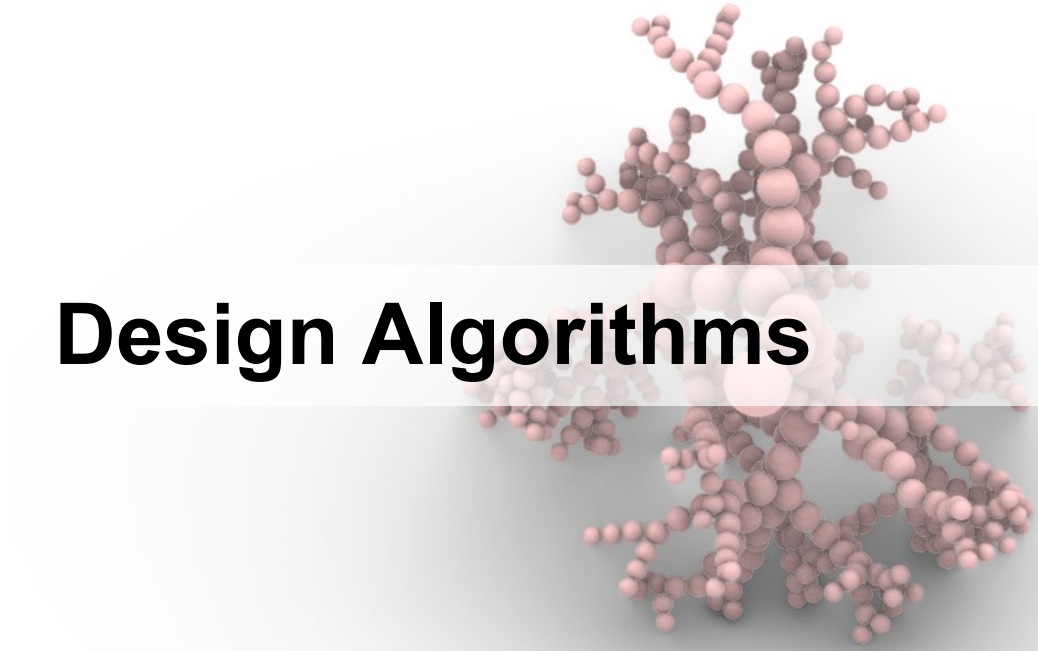
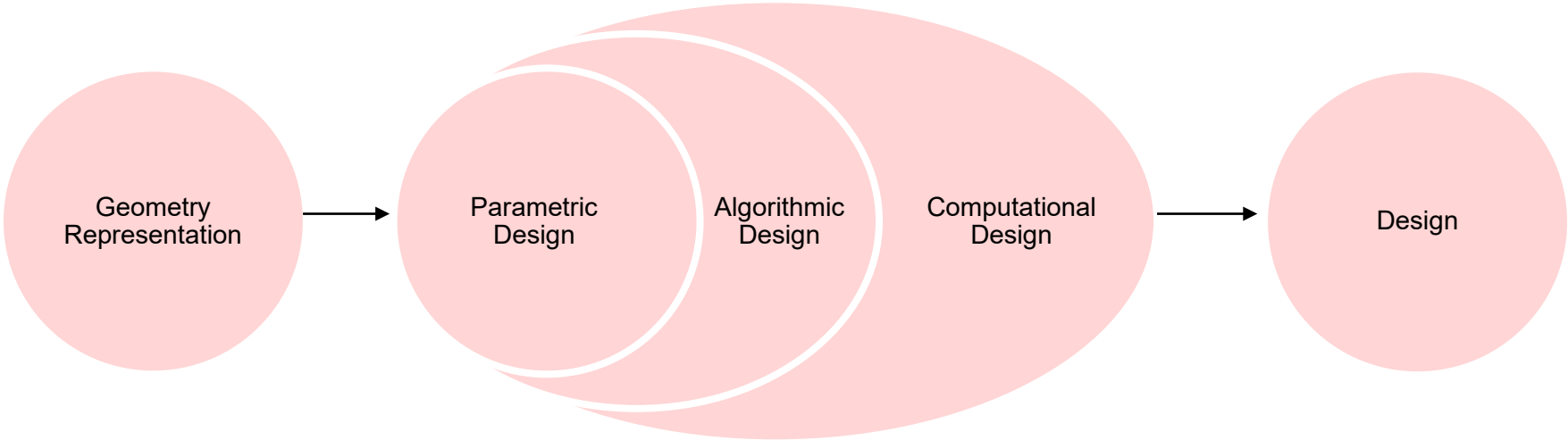
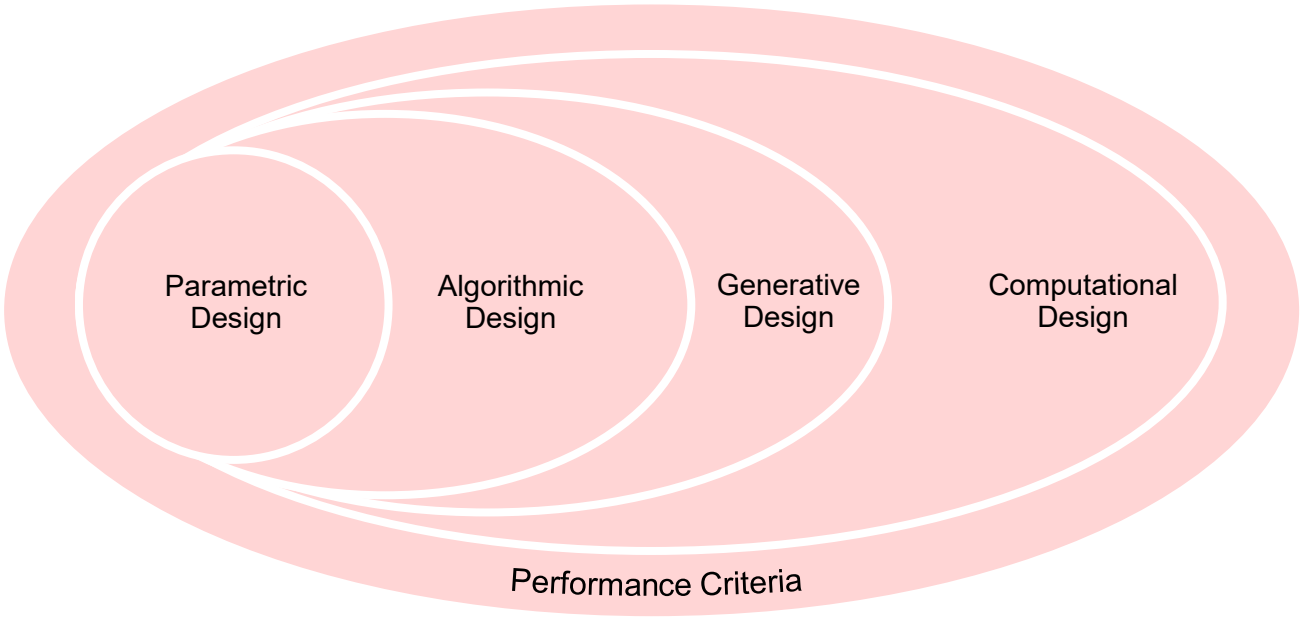


