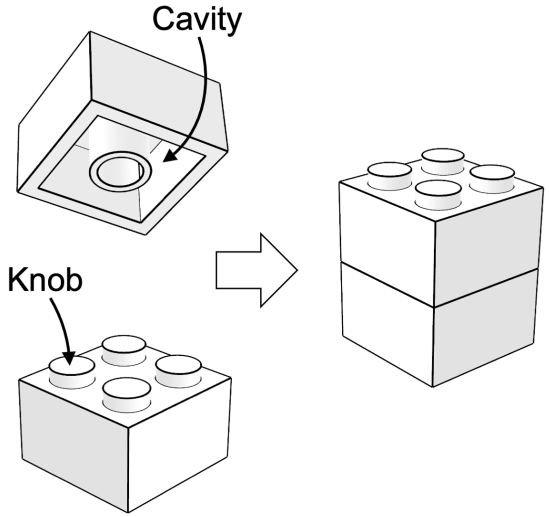
[Wang et al. 2021]

Friction

Friction

- Friction prevents the relative movement of adjacent parts.
- Many assemblies that use snap joints need friction to stay stable.

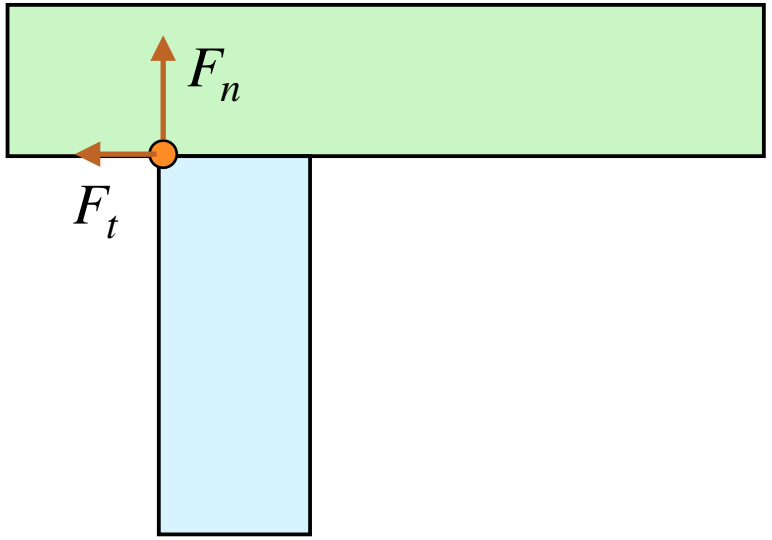




Snap Joint

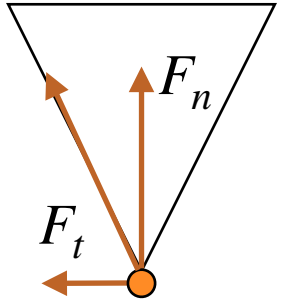
Coulomb Friction

- The resultant force must be within the friction cone.



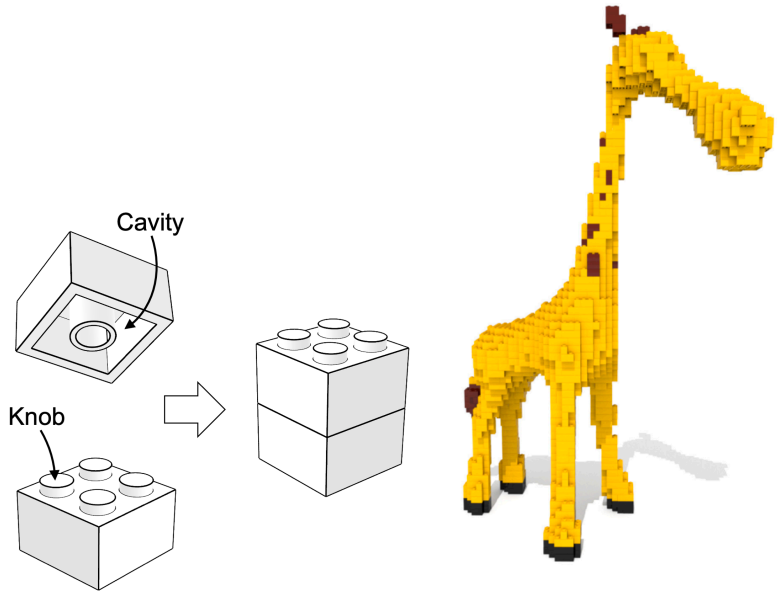
Friction Cone

$$F_t \leq \mu F_n$$

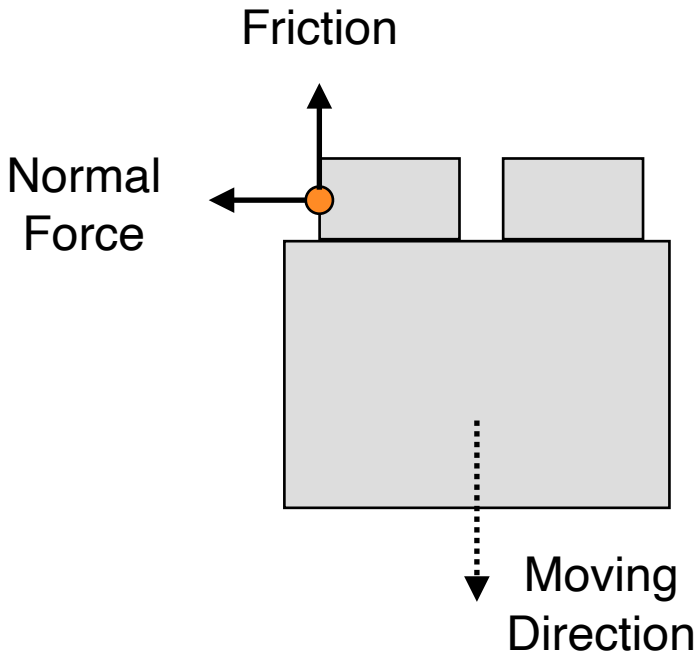


Friction for LEGOs

- For Legos, the normal forces are constant.
- The friction forces must be within a precomputed range.

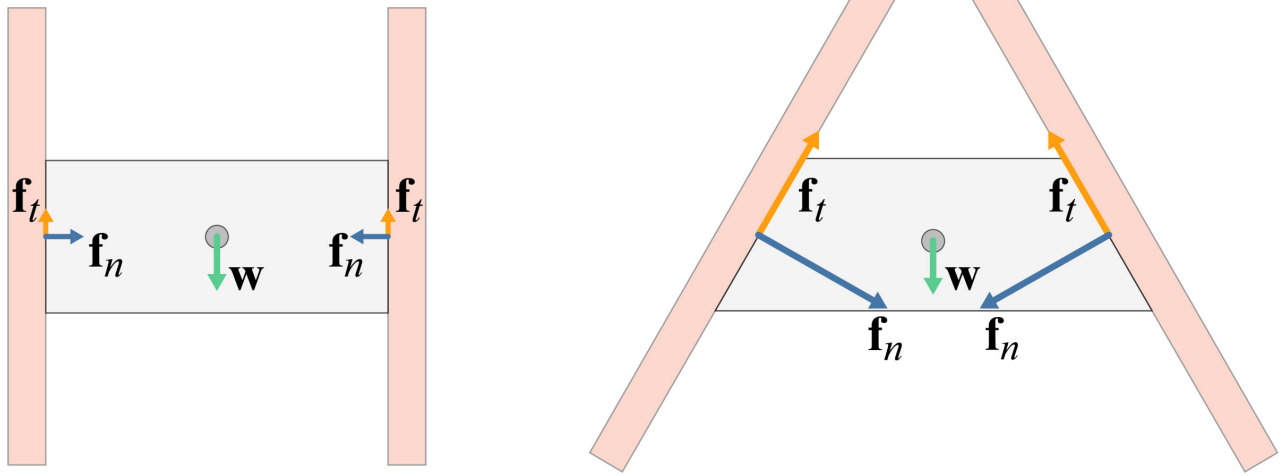


[Luo et al. 2015]



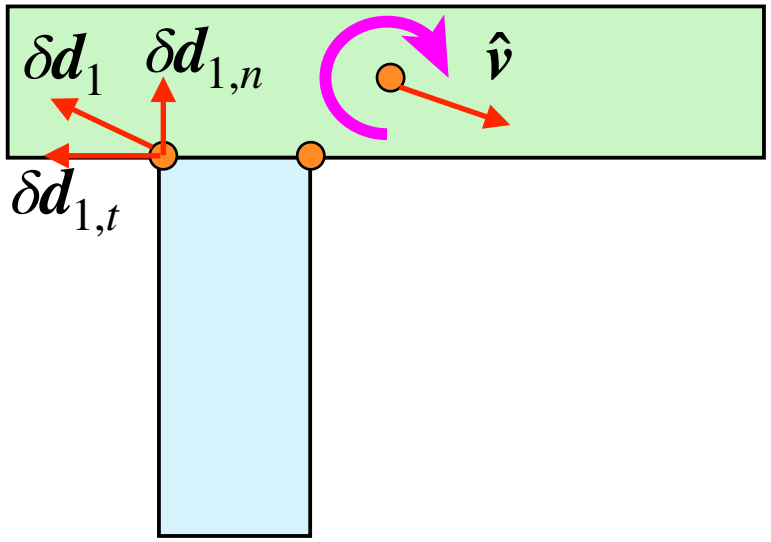
Limitations of Coulomb Friction

- The Coulomb friction may produce unrealistic force configurations.
- The most well-known failure case is the sliding issue.



Additional Physical Principles

- Adding more constraints to regulate the friction helps avoid unrealistic cases.



$$\delta \boldsymbol{d} = B_{\text{in}} \hat{\boldsymbol{v}}$$

Complementary Condition:

$$\delta d_{1,n} \cdot f_{1,n} = 0$$

Maximum Dissipation

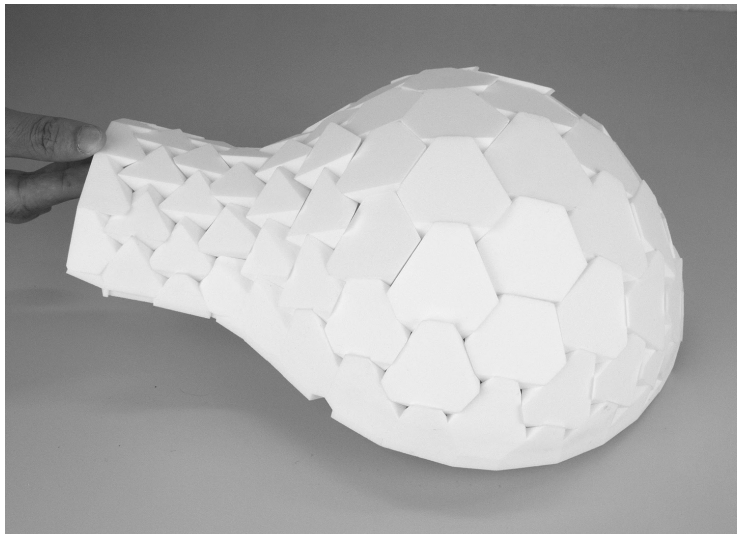
$$f_{1,t} = -\alpha_1 \delta d_{1,t}$$

Part 3: Design for More Types of Stability

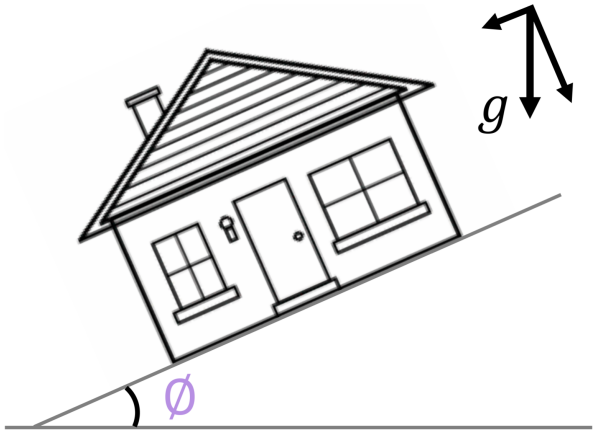
Lateral Stability

Lateral Stability

- Assemblies with later stability are in equilibrium for a cone of gravity direction.

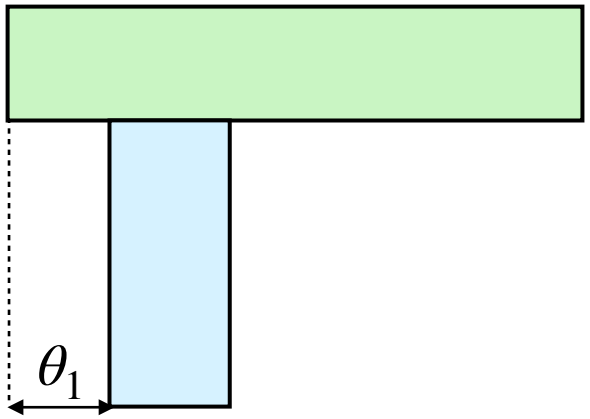


[Wang et al. 2019]

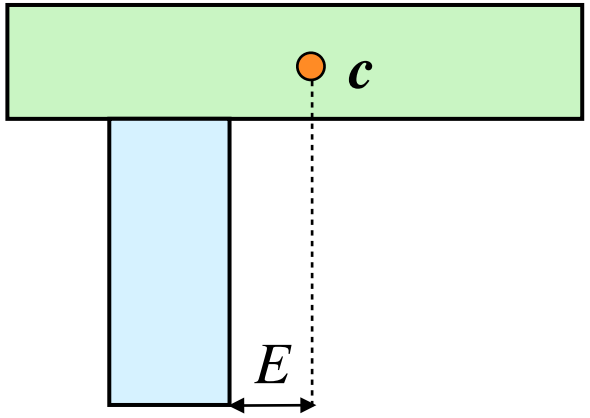


Recap: Gradient-based Stability Optimization

- Come up with new infeasibility energy for lateral stability.



