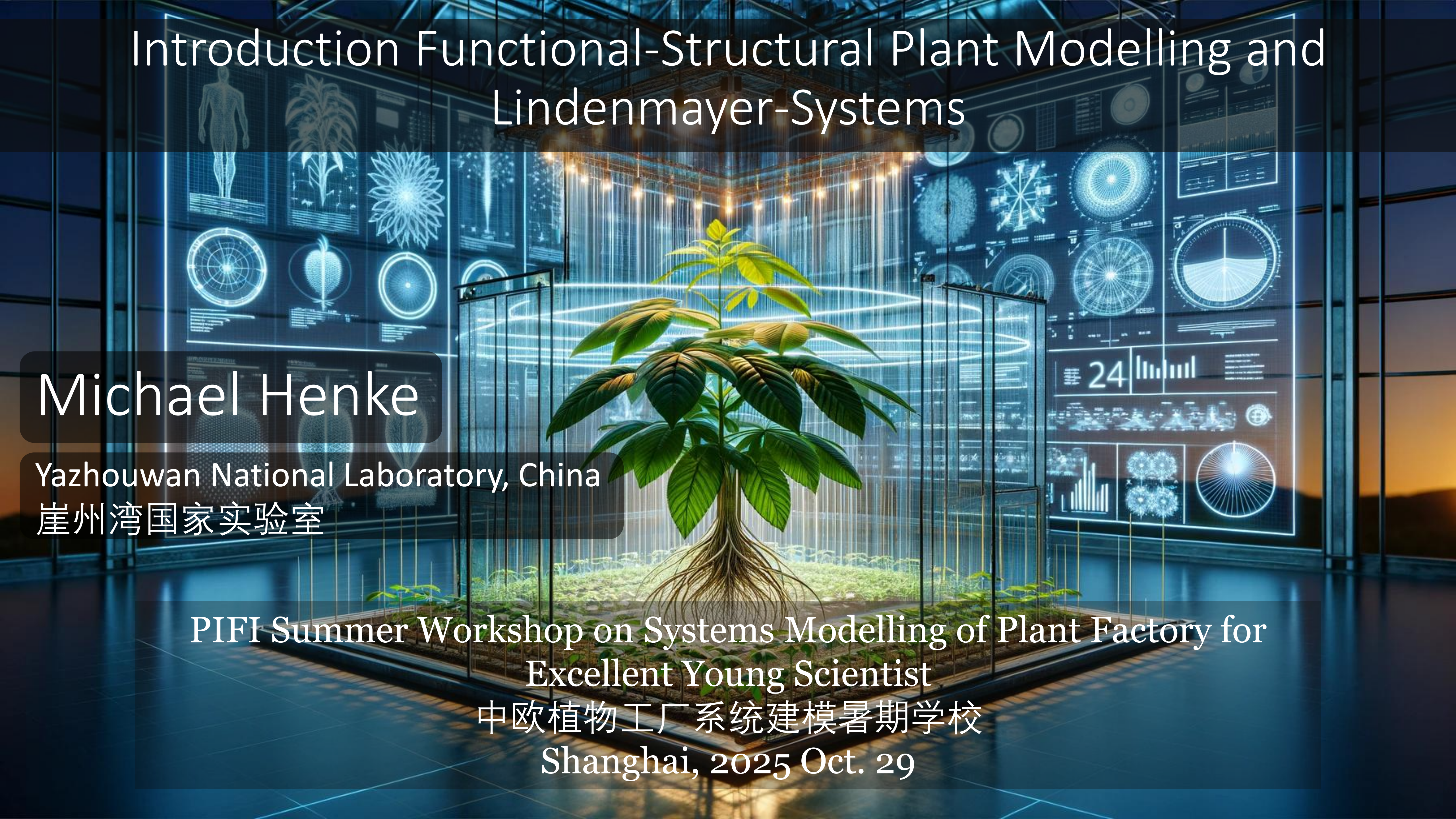
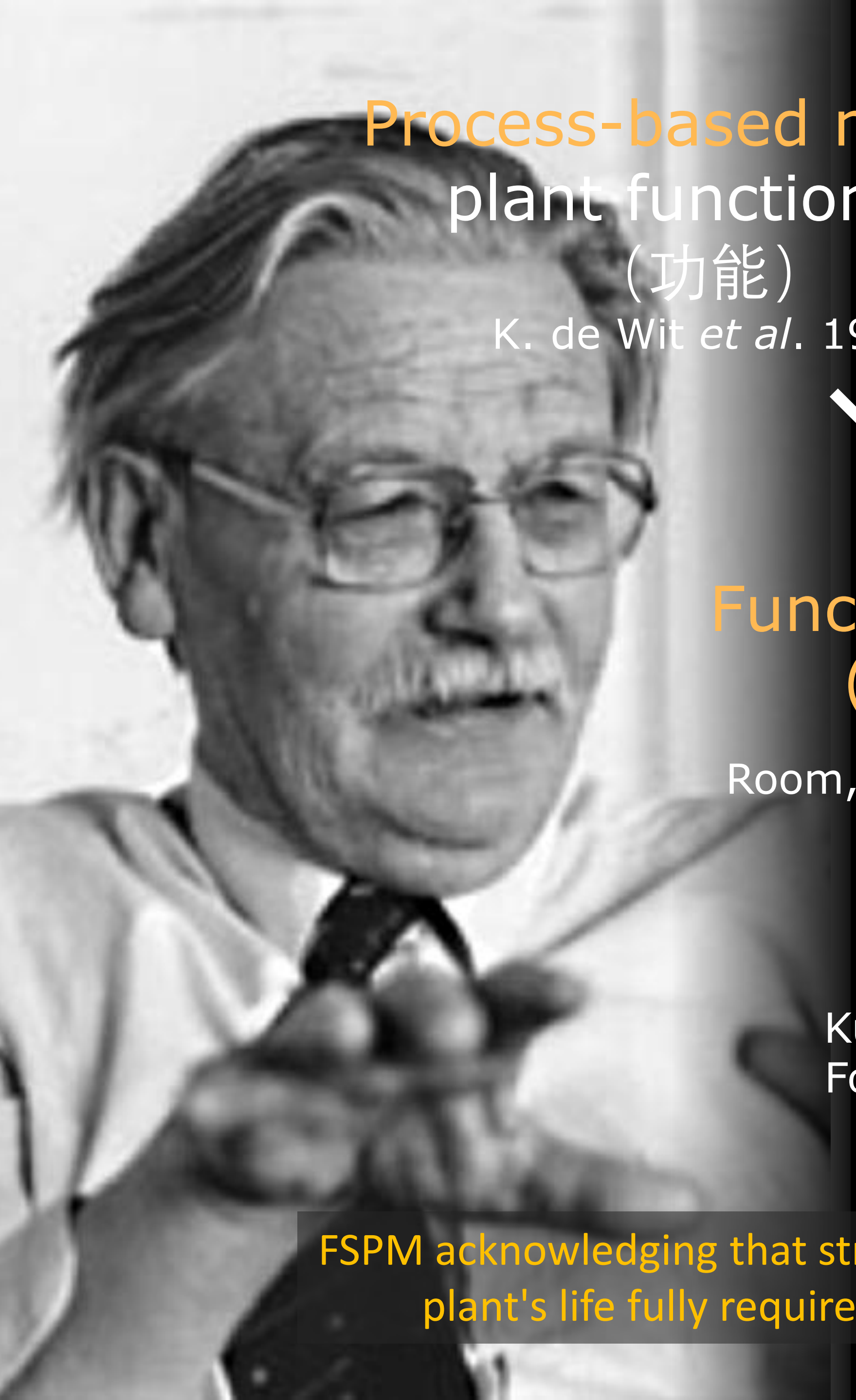


Introduction Functional-Structural Plant Modelling and Lindenmayer-Systems

Michael Henke

Yazhouwan National Laboratory, China
崖州湾国家实验室

PIFI Summer Workshop on Systems Modelling of Plant Factory for Excellent Young Scientist
中欧植物工厂系统建模暑期学校
Shanghai, 2025 Oct. 29



Process-based models
plant functioning
（功能）
K. de Wit *et al*. 1965

Architectural models
plant structure
（结构）
A. Lindenmayer *et al*. 1968

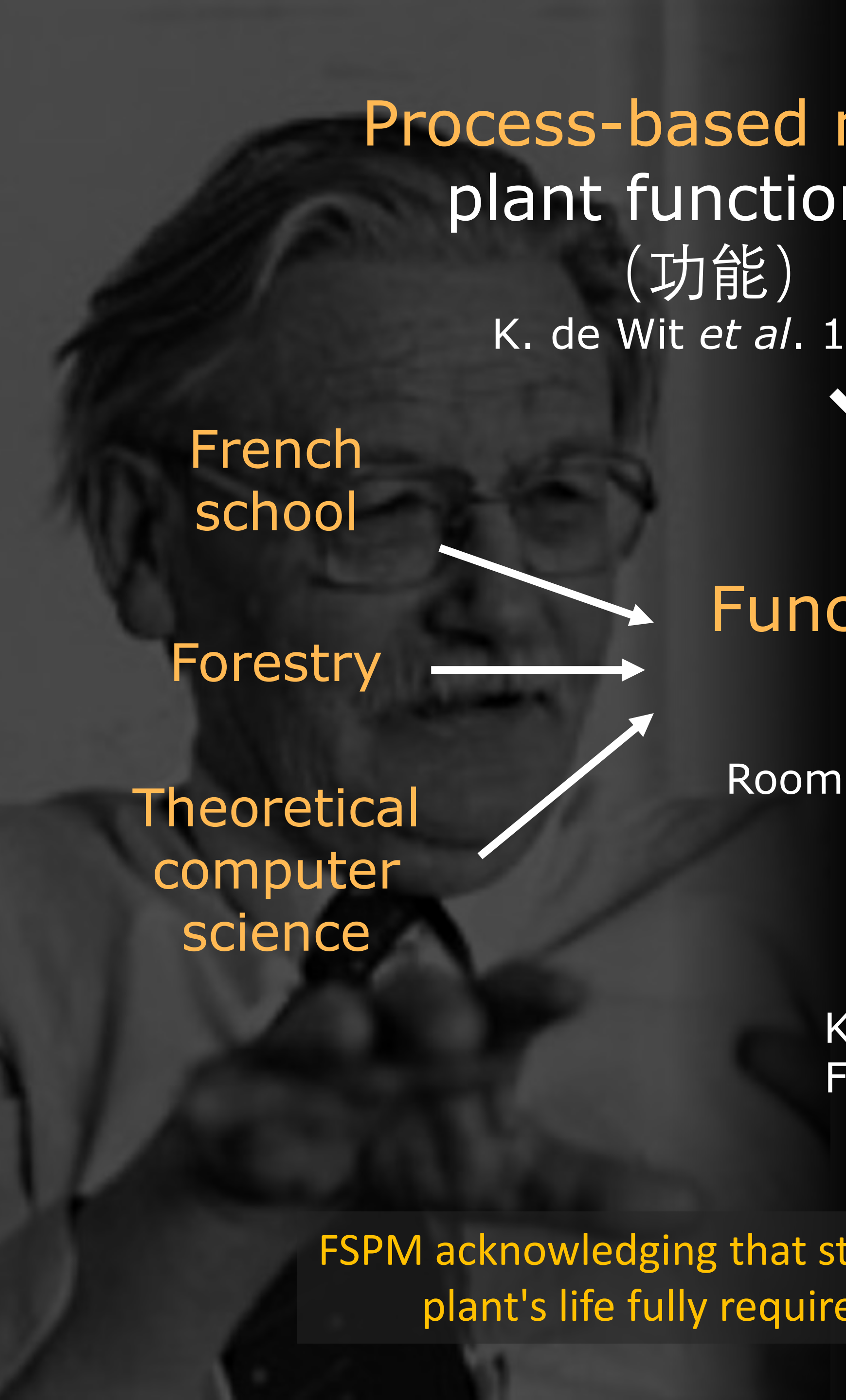
Functional-Structural Plant Models
（功能一结构植物模型 FSPM）

