1. GeoGebra入门

1.1 GeoGebra 界面与基本操作

1.1.1 介绍界面和基本操作

- **GeoGebra的安装**：在线版本、离线安装。已知问题：GeoGebra6 classical 在Mac中存在首次启动后，不能将窗口前置显示的问题。建议使用GeoGebra 5经典版。
- **GeoGebra的基本界面**：如何隐藏/显示坐标轴、网格、标签；如何保存、导出文件。
- **GeoGebra的基本操作**：使用工具构造各种几何对象，并观察代数区域。

1.2 绘制基本几何图形

1.2.1 基本图形

- 绘制点、线段、圆、椭圆、抛物线、多边形等；
- 尝试绘制不同的几何图形
- 探索GeoGebra的更多功能

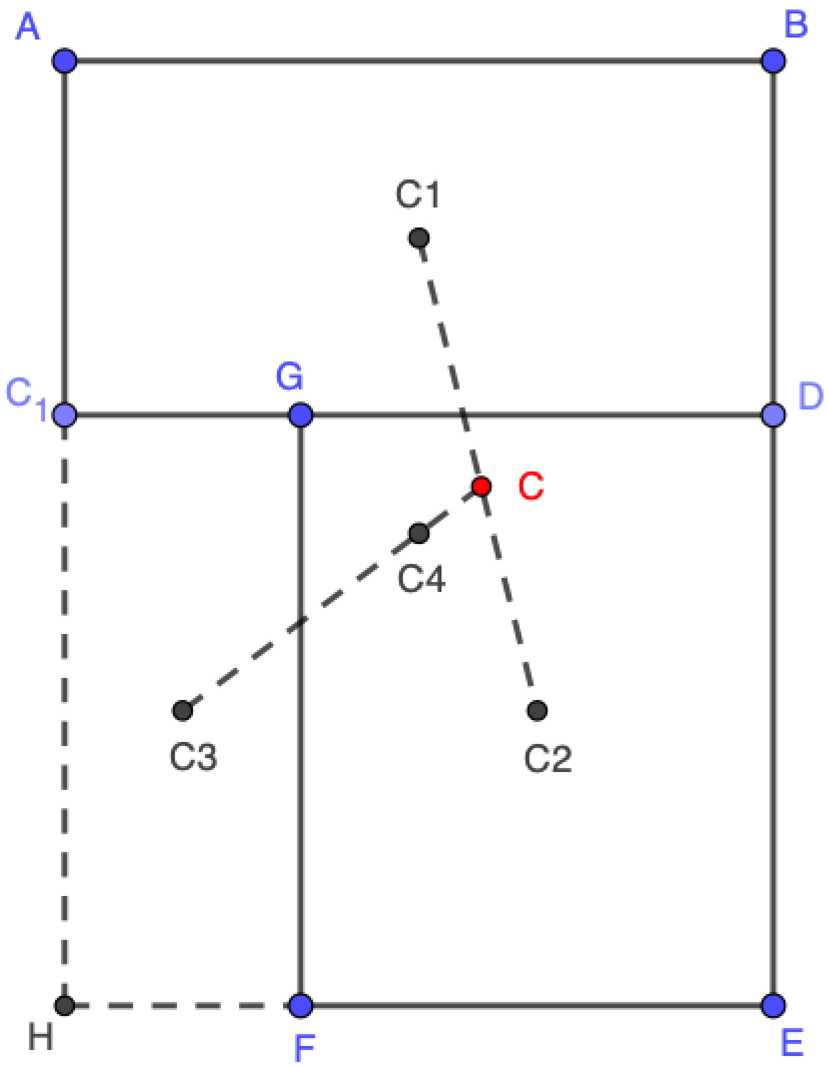
1.2.2 探索多边形的重心

回忆一个三角形ABC的重心就是三条边中线的交点G。此时，$\overrightarrow{GA}+\overrightarrow{GB}+\overrightarrow{GC}=0.$

我们希望回答如下问题：

1. 在三角形ABC中，满足：$\overrightarrow{GA}+\overrightarrow{GB}+\overrightarrow{GC}=0$ 的点G是否唯一？若唯一，它是否就是ABC的重心？
2. 利用1, 我们猜测任何多边形$A_1A_2\dots A_n$的重心G都满足$\sum_{i=1}^{n}\overrightarrow{GA_i}=0.$ 而且反过来也对。
3. 对矩形，平行四边形，梯形等简单的四边形，检验上述定义的重心是否正确？
4. 你能对两个矩形拼接而成的L型图形，找到该图形的重心吗？
5. 如何定义一般的多边形的重心？
6. 过重心的直线是否将图形平分为等面积的两部分？

示例: 利用GeoGebra作出两个矩形拼接成的L形图形的重心。



背景知识

1. 质心是空间中质量分布中心的唯一点，其性质是相对于该点的加权位置向量总和为零。
2. 假设一个粒子系统：$P_1, \ldots, P_n$, 每个粒子P_i的质量为m_i, 位置为$\vec{r}_i$, 则其质心坐标$\vec{R}$满足

$$\sum_{i=1}^{n} m_i(\vec{r}_i - \vec{R}) = 0 \implies \vec{R} = \frac{\sum_{i=1}^{n} m_i \vec{r}_i}{\sum_{i=1}^{n} m_i}.$$