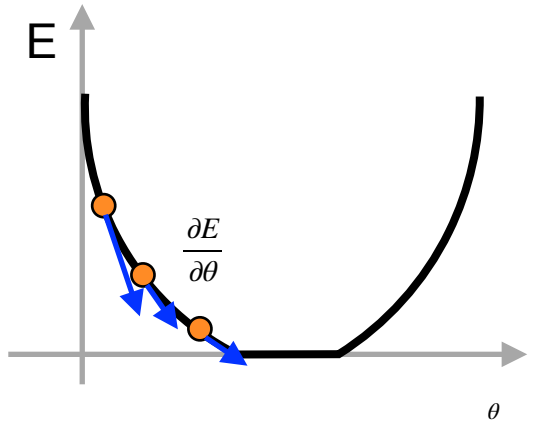
Step 1
Geometrical Property



Step 2
Infeasibility Energy

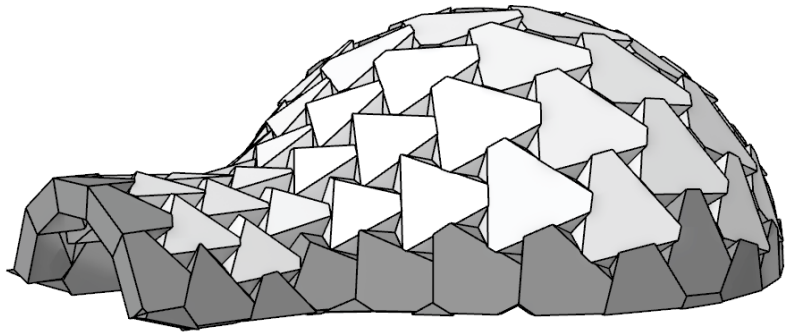
$$\boldsymbol{\theta} \xrightarrow{\partial} \boldsymbol{c} \xrightarrow{\partial} E$$