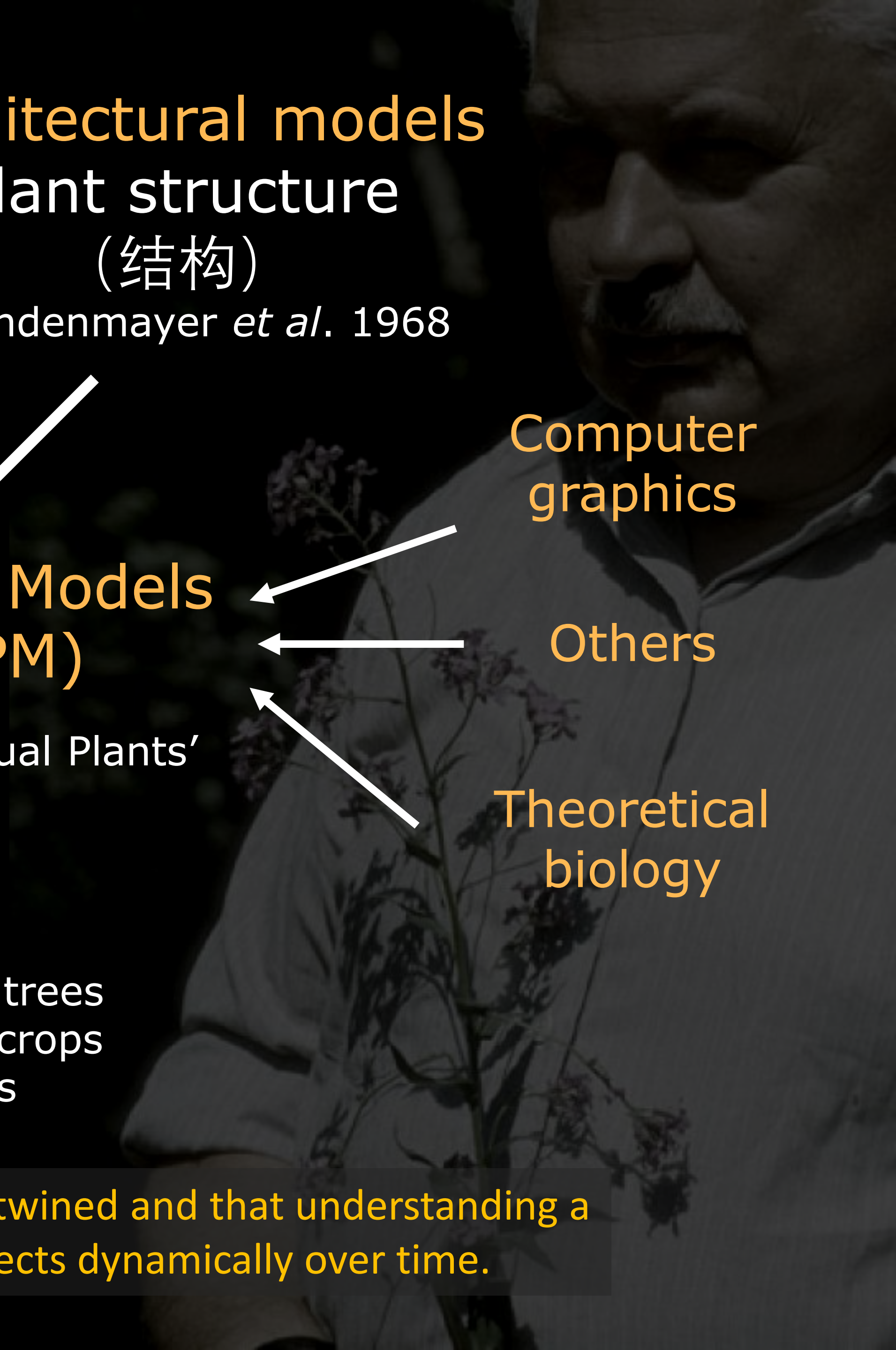
Room, Hanan, Prusinkiewicz 1996: ‘Virtual Plants’

Kurth 1994, Perttunen *et al*. 1996: trees
Fournier and Andrieu 1998: cereal crops
Pagès *et al.* 1994: root systems

FSPM acknowledging that structure and function are deeply intertwined and that understanding a plant's life fully requires a model that can simulate both aspects dynamically over time.

Based on slides by J. Evers, WUR





Process-based models
plant functioning
（功能）
K. de Wit *et al*. 1965

Architectural models
plant structure
（结构）
A. Lindenmayer *et al*. 1968

French school

Forestry

Theoretical computer science

Computer graphics

Others

Theoretical biology

Functional-Structural Plant Models
(功能一结构植物模型 FSPM)

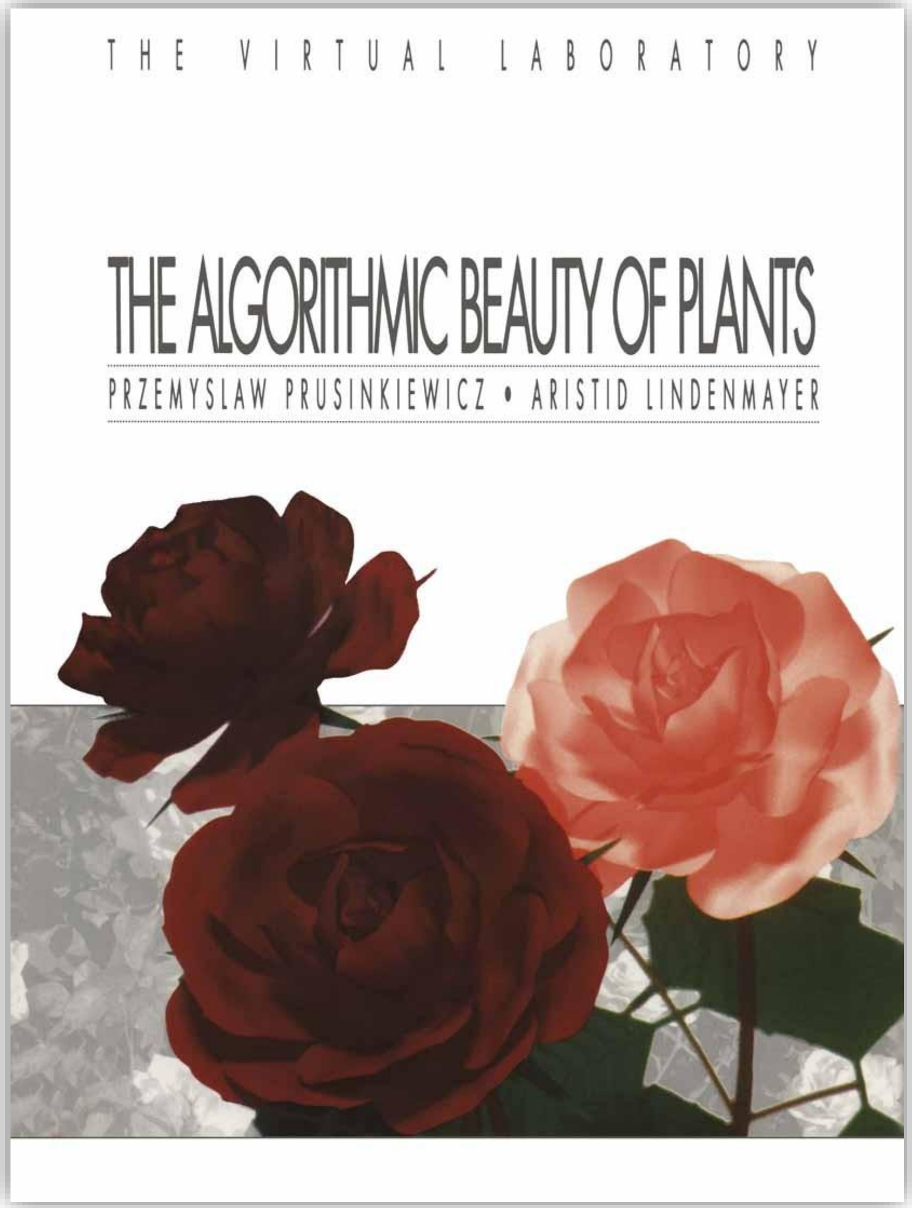
Room, Hanan, Prusinkiewicz 1996: 'Virtual Plants'

Kurth 1994, Perttunen *et al*. 1996: trees
Fournier and Andrieu 1998: cereal crops
Pagès *et al.* 1994: root systems

FSPM acknowledging that structure and function are deeply intertwined and that understanding a plant's life fully requires a model that can simulate both aspects dynamically over time.

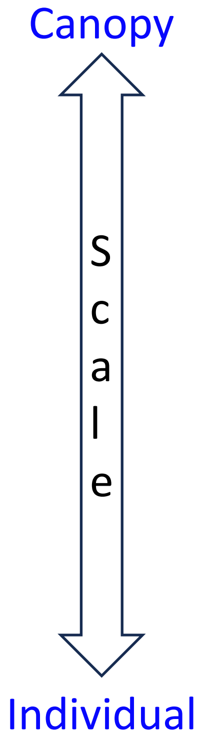
Further reads

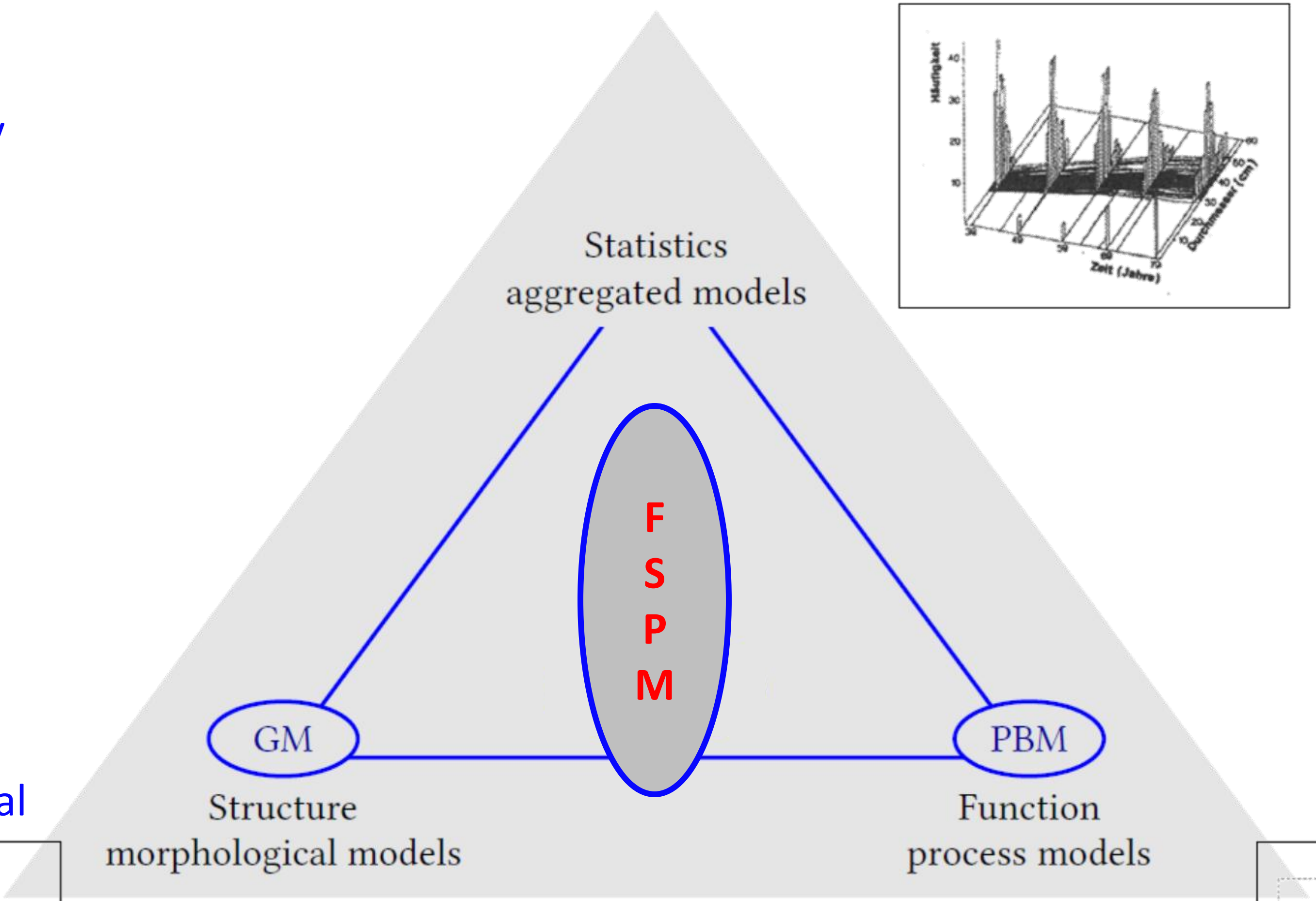
"The Algorithmic Beauty of Plants" by Prusinkiewicz and Lindenmayer (1990)

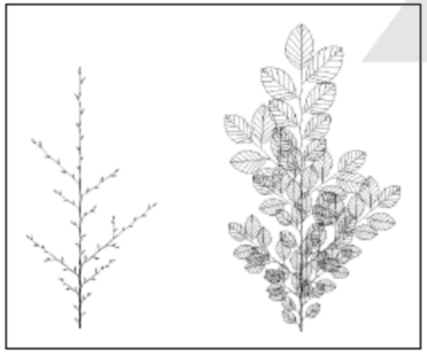


http://algorithmicbotany.org/papers/abop/abop.pdf

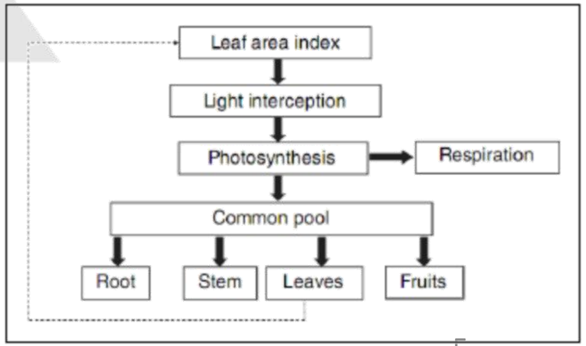
FSPM – Classification







Kurth (1999)



de Reffye *et al.* (2008)