Design Algorithms

Computational Design



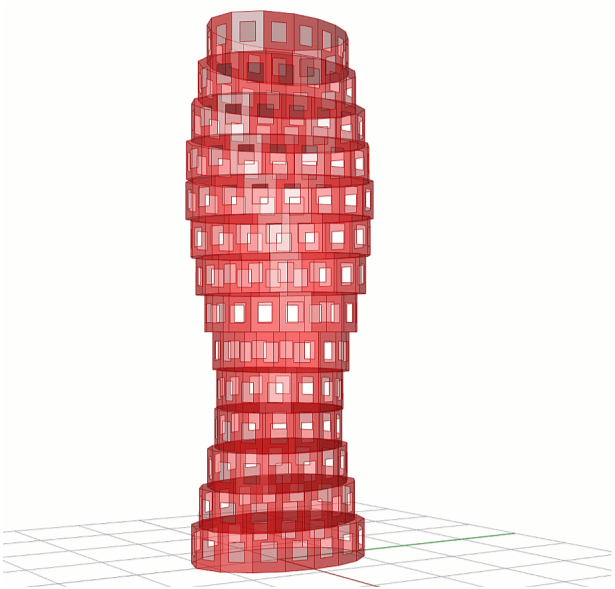
Computational Design

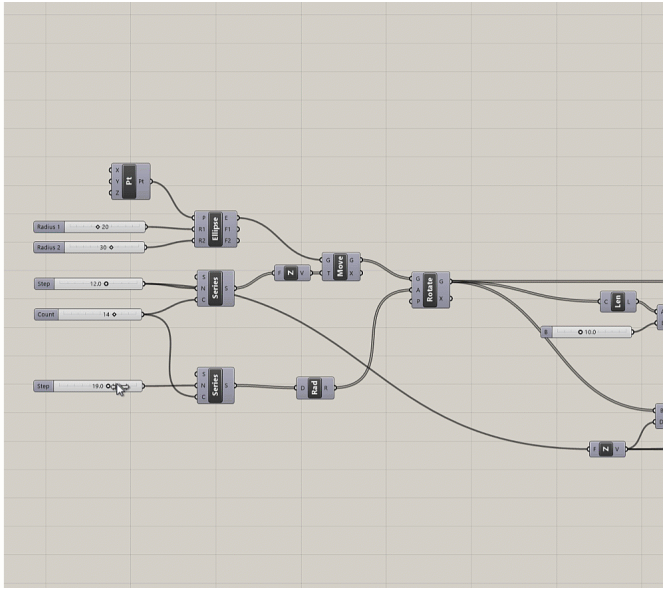


Computational design concepts

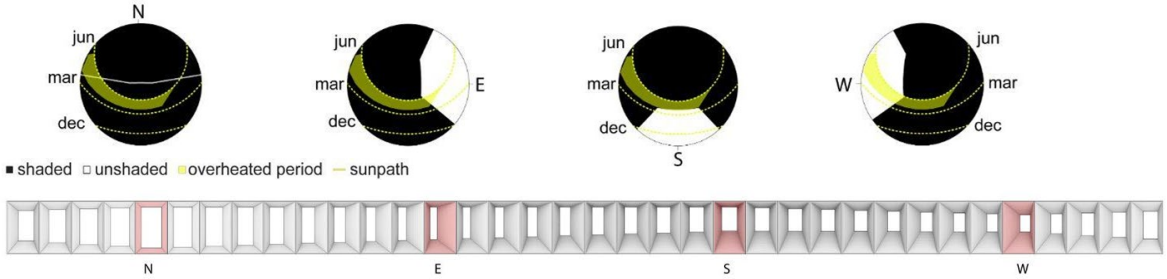
Design Concept	Focus	Role of Designer
Parametric	Relationships	Defines parameters & dependencies
Performance-based	Evaluation	Defines criteria & trade-offs
Algorithmic	Process	Designs the rules
Generative	Exploration	Curates and selects outcomes