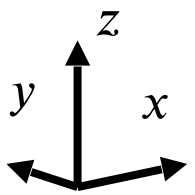
Step 3
Sensitivity Analysis



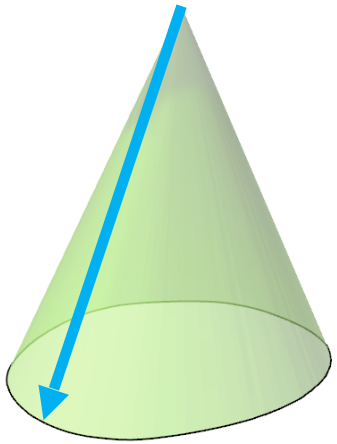
Step 4
Numerical Optimization

Feasible Gravitational Cone



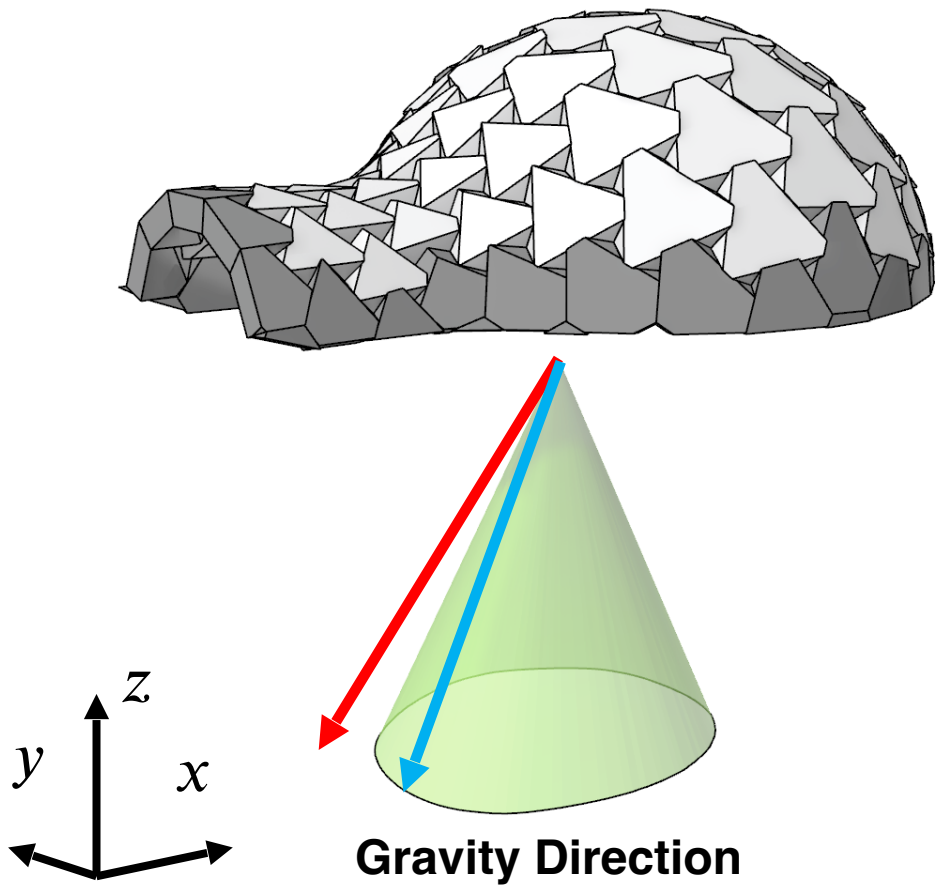


A gravity direction for which the structure is in equilibrium



Convex feasible cone

Lateral Infeasibility Measurement

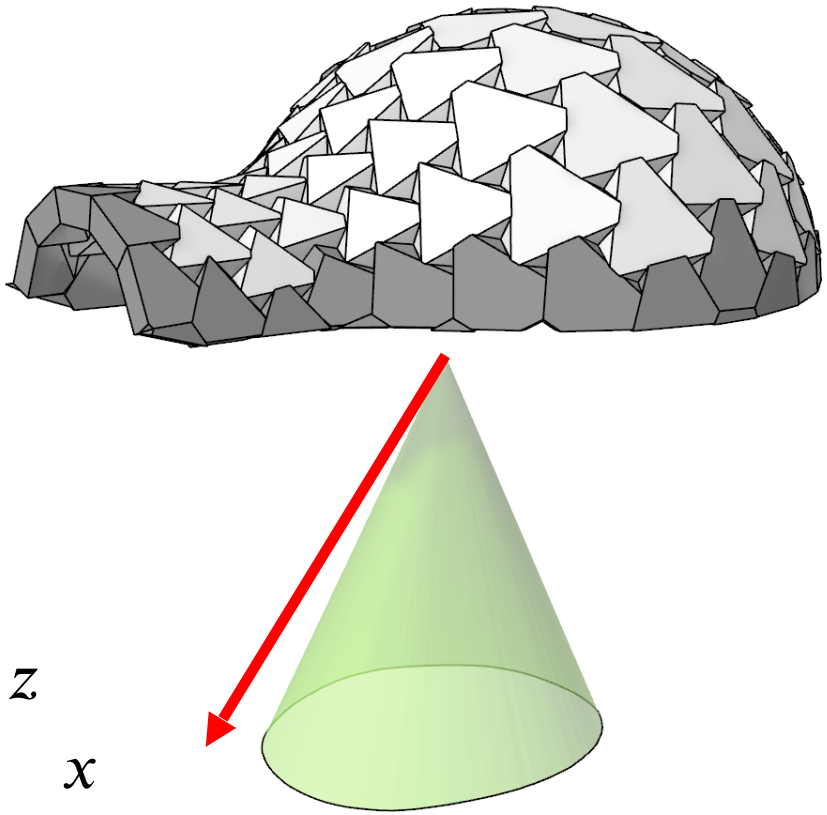


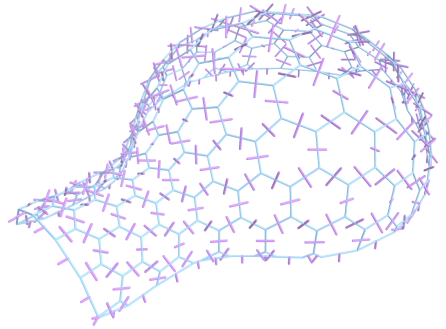
Structural Infeasibility

$E(\ \swarrow\ ) = 0$

$E(\ \downarrow\ ) > 0$

Stability Optimization Algorithm

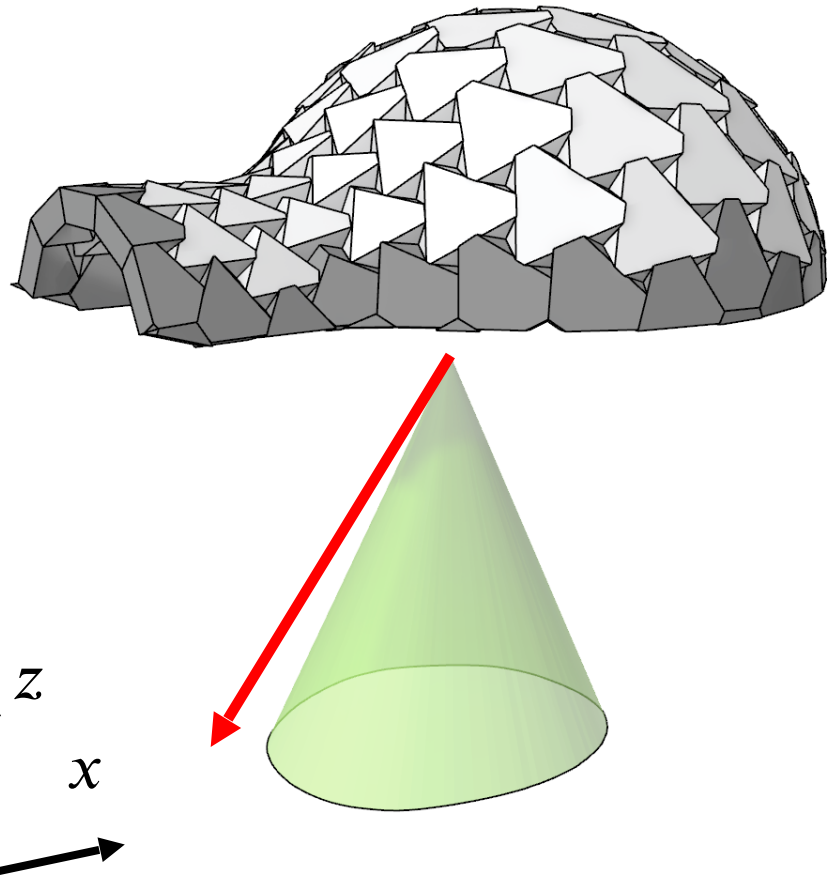




$$\partial E(\downarrow)$$

with respect to the design parameters

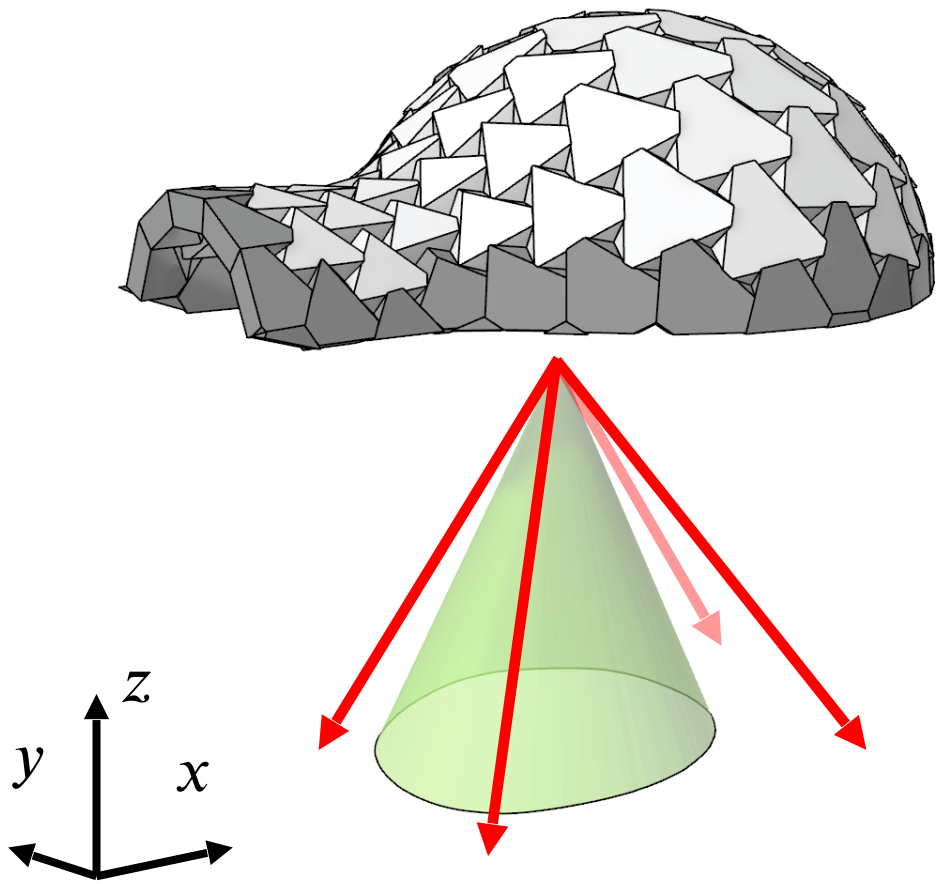
Stability Optimization Algorithm



$$\min E(\ \downarrow\ )$$

Solved by a **BFGS** Solver

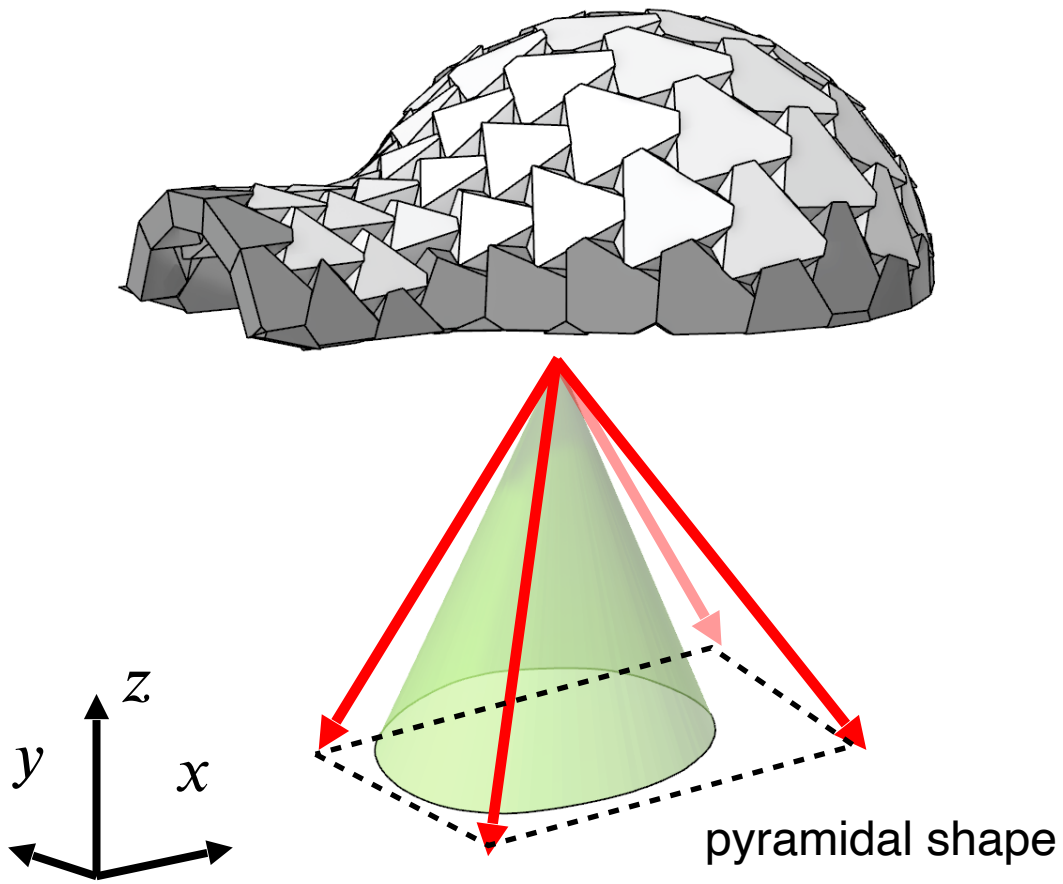
Lateral Infeasibility Measurement



$$\min E(\ \searrow\ ) + E(\ \searrow\ ) + E(\ \searrow\ ) + E(\ \searrow\ )$$

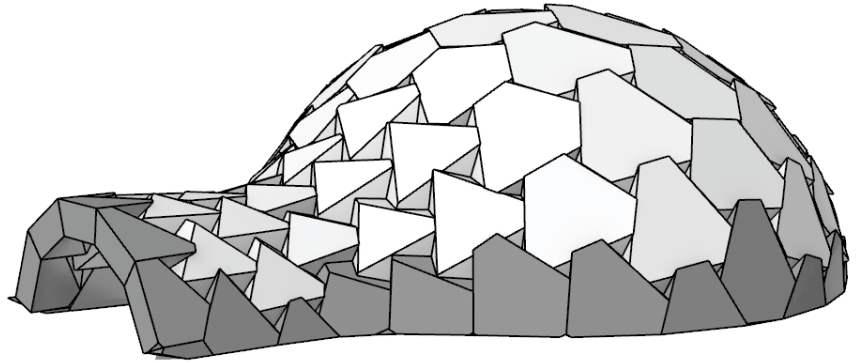
Contact Area $\geq$ User defined value

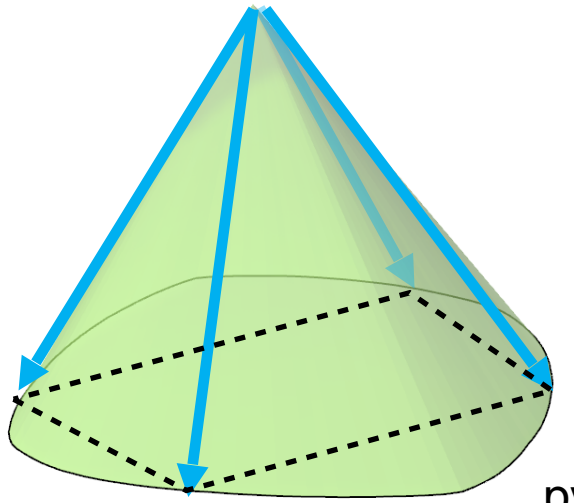
Lateral Infeasibility Measurement

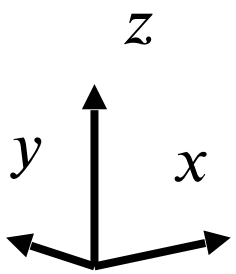


Due to the convexity of the feasible cone

Lateral Infeasibility Measurement







pyramidal shape

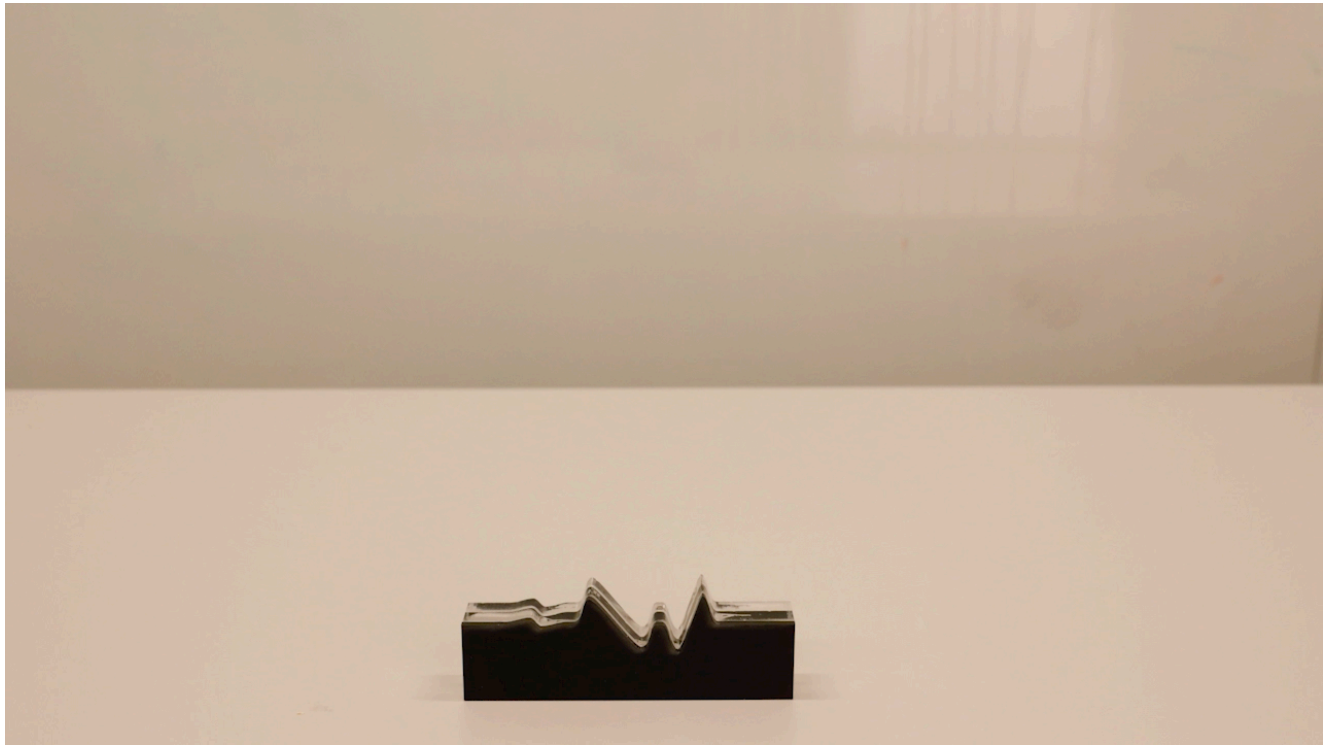
The new feasible cone
will cover the pyramidal shape

Result

Scaffold-free Assembly

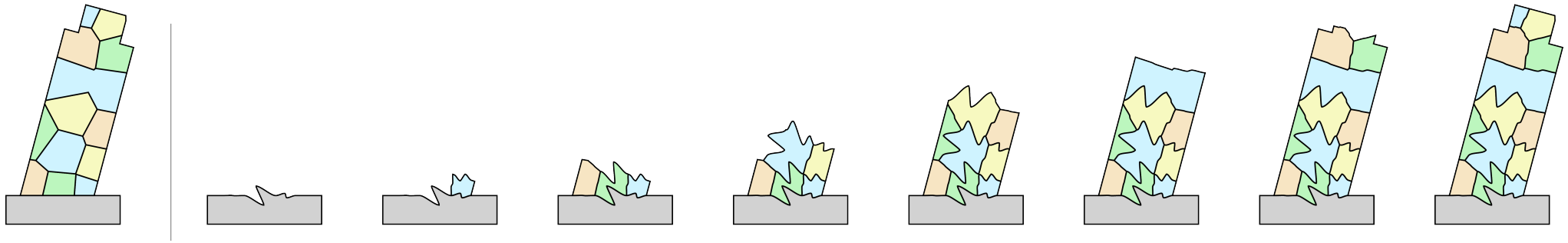
Scaffold-free Assembly

- Making the assembling process stable.



Infeasibility Energy

- The infeasibility energy is the summation of all the infeasibility energy of the structure at each assembling stage.

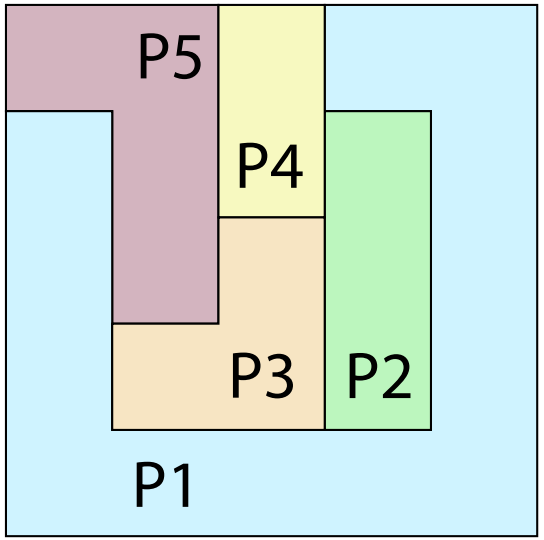


$$E = E_1 + E_2 + E_3 + E_4 + E_5 + E_6 + E_7 + E_8$$

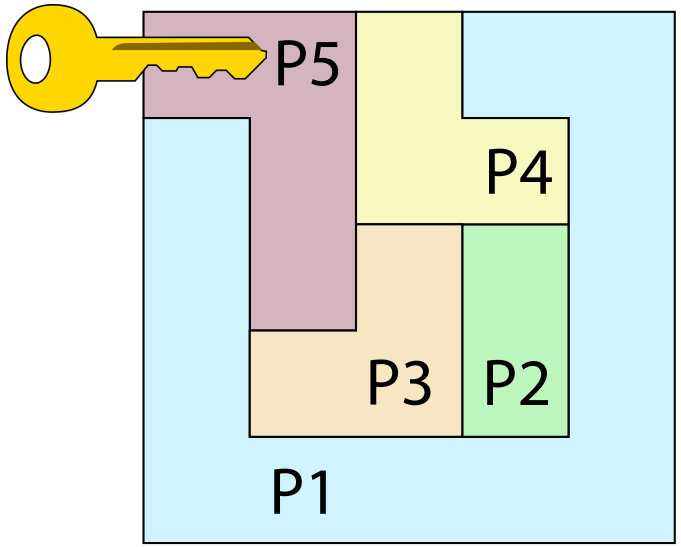
Globally Interlocking Assemblies

Recap: Globally Interlocking

Once the key and a part of the reset are fixed, no parts can be taken out from the assembly.



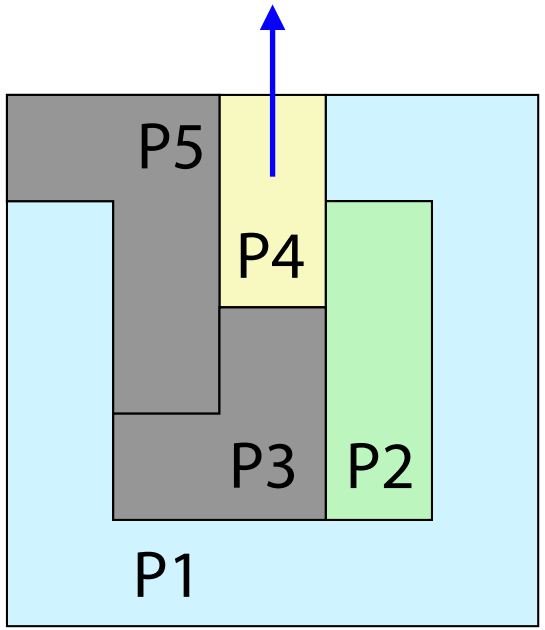
Non Interlocking



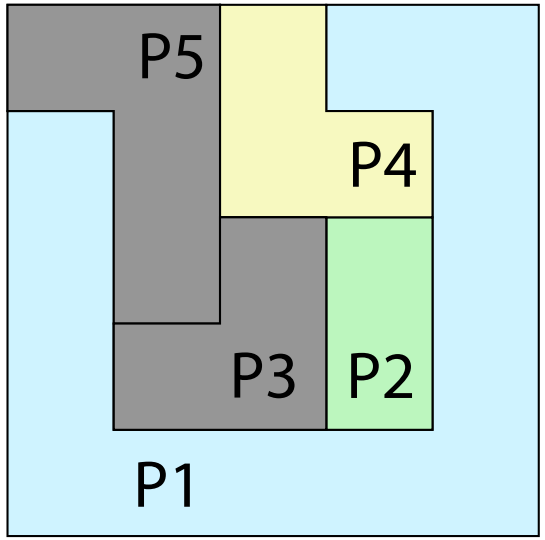
Interlocking

Recap: Globally Interlocking

Once the key and a part of the reset are fixed, no parts can be taken out from the assembly.



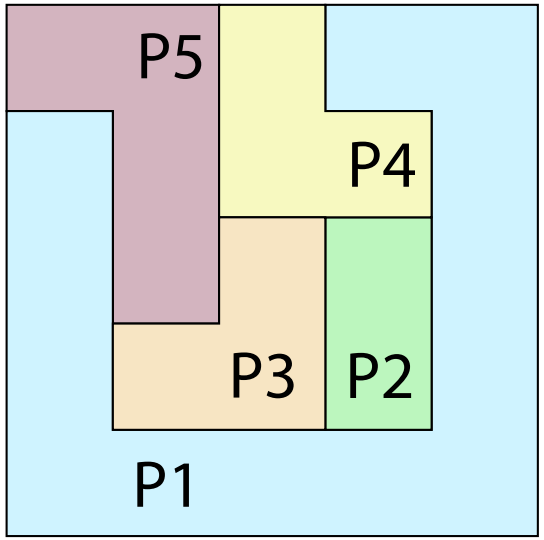
Non Interlocking

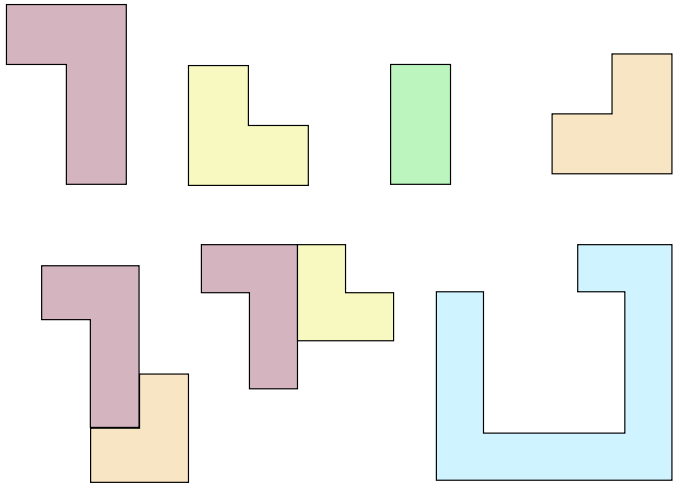


Interlocking

Recap: Classic Interlocking Test

Classic method examines every subset of parts, which has **exponential time complexity.**



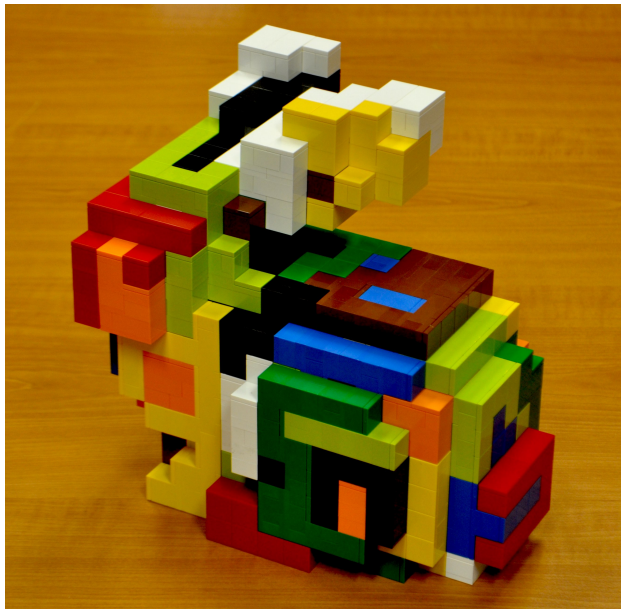


…

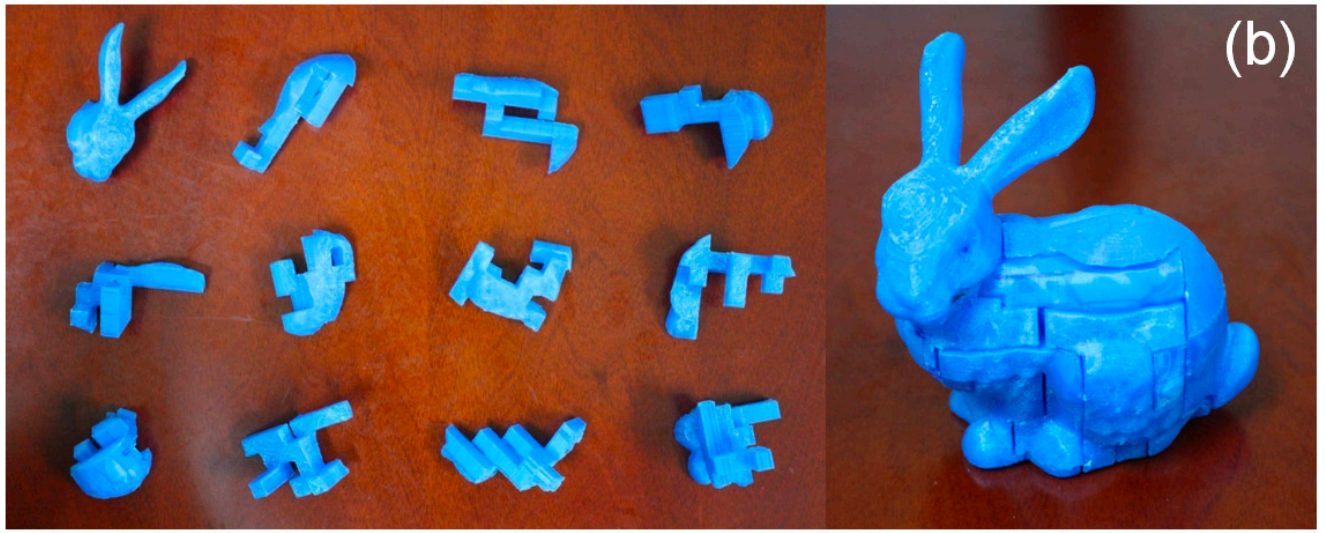
[Song et al. 2012]

Shape Decomposition

- When the input is a target shape, computational design of interlocking assemblies can be formulated as a shape decomposition problem.