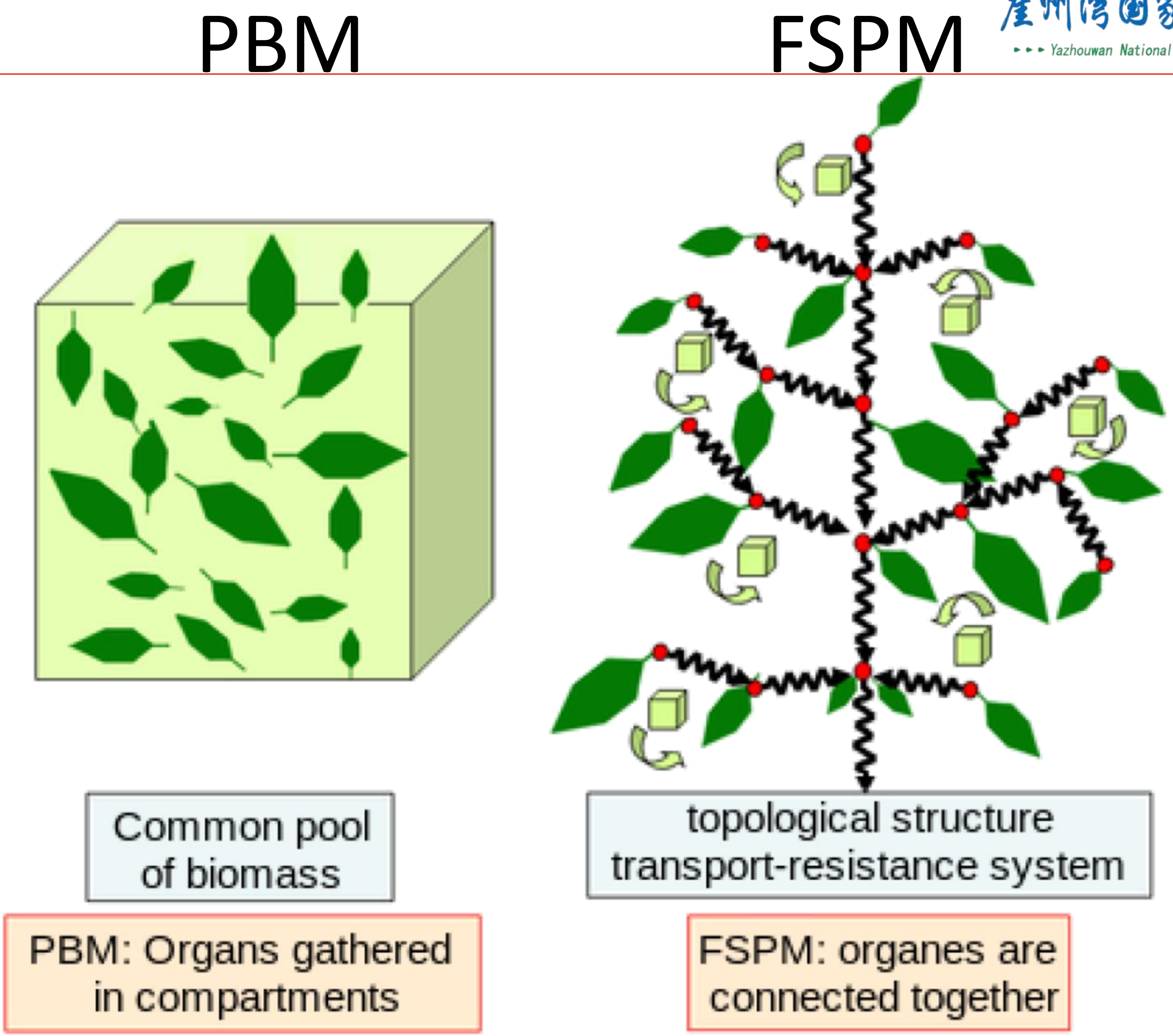
Extended model triangle of plant models
based on Kurth (1994b)

Source and sink principle

The organ compartments are usually limited to organ types in PBM, competing for a common biomass pool, while in FSPM, each organ is individualised.

The assimilates produced by local sources are transported in the direction of sinks according to their sink strength.

➔ Major drawback of PBMs have their origin in the total neglect or oversimplification of 3d plant architecture and its plasticity.



Drawing: de Reffye, CIRAD, http://greenlab.cirad.fr/GLUVED/html/P1_Prelim/Model/Model_FSPM_001.html

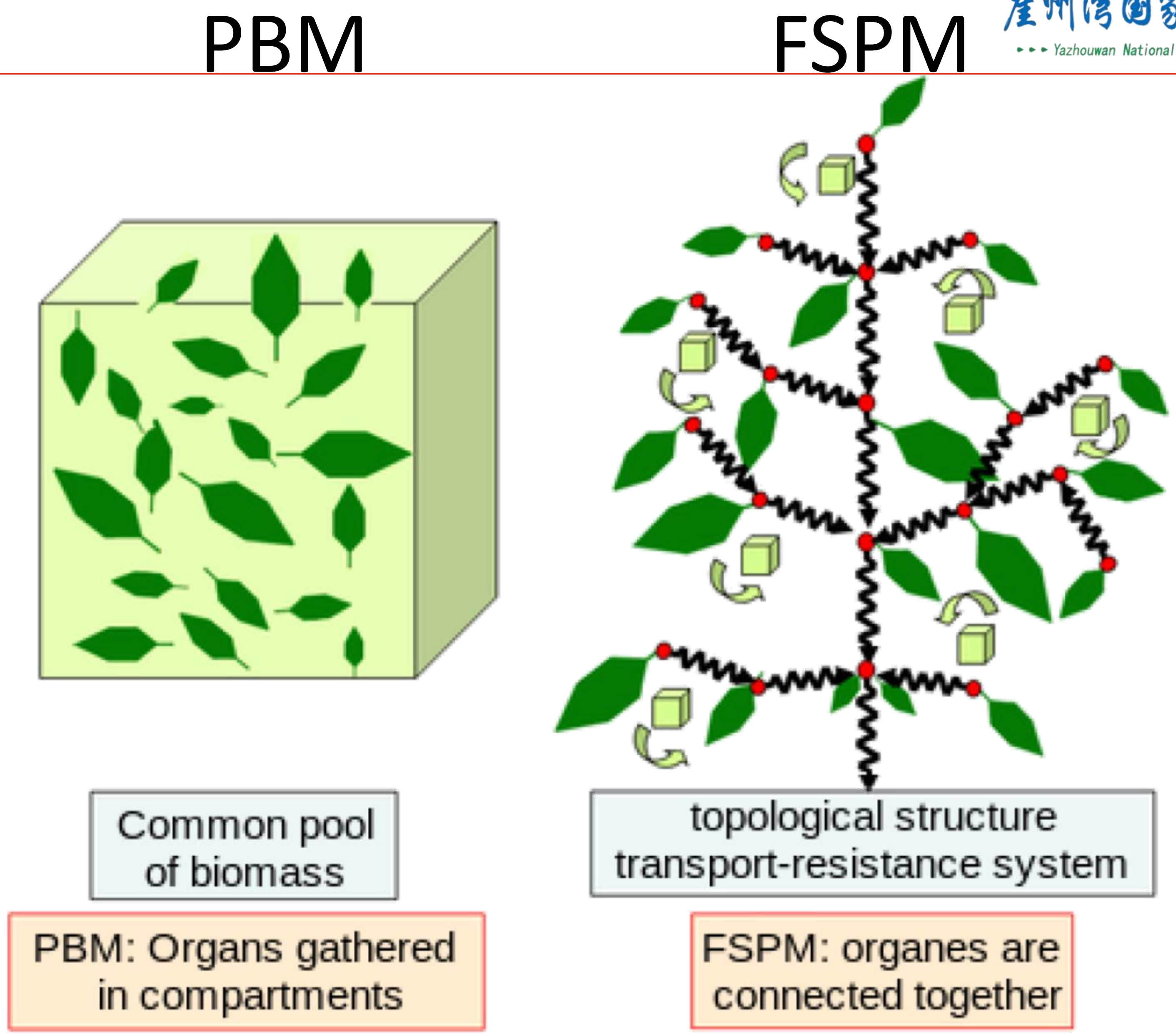
崖州湾国家实验室
Yazhouwan National Laboratory

Source and sink principle

The organ compartments are usually limited to organ types in PBM, competing for a common biomass pool, while in FSPM, each organ is individualised.

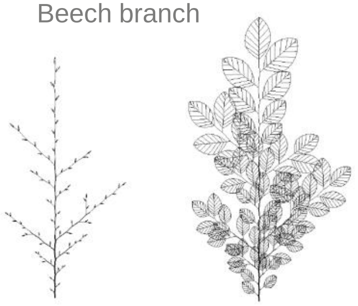
The assimilates produced by local sources are transported in the direction of sinks according to their sink strength.

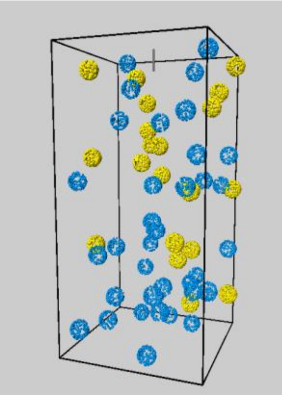
➔ Major drawback of PBMs have their origin in the total neglect or oversimplification of 3d plant architecture and its plasticity.



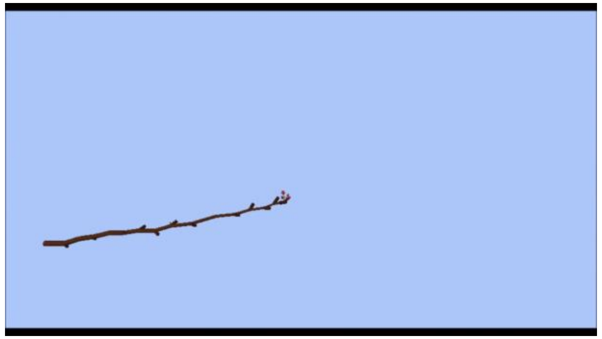
Drawing: de Reffye, CIRAD, http://greenlab.cirad.fr/GLUVED/html/P1_Prelim/Model/Model_FSPM_001.html

Functional-Structural Plant Models – Scales 规模影响因素算力与研究目标


Kurth 1999
Beech branch


Henke *et al.* 2009
Root plasticity

Organ level
器官

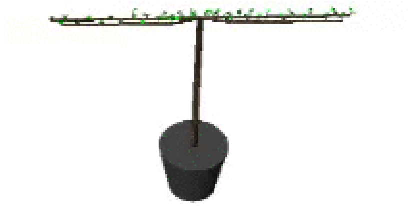

Buck-Sorlin *et al.* 2009
Apple branch and fruit

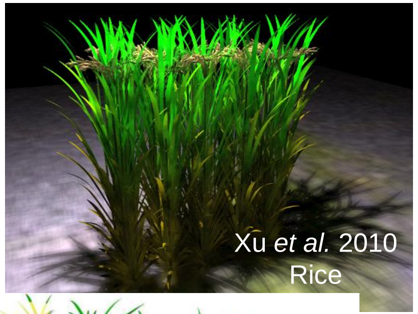

Evers *et al.* 2009
Arabidopsis

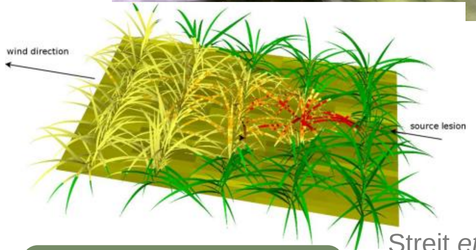

Groer *et al.* 2009
Rapeseed Oil

Individual plants
单株植物


Buck-Sorlin *et al.* 2009
Cut Rose


Zhu *et al.* 2015
Grape Wine


Xu *et al.* 2010
Rice



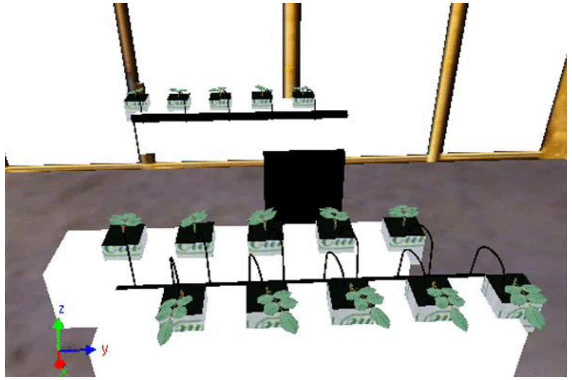
Streit *et al.* 2017
Wheat

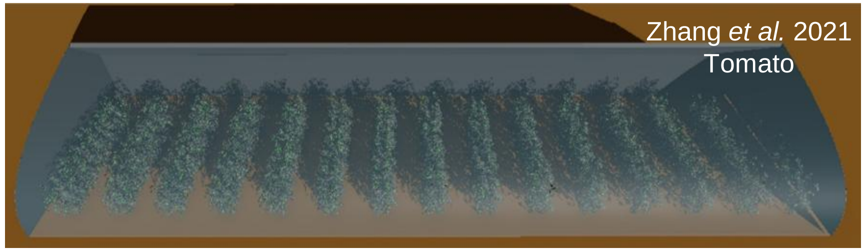
Small plant stands
小群体

Evers *et al.* 2011
Wheat


Zhu 2015
Maize and wheat intercropping

Canopy level
冠层群体


Buck-Sorlin *et al.* 2009
Cut Rose


Zhang *et al.* 2021
Tomato

Henke, Shanghai, 2025 Oct. 29

Functional-Structural Plant Models – Components (1/7) 重构植物结构

Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Model Parameterization
Validation
Data acquisition
Phenotyping

Modelling Environment
Simulation platform

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)

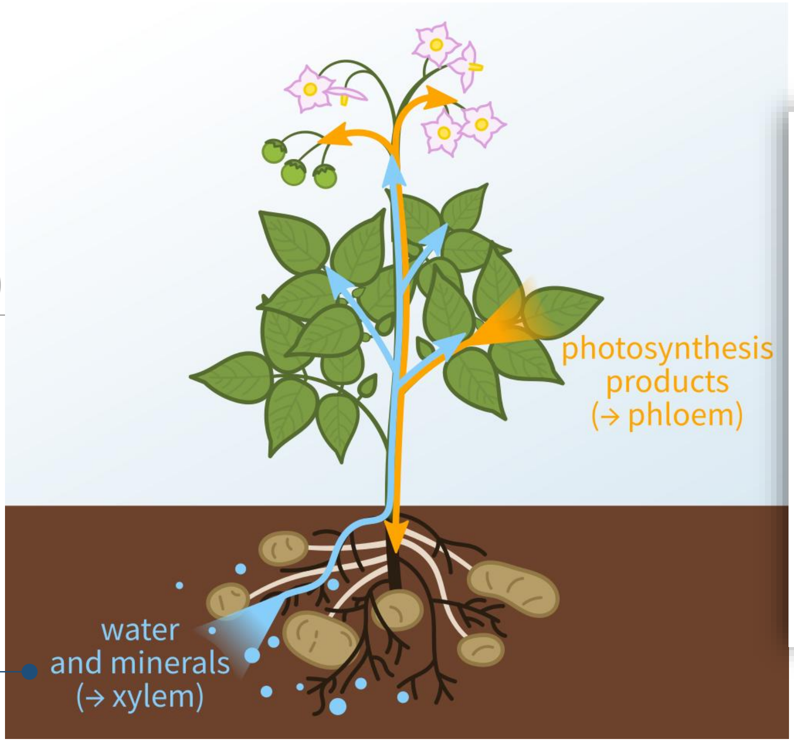
Genetic Control
'Virtual Breeder'
Phenotype prediction

