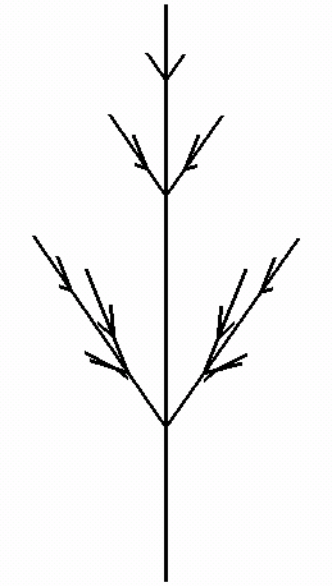
Modeling and visualization of plants in 3D

The purpose of this section is to introduce:

- Structure of L-systems that capture basic branching patterns: monopodial, sympodial, and polypodial,
- Extension of turtle geometry to 3D,
- Basic computer graphics techniques for viewing and rendering 3D objects,
- Modeling of plant organs using parametric surfaces,
- Incorporation of growth functions into developmental plant models,
- The L-studio editors of materials, surfaces and functions.

Planation

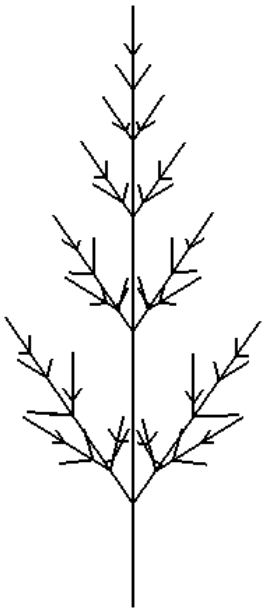
- Relation between opposite and decussate branching



A transition from 3D to 2D, or planation, is one of evolutionary modifications predicted by Zimmermann's telome theory of plant evolution (*c.f.* W. N. Stewart and G. W. Rothwell, *Paleobotany and the evolution on plan*ts (second edition), Cambridge University Press, 1993). Here this transition is used to illustrate the relation between opposite and decussate branching patterns.

Decussate branching

- Manipulating the turtle in 3D
- Turtle symbol /



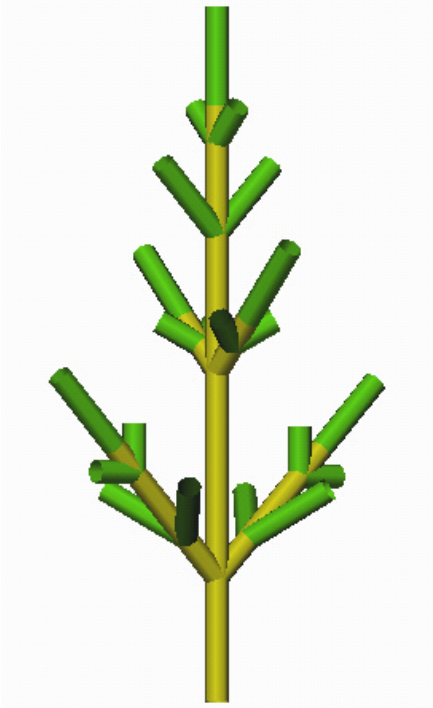
The planar branching structure with the opposite branching pattern is transformed into three-dimensional branching pattern with the decussate phyllotaxis by rotating the turtle 90 degrees around the header vector between consecutive metamers.

```
#define STEPS 400
#define dt 0.02

#define D 2
#define R 1.5

#define RATE (R^dt)

Lsystem: 1
derivation length: STEPS
Axiom: A(0)
A(t) --> A(t+dt)
I(x) --> I(x*RATE)
decomposition
maximum depth: 2
A(t) : t >= D --> /(90) M(t-D) A(t-1)      the only change needed
M(t) : 1 {x=0.5*R^t;}
     --> I(x)[+A(t)][-A(t)]I(x)
homomorphism
A(t) --> ;F(t)
I(x) --> F(x)
endlsystem
```

The 3D geometry of this model is difficult to appreciate, since this model is still rendered as a line drawing. Our visual system needs more cues. One method is to rotate the object. Another is to render the object more realistically.