3. 对空间中的连续体Q（例如一个密度均匀的平板），假设位置为$\vec{r}$处的密度位$\rho(\vec{r})$, 则该连续体Q的密度为：

$$\iiint_{\vec{r}\in Q}\rho(\vec{r})(\vec{r}-\vec{R})dV_Q=0 \implies \vec{R}=\frac{1}{M}\iiint_Q\rho(\vec{r})\vec{r}dV,$$

这里$M=\iiint_Q 1dV$为Q的总质量。

1.2.3 探索三角形的各种心

练习：用GeoGebra作图：

1. 一个平面三角形ABC;
2. 该三角形的垂心H.
3. 该三角形的重心G.
4. 该三角形的内心I.
5. 该三角形的外心O.
6. 该三角形的旁心P.
   最终的图形只有三角形的三条边和三个顶点，以及这五个心。所有的点都有标签，边不要标签。

思考，除了这些心，三角形ABC还有哪些心?

练习：创建一个自定义工具，使得可以非常容易的构造给定三角形的旁心。

练习：利用尺规作图，将任一线段AB三等分。

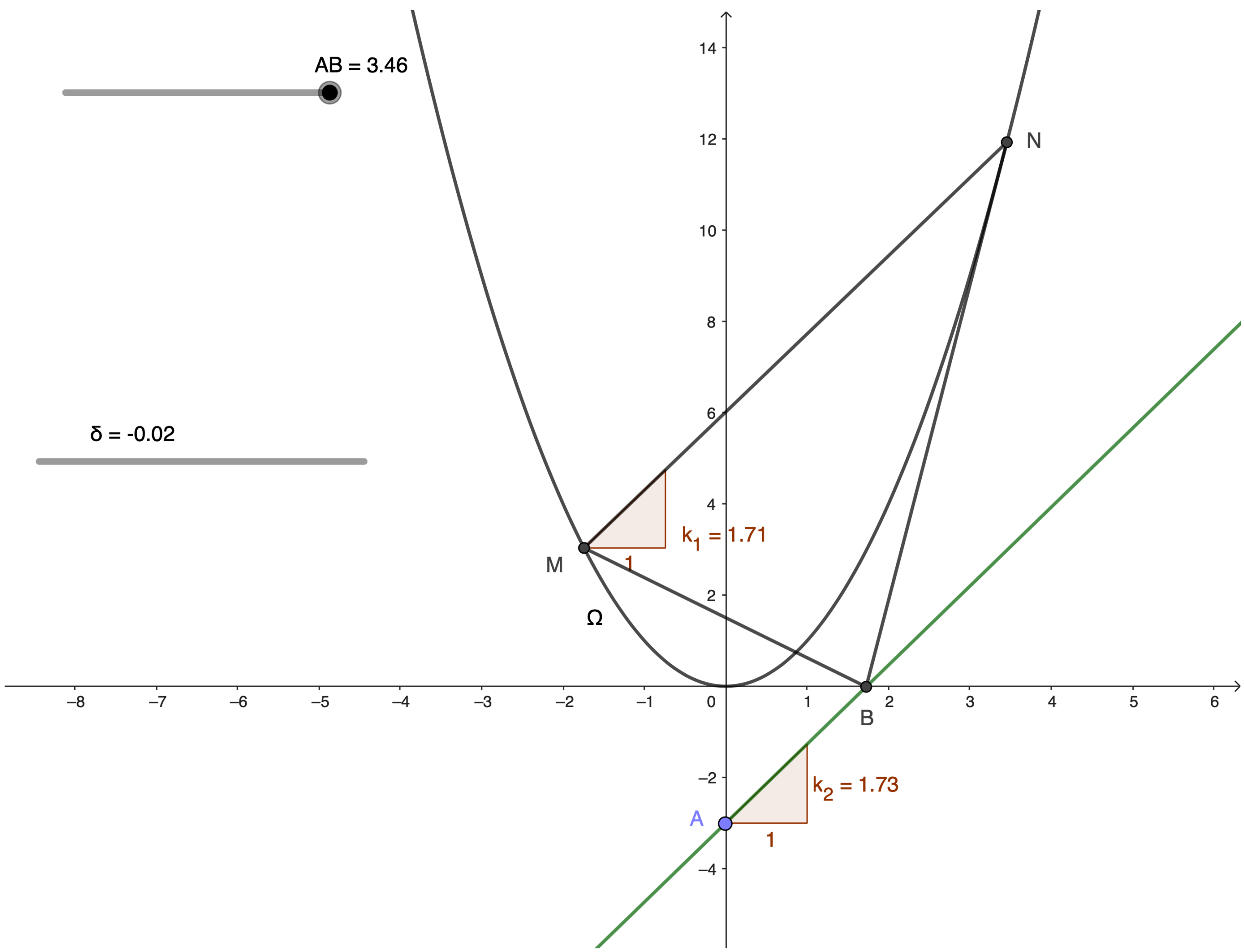
1.3 GeoGebra交互式教学的应用

1.3.1 GeoGebra的基础操作

示例: 2024重庆高中数学联赛初试，第十题

已知抛物线Ω: $y=x^2$，动线段AB在直线$y=\sqrt{3}x-3$上且B在A的右侧，满足$|AB|=2\sqrt{3}$. 过A作Ω的切线，令左边的切点为M. 过B作Ω的切线，令右边的切点为N. 求当$MN \parallel AB$时，A点的横坐标。