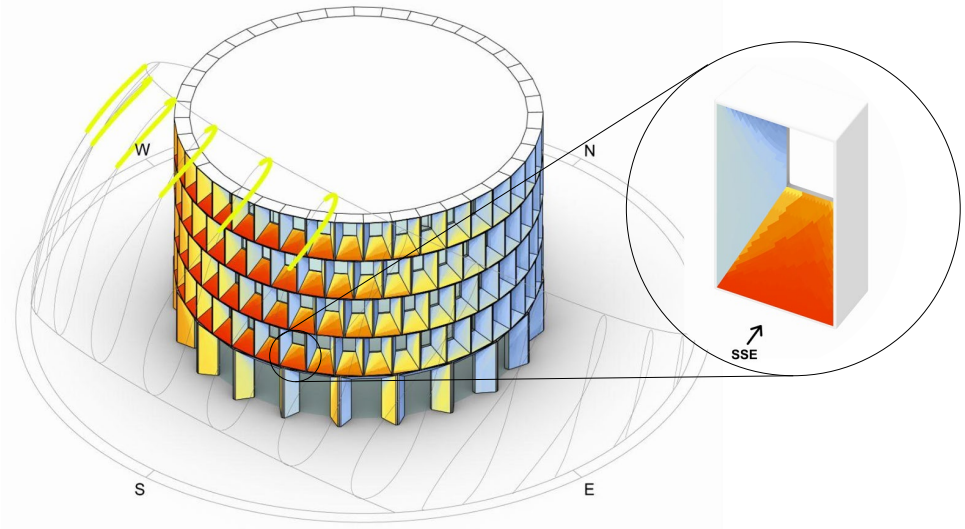
Parametric design





Performance-based parametric design





Performance-based parametric design

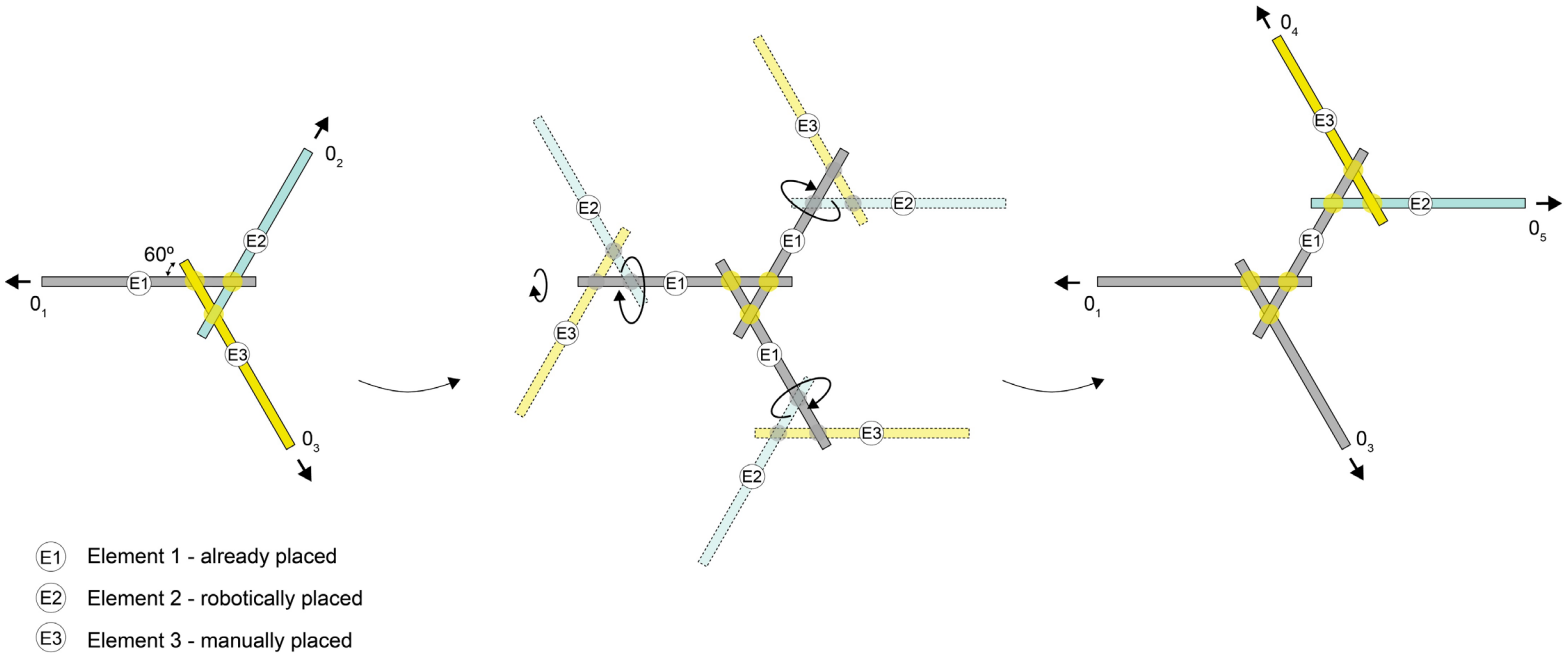
Computational design

Design Concept	Focus	Role of Designer
Parametric	Relationships	Defines parameters & dependencies
Performance-based	Evaluation	Defines criteria & trade-offs
Algorithmic	Process	Designs the rules
Generative	Exploration	Curates and selects outcomes

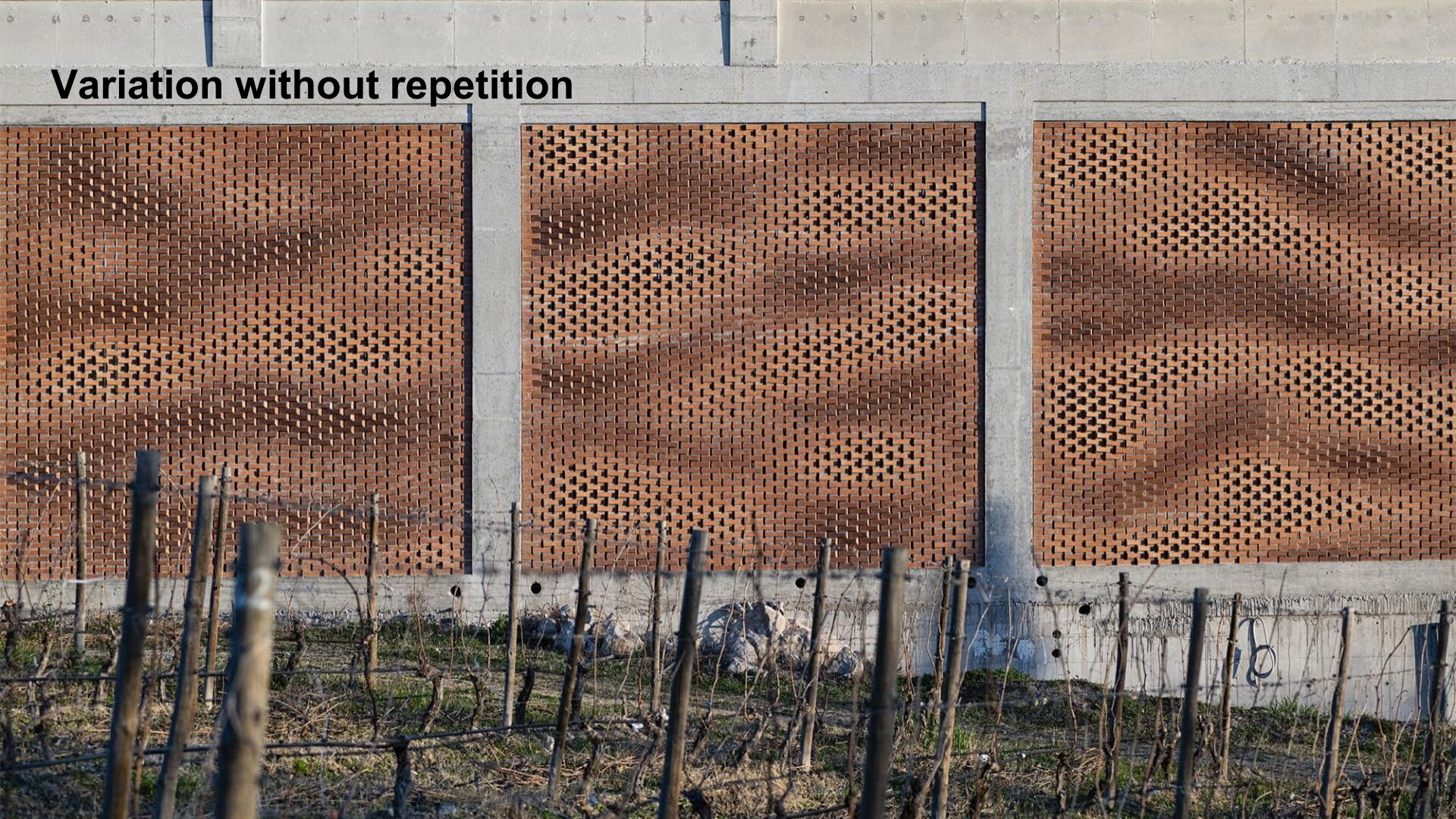
Why “Design Algorithms”?