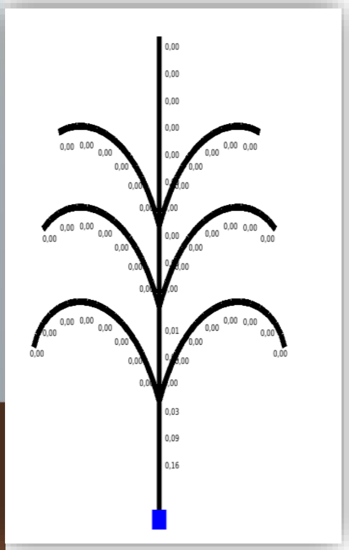
Functional-Structural Plant Models – Components (2/7) 模拟生理功能

Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

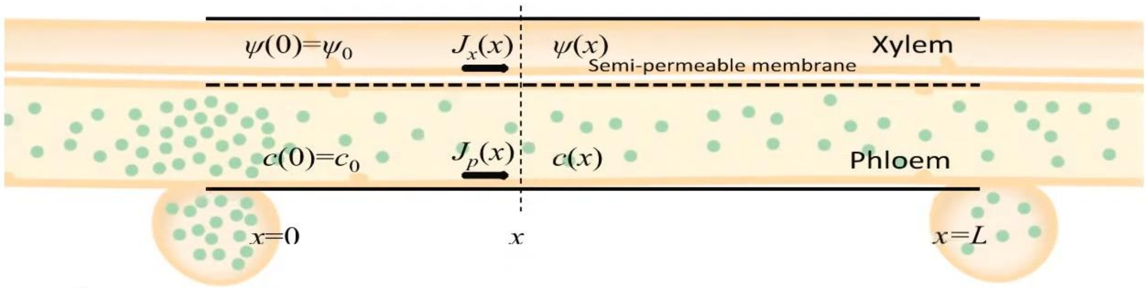
Model Parameterization
Validation
Data acquisition
Phenotyping





Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)



$$\begin{cases} J_x(x) = -\dfrac{1}{R_x}\dfrac{d\psi(x)}{dx} \\ J_p(x) = -\dfrac{1}{R_p(c(x))}\dfrac{d}{dx}(\psi(x)-\Pi(c(x))) \end{cases}$$

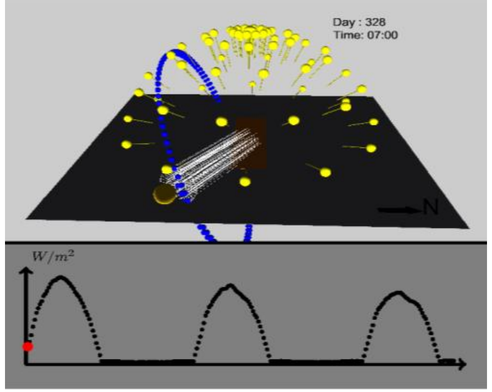
$$\begin{cases} J_{tot} = J_x(x) + J_p(x) = const \\ J_s = c(x)J_p(x) = const \end{cases}$$

Simulation platform

Genetic Control
'Virtual Breeder'
Phenotype prediction

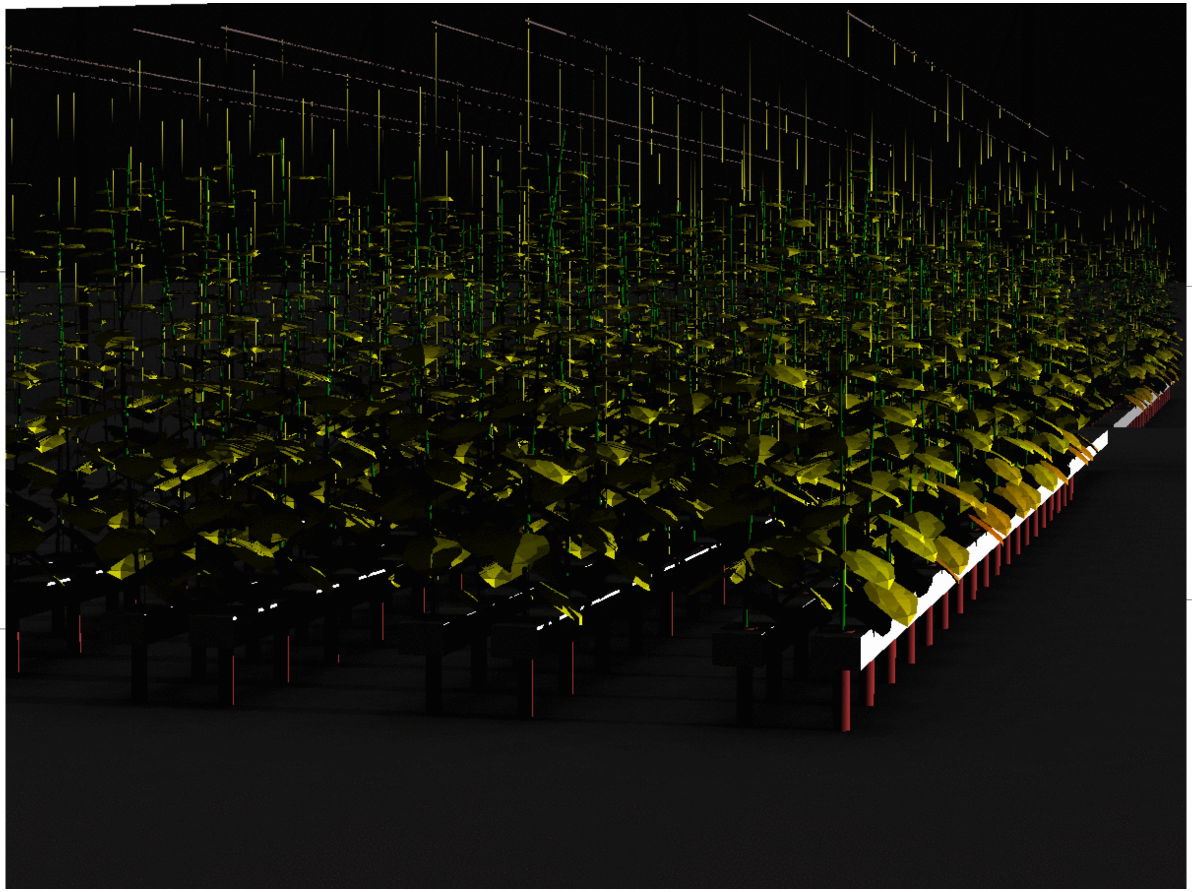
Functional-Structural Plant Models – Components (3/7) 模拟环境交互

崖州湾国家实验室
Yazhouwan National Laboratory



Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources



Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Growth and Development
Organ formation & extension
Allocation (sinks)

Model Parameterization
Validation
Data acquisition
Phenotyping

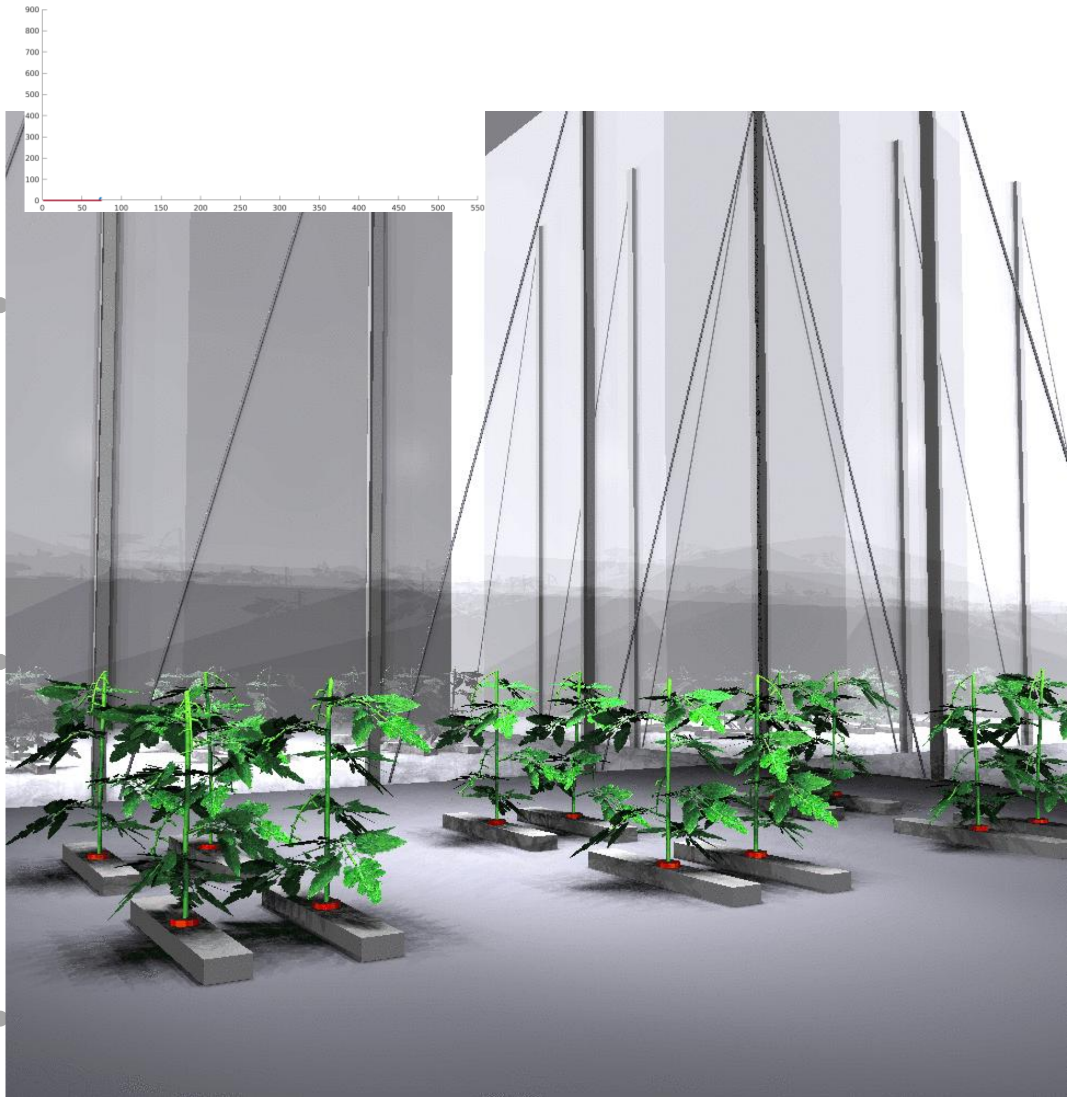
Genetic Control
'Virtual Breeder'
Phenotype prediction

Functional-Structural Plant Models – Components (4/7) 生长发育模式

Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Model Parameterization
Validation
Data acquisition
Phenotyping



Modelling Environment
Simulation platform

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)

Genetic Control
'Virtual Breeder'
Phenotype prediction

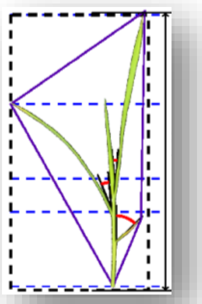
Functional-Structural Plant Models – Components (5/7) 设置模型参数

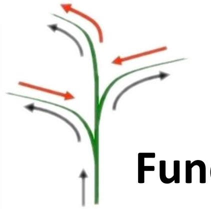
Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Model Parameterization
Validation
Data acquisition
Phenotyping

Structural
Architecture
Developmental
Plasticity



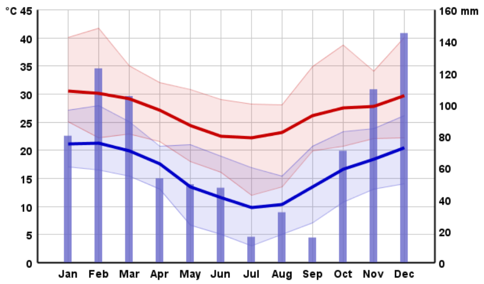


Functional
Physiological processes
Biomass allocation
Growth Curves

DATA

Genotypic

Environmental
Climate
Soil
Stressors



Experimental
Management
Treatment
Greenhouse

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)

Genetic Control
'Virtual Breeder'
Phenotype prediction

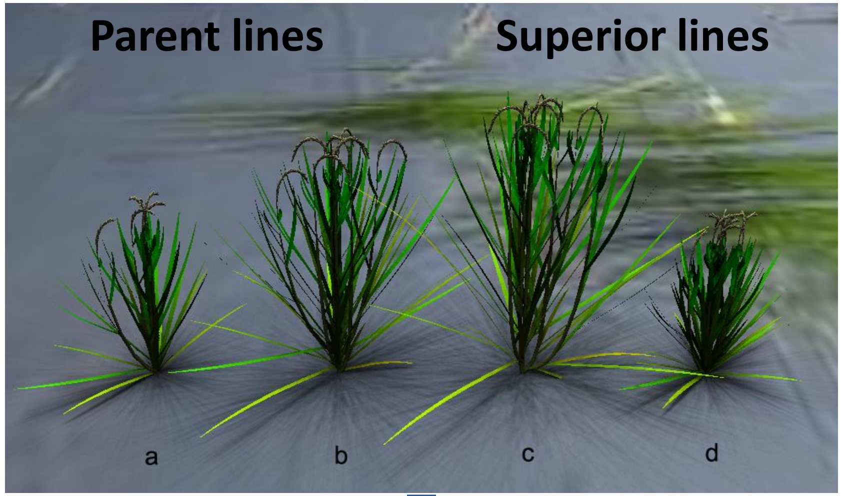
Modelling Environment
Simulation platform