根据题意，利用Geogebra作图如下, 在线版本。



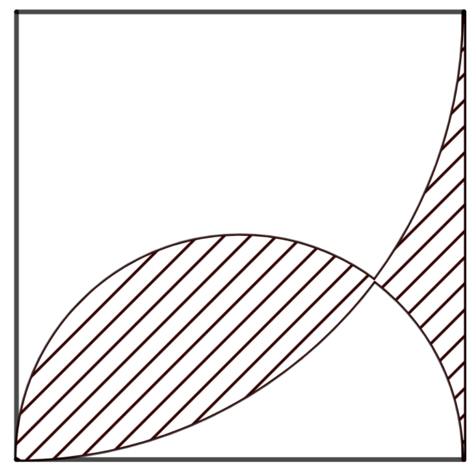
练习:

1. 利用GeoGebra探索上述例子中的答案；
2. 通过将上述二次曲线换成椭圆，你可以得到更加一般的结论吗?
3. (必做!!) 利用GeoGebra作图探索其他数学竞赛试题。
4. (必做!!) **大作业**：参考周向宇院士的论文@周2022中国，利用GeoGebra复原商高关于勾股定理的证明。

1.4 轨迹与阴影

1.4.1 轨迹与阴影

示例: 如图所示，通过先构造一个正方形$ABCD$，然后以AB为直径画半圆弧、以D为圆心画圆弧AC、线段BC构成一个封闭图形。试将该图形利用斜线进行标记。

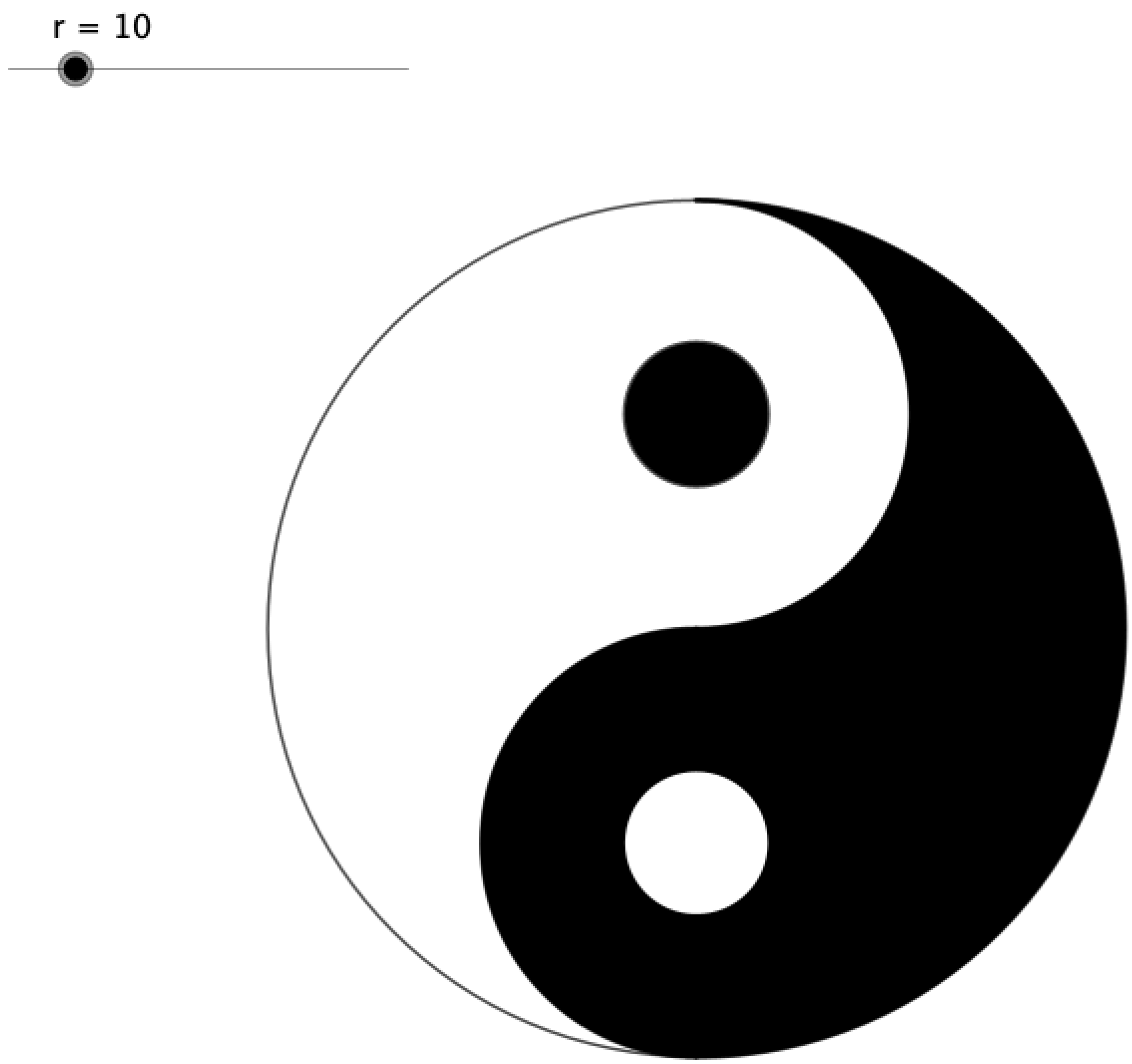


1.4.2 轨迹与阴影（高级）

要点如下：

- 我们利用这三段路径生成了一个列表`l1={c,d,f}`
- 通过`Point(l1)`得到路径上任意的一个点E
- 通过`Locus(E+0,E)`得到点E的轨迹
- 通过设置该轨迹的属性：显示轨迹、颜色、透明度、样式得到效果图。