Managing geometric complexity

From design principals to fabrication logic



Variation without repetition



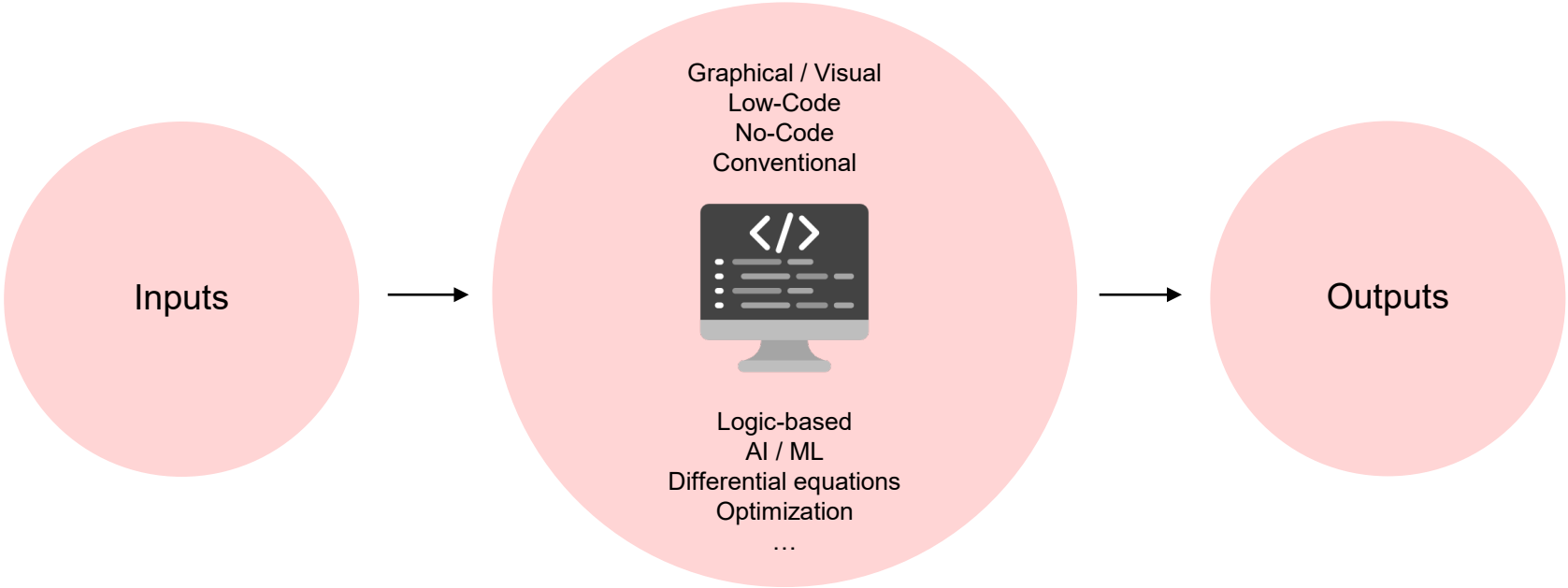
Material behavior as a design driver

Coordinating human-robot construction

. following the instructions of a second human who operates the AR app.


3. Robot 1 places an element and holds it.

What is an algorithm in design?