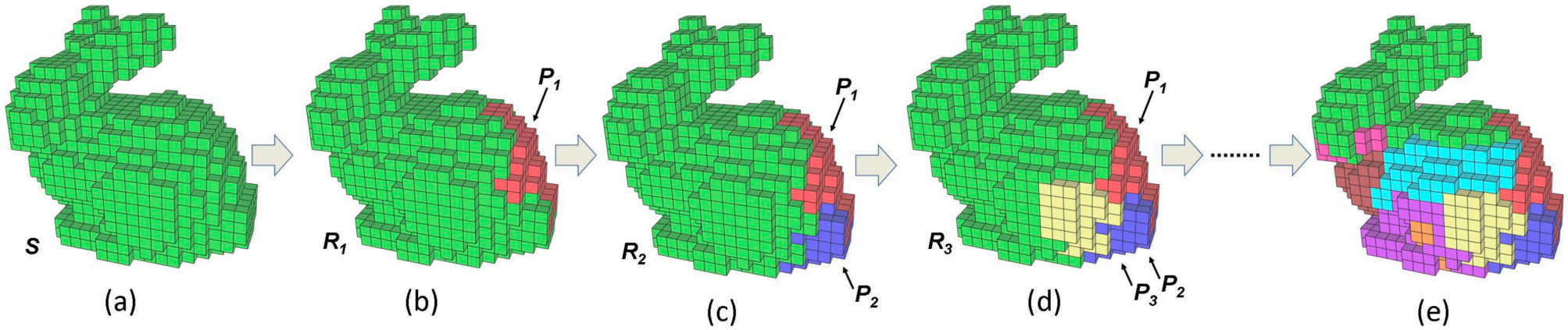
[Song et al. 2012]



[Song et al. 2015]

Shape Decomposition

- When the input is a target shape, computational design of interlocking assemblies can be formulated as a shape decomposition problem.

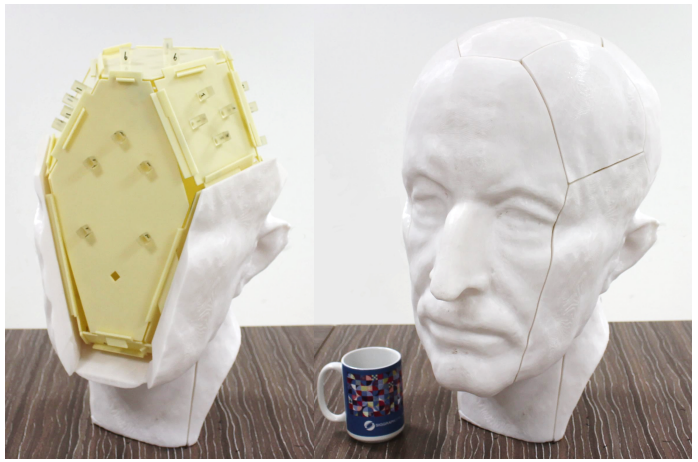


[Song et al. 2012]

Joint Planning

- When the input is a set of initial parts without joints, designing interlocking assemblies can be formulated as a joint planning problem.

[Fu et al. 2015]

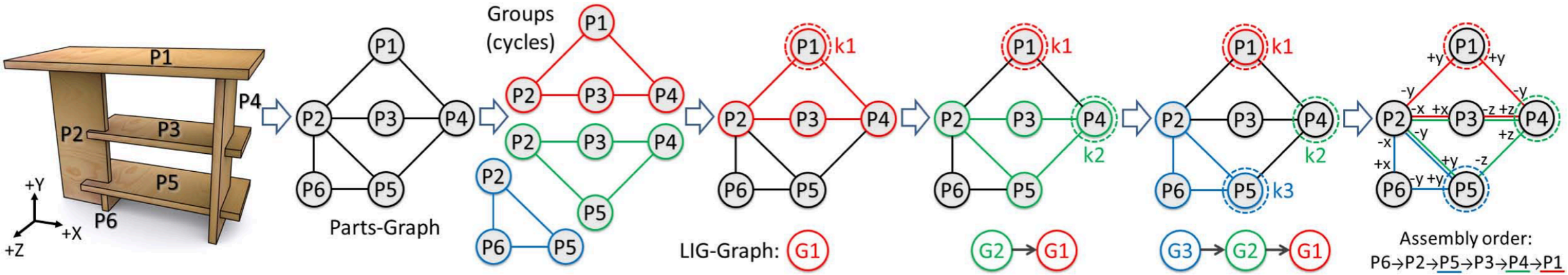


[Song et al. 2016]

[Yao et al. 2017]

Joint Planning

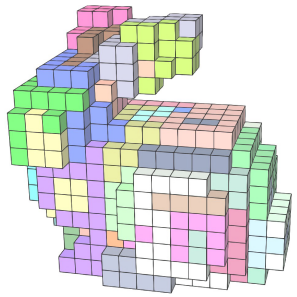
- Fu et al. computed an interlocking joint configuration following the LIG-based approach.



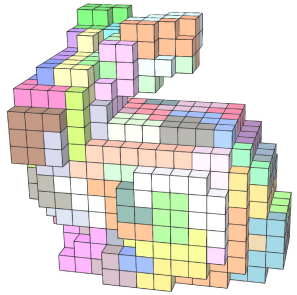
[Fu et al. 2015]

DBG-based Interlocking Design

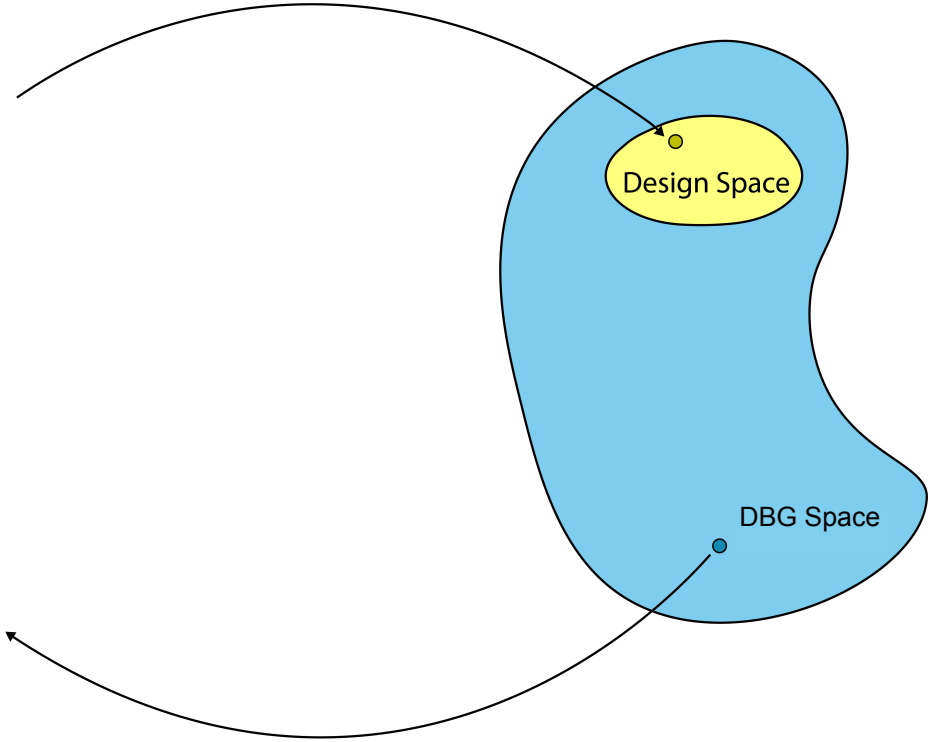
- The DBG approach allow exploring the full search space of interlocking configurations.



40-part Bunny
[Song et al. 2012]



80-part Bunny
[Wang et al. 2018]



[Wang et al. 2018]

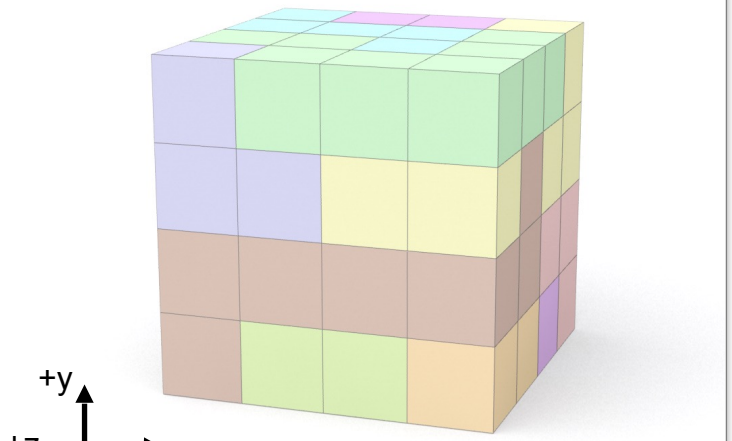
DBG-based Interlocking Design

- Wang et al. use the base DBG to test and design interlocking assemblies.

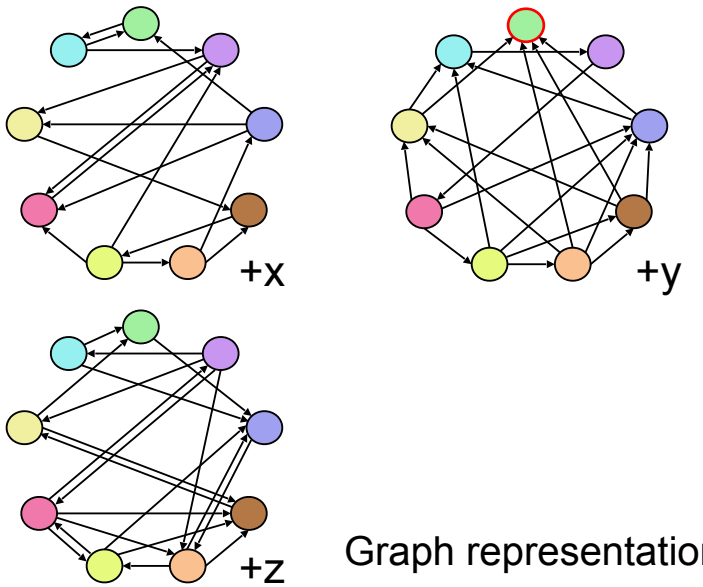
The 3D assembly is interlocking

=

All graphs are strongly connected
(note: ignore the key when check connectivity)



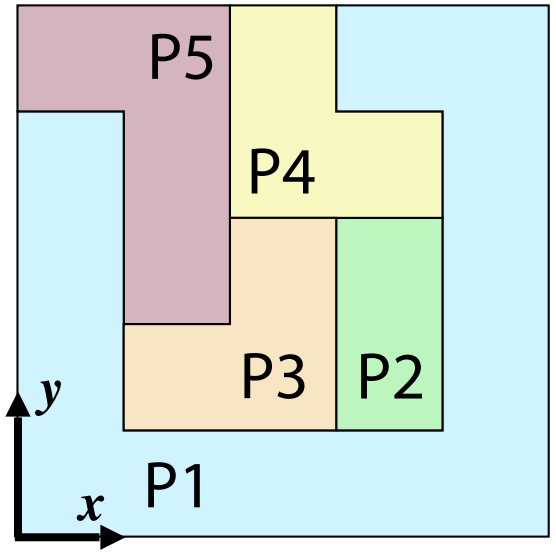
Interlocking geometry



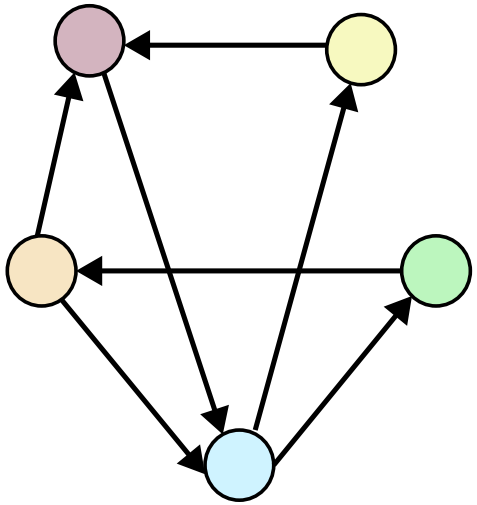
Graph representation

[Wang et al. 2018]

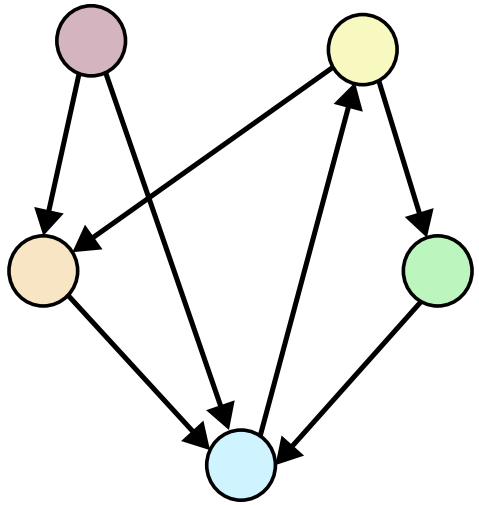
Directional Blocking Graphs



G(+x)



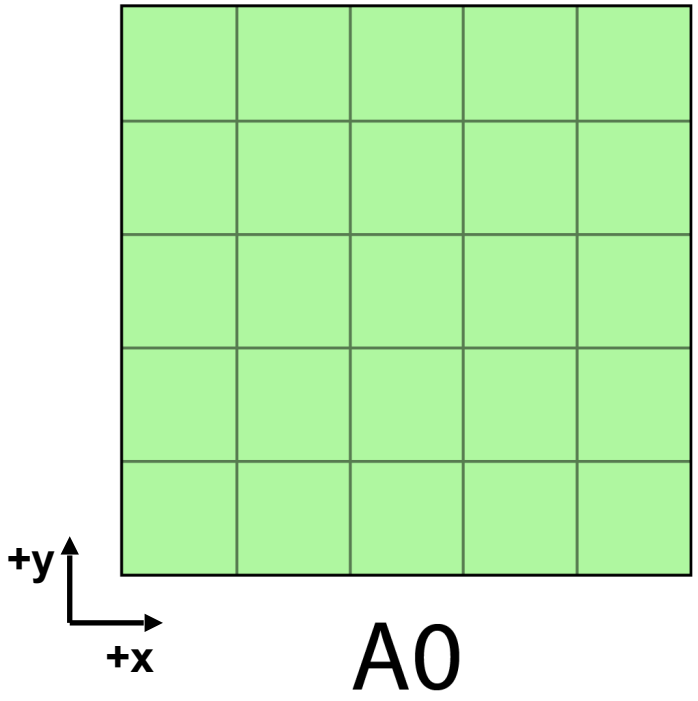
G(+y)



An assembly is global Interlocking when its directional blocking graphs are **strongly connected** expect the key

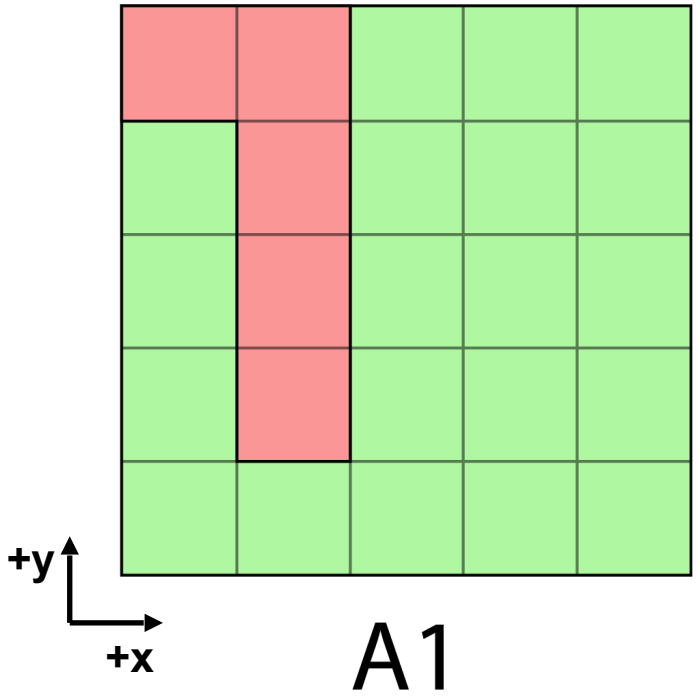
Interactive Design Framework

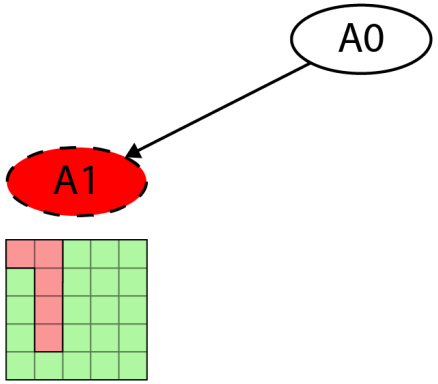
Example: design a 5-part interlocking assembly by partitioning a 5x5 square.