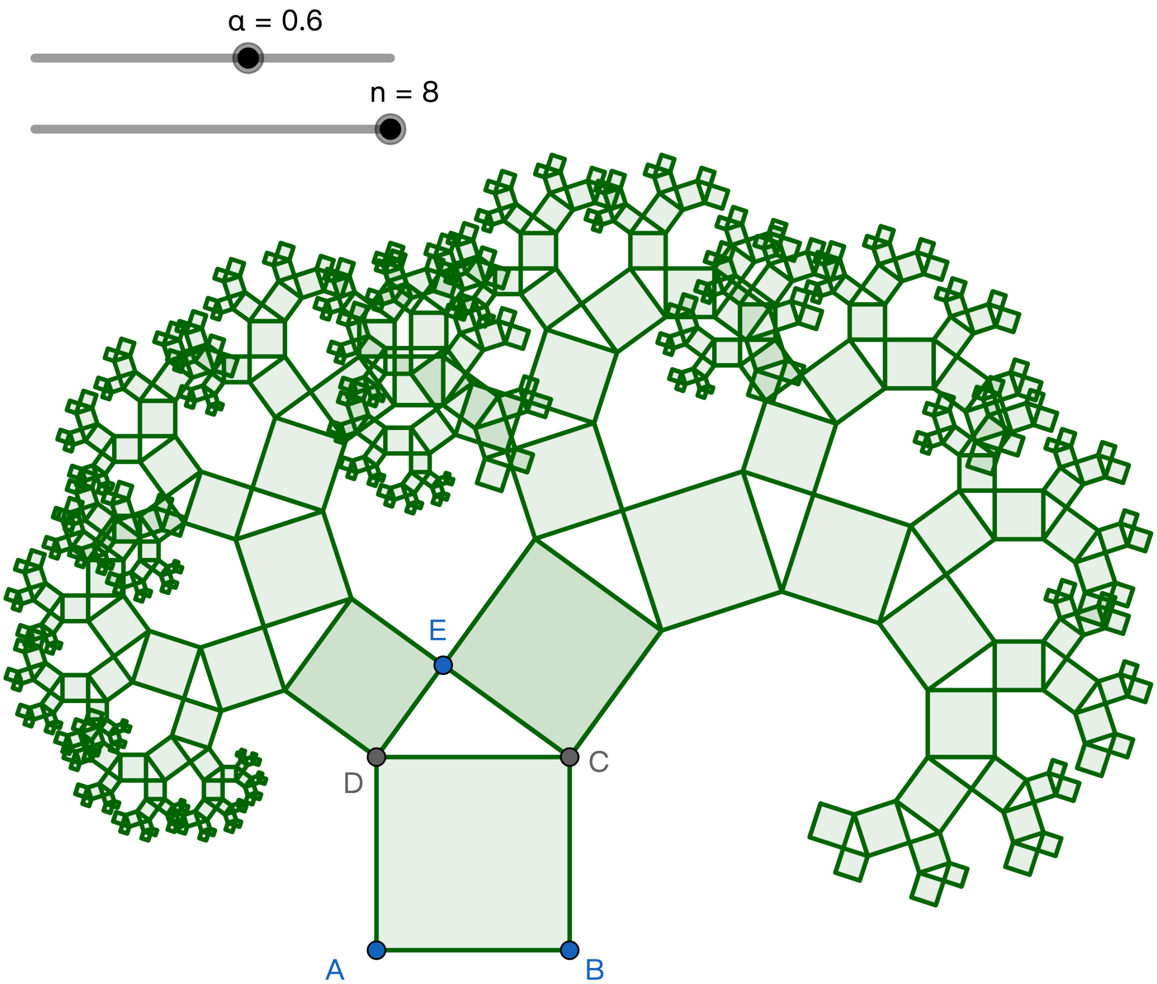
练习: 利用轨迹生成太极图



2 GeoGebra 高级功能

2.1 迭代

2.1.1 GeoGebra迭代生成勾股树



直接利用代数输入区，完成以下操作:

- 设定点 $A=(-8,-30)$, $B=(-2,-30)$;
- 生成以 AB 为底边的正方形;
- 定义参数 α 初始值为 0.6, 范围为 $[0,1]$;
- 以顶点 C,D 为端点画圆;

- 在圆弧上取点 E, 使得它依赖于参数 α;
- 分别以 DE、EC 为底边构造正方形;
- 定义迭代次数参数 n, 范围为 $[2,8]$, 初始值为 6;
- 使用迭代列表生成勾股树。