Functional-Structural Plant Models – Components (6/7) 基因控制参数

Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Model Parameterization
Validation
Data acquisition
Phenotyping



Simulated population with different QTL genotype for plant height

Xu *et al.* 2012

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)

Genetic Control
'Virtual Breeder'
Phenotype prediction

Integrate genetics into FSPM
G×P×E plant model

Functional-Structural Plant Models – Components (7/7) 模型运行环境

Architecture Modelling
Plant morphology
Lindenmayer-System (L-System)

Physiological Processes
Uptake & Transport
Photosynthesis (sources)
Leaf-gas exchange

Model Parameterization
Validation
Data acquisition
Phenotyping

Environmental Interactions
Light modelling
Sun & Sky
Artificial light sources

Growth and Development
Organ formation & extension
Allocation (sinks)

Genetic Control
'Virtual Breeder'
Phenotype prediction

Modelling Environment
Simulation platform

Interactive Modelling Platform GroIMP 交互式3D建模平台

崖州湾国家实验室
Yazhouwan National Laboratory

- **Developed by the group Prof. Kurth, Göttingen, Germany**
- **2021 20th Anniversary**
- **Numerous PhD, master and bachelor thesis**
- **Over 680 hits on Google Scholar**
- **Free & Open-source**
- **Continuously maintained and further developed**

GroIMP WeChat Group

Valid until 11/5

Hosted on GitLab: https://gitlab.com/grogra/groimp

Selected User
著名用户代表

FSPM – Orientations and Application areas

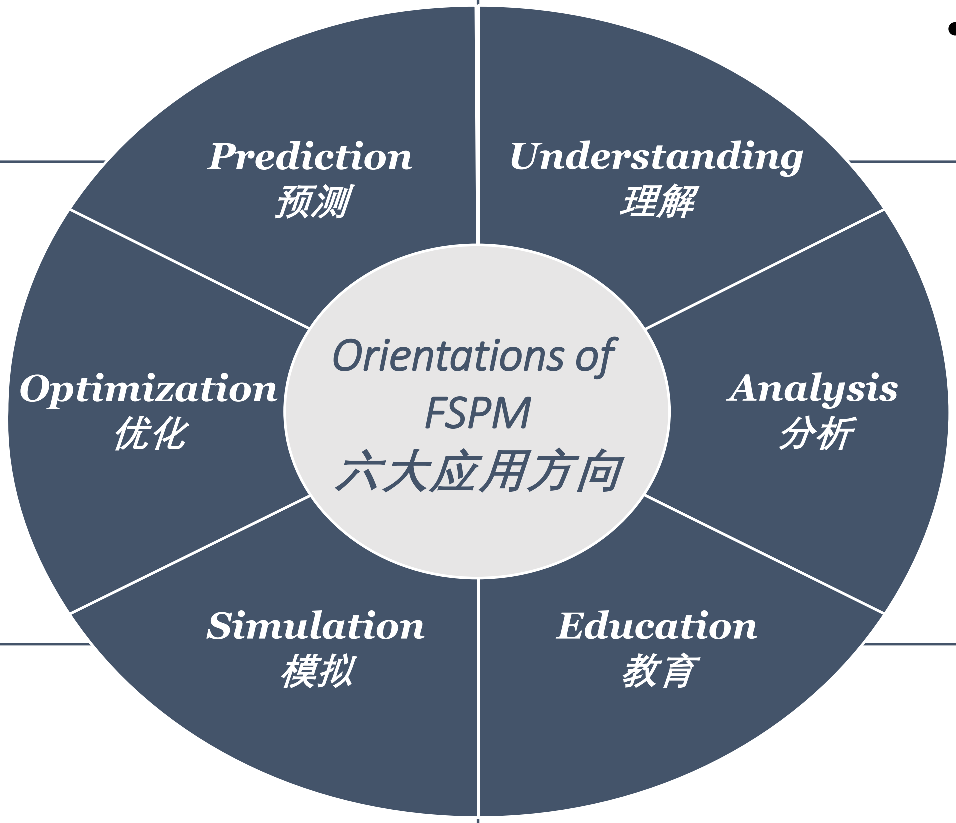
崖州湾国家实验室
Yazhouwan National Laboratory

Agriculture （农学）
- Optimize crop yield
- Plant architecture
- Planting strategies
- Management tactics

Breeding （育种）
- Output prediction
- Trait selection
- Identifying desirable traits

Plant Biology （生物学）
- Fundamental research,
- Exploration of processes
- Phenotypic plasticity
- Monitoring and reporting

Horticulture （园艺学）
- Design of desired aesthetic qualities
- Design of desired growth characteristics, such as compactness
- Design of desired branching patterns



Urban Planning （城市农业）
- Design green spaces that optimize plant growth, contribution to urban climate

Ecology （生态学）
- Interactions between plants and their environment
- Competition for resources, plant-animal interactions
- Dynamics of plant communities

Forestry （林业学）
- Simulate forest stand dynamics
- Simulate management strategies

Education （教育学）
- Teaching about plant biology and the complexity of plant-environment interactions

FSPM – Orientations and Application areas

崖州湾国家实验室
Yazhouwan National Laboratory

Agriculture （农学）
- Optimize crop yield
- Plant architecture
- Planting strategies
- Management tactics

Breeding （育种）
- Output prediction
- Trait selection
- Identifying desirable traits

Plant Biology （生物学）
- Fundamental research,
- Exploration of processes
- Phenotypic plasticity
- Monitoring and reporting

Invaluable in any field where **understanding** and **predicting** plant growth is crucial.

可被应用到任何涉及**理解**和**预测**植物生长发育相关的研究领域

特别是 Key technology for advancing fields like:
smart agriculture 智能农业,
greenhouse management 温室管理,
decision support 决策系统,
etc.

Horticulture （园艺学）
- Design of desired aesthetic qualities
- Design of desired growth characteristics, such as compactness
- Design of desired branching patterns

Urban Planning （城市农业）
- Design green spaces that optimize plant growth, contribution to urban climate

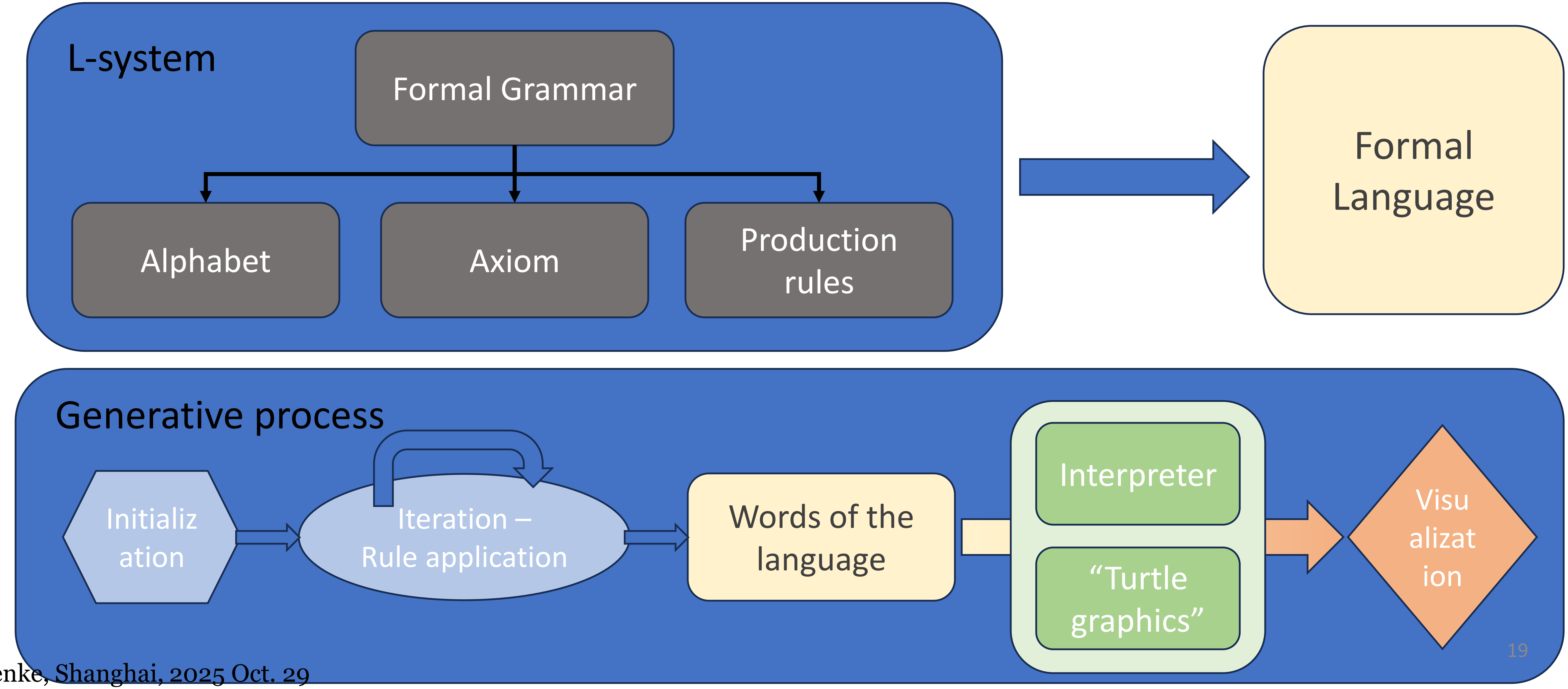
Ecology （生态学）
- Interactions between plants and their environment
- Competition for resources, plant-animal interactions
- Dynamics of plant communities

Education （教育学）
- Teaching about plant biology and the complexity of plant-environment interactions

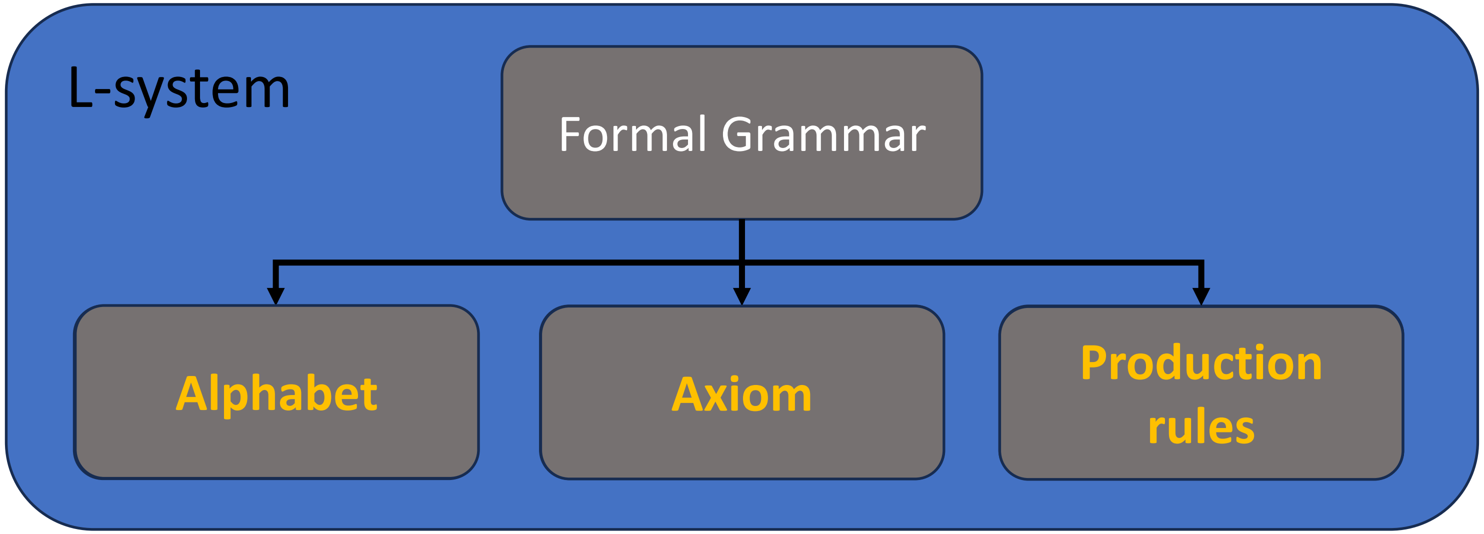
Forestry （林业学）
- Simulate forest stand dynamics
- Simulate management strategies

Lindenmayer-Systems – Definition

A Lindenmayer-system, or L-system, is a **parallel string rewriting system** and **formal grammar** used to generate complex structures and to model the growth of biological organisms, such as plants.