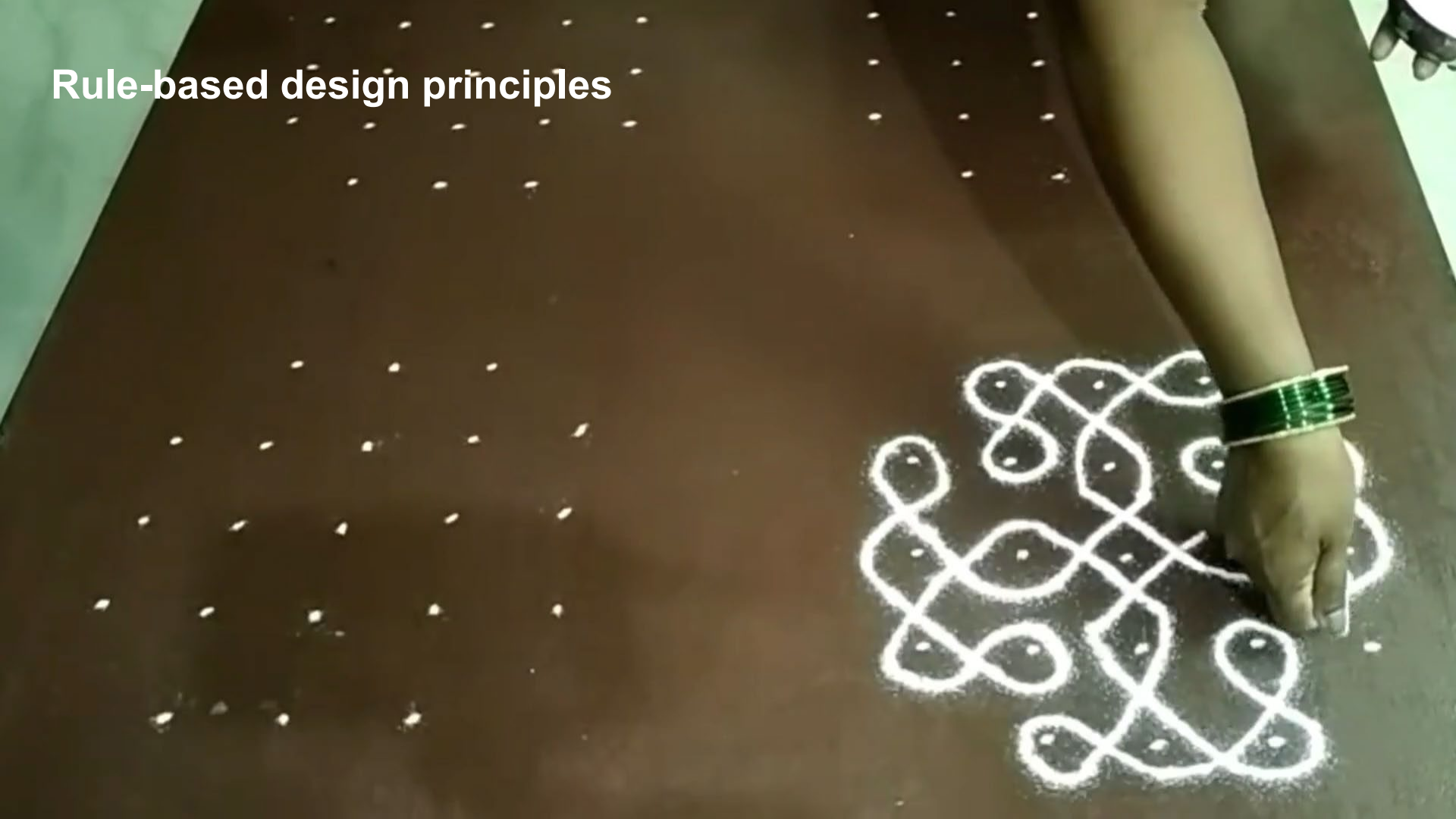
Algorithmic design approaches

- Rule-based & Procedural design
- Process- and Sequence-based design
- Growth-based design
- Agent-based design
- Constraint- and Feasibility-based design

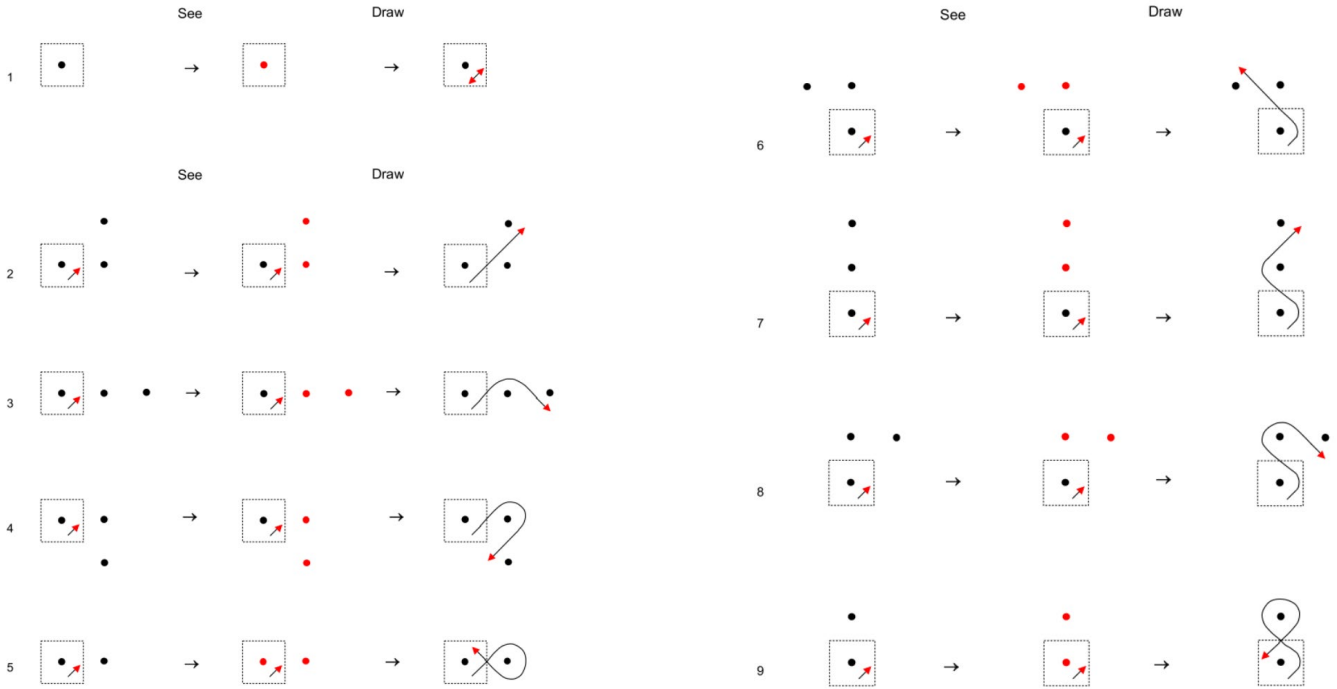
Algorithmic design approaches

- **Rule-based & Procedural design**
- Process- and Sequence-based design
- **Growth-based design**
- Agent-based design
- Constraint- and Feasibility-based design

Rule-based design principles

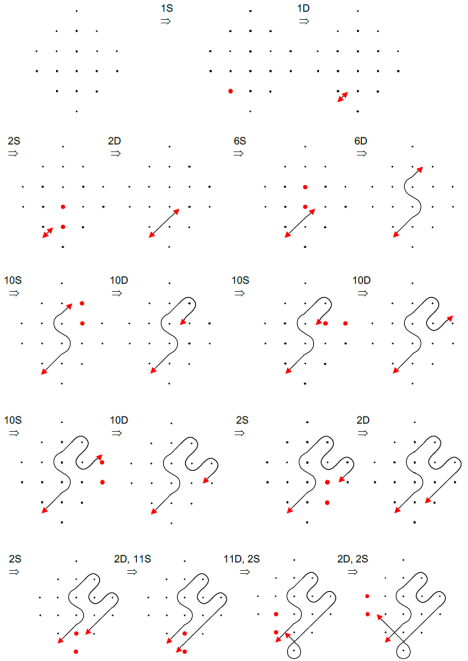


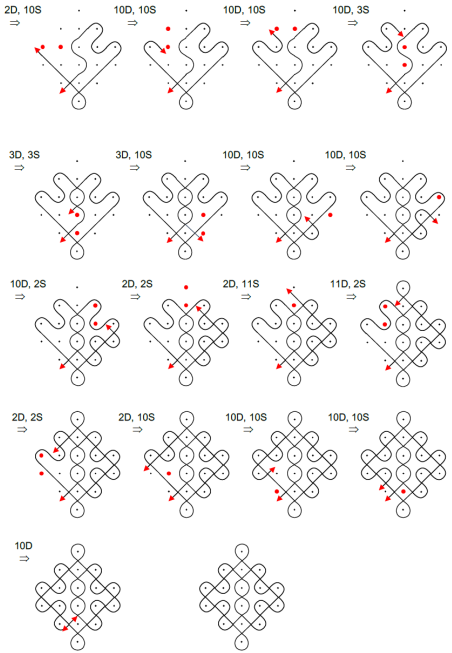
Rule-based design translation



The rules of making grammar
Source: Craft, Performance, and Grammars (Knight 2018)