以下是完整代码:

```
A=(-8, -30)
B=(-2, -30)
p=Polygon(A, B, 4)
alpha=0.6
Semicircle(Vertex(p, 4), Vertex(p, 3))
E=Point(Semicircle(Vertex(p, 4),
Vertex(p, 3)), alpha)
pl=Polygon(Vertex(p, 4),
Point(Semicircle(Vertex(p, 4), Vertex(p,
3)), alpha), 4)
pr=Polygon(Point(Semicircle(Vertex(p,
4), Vertex(p, 3)), alpha), Vertex(p, 3),
4)
n=8
IterationList(Flatten(Zip({Polygon(Verte
x(p, 4), Point(Semicircle(Vertex(p, 4),
Vertex(p, 3)), alpha), 4),
Polygon(Point(Semicircle(Vertex(p, 4),
Vertex(p, 3)), alpha), Vertex(p, 3),
4)}, p, P)), P, {{p}}, n)
```

---

1. 根据在线文档，理解上述代码中的 Flatten、Zip、IterationList 命令；

2. 尝试使用迭代列表生成其他分形图像：例如雪花曲线（Koch曲线）、分形二叉树、黎曼积分的上和与下和等。

2.1.2 迭代代码解释

Zip: `Zip( <Expression>, <Var1>, <List1>, <Var2>, <List2>, ...)`

- 通过将变量 Var 替换为随后的 List，返回 Expression 计算结果后的列表。
- 输入：`Zip(a^2, a, {1,2,5})`
  结果：`{1,4,25}`
- 输入：`Zip(Midpoint(A, B), A, {P, Q}, B, {R, S})`
  结果：PR, QS 之中点列表
- 输入：`Zip(Degree(a), a, {x^2, x^3, x^6})`
  结果：`{2,3,6}`
- 输入：`Zip(Simplify(a*x^(b-1)), a, {1, 2, 5}, b)`
  结果：$\{1, 2x, 5x^{2}\}$
- 输入：`Zip(f(2), f, {x+1,x+3})`
  结果：`{3, 5}`

IterationList: `IterationList( <Function>, <Start Value>, <Number of Iterations> )`

- 重复计算函数的值，返回循环次数加一长度的列表，列表的第一个元素就是给定的初值。

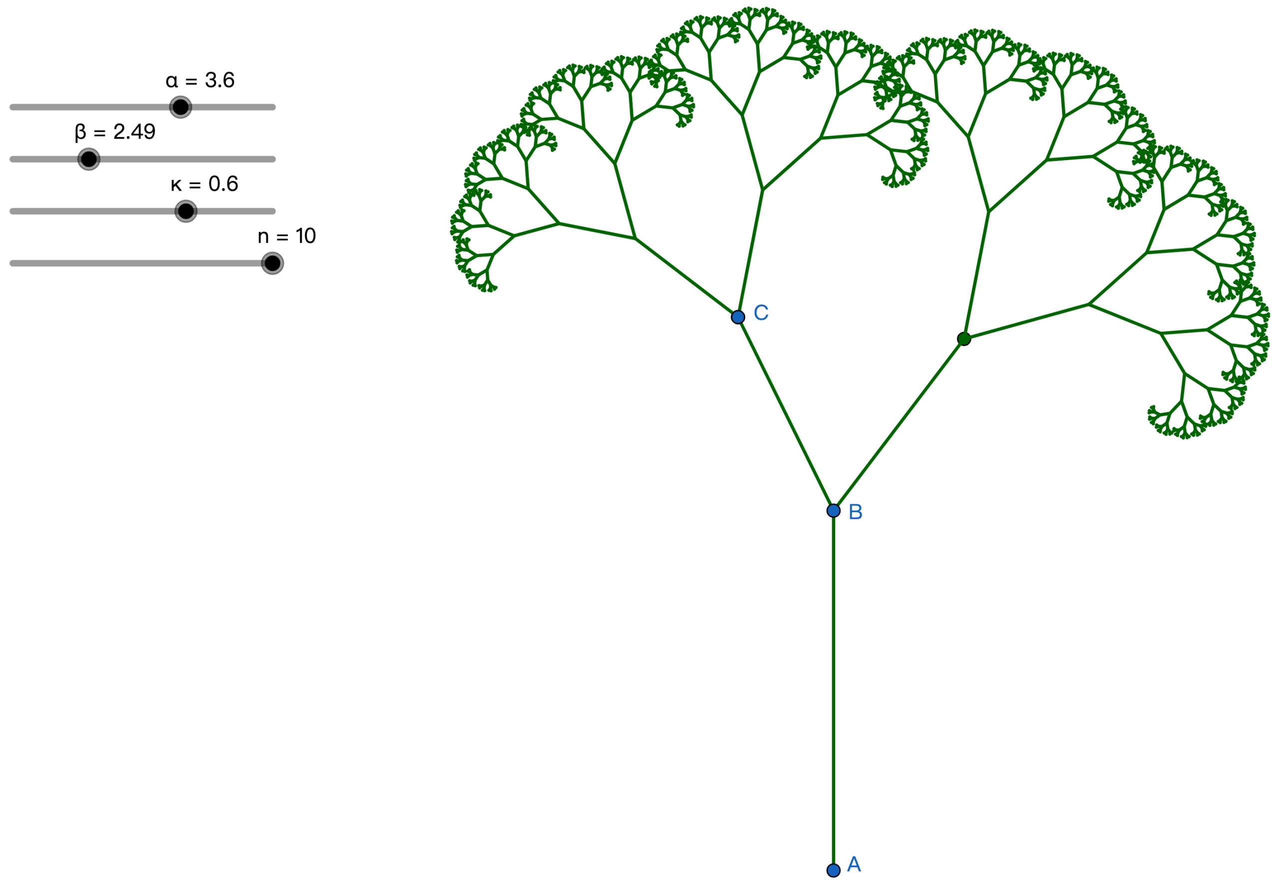
- 输入： `IterationList(a^2+1,1,3)`
  输出： `{1,2,5,26}`
- 输入： `IterationList(a^(2)+b^(2),a,b,{1,1},5)`
  输出： `{1,1,2,5,29,866}`