Lindenmayer-Systems – Definition



Alphabet: Σ, set of symbols (terminals and non-terminals)
Axiom: α, start symbol, element of the alphabet
Production rules: δ, set of rules
left-hand side → right-hand side
search pattern → replacement

Lindenmayer-Systems – Example

L-system

Formal Grammar

- Alphabet: $\Sigma = \{A, B, F, +, -, [,]\}$
- Axiom: $\alpha = A$
- Production rules: $\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \to F[-B][+B]A, \\ B \to FB \end{Bmatrix}$$

Production rules: δ, set of rules

left-hand side → right-hand side

search pattern → replacement

$$A \to F[-B][+B]A$$

- Parallel string rewriting system
- Matching & rewriting:
  searching for all occurrences of the search pattern, and simultaneously replacing them by the right-hand side of the rule
- Iterative (production/generation) process
- Starting from the initial state, the axiom α

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$$\Sigma = \{A, B, F, +, -, [,]\}$$

$$\alpha = A$$

$$\delta = \left\{ \begin{matrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{matrix} \right\}$$

matching & rewriting

Step		String
0	α =	A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$$\Sigma = \{A, B, F, +, -, [,]\}$$

$$\alpha = A$$

$$\delta = \left\{ \begin{matrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{matrix} \right\}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

崖州湾国家实验室
Yazhouwan National Laboratory

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \left\{ \begin{matrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{matrix} \right\}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
4		F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

崖州湾国家实验室
Yazhouwan National Laboratory

L-system $G = \{\Sigma, \alpha, \delta\}$

$$\Sigma = \{A, B, F, +, -, [,]\}$$

$$\alpha = A$$

$$\delta = \left\{ \begin{matrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{matrix} \right\}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
4		F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

崖州湾国家实验室 Yazhouwan National Laboratory

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \left\{ \begin{matrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{matrix} \right\}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
4		F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
5		F[-FFFFB][+FFFFB]F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

$\Sigma = \{A, B, F, +, -, [,]\}$

$\alpha = A$

$$\delta = \begin{Bmatrix} A \to F[-B][+B]A, \\ B \to FB \end{Bmatrix}$$

matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
4		F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
5		F[-FFFFB][+FFFFB]F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
…		…

- Parallel string rewriting system
- Matching & rewriting:
  searching for all occurrences of the search pattern, and simul-taneously replacing them by the right-hand side of the rule
- Iterative (production/generation) process
- Starting from the initial state, the axiom α

Lindenmayer-Systems – Rule application / Derivation steps

L-system $G = \{\Sigma, \alpha, \delta\}$

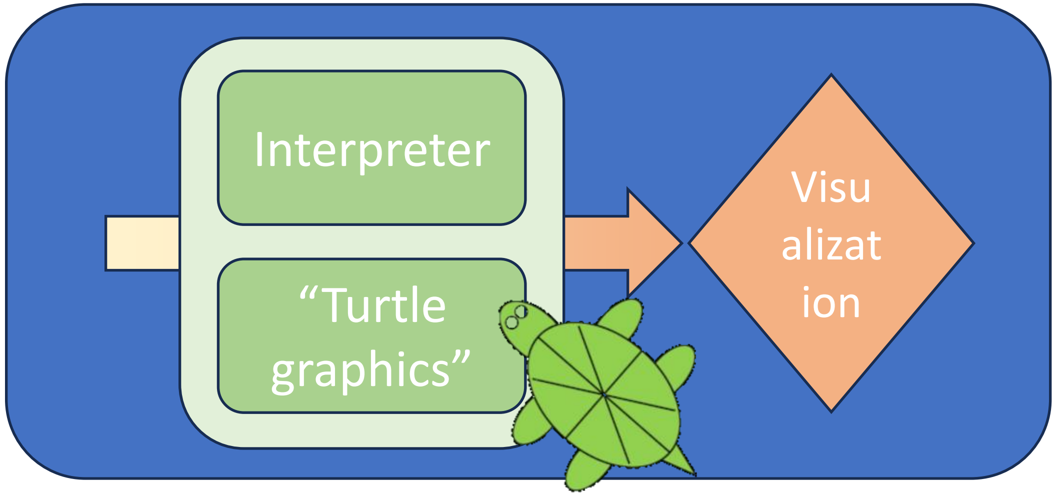
$$\Sigma = \{A, B, F, +, -, [,]\}$$

$$\alpha = A$$

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

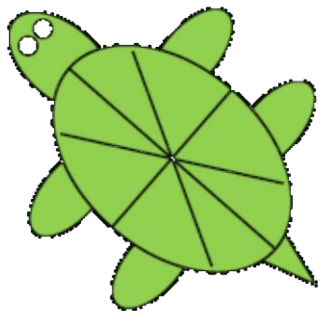
matching & rewriting

Step		String
0	α =	A
1		F[-B][+B]A
2		F[-FB][+FB]F[-B][+B]A
3		F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
4		F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
5		F[-FFFFB][+FFFFB]F[-FFFB][+FFFB]F[-FFB][+FFB]F[-FB][+FB]F[-B][+B]A
…		…



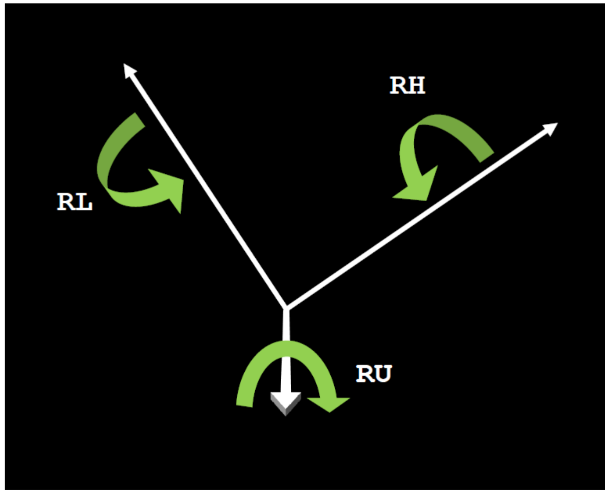
Turtle graphics

- Turtle = **Interpreter**, that listens to commands and executes them, e.g., interprets them as geometry → **visualization**
- **Virtual pen**, that can be moved, oriented, and draw
- Turtle has three properties:
  - a position (x, y, z)
  - an orientation (roll, pitch, yaw)
  - a pen in its tail with attributes such as colour and line width

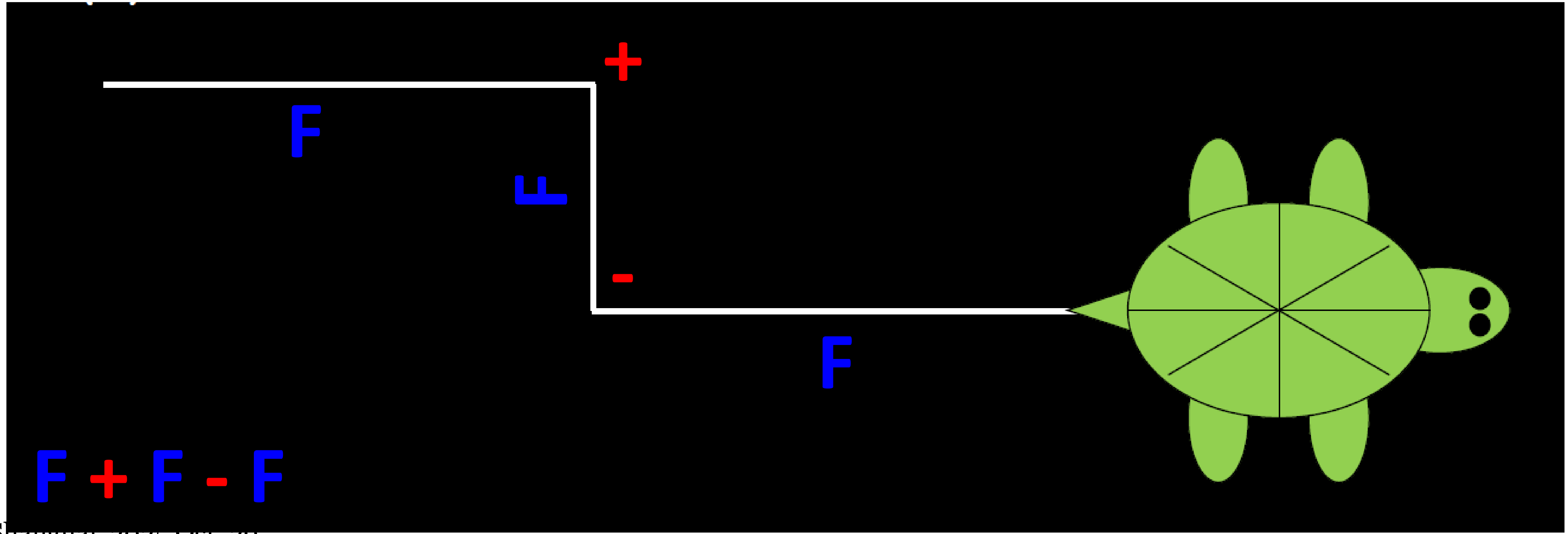


Wikipedia: https://en.wikipedia.org/wiki/Turtle_graphics

Turtle graphics



- Turtle = **Interpreter**, that listens to commands and executes them, e.g., interprets them as geometry → **visualization**
- **Virtual pen**, that can be moved, oriented, and draw
- Turtle has three properties:
  - a position (x, y, z)
  - an orientation (roll, pitch, yaw)
  - a pen in its tail with attributes such as colour and line width

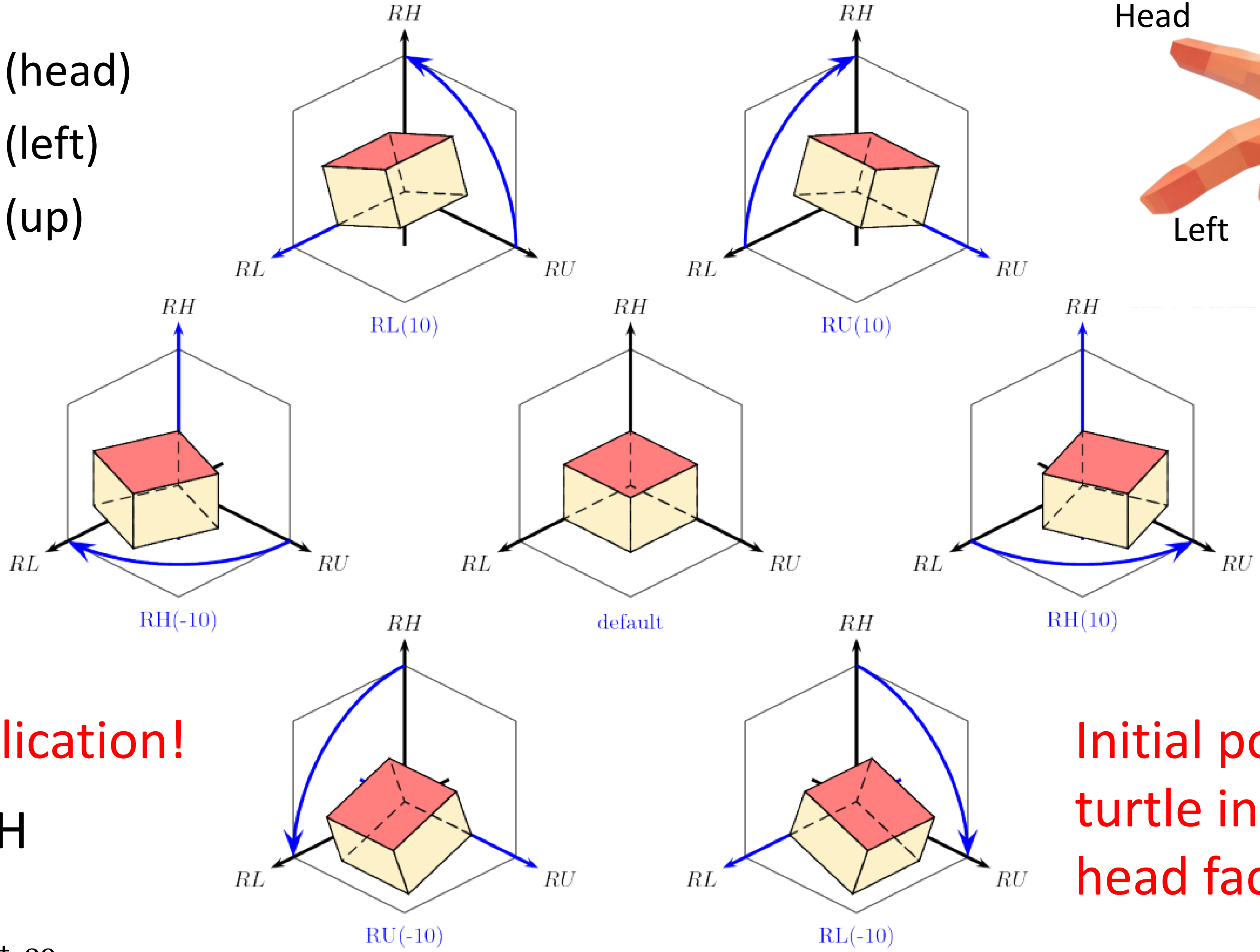


Turtle graphics – Euler angles

Roll – RH (head)

Pitch – RL (left)

Yaw – RU (up)



Order of application!

RH RL ≠ RL RH

Initial position of the turtle in GroIMP is head facing the Z-axis!