3D viewing and rendering

- 2D lines vs. 3D cylinders
  - Radius specification in world units
  - Wire-frame vs. shaded models
- Rendering
  - Phong shading model
  - Material editor
  - Light specification
- Viewing
  - Hidden surface elimination
  - Z-buffer
  - Front and back plane



To present objects more realistically, we need to simulate its interaction with the light. This involves simulation of light propagation in the medium, and simulation of the optical properties of the materials of which the models is supposed to be built. These problems have been (and are being) extensively researched in computer graphics. Cpfg supports a simply local illumination model called Phong shading. It was implemented using calls to the OpenGL graphics library, and cpfg viewing and rendering parameters correspond closely to those found in OpenGL.

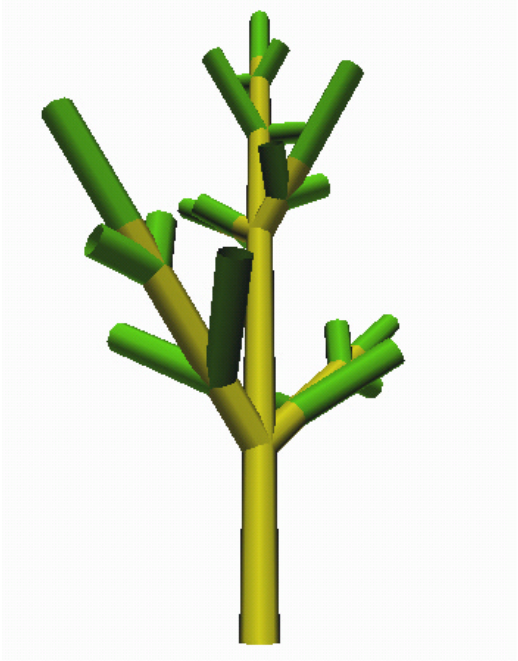
In L-studio/cpfg material parameters are specified using the material editor. Light parameters are listed in the view file. Experiment with the settings as suggested below:

```
angle increment: 45
initial line width: 0.5
initial color: 1
color increment: 1
line style: cylinder                    polygon or cylinder
render mode: shaded                     shaded or wireframe
front distance: -100                    try –1 or –10000
back distance: 100                      try –1 or –10000
z buffer: on                            on or off
scale factor: 0.9800
light: V: 1,0.5,1                       V: direction of incoming light
light: V: -1,-0.5,-1 D: 0.7 0.7 0.7     D: diffuse light intensity
```

Specification of material and light parameters that result in appealing images is a true art.

Perspective viewing

- Perspective projections
- Projection parameters

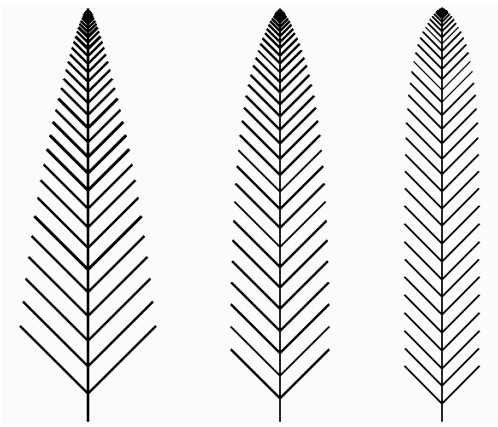


Cpfg also supports perspective viewing, although it is more difficult to properly set parameter values. Specification of a view in the the view file may include the following lines:

```
projection: perspective                 perspective or parallel
view reference point: 0,5,0             a point at the view center
viewpoint: 0,5,20                       position of the observer
viewing angle: 60                       defines the field of view
```