Rule-based design outputs



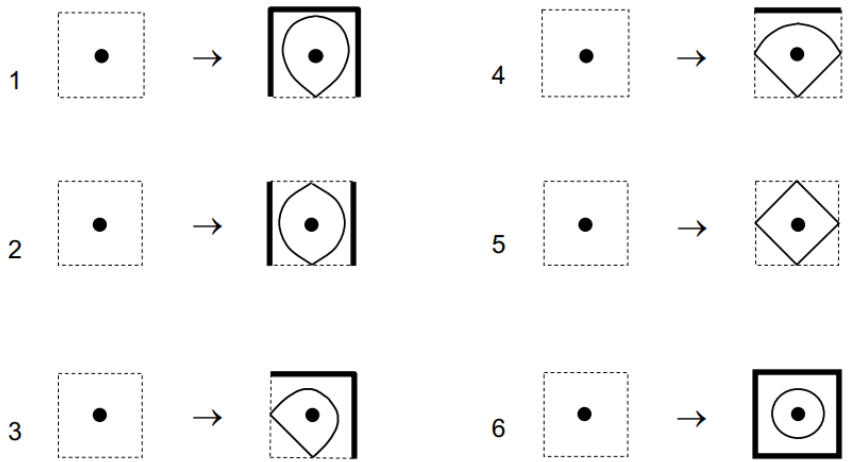


Computation of a new kolam design with making grammar
Source: Craft, Performance, and Grammars (Knight 2018)

Try it yourself

Simplified modular motif shape grammar:

- **A. Understand the grammar**
- B. Decode the designs
- C. Create your own designs



Computation of a new kolam design with making grammar
Source: Craft, Performance, and Grammars (Knight 2018)

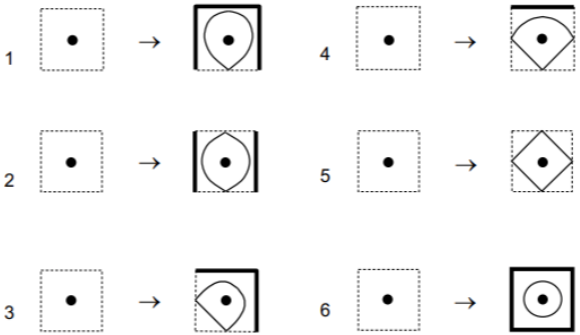
Try it yourself

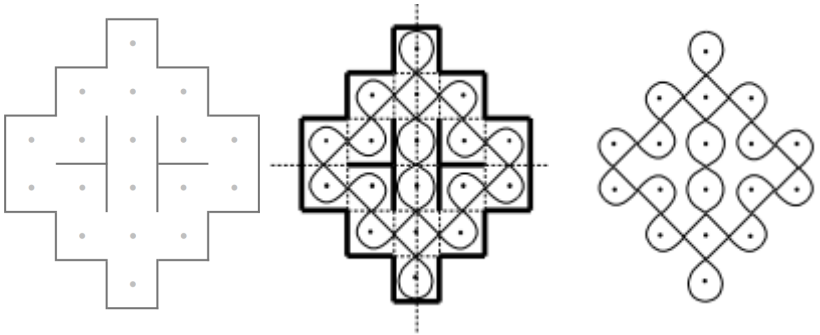
Simplified modular motif shape grammar:

A. Understand the grammar

B. Decode the designs

C. Create your own designs





Computation of a new kolam design with making grammar
Source: Craft, Performance, and Grammars (Knight 2018)

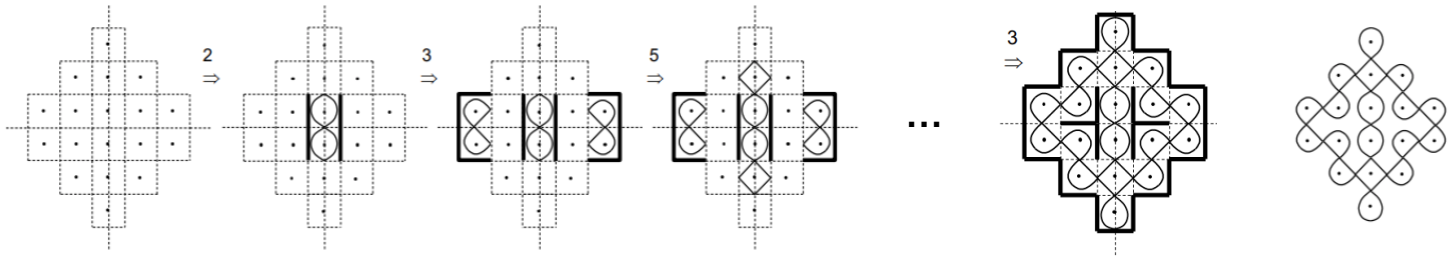
Try it yourself

Simplified modular motif shape grammar:

A. Understand the grammar

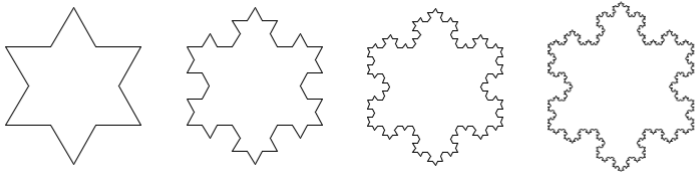
B. Decode the designs

C. Create your own designs

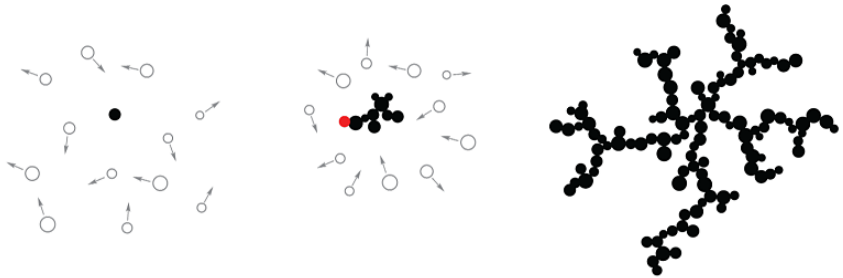


Computation of a new kolam design with making grammar
Source: Craft, Performance, and Grammars (Knight 2018)

Growth-based design



Recursive growth



Sean Collier Memorial at MIT, 2000

Dendrite growth

Examples

Recursive Growth (lines)