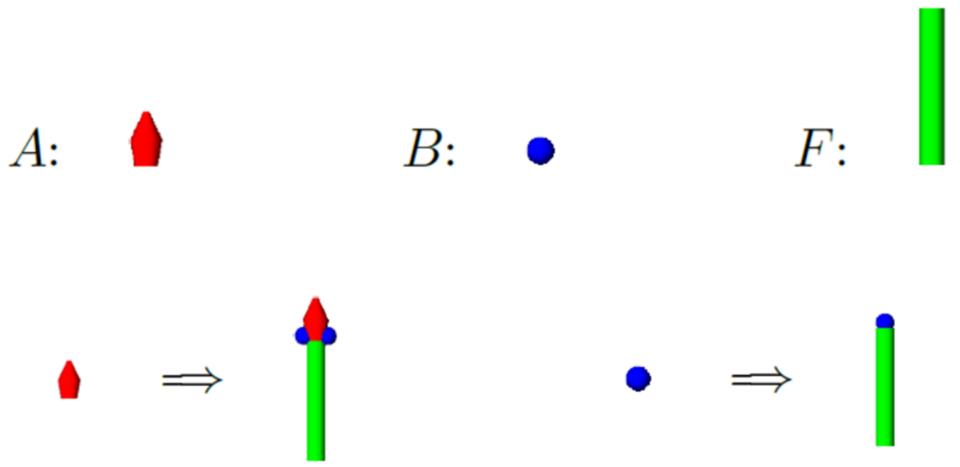
Turtle graphics – Geometric interpretation – Example

Alphabet

$$\Sigma = \{A, B, F, +, -, [,]\}$$

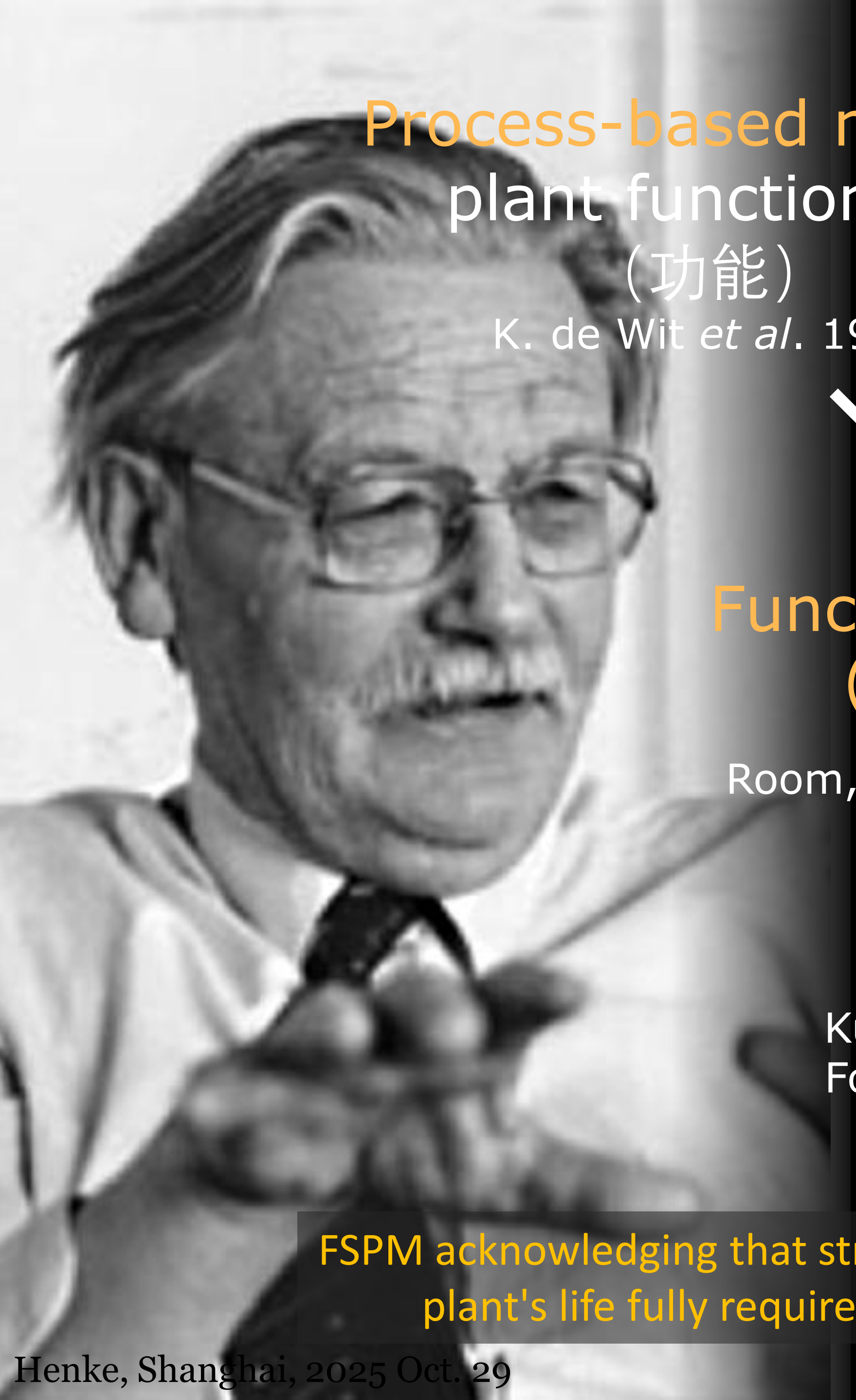
Rules

$$\delta = \begin{Bmatrix} A \to F[-B][+B]A, \\ B \to FB \end{Bmatrix}$$



[Open a branch: remember the position

] Close a branch: jump back to the position where the branch was opened



Process-based models
plant functioning
（功能）
K. de Wit *et al*. 1965

Architectural models
plant structure
（结构）
A. Lindenmayer *et al*. 1968

Functional-Structural Plant Models
（功能一结构植物模型 FSPM）

Room, Hanan, Prusinkiewicz 1996: 'Virtual Plants'

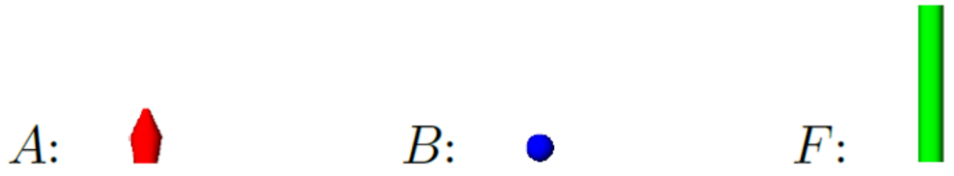
Kurth 1994, Perttunen *et al*. 1996: trees
Fournier and Andrieu 1998: cereal crops
Pagès *et al.* 1994: root systems

FSPM acknowledging that structure and function are deeply intertwined and that understanding a plant's life fully requires a model that can simulate both aspects dynamically over time.

Turtle graphics – Geometric interpretation – Example

Alphabet

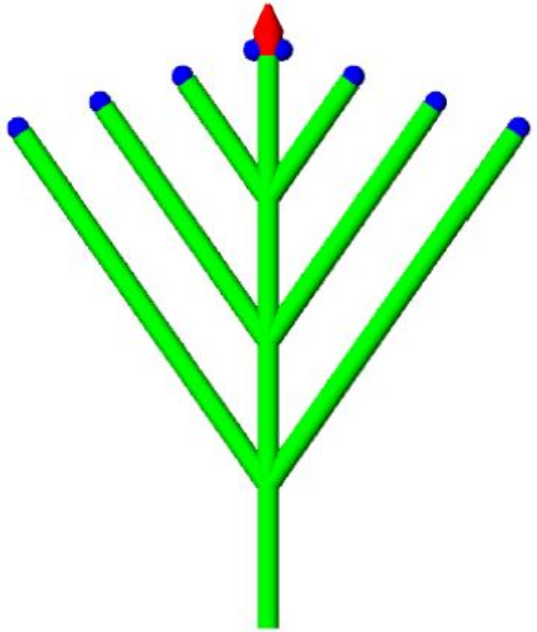
$$\Sigma = \{A, B, F, +, -, [,]\}$$



Rules

$$\delta = \begin{Bmatrix} A \rightarrow F[-B][+B]A, \\ B \rightarrow FB \end{Bmatrix}$$

Step		String
0	α =	A
1		
2		
3		
4		
5		
…		



Turtle graphics – Turtle Commands

Basic turtle commands:

M	Move forward default distance
F	Go forward default distance and draw a line of default length and width
F(x)	Go forward distance x and draw a line of length x and default width
F(x, y)	Go forward distance x and draw a line of length x and width y
RU(x)	Rotate x degrees around the up axis
RL(x)	Rotate x degrees around the left axis
RH(x)	Rotate x degrees around the head axis
+	**Turn left by n/90 degree (2D)**
-	Turn right by n/90 degree (2D)
[	Open a branch, **remember the position**
]	Close a branch, jump back to the position where the branch was opened

XL turtle geometry and graph construction

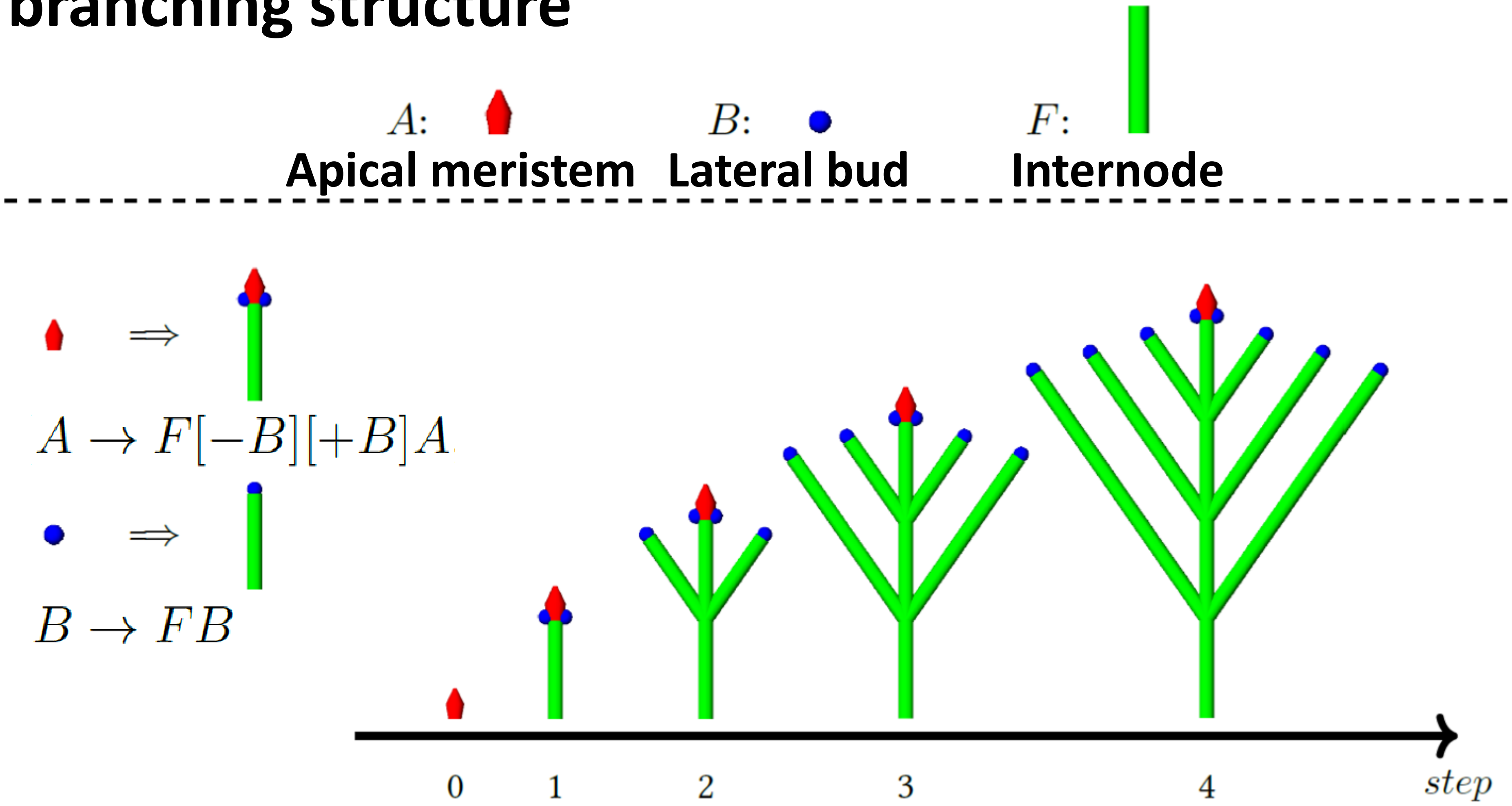
https://wiki.grogra.de/doku.php?id=tutorials:xl-turtle-geometry

List of Turtle commands

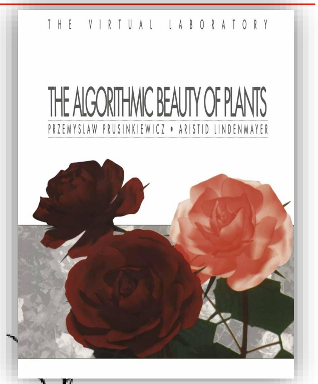
https://wiki.grogra.de/doku.php?id=groimp-platform:turtle_commands

Associating symbols with “real-world”, e.g., biological-related meanings.

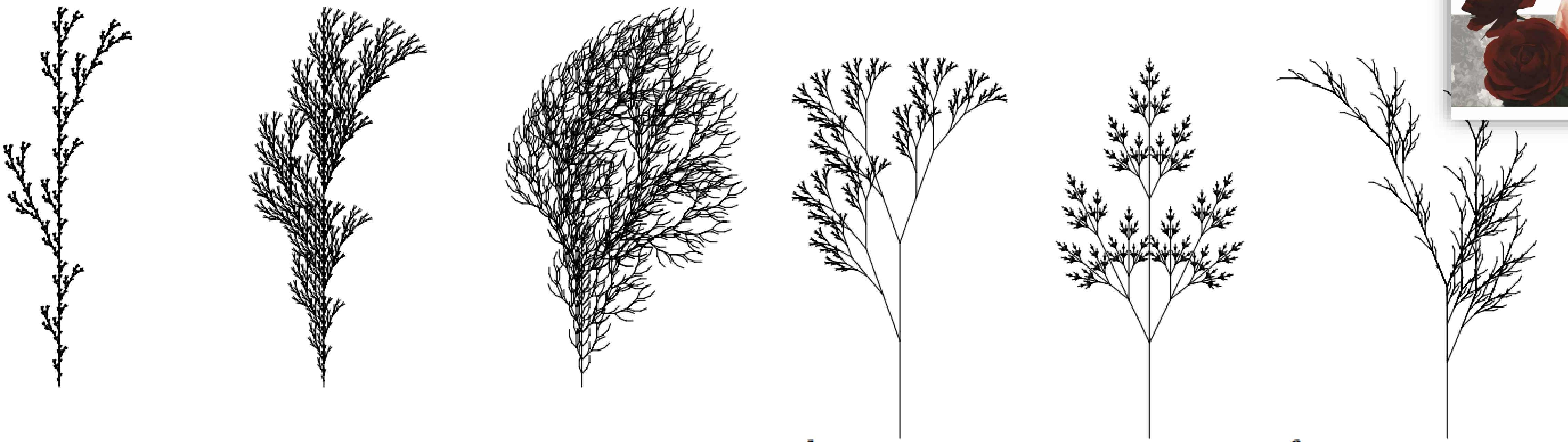
Simple branching structure



Examples



"The Algorithmic Beauty of Plants" by Prusinkiewicz and Lindenmayer (1990)



a
n=5, δ=25.7°
F
F → F[+F]F[-F]F

b
n=5, δ=20°
F
F → F[+F]F[-F][F]

c
n=4, δ=22.5°
F
F → FF-[-F+F+F]+[+F-F-F]

d
n=7, δ=20°
X
X → F[+X]F[-X]+X
F → FF

e
n=7, δ=25.7°
X
X → F[+X][-X]FX
F → FF

f
n=5, δ=22.5°
X
X → F-[[X]+X]+F[+FX]-X
F → FF

P. 25, Figure 1.24: Examples of plant-like structures generated by bracketed OLsystems. L-systems…

http://algorithmicbotany.org/papers/abop/abop.pdf

Context-sensitive L-Systems

- Restricting rule application to the context of the symbol (within the word).
- The targeted symbol only replaced, it the context if fullfilled.