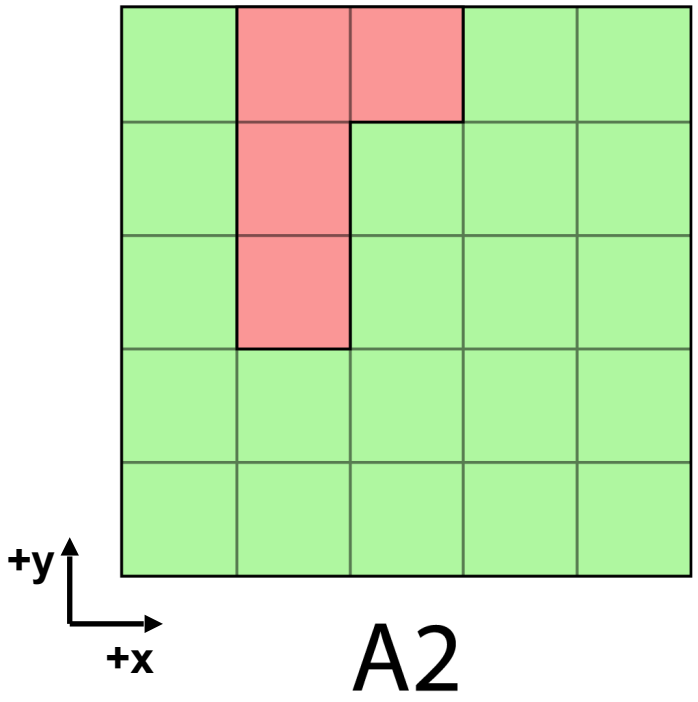
Interactive Design Framework

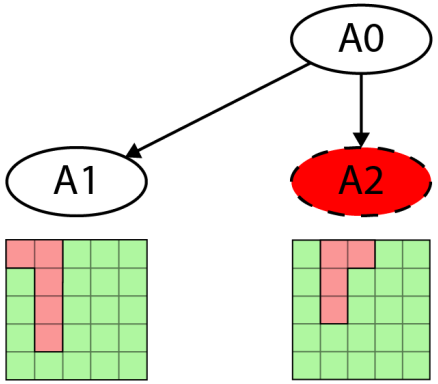
First, construct the key part.





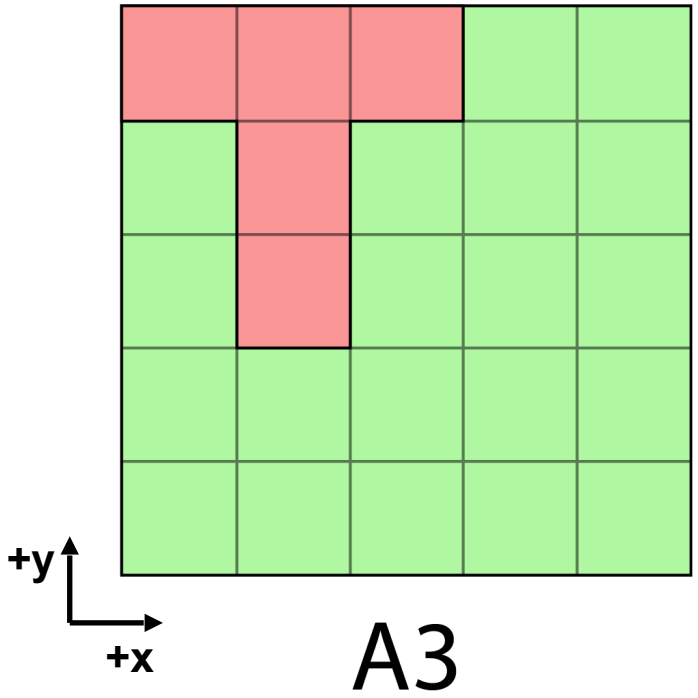
Interactive Design Framework

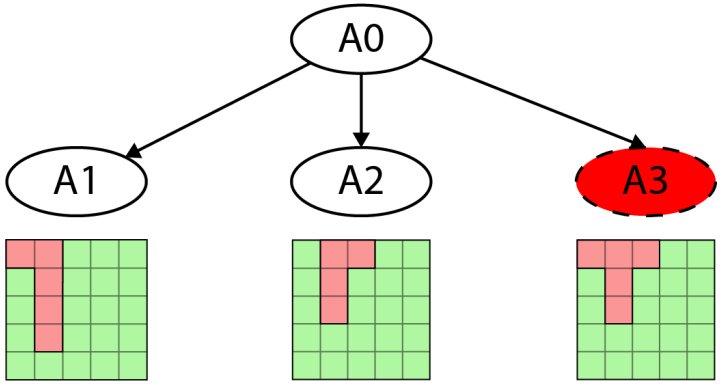




Interactive Design Framework

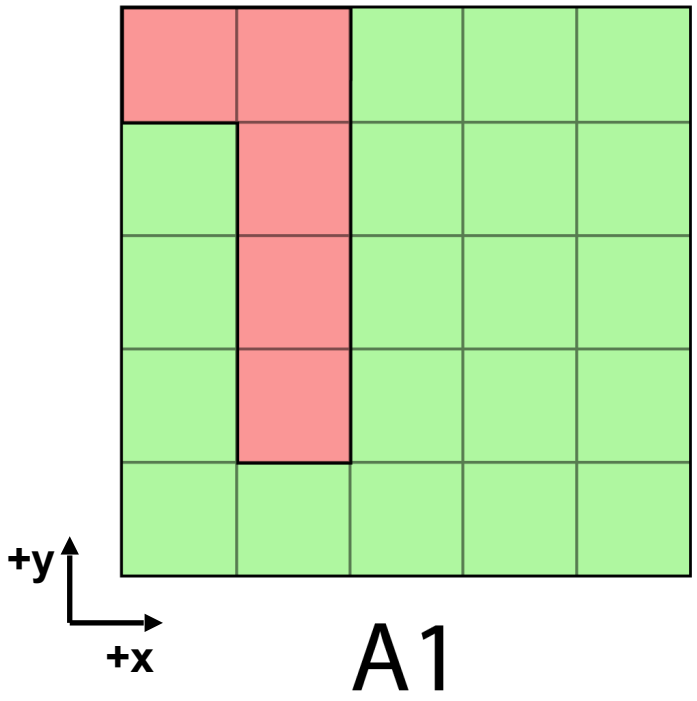
There are many possibilities and we only select a few candidates.

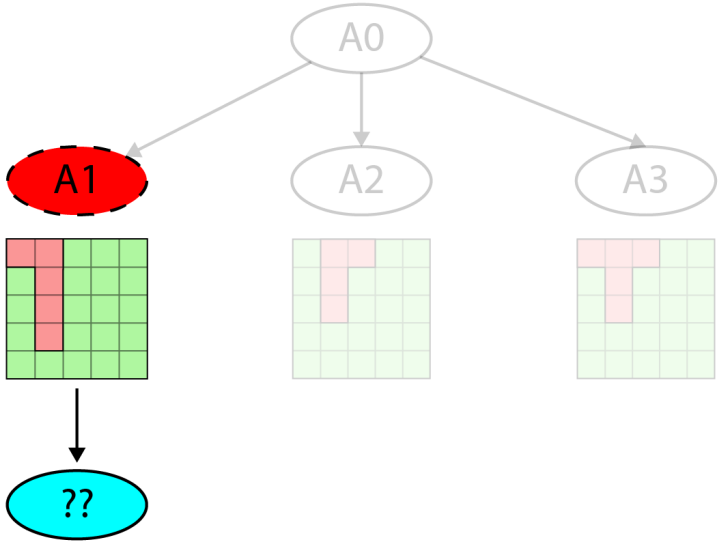




Interactive Design Framework

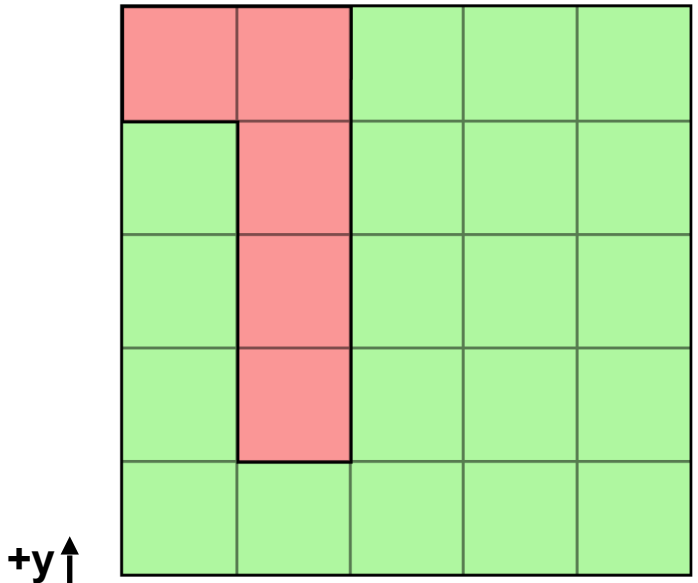
How to design a 3-part interlocking assembly by partitioning A1?



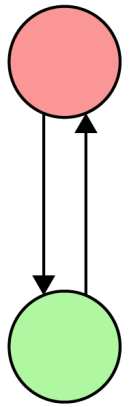


Graph Design

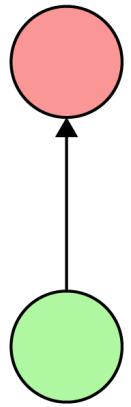
Construct the base directional blocking graphs for A1.



A1



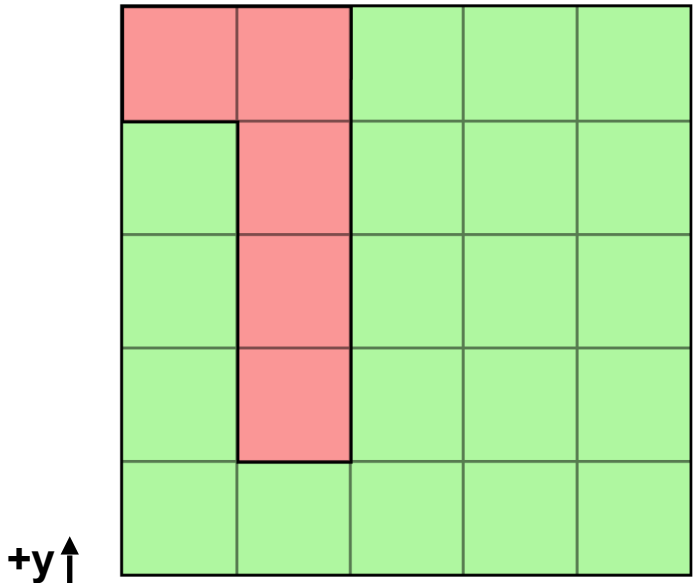
G(+X)



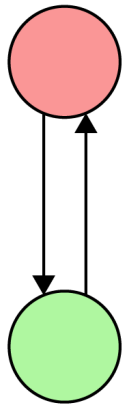
G(+Y)

Graph Design

Construct the base directional blocking graphs for A1.



A1

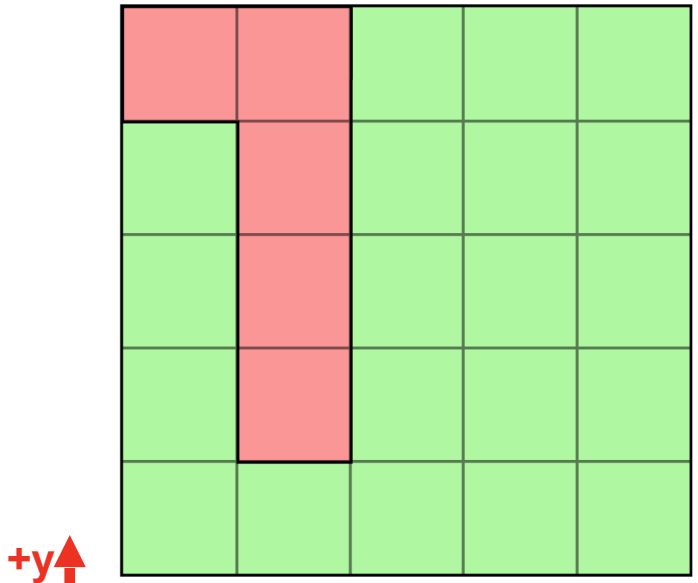


G(+X)

G(+Y)

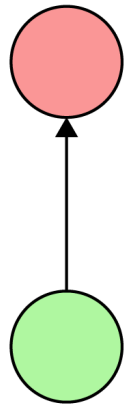
Graph Design

Construct the base directional blocking graphs for A1.



A1

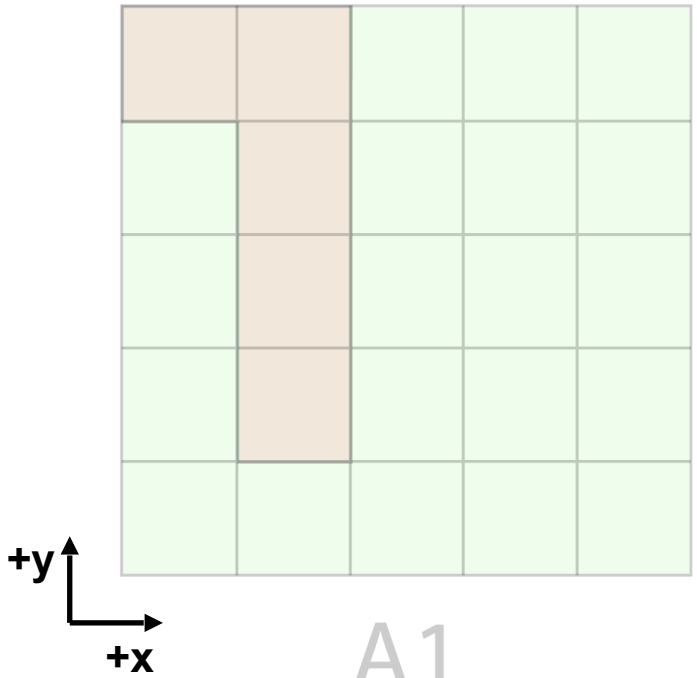
G(+X)

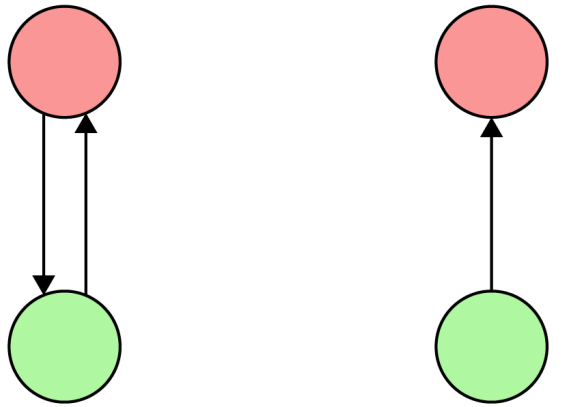


G(+Y)

Graph Design

Ignore geometry of the assembly.