2.2 分形二叉树

2.2.1 分形二叉树

```
// 定义起始点 A 和 B
A = (50, -50) // 左下角点
B = (50, -10) // 右下角点

// 创建初始线段 f
f = Segment(A, B)

// 定义旋转角度和缩放因子
alpha = 2.72 // 第一个旋转角度
beta = 3.73  // 第二个旋转角度
kappa = 0.7  // 缩放因子

// 定义迭代函数
// Flatten: 扁平化嵌套列表
// Zip: 将多个列表按元素配对
// Segment: 生成线段
// Point(f, 1): 获取线段 f 的末端点
// Dilate: 进行缩放
// Rotate: 进行旋转
// theta: 旋转角度
// n: 迭代次数
```

```
// 生成分形二叉树的迭代列表
IterationList(
    Flatten(
        Zip(
            Zip(
                Segment(Point(f, 1),

Dilate(Rotate(Point(f, 0), theta,
Point(f, 1)), kappa, Point(f, 1))
                ),
                theta, {alpha, beta} //
旋转角度列表
            ),
            f, f1 // 将线段与其生成的分支配
对
        )
    ),
    f1, {{f}}, n // 迭代次数和初始线段
)
```

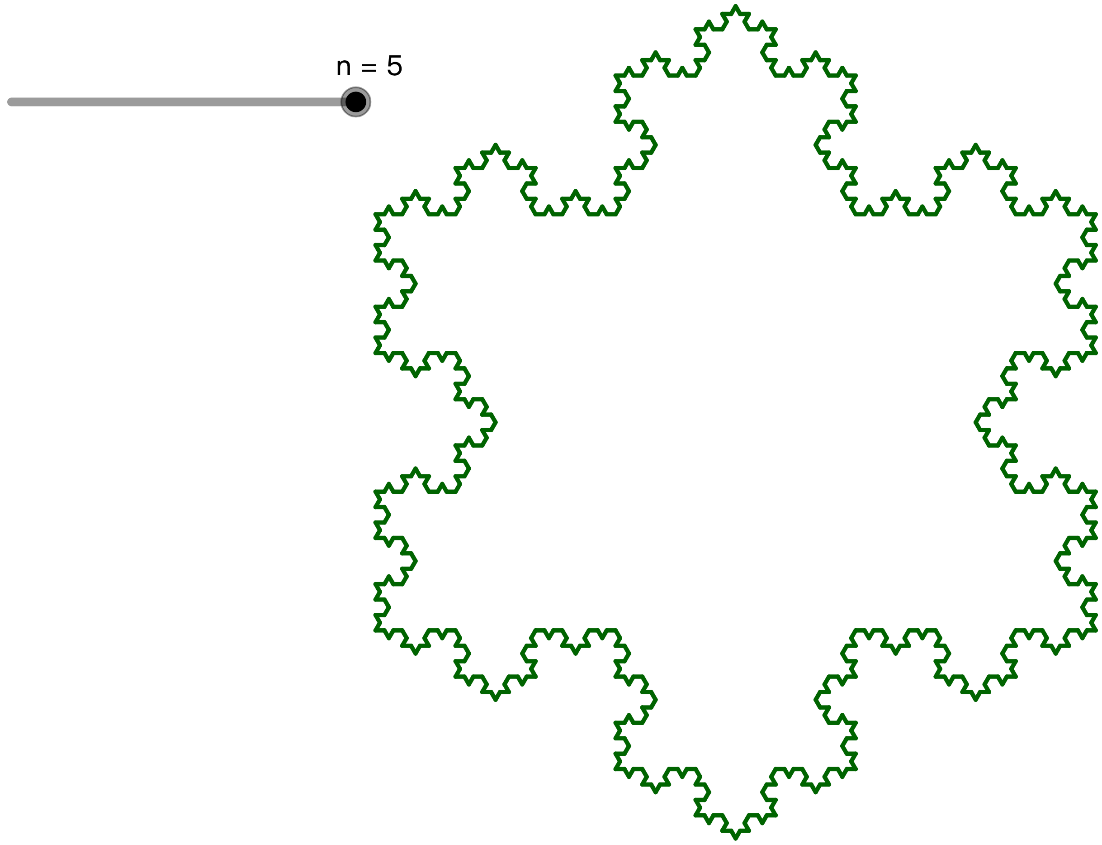

α = 3.6
β = 2.49
κ = 0.6
n = 10
C
B
A

2.3 Koch曲线

2.3.1 Koch曲线

```
A=(0,0)
B=(21,0)
C=Rotate(B,pi/3,A)
f=Segment(A,B)
g=Segment(B,C)
h=Segment(C,A)
IterationList(
Flatten(
Zip(
Zip(
Sequence(