Example rule:

B<A>C → AA

Transform "**A**" to "**AA**", but only if the "**A**" occurs between "**B**" and "**C**" in the input word/sting.

- Adding a list of parameters to the symbols
- Parameters could be, e.g., stings, numbers, etc.
- Symbols coupled with parameter are called ***modules***

Production rules can use the parameters in two ways:

1. as conditional statement determining whether the rule will apply
2. to modify the actual parameters

Example rule:

B(x), x<5 → FB(x+1)

Bud break conditions:
Only if condition is met, a new phytomer is built.

Transform "**B**" to "**FB**", but only if "**x<5**" (condition), and initialze the new "**B**" with "**x+1**" (parameter update).

Conditions to control rule application

```
b:Bud(rank, order, indiID), (b.isGrowingConditions()) ==>
  RV(-0.15)
  Internode(rank, order, indiID)
  [Leaf(rank, order, indiID)]
  if(b.isBranchingConditions()) (
    [RL(30) Shoot0(indiID) Bud(rank+1, order+1, indiID)]
  )
  Rotate(random(-5,5), random(-5,5), PHYLLOTAX + random(-5, 5))
  Bud(rank+1, order, indiID);
```

Growing Conditions:

```
public boolean isGrowingConditions() {
  ...
  return
    // architectural conditions
    rank<11 && order<=2 &&
    // absorbed sensed power
    absorbedSensedPower &&
    // average SSR as growth regulation
    (indi.getAverageSourceSinkRatio()>0.66) &&
    // plastochron
    plastochron<=0 ||
    // average SSR as growth regulation: building new sinks,
    // if avgSSR is to big and the bud is old enough
    (indi.getAverageSourceSinkRatio()>MAX_AVERAGE_SOURCE_SINK_RATIO
         && tempsum>200);
}
```

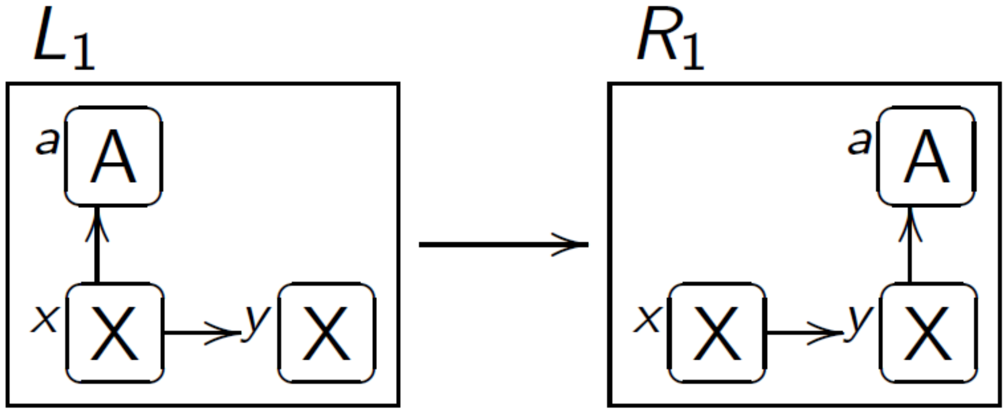
FSPM: Bud break conditions

- Age, temperature sum
- Temperature
- Hormone concentration, e.g., auxin
- Water content
- etc.

- L-systems are quite successful but based on plain strings.
- Relational growth grammars (RGG) combining L-systems and graphs
  ➔ Graph manipulation rules

Example rule (simplified):

x:**X[a:A][y:X]** ➔ **x[y[a]]**



Move the “**A**” from the frist “**X**” to the second “**X**”.

Lindenmayer-Systems – Summary

- Parallel string rewriting systems and formal grammar
- Iterative generation process
- Turtle graphics for visual interpretation
- Extensions
  - Context-sensitive grammars
  - Parametric grammars
  - RGG: Graph-based L-systems

Where to find more information?

崖州湾国家实验室
Yazhouwan National Laboratory

GroIMP Wiki pages:

https://wiki.grogra.de

- User guides
- Tutorials
- Example Gallery
- Developer Guides
- …
- FAQ
- Where to find Information
- Workshops

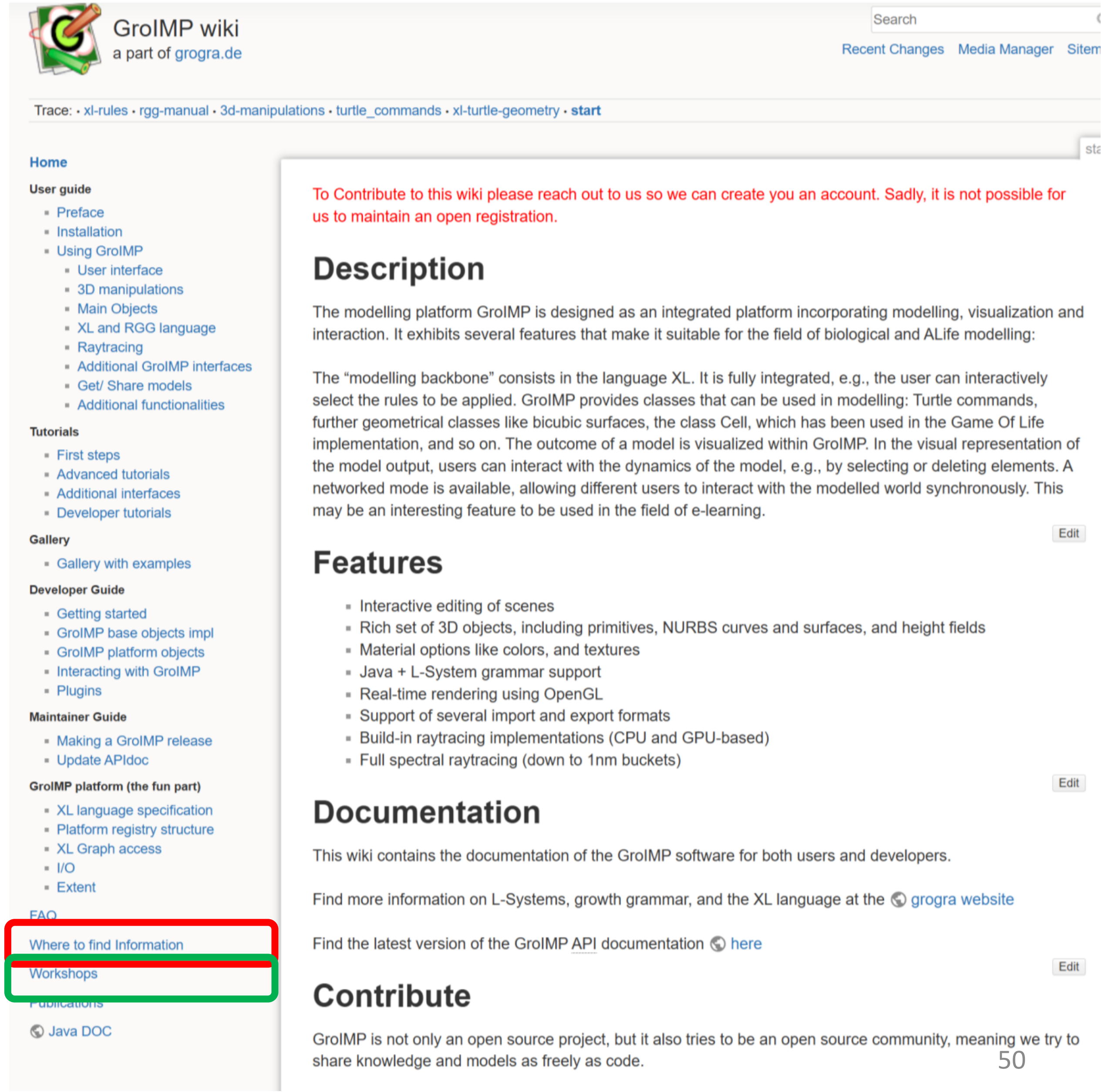
workshops:summer_school_sh_25

Sion-Europe summer workshop on systems modelling of plant factory for excellent young scientist

GroIMP at the 'Sion-Europe summer workshop on systems modelling of plant factory for excellent young scientist' Oct 27~31, 2025, Shanghai, China (link).

Table of Contents

- Sion-Europe summer workshop on systems modelling of plant factory for excellent young scientist
- Technical requirements
  - GroIMP
- Discussion



GroIMP wiki
a part of grogra.de

Search
Recent Changes Media Manager Sitem

Trace: • xl-rules • rgg-manual • 3d-manipulations • turtle_commands • xl-turtle-geometry • **start**

Home

User guide
- Preface
- Installation
- Using GroIMP
  - User interface
  - 3D manipulations
  - Main Objects
  - XL and RGG language
  - Raytracing
  - Additional GroIMP interfaces
  - Get/ Share models
  - Additional functionalities

Tutorials
- First steps
- Advanced tutorials
- Additional interfaces
- Developer tutorials

Gallery
- Gallery with examples

Developer Guide
- Getting started
- GroIMP base objects impl
- GroIMP platform objects
- Interacting with GroIMP
- Plugins

Maintainer Guide
- Making a GroIMP release
- Update APIdoc

GroIMP platform (the fun part)
- XL language specification
- Platform registry structure
- XL Graph access
- I/O
- Extent

FAQ
Where to find Information
Workshops
Publications
Java DOC

To Contribute to this wiki please reach out to us so we can create you an account. Sadly, it is not possible for us to maintain an open registration.

Description

The modelling platform GroIMP is designed as an integrated platform incorporating modelling, visualization and interaction. It exhibits several features that make it suitable for the field of biological and ALife modelling:

The "modelling backbone" consists in the language XL. It is fully integrated, e.g., the user can interactively select the rules to be applied. GroIMP provides classes that can be used in modelling: Turtle commands, further geometrical classes like bicubic surfaces, the class Cell, which has been used in the Game Of Life implementation, and so on. The outcome of a model is visualized within GroIMP. In the visual representation of the model output, users can interact with the dynamics of the model, e.g., by selecting or deleting elements. A networked mode is available, allowing different users to interact with the modelled world synchronously. This may be an interesting feature to be used in the field of e-learning.

Edit

Features

- Interactive editing of scenes
- Rich set of 3D objects, including primitives, NURBS curves and surfaces, and height fields
- Material options like colors, and textures
- Java + L-System grammar support
- Real-time rendering using OpenGL
- Support of several import and export formats
- Build-in raytracing implementations (CPU and GPU-based)
- Full spectral raytracing (down to 1nm buckets)

Edit

Documentation

This wiki contains the documentation of the GroIMP software for both users and developers.

Find more information on L-Systems, growth grammar, and the XL language at the grogra website

Find the latest version of the GroIMP API documentation here

Edit

Contribute

GroIMP is not only an open source project, but it also tries to be an open source community, meaning we try to share knowledge and models as freely as code.

感谢观看
Thank you!

GroIMP WeChat Group

Valid until 11/5

Special thanks to GroIMP User Groups!

WAGENINGEN UNIVERSITY & RESEARCH

Jochem Evers
Katarína Smoleňová
Leo Marcelis
Pieter de Visser

Plant & Food RESEARCH RANGAHAU AHUMARA KAI

Zhu Junqi

IB-CAS

Dai Zhanwu

Leibniz Universität Hannover

Hartmut Stützel
Magnus Adler

IPK GATERSLEBEN

Evgeny Gladilin

浙江工業大學 ZHEJIANG UNIVERSITY OF TECHNOLOGY

Xu Lifeng

Winfried Kurth
Ole Kniemeyer
Gaëtan Heidsieck
GroIMP Developer Team

HUMBOLDT-UNIVERSITÄT ZU BERLIN

Chen Tsu-Wei

浙江大學 ZHEJIANG UNIVERSITY

Chen Ming

Hochschule Geisenheim University

Kathrin Kahlen
Christopher Bahr

UNIVERSITÄT HOHENHEIM

Sebastian Munz

AGRO CAMPUS OUEST

Gerhard Buck-Sorlin

THE UNIVERSITY OF WESTERN AUSTRALIA

Michel Renton

UNIVERSITEIT GENT

Kathy Steppe
Jonas Coussement

中国農業大學 China Agricultural University

Ma Yuntao
Zhang Lizhen

上海市農業科学院 SHANGHAI ACADEMY OF AGRICULTURAL SCIENCES

Qian Tingting