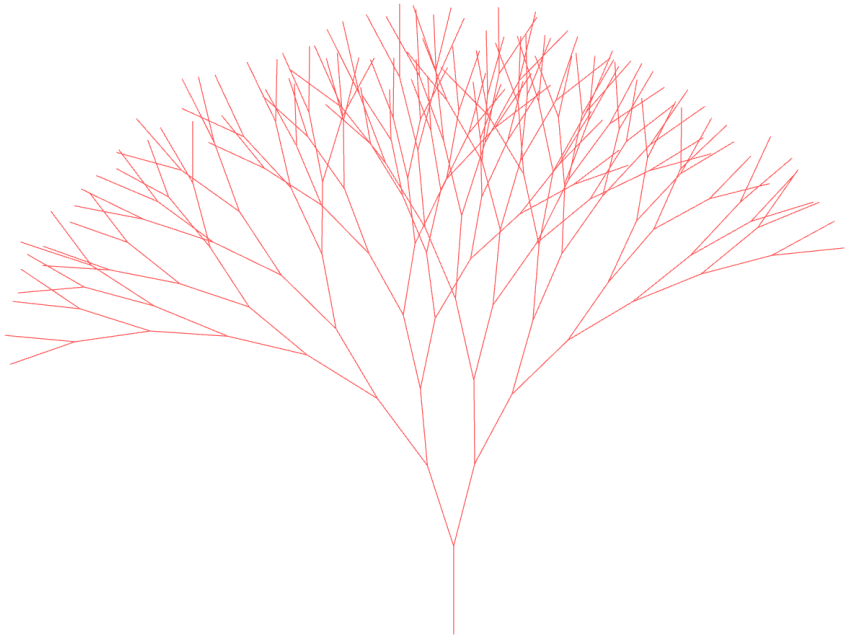
Learning objectives

1. Constructing **branching line systems** using **recursion**.
2. **Rotating and scaling vectors** to control directional growth.
3. Applying **stochastic rules to generate variation** in recursive structures.

Recursive Growth (lines)

Learning objectives

1. Constructing **branching line systems** using **recursion**.
2. **Rotating and scaling vectors** to control directional growth.
3. Applying **stochastic rules to generate variation** in recursive structures.



Recursive Growth (lines)

Learning objectives

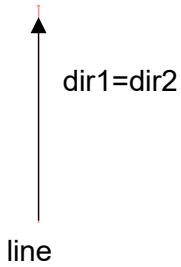
1. Constructing **branching line systems** using **recursion**.
2. **Rotating and scaling vectors** to control directional growth.
3. Applying **stochastic rules to generate variation** in recursive structures.

line

Recursive Growth (lines)

Learning objectives

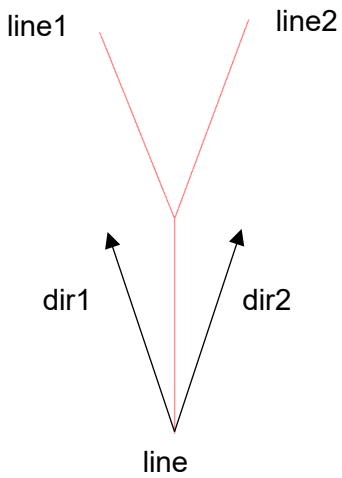
1. Constructing **branching line systems** using **recursion**.
2. **Rotating and scaling vectors** to control directional growth.
3. Applying **stochastic rules to generate variation** in recursive structures.



Recursive Growth (lines)

Learning objectives

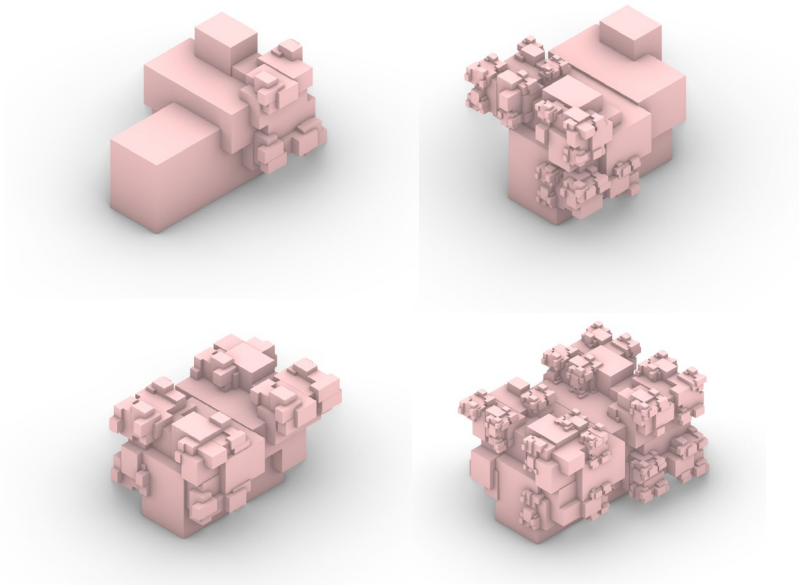
1. Constructing **branching line systems** using **recursion**.
2. **Rotating and scaling vectors** to control directional growth.
3. Applying **stochastic rules to generate variation** in recursive structures.



Recursive Growth (boxes)

Learning objectives

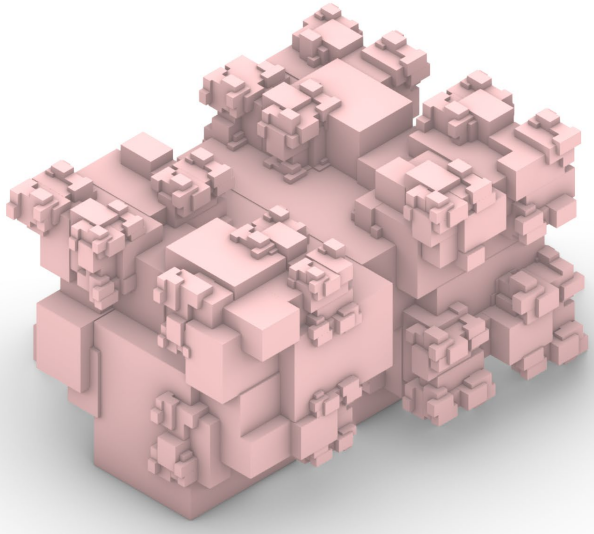
1. Generating **volumetric boxes from recursive line growth**.
2. **Constructing local coordinate frames** using cross products.
3. Creating bounding **volumes aligned to dynamically oriented geometry**.



Recursive Growth (boxes)

Learning objectives

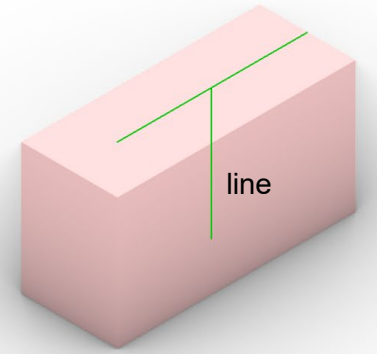
1. Generating **volumetric boxes from recursive line growth**.
2. **Constructing local coordinate frames** using cross products.
3. Creating bounding **volumes aligned to dynamically oriented geometry**.