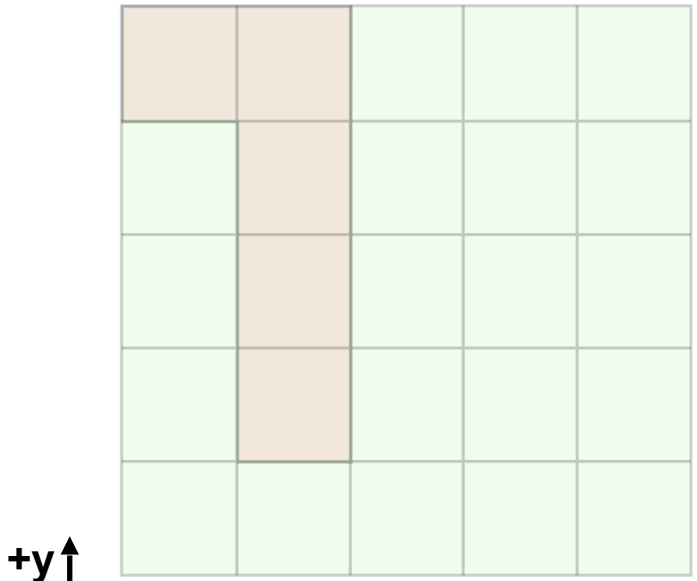


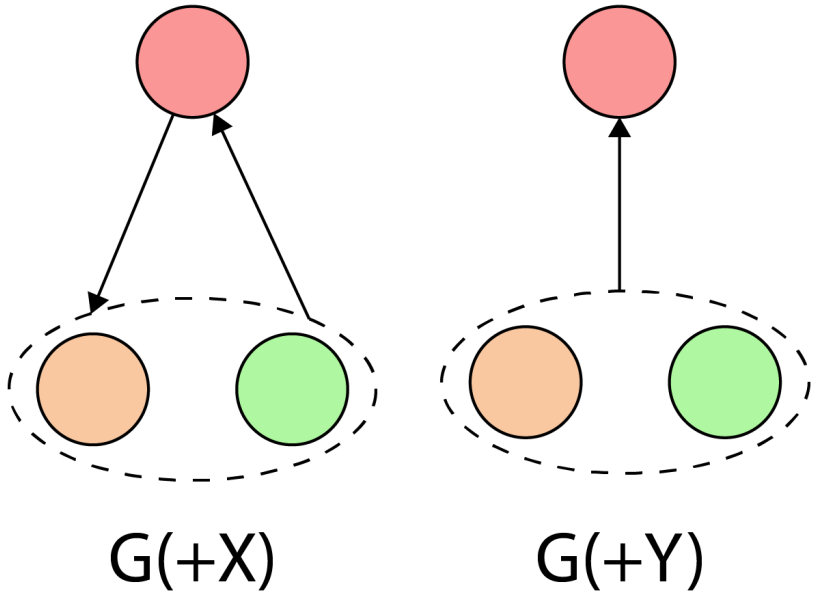
G(+X)

G(+Y)

Graph Design

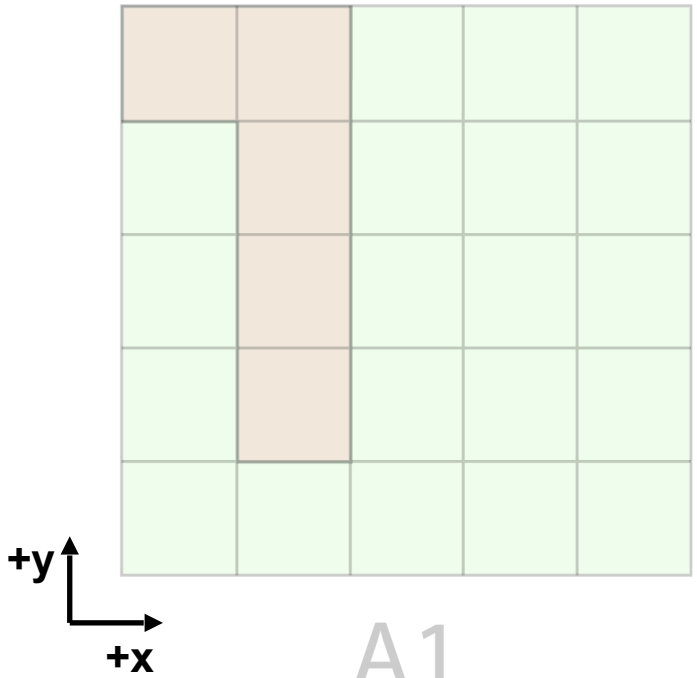
How to make graphs strongly connected?

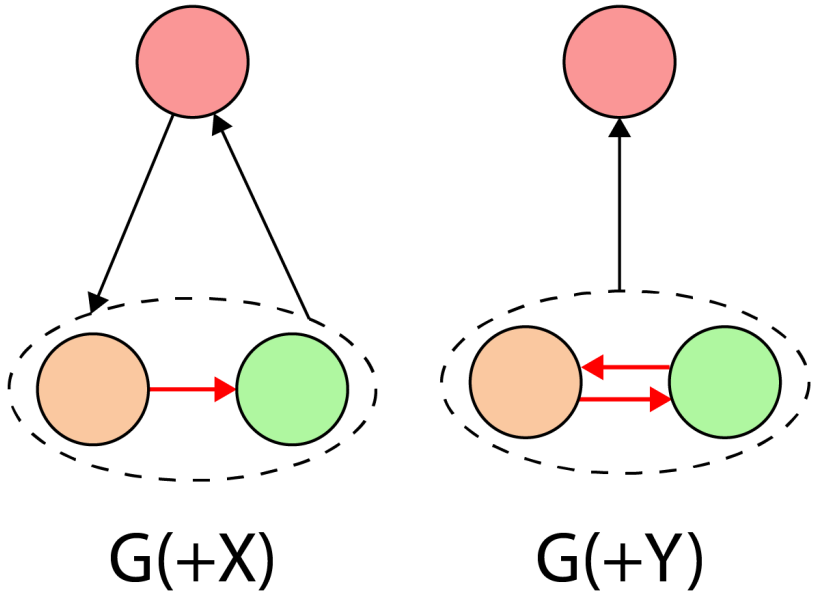




Graph Design

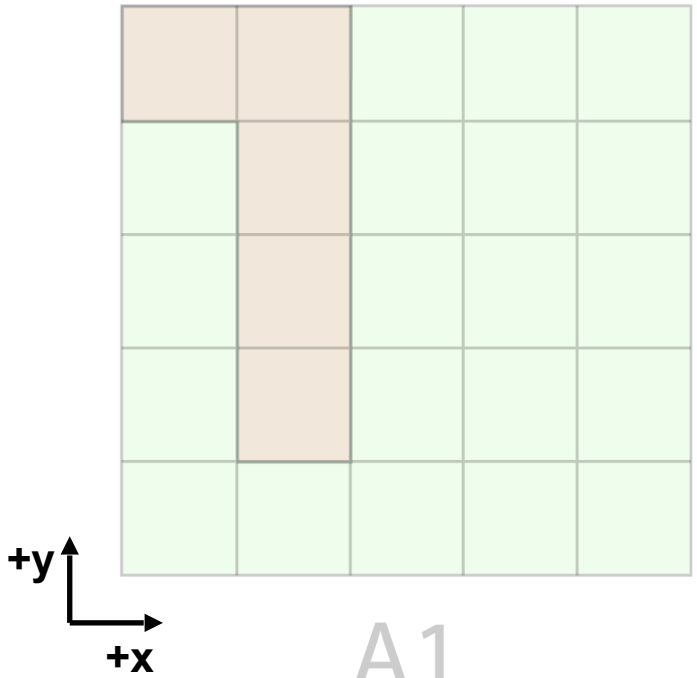
First choose the internal blocking relation between two split nodes.

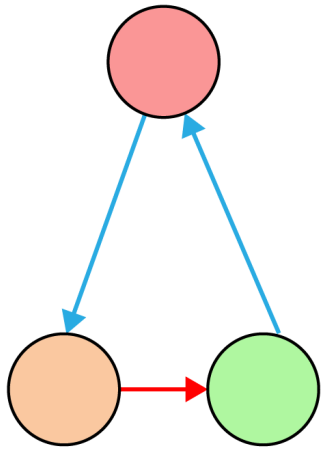




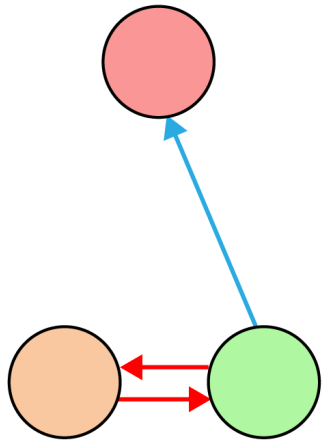
Graph Design

Then distribute the external blocking relations to the two split nodes.





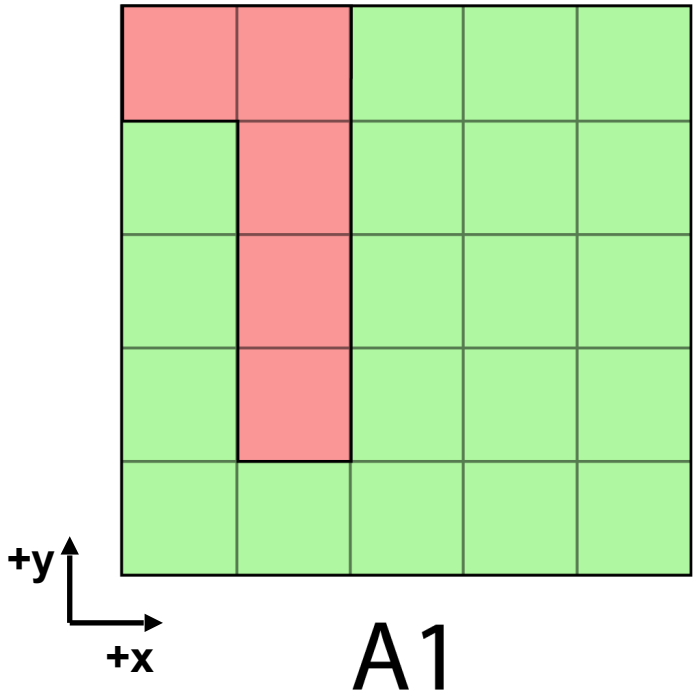
G(+X)



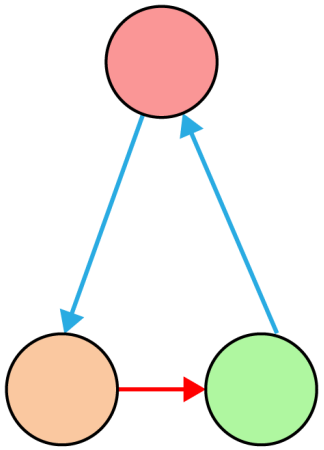
G(+Y)

Geometry Realization

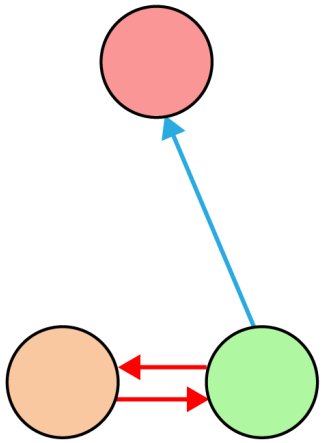
Find geometry corresponding to the designed base DBGs.



A1



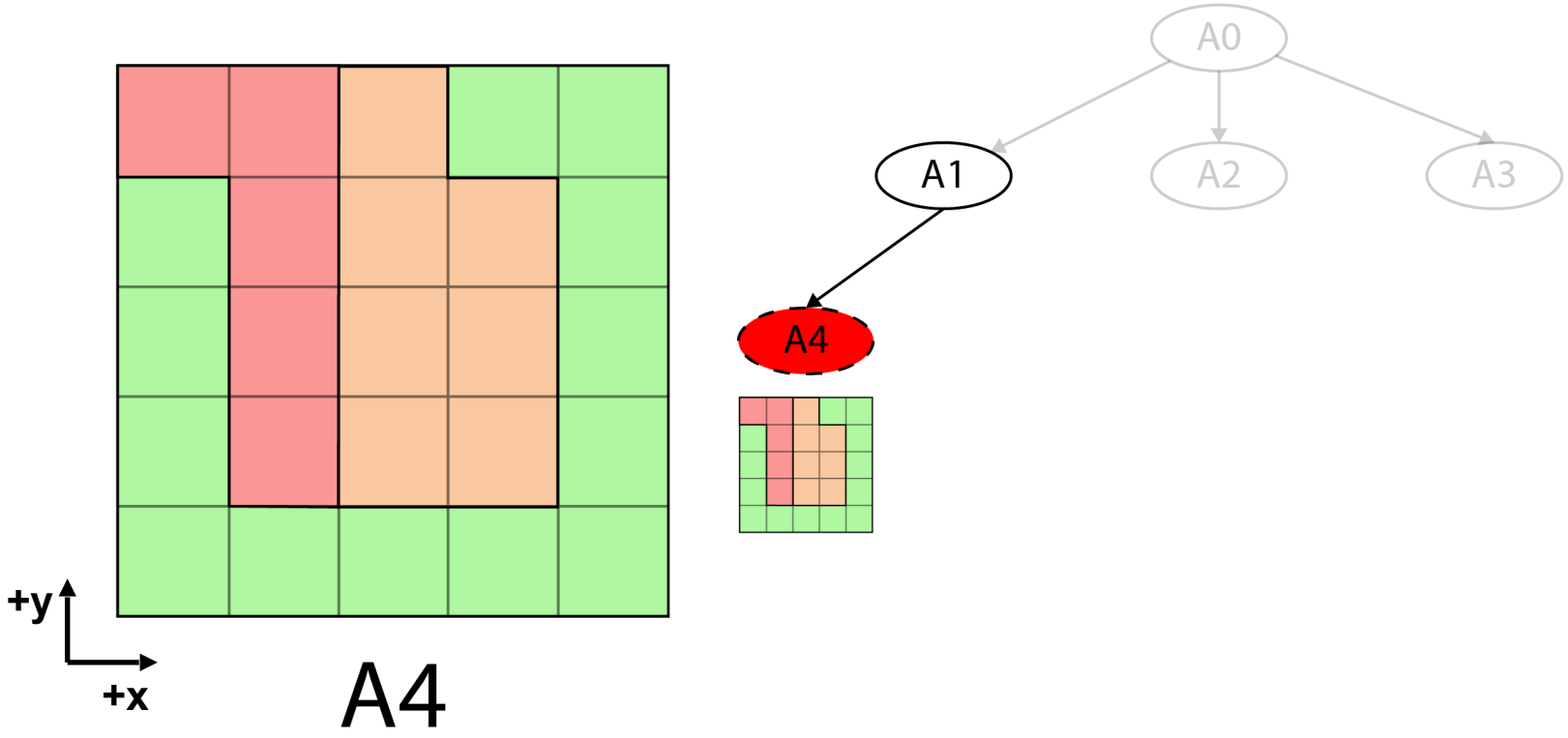
G(+X)



G(+Y)

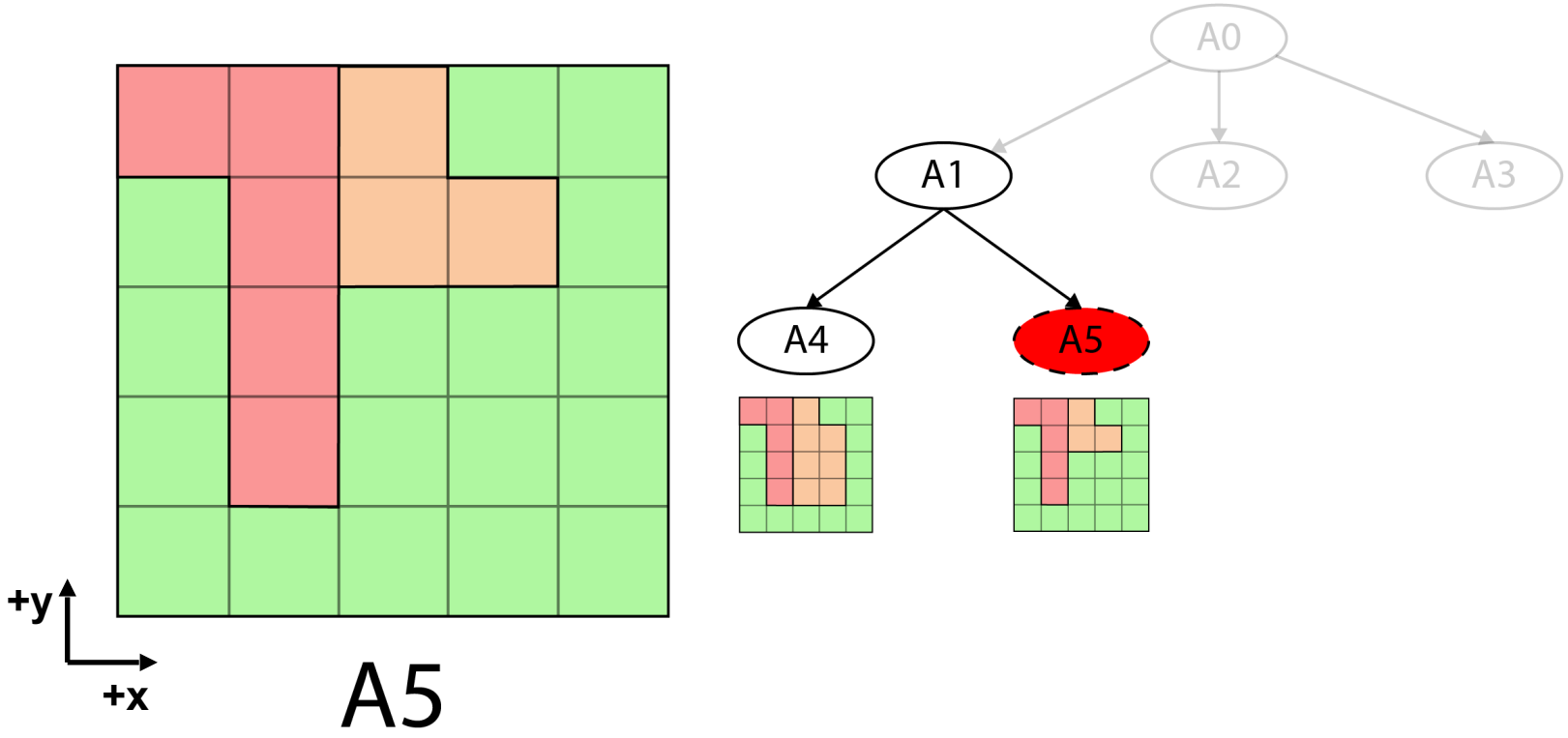
Interactive Design Framework

Generate candidates of a 3-part interlocking assembly.