Segment(Element(l1, u), Element(l1, u +
1)), u, 1, 4
), l1, {
   /*将顶点插入到三等分点序列*/
   Insert(
   /*得到三等分线段的顶点*/
   Rotate(Point(ff, 2 / 3), -pi/3,
Point(ff, 1 / 3)),
   /*生成线段ff上的三等分点序列，包括端点*/
   Sequence(Point(ff, v), v, 0, 1, 1 / 3)
   /*3表示插入在序列的第三个位置*/
```

```
, 3)}
), ff, f1)
),
f1, {{f, g, h}}, n)
/*最后一个迭代完整代码如下*/
Element( IterationList( Flatten( Zip(
Zip( Sequence( Segment( Element( l3, u),
Element( l3, u+1)), u, 1, 4), l3,
{Insert( Rotate( Point( ff, 2/3), -pi/3,
Point( ff, 1/3)), Sequence( Point( ff,
v), v, 0, 1, 1/3), 3)}), ff, f1)), f1,
{{f, g, h}}, n), n)
```



2.4 涂色

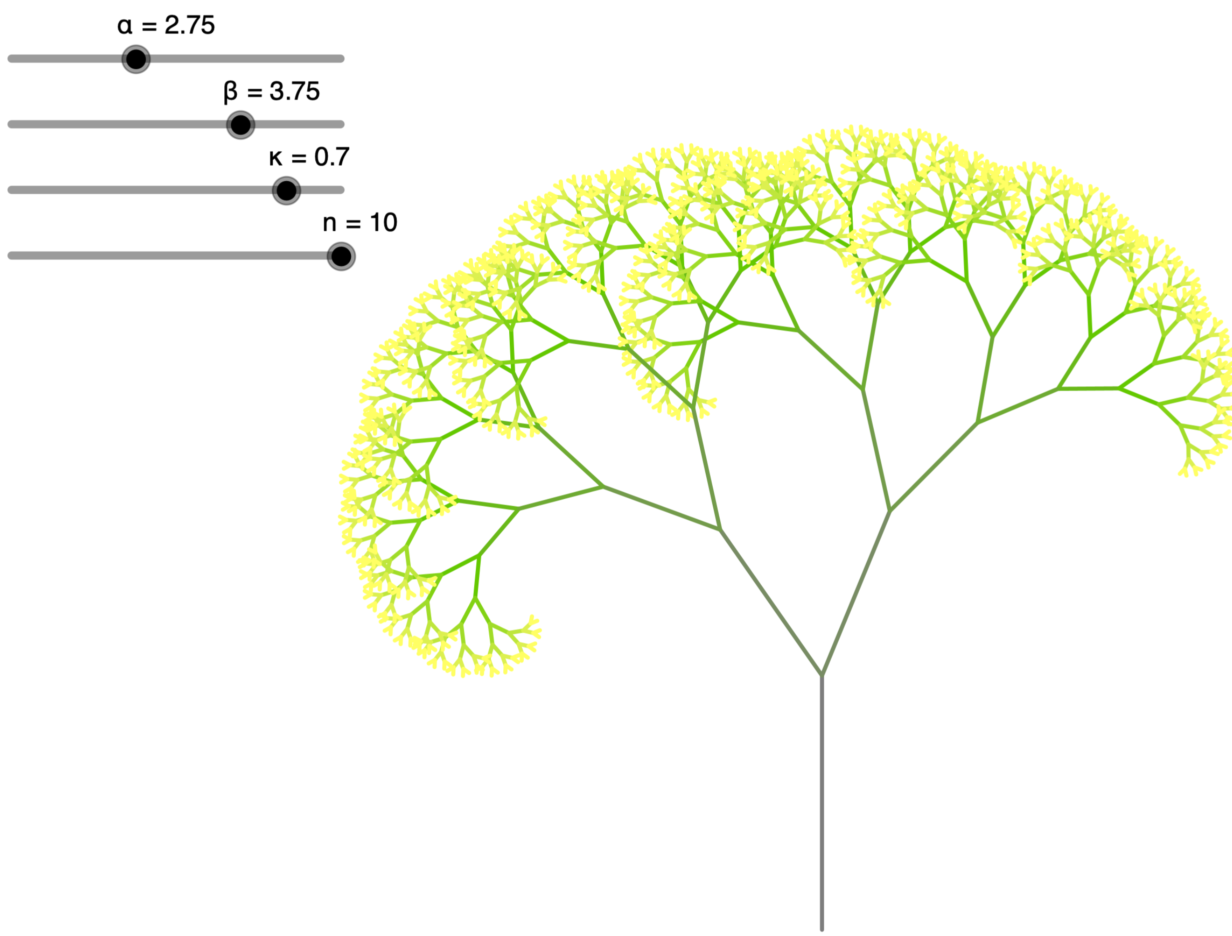
2.4.1 涂色原理

- 考虑上面迭代的例子，其实迭代的最终结果是一个列表，列表的第 k 个元素就是迭代中第 k 步生成的图形。
- 因此，我们可以对每步生成的图形进行不同的染色，这样就得到了渐变色。
- 渐变色的构造如下：利用颜色的 RGB 值，构造一个折线段：

```
/*颜色的RGB/255，归一化到区间[0,1]*/
h=Polyline((128, 128, 128) / 255,
(102, 204, 0) / 255, (255, 255, 102)
/ 255)
/*此时折线段上的点的坐标就是该点处颜色的RGB
值*/
Point(h,0.3)
/*表示整个折线长度的0.3处点的RGB值*/
```