Monopodial branching

- Monopodial branching pattern
- Growth functions
- Graphical function editor



So far, we have only considered models with a single apex type. This resulted in highly recursive, polypodial branching structures. In reality, different apices may have different fates. In the case of monopodial branching, lateral buds produce organs that do not branch further.

```
#define STEPS 400
#define dt 0.05

#define PL 0.500000                 plastochron
#define MAX_INT_LEN 0.500000        maximum internode length
#define MAX_LEAF_LEN 1.500000       maximum leaf length
#define DURATION 20                 growth time to maturity

Lsystem: 1
derivation length: STEPS
Axiom: #(0.05)A(0)
A(t) --> A(t+dt)
I(t) --> I(t+dt)
L(t) --> L(t+dt)
decomposition
A(t) : {t=t-PL;} t>0 -->
     I(t)[+(45)L(t)][-(45)L(t)]A(t)          L denotes a leaf
homomorphism
I(t) --> F(MAX_INT_LEN*func(1,t/DURATION))   internode growth
L(t) --> ;F(MAX_LEAF_LEN*func(2,t/DURATION)) leaf growth
endlsystem
```

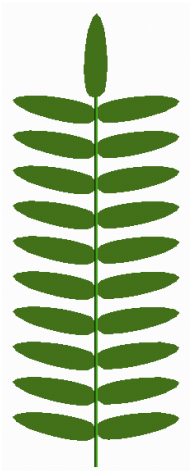
The keyword func denotes a call to a graphically defined function. They can be edited using the L-studio function editor. The files representing these functions are identified in the view file using statements:

```
function: internode.func 100       function #1
function: leaf-length.func 100     function #2
```

The optional parameter (100) affects the accuracy of function representation by cpfg.

Single-compound leaf

- Parametric surfaces
- Surface editing in 2D
- Turtle symbol ~
- Symbols with multiple parameters
- Logical operators
- Determinate growth



In this model, individual leaflets are modeled using parametric surfaces. They are asuumed to be flat, which makes specification of leaflet shape using an interactive surface editor particularly easy.

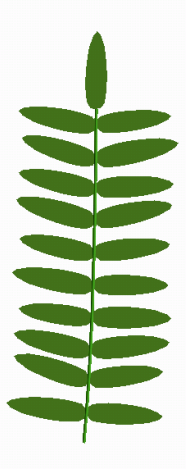
```
#define STEPS 800
#define dt 0.05
#define PL 1.500000
#define DURATION 15
#define TOTAL 10
#define ANGLE 80
#define INT_SIZE 1.0
#define LEAF_SIZE 0.6

Lsystem: 1
derivation length: STEPS
Axiom: #(0.1)A(0,0)
A(t,n) --> A(t+dt,n)                          n is the leaf (pair) number
I(t,x) --> I(t+dt,x)                          x is the maximum size
L(t,n,x) --> L(t+dt,n,x)
decomposition
A(t,n) : {t=t-PL;} t>0 && n<TOTAL -->         conjunction of two conditions
     I(t,INT_SIZE)
     [+(ANGLE)L(t,n,LEAF_SIZE)]
     [-(ANGLE)L(t,n,LEAF_SIZE)]
     A(t,n+1)
A(t,n) : {t=t-PL;} t>0 && n>=TOTAL -->
     I(t,0.5)L(t,n,LEAF_SIZE)                 terminal leaflet
homomorphism
I(t,x) --> F(x*func(1,t/DURATION))
L(t,n,x) --> ~l(x*func(2,t/DURATION))         reference to surface ~l
endlsystem
```

The surface denoted ~l has been identified by the following statement in the view file:

```
surface: l leaf.s 1 8 4          surface: id fil
```

Random variation

- Stochastic modeling
- Random function srand()
- Orthotropism



It is unrealistic to assume that all leaflets will have exactly the same parameter values. In this model, random functions with normal distribution are used to introduce unorganized variation. This has been achieved by replacing constant definitions from the previous model,

```
#define ANGLE 80
#define INT_SIZE 1.0
#define LEAF_SIZE 0.6
```

with the statements:

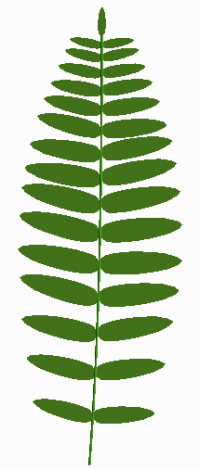
```
#define ANGLE nran(80,4)              mean, standard deviation
#define INT_SIZE nran(1,0.05)
#define LEAF_SIZE nran(0.6,0.02)
```