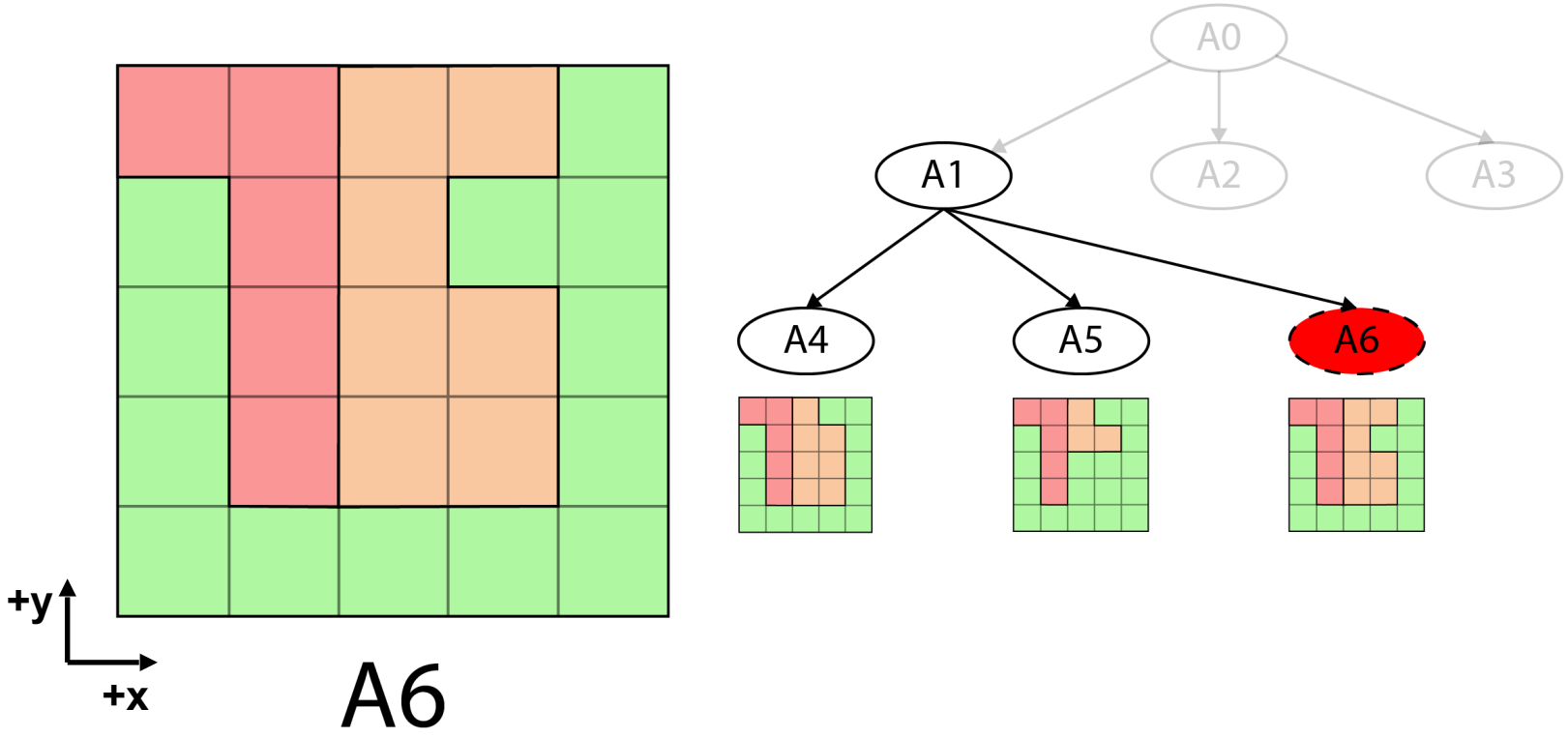
Interactive Design Framework

Generate candidates of a 3-part interlocking assembly.



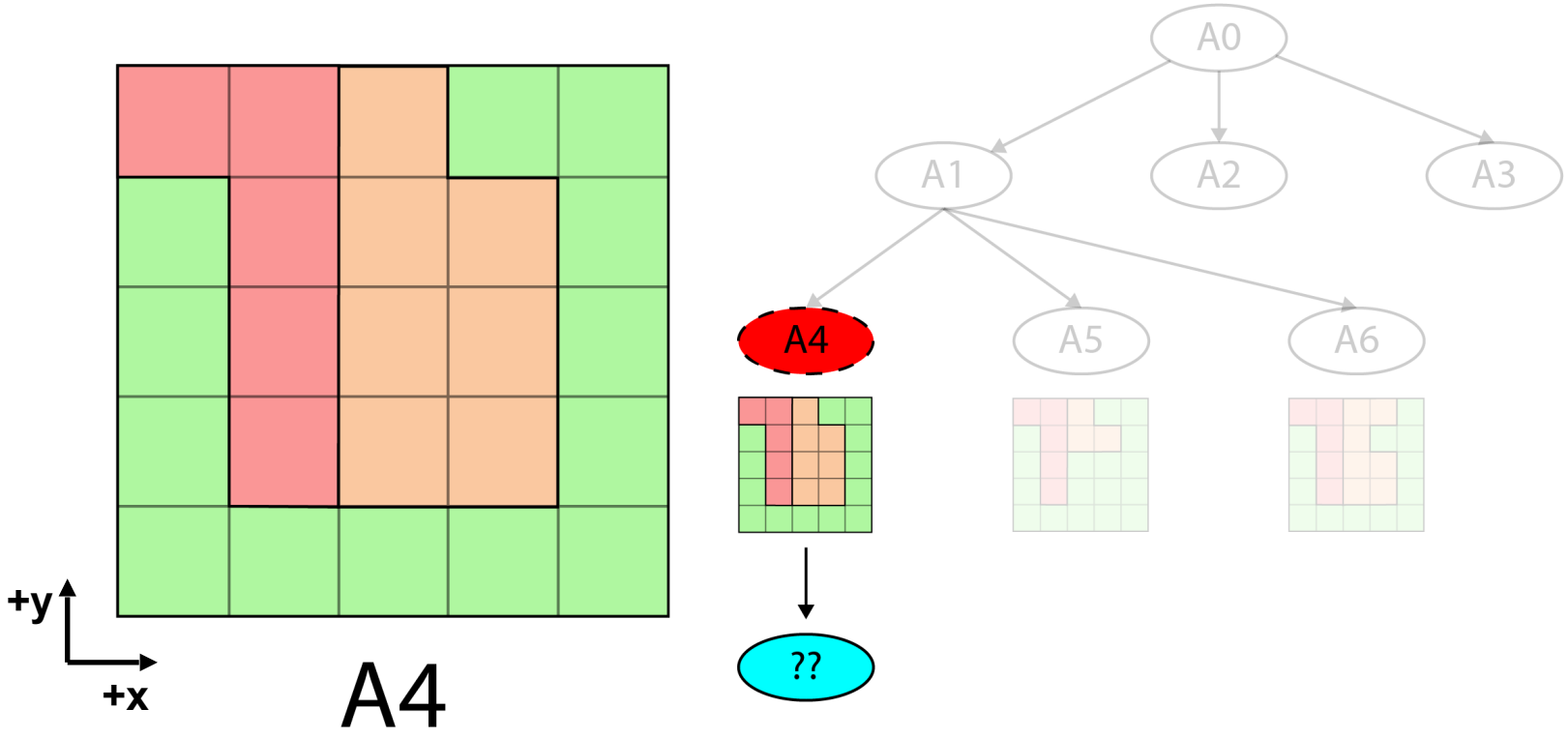
Interactive Design Framework

Generate candidates of a 3-part interlocking assembly.



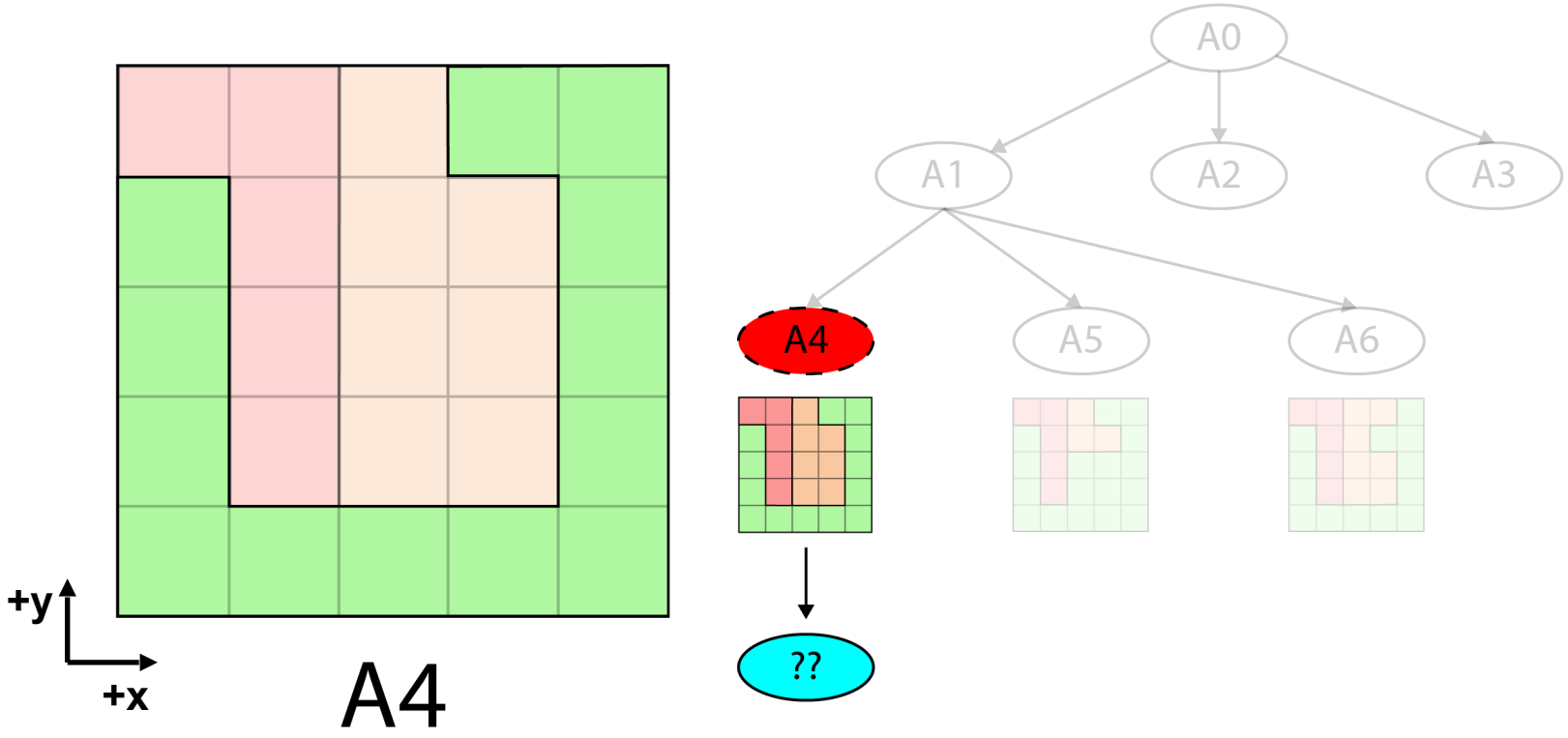
Interactive Design Framework

Can we further generate a 4-part interlocking assembly?



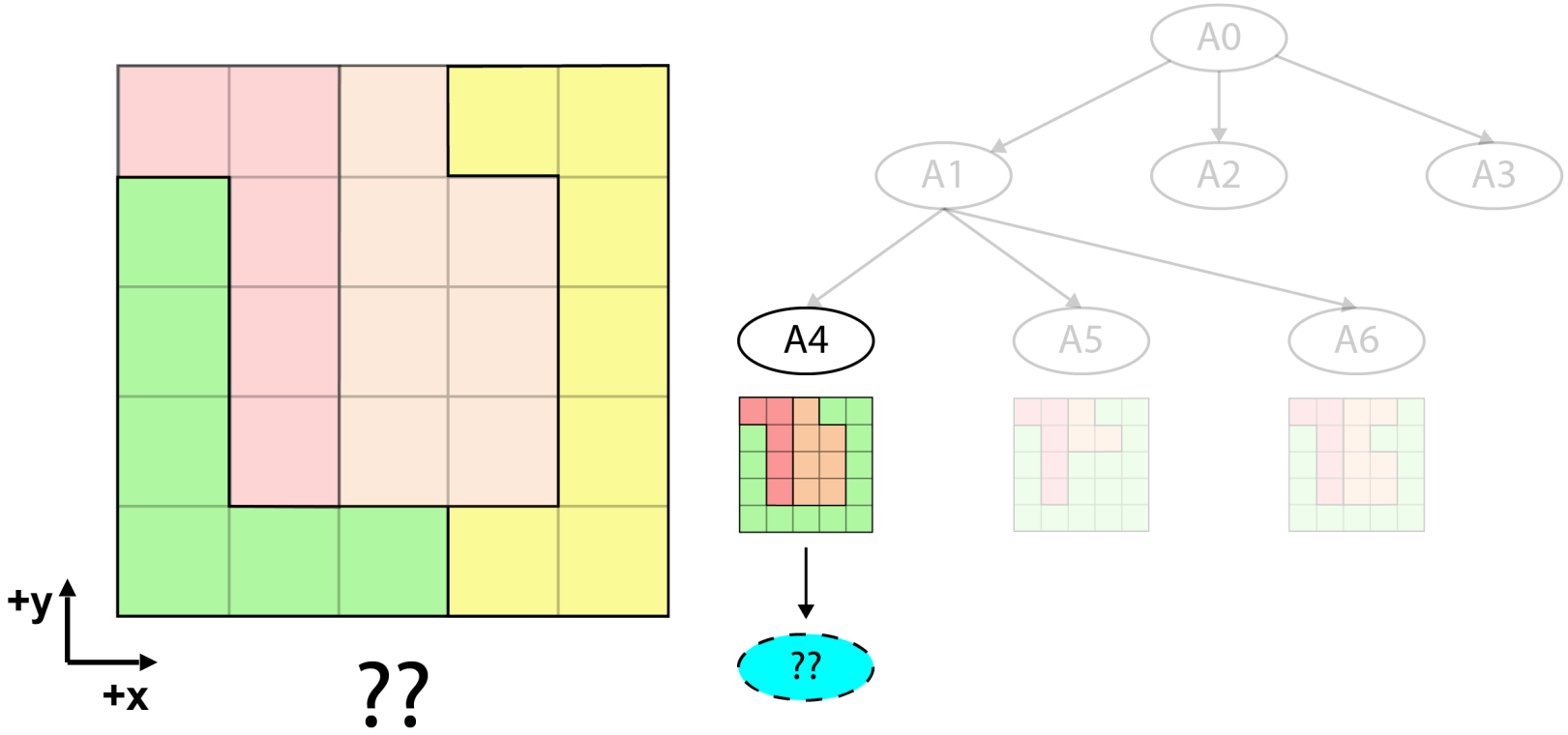
Interactive Design Framework

Can we further generate a 4-part interlocking assembly?