- 动态染色
  - 利用表格，取得迭代列表中第 k 步的元素
  - 根据 k 在折线段上取 $(k-1)/n$ 处的点
  - 根据该点的 RGB 动态染色第 k 步的元素
  - 具体代码如下：

```
/*取元素*/
B1=Element(l2,A1)
/*右键设置该元素属性:高级/动态颜色*/
/*红: x(Point(h, (A1 - 1) / n)) */
/*绿: y(Point(h, (A1 - 1) / n)) */
/*蓝: z(Point(h, (A1 - 1) / n)) */
/*最后下拉填充表格得到B2...Bn的元素*/
```



作业

1. 自行对勾股数等迭代图形涂色。
2. 你能想到涂色的其他应用吗?
3. (必做!!)**大作业**：对 Julia 集

3 GeoGebra 在教学中的应用

3.1 教学案例分析

黎曼积分

回忆，给定连续函数$y=f(x)$, $x\in[a,b]$. 我们可以将区间$[a,b]$等分为n等份。然后定义和

$$\bar{S}_n=\sum_{k=1}^{n}\frac{1}{n}(b-a)\max_{x\in\left[a+(k-1)\frac{b-a}{n},a+k\frac{b-a}{n}\right]}f(x),$$
$$\underline{S}_n=\sum_{k=1}^{n}\frac{1}{n}(b-a)\min_{x\in\left[a+(k-1)\frac{b-a}{n},a+k\frac{b-a}{n}\right]}f(x).$$

可以证明$n\to\infty$时，上述$\bar{S}_n$以及$\underline{S}_n$的极限都存在。分别称为函数f的黎曼积分的上和与下和。

3.2 设计互动式教学活动

黎曼和的代码实现

利用GeoGebra可以实现上述动态效果， 完整代码如下：

```
f: y=0.05 x^(3)-3 x+1
A=(-9,0)
B=(9,0)
n=10
Zip(Polygon(u, (x(u), y(Min(f, x(u),
x(v)))), (x(v), y(Min(f, x(u), x(v)))),
v), u, Sequence(A+((B-A)/(n)) i, i, 0,
n-1), v, Sequence(A+((B-A)/(n)) i, i, 1,
n))
Zip(Polygon(u, (x(u), y(Max(f, x(u),
x(v)))), (x(v), y(Max(f, x(u), x(v)))),
v), u, Sequence(A+((B-A)/(n)) i, i, 0,
n-1), v, Sequence(A+((B-A)/(n)) i, i, 1,
n))
b=Sum(Zip(x(((B-A)/(n))) y(Min(f, x(u),
x(v))), u, Sequence(A+((B-A)/(n)) i, i,
0, n-1), v, Sequence(A+((B-A)/(n)) i, i,
1, n)))
a=Sum(Zip(x(((B-A)/(n))) y(Max(f, x(u),
x(v))), u, Sequence(A+((B-A)/(n)) i, i,
0, n-1), v, Sequence(A+((B-A)/(n)) i, i,
1, n)))
text1=TableText({a,b,c})
```

```
text2=TableText({"Upper","Lower","Difference"})
```

3.3 学生作品展示与评价

黎曼积分的代码解释

代码解释如下：

- `l1=Sequence(A+ (B-A)/n*i,i,0,n)`：生成AB线段的n等分点列表。
- `C=Min(f,x(Element(l1,1)),x(Element(l1,2)))`：构造第一个区间的最小值点；
- `D=Max(f,x(Element(l1,1)),x(Element(l1,2)))`：构造第一个区间的最大值点；
- 构造最小值矩形的左、右端点:

```
E=(x(Element(l1, 1)), y(Min(f, x(Element(l1, 1)), x(Element(l1, 2)))))
F=(x(Element(l1, 2)), y(Min(f, x(Element(l1, 1)), x(Element(l1, 2)))))
```

- 构造最小值矩阵:

```
poly1=Polygon(Element(l1, 1),
(x(Element(l1, 1)), y(Min(f,
x(Element(l1, 1)), x(Element(l1, 2))))),
(x(Element(l1, 2)), y(Min(f,
x(Element(l1, 1)), x(Element(l1, 2))))),
Element(l1, 2))
```

- 利用 `Zip` 构造所有最小值矩阵:

```
l2=Zip(Polygon(u, (x(u), y(Min(f, x(u),
x(v)))), (x(v), y(Min(f, x(u), x(v)))),
v), u, Sequence(A+((B-A)/(n)) i, i, 0,
n-1), v, Sequence(A+((B-A)/(n)) i, i, 1,
n))
```

- 将所有的最小值矩阵的面积求和:

```
b=Sum(Zip(x(((B-A)/(n))) y(Min(f, x(u),
x(v))), u, Sequence(A+((B-A)/(n)) i, i,
0, n-1), v, Sequence(A+((B-A)/(n)) i, i,
1, n)))
```

黎曼积分-上和与下和