Recursive Growth (boxes)

Learning objectives

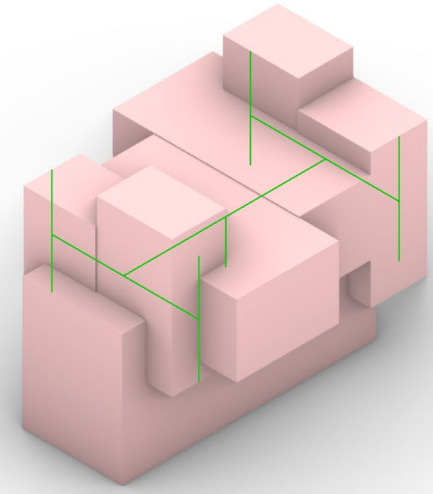
1. Generating **volumetric boxes from recursive line growth**.
2. **Constructing local coordinate frames** using cross products.
3. Creating bounding **volumes aligned to dynamically oriented geometry**.



Recursive Growth (boxes)

Learning objectives

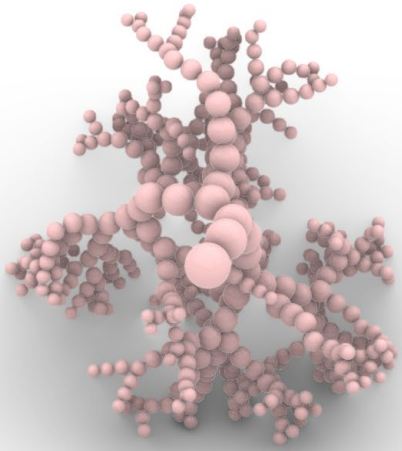
1. Generating **volumetric boxes from recursive line growth**.
2. **Constructing local coordinate frames** using cross products.
3. Creating bounding **volumes aligned to dynamically oriented geometry**.



Dendrite Growth

Learning objectives

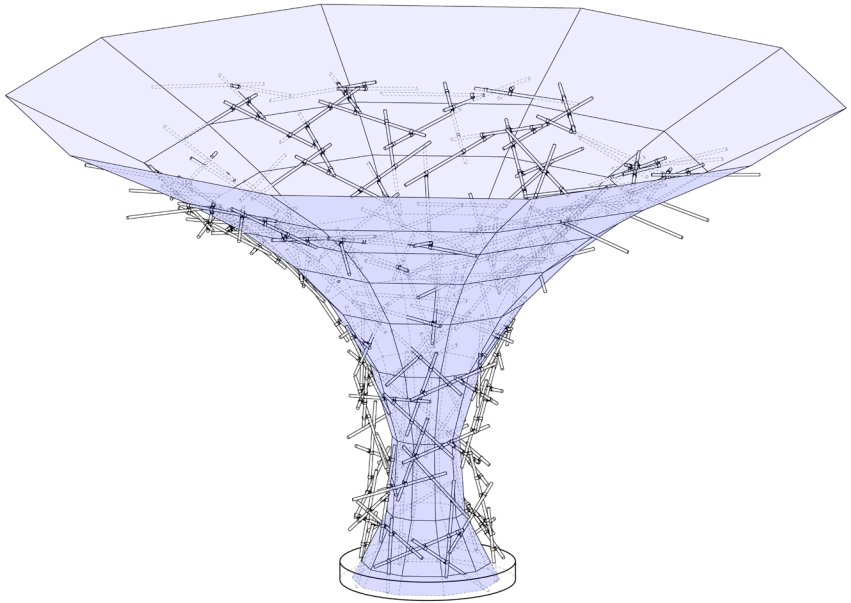
1. **Constructing sphere aggregates** through iterative tip-based growth.
2. Deflecting growth directions in three-dimensional space.
3. Preventing geometric overlap using **spatial collision checks**.



Constrained Growth

Learning objectives

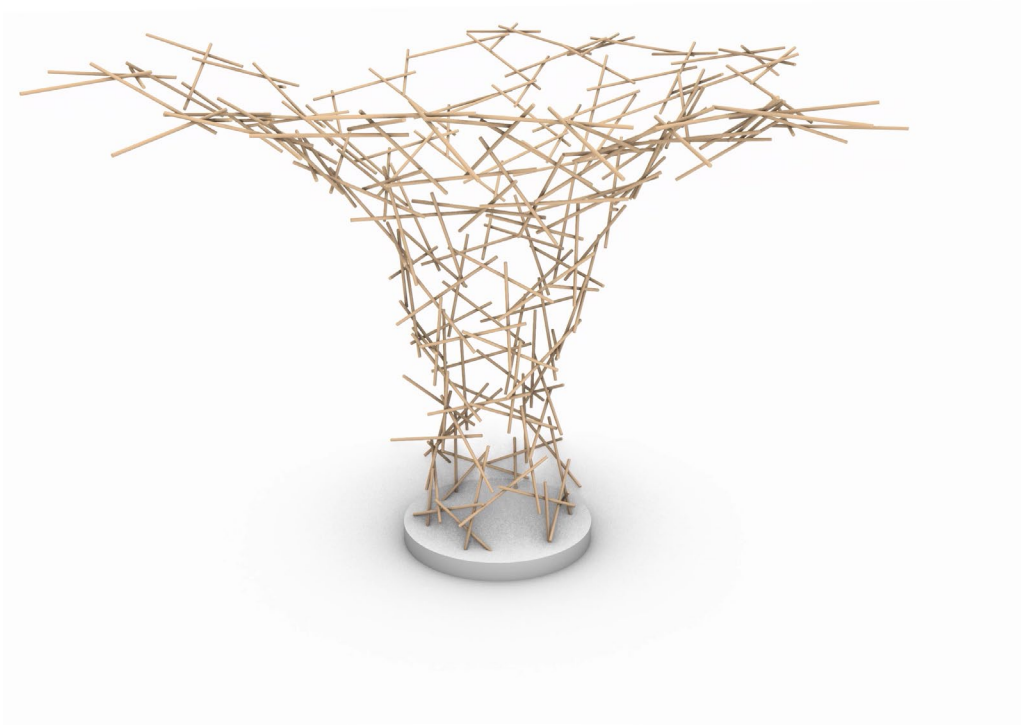
1. **Creating a custom spatial growth algorithm.**
2. Introducing **design criteria** and **growth constraints**.
3. Preventing geometric overlap using **spatial collision checks**.



Constrained Growth

Learning objectives

1. **Creating a custom spatial growth algorithm.**
2. Introducing **design criteria** and **growth constraints**.
3. Preventing geometric overlap using **spatial collision checks**.



Enjoy creating!