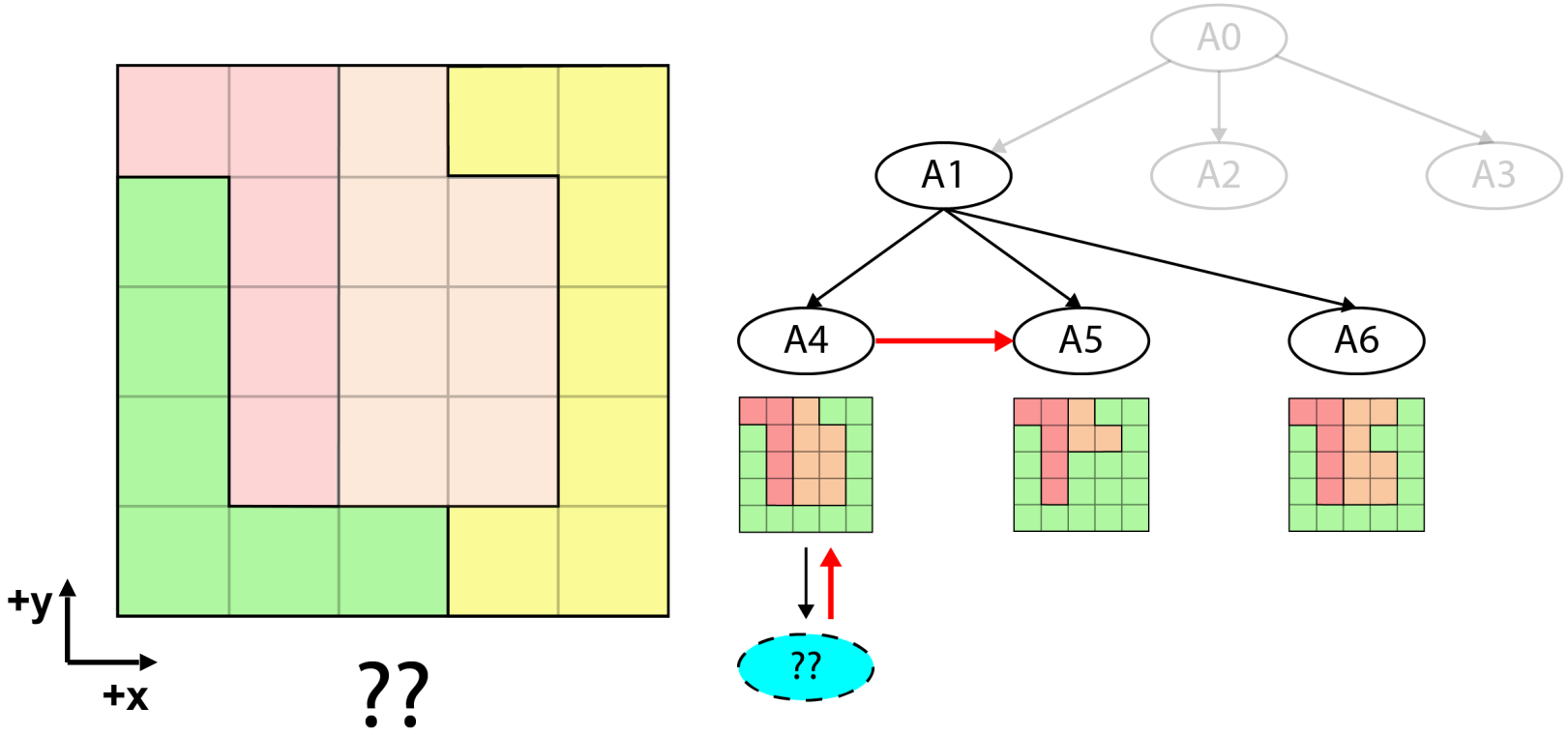
Interactive Design Framework

No matter what the partition is, this candidate cannot be interlocking.



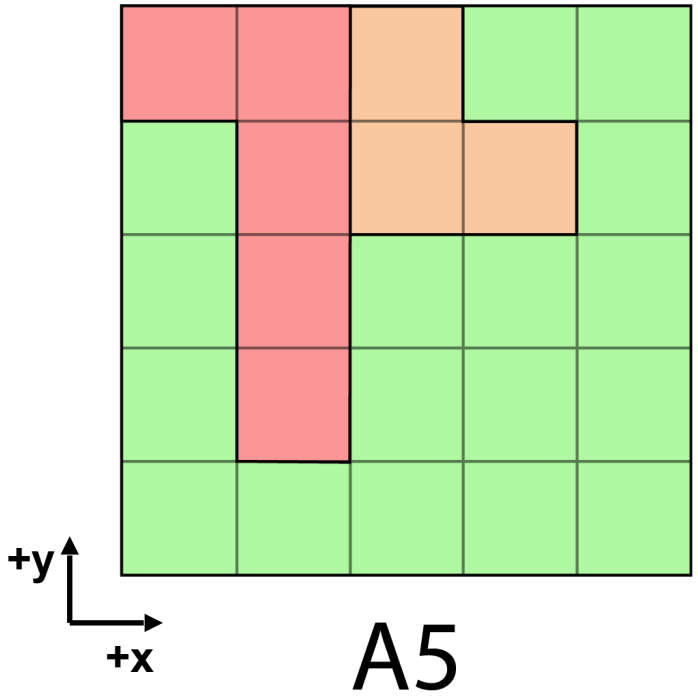
Interactive Design Framework

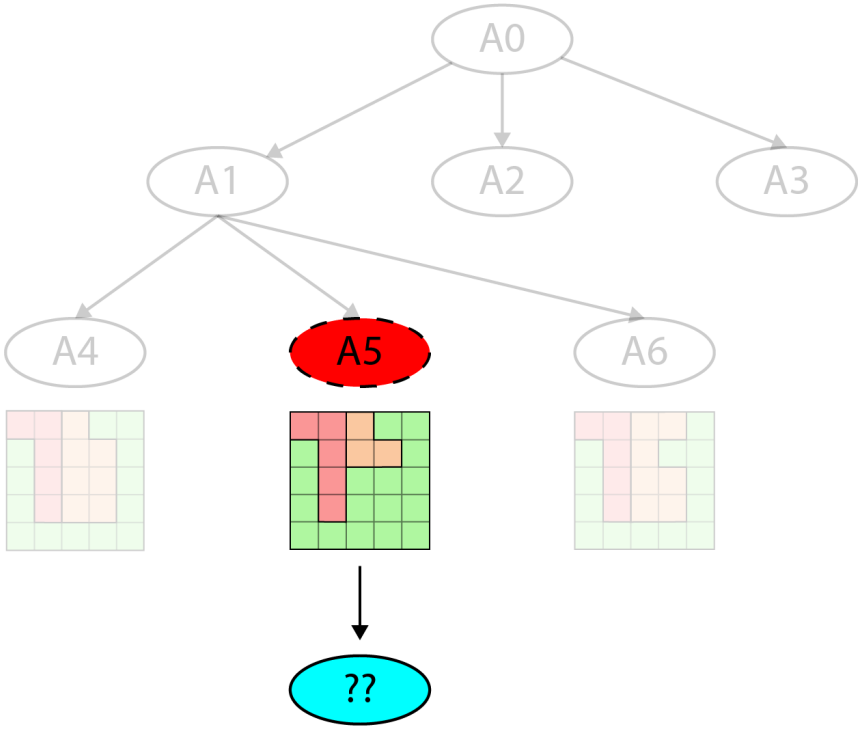
So we need to do backtracking.



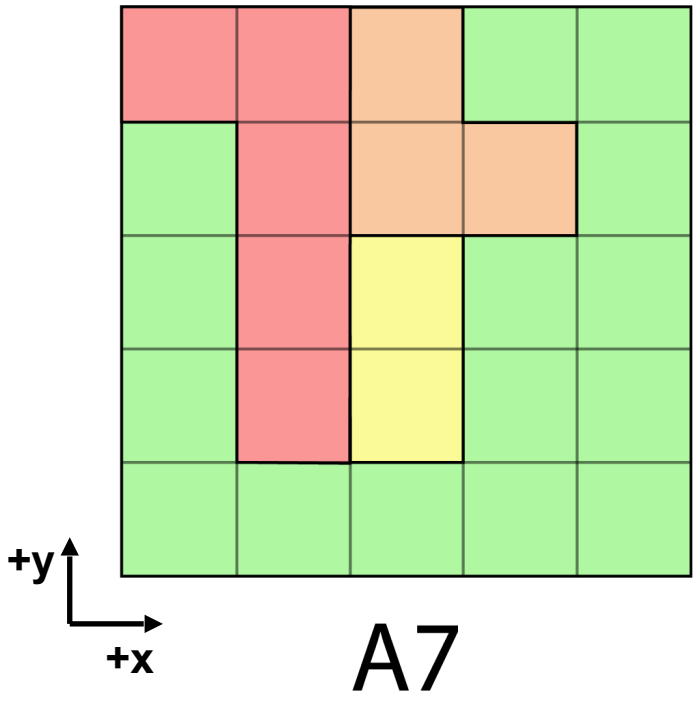
Interactive Design Framework

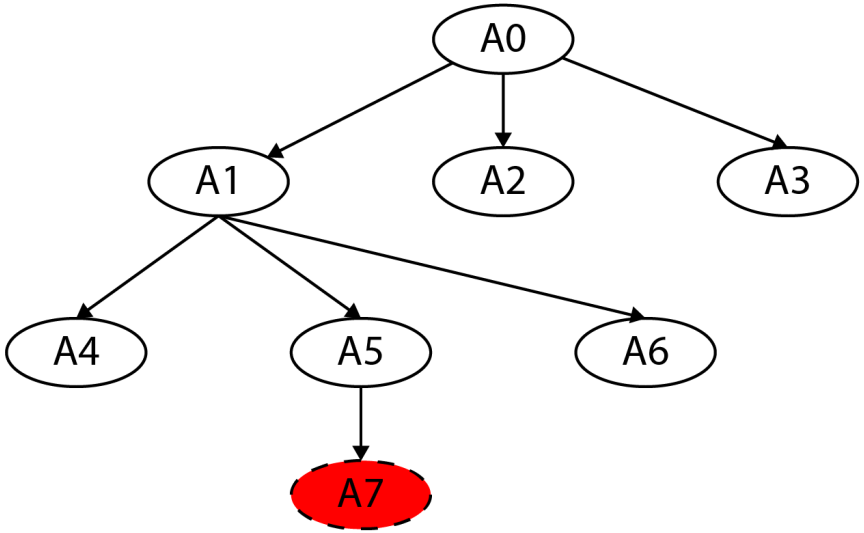
Can we generate a 4-part interlocking assembly?





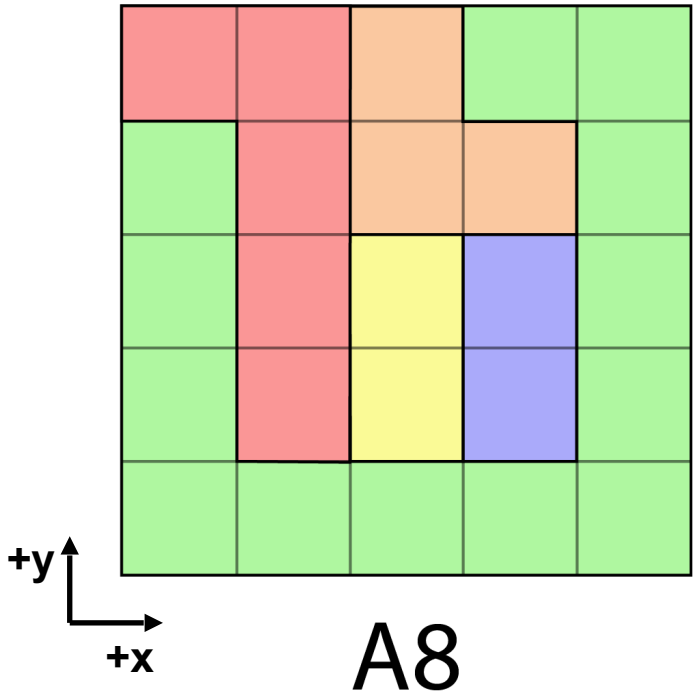
Interactive Design Framework

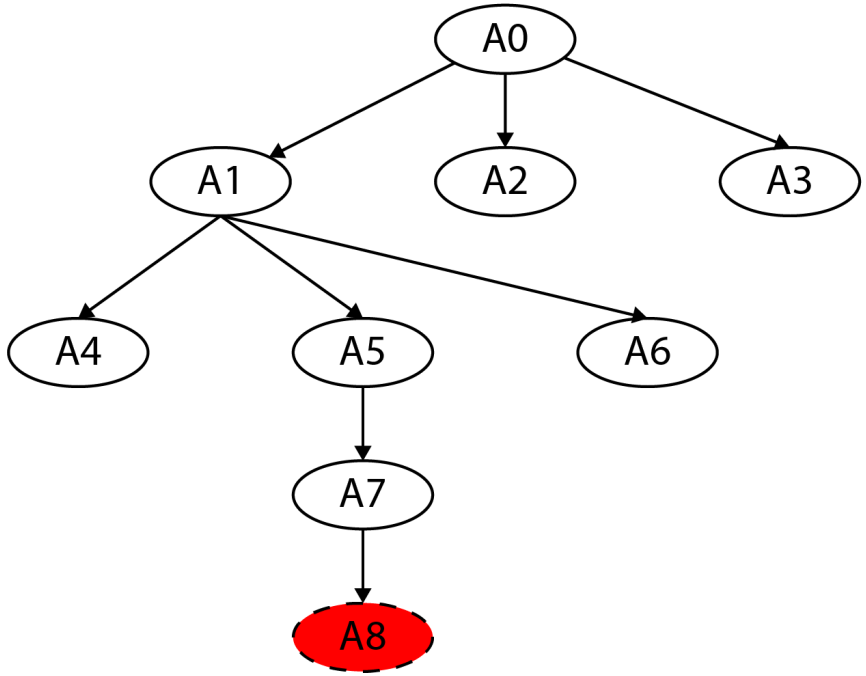




Interactive Design Framework

Succeed to find a 5-part interlocking assembly and terminate.

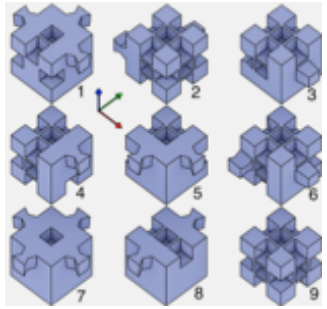


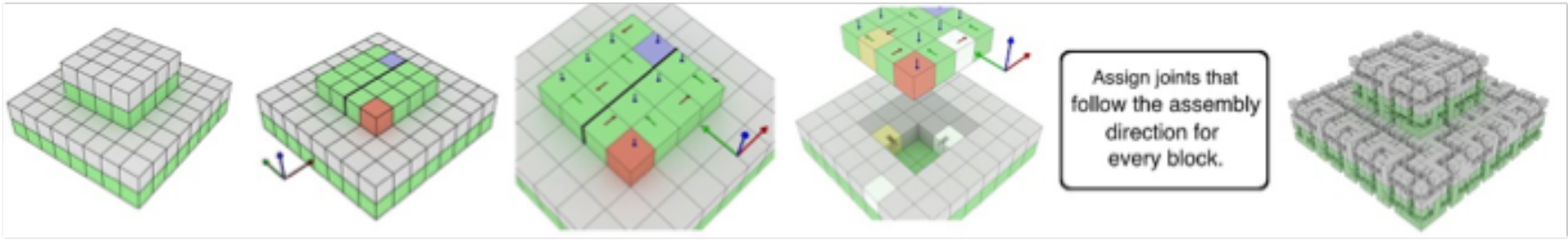


Result

Interlocking Voxels

- Zhang et al. designed a set of nine tileable blocks with integral joints, called interlocking voxels.





[Zhang et al. 2016]

Conclusion

- A general workflow for stability optimization problems.
- Force and Kinematic-based Equilibrium method
- Lateral stability, scaffold-free assembly, globally interlocking assemblies.

Thank you!