Upward curving of the stem was achieved using a simple tropism model, invoked by the following vie file statement:

```
tropism: T: 0,1,0 E: 0.2              T: tropism direction E: elasticity
```

Organized variation

- Positional information
- Acrotony, mesotony, and vigor
- Macro function definitions



Metamer parameters may depend not only on random factors, but also on the position of metamers on the stem. This results in organized variation of organs. In this model, the final sizes of internodes and leaves are functions of the metamer number. The model has been implemented by replacing constant definitions from the original model,

```
#define ANGLE 80
#define INT_SIZE 1.0
#define LEAF_SIZE 0.6
```

as follows:

```
#define ANGLE nran(80,2)                    random, as before
#define INT_SIZE(n) func(3,n/TOTAL)         macro function definition
#define LEAF_SIZE(n) func(4,n/TOTAL)
```

Note that this approach makes it possible to capture acrotonic and mesotonic structures. Basically, this is a descriptive approach (we express the maximum size as a function, without justifying where it comes from), the functions INT_SIZE(n) and LEAF_SIZE(n) could also be interpreted as returning vigor values.

Vegetative shoot with spiral phyllotaxis

- Spiral phyllotaxis
- Turtle symbol &
- Parametric surfaces in 3D
- Non-uniform scaling of surfaces
- Interpolation of material properties
- Predefined functions floor()

This model uses graphically defined functions of time to control:

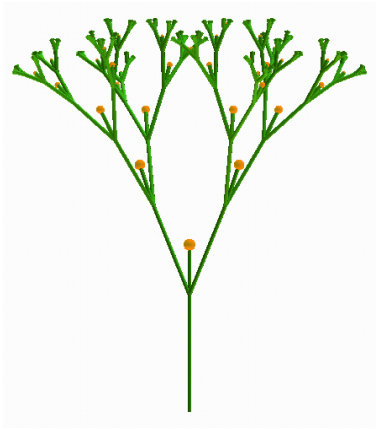
- the elongation of internodes,
- the growth of leaves in length and width,
- the branching angles between the stem and the leaves, and
- the gradual changes of leaf colors.

Leaves, modeled using cubic patches, are arranged into a spiral phyllotactic pattern. Slight orthotropism improves the appearance of the model.

```
#define STEPS 450
#define dt 0.05
#define PL 0.500000
#define MAX_INT_LEN 0.500000
Lsystem: 1
derivation length: STEPS
Axiom: -(5)#(0.15)A(0)                          stem leans to the right
A(t) --> A(t+dt)
I(t) --> I(t+dt)
L(t) --> L(t+dt)
decomposition
A(t) : {t=t-PL;} t>0 -->
     I(t)[L(t)]/(137.5)A(t)                     spiral phyllotaxis
homomorphism
I(t) --> F(MAX_INT_LEN*func(1,t/15))
L(t) : 1 {len = func(2,t/10);
          wid = func(3,t/20);
          ang = 90*func(4,t/30);
          col = 32+floor(31*func(5,t/20));}     integer material index
     --> &(ang);(col)~l(wid,len,len)            non-uniform scaling
endlsystem
```

Sympodial branching

- Sympodial branching pattern
- Turtle symbol &O (sphere)



The follwing L-system capture the generic architecture of a sympodial branching structure.

```
#define STEPS 180
#define dt 0.05

#define PL 1.00000          plastochron
#define D 10                elongation time
#define ANG 25              branching angle

Lsystem: 1
derivation length: STEPS
Axiom: #(0.03)\(60)A(0)

A(t) --> A(t+dt)
I(t) --> I(t+dt)
K(t) --> K(t+dt)

decomposition

A(t) : {t=t-PL;} t>0 -->
     I(t)/(90)[+(ANG)A(t)][-(ANG)A(t)]K(t)

homomorphism
maximum depth: 3

I(t) --> F(func(1,t/D))
K(t) --> F(0.4*func(2,t/D))
     ;@O(0.1*func(2,t/D))                    @O is a sphere

endlsystem
```

The key feature of this model is the form of its decomposition rule. It is instructive to compare it with the rule that captures monopodial branching.

```
A(t) : {t=t-PL;} t>0 -->
     I(t)[+(45)L(t)][-(45)L(t)]A(t)
```