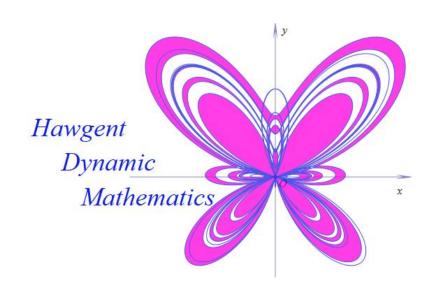


# Hawgent 皓骏动态数学课程系列

# 动态数学范例教程

深入学科,彻底突破数学教学和数学学习中的重点难点问题 开展数学实验、数学教学、数学学习和数学研究的必备工具



# 皓骏(广州)数学技术中心

Hawgent Technology Centre in Mathematics

## 内容介绍

Hawgent 皓骏提供了功能强大、操作简便的工具,但并不希望作为数学教师的我们将大量的时间和精力耗费在学习技术和制作资源方面,因此也提供了大量与课程教学、探究实验、自主学习等方面的资源,从小学到高中,从课内到课外,具有几千件,并且还会不断补充和更新。

不过,仍然有许多数学教师或者在校师范生希望自己尝试动手做一做,或者按照自己的设计制作出更有个性化的作品。Hawgent 皓骏团队为了满足部分人的愿望,于是就编写了这本教程。

这本教程以学习制作常用的教学资源为主线、以介绍 Hawgent 皓骏动态数学软件的主要功能为目标,故为范例教程。

这是第一版本,必定还存在许多的不足。在使用过程中,如有任何问题或建议,还望不吝赐教。

欢迎联系我们。

邮件: 11033149@qq.com

QQ 群: 367878041

# 目录

第一	-部分 动态数学入门	1
	1.1 什么是动态数学	1
	1.2 了解 Hawgent 皓骏	2
	1.3 认识软件界面	7
第二	二部分 几何图形与变换	10
	2.1 经过一点的直线	10
	2.2 三角形的三条高线	12
	2.3 测量三角形的内角和	13
	2.4 三角形内角的直观说明	15
	2.5 勾股定理的直观说明	19
	2.6 在平面上滚动的车轮	23
	2.7 圆在圆上滚动	27
	2.8 翻折得到轴对称图形	32
	2.9 等分圆周设计的图案	35
	2.10 直线与圆的位置关系	39
	2.11 圆与圆的位置关系	41
	2.12 研究圆幂定理	43
	2.13 毕达哥拉斯树	46
	2.14 二叉树	48
	2.15 多边形的密铺	50
	2.16 密铺曲线	53
	2.17 平面中的变换	61
第三	E部分 函数、方程及图像	71
	3.1 二次函数的图像	71
	3.2 反比例函数的图像	
	3.3 指数函数及其反函数的图像	76
	3.4 参数方程曲线	77
	3.5 极坐标方程曲线	81
	3.6 分段函数的图像	82
	3.7 描点连线画函数图像	
	3.8 美丽的玫瑰线	
	3.9 根据通项公式画数列的图像	
	3.10 根据递推公式画数列的图像	94
第四	<b>U部分 解析几何问题</b>	
	4.1 固定长度的线段在滑动	
	4.2 斜率之积为定值的两直线交点的轨迹	99
	4.3 根据定义构造椭圆和双曲线	
	4.4 根据定义构造抛物线	
	4.5 构造标准圆锥曲线	
	4.6 研究圆锥曲线的光学性质	
	4.7 多驱动点下的轨迹	108
第王	I部分 三角函数关系	116

5.1 利用正弦线画正弦函数的图像	116
5.2 y=sin(x)-> 3sin (2x+pi/4)的动态变换	118
5.3 由两边及一边对角解三角形	123
5.4 三角函数曲线的变换	128
5.5 正弦波的叠加	131
第六部分 算法与编程	135
6.1 Hawgent 皓骏的编程环境	135
6.2 了解系统中的函数	143
6.3 求最大公约数的程序	149
6.4 判断是否为闰年的程序	150
第七部分 概率与统计	152
7.1 随机得到一个 100 以内的自然数	152
7.2 模拟随机抛硬币实验	154
7.3 掷骰子实验	162
7.4 利用抛豆估计 π	165
7.5 模拟转盘游戏	169
第八部分 专题介绍	174
8.1 变量尺	174
8.2 属性对话框	175
8.3 案例的操作	183
8.4 截图工具	185
8.5 对象间的关联	187
8.6 视口的作用	190
8.7 对象组	192
8.8 作图规则	194
8.9 对象的轨迹	199
8.10 对象的跟踪	203
8.11 两类迭代	204
8.12 文本与公式	212
8.13 初识测量	214
8.14 插入多媒体	223
8.15 语音合成	225
8.16 统计表格	227
8.17 动作按钮	
8.18 图形变换	235
8.19 程序设计	

## 第一部分 动态数学入门

## 1.1 什么是动态数学

"任意三角形 ABC…"、"点 P 在圆上 O 上运动过程中…"、"周长固定的长方形"…,这些都是我们常用的几何语言。这些语言所对应的情景,在传统教学手段下,只能要求学生充分发挥想象能力。相对于我们所生活的多姿多彩的空间而言,高度抽象化了的语言以及图形对于刚学习数学学生来说是难以接受的。

计算机的出现使这种现象渐渐发生改变。利用计算机所作的图形不仅直观、形象,而且 具有动态性,易于观察和理解。

若利用计算机所作的几何图形,在被拖动过程中仍保持几何性质不变,如三角形还是三角形、圆上的点还是圆上的点、平行还是平行、垂直还是垂直、中点还是中点...,那么这样的图形就称为动态几何图形。

构造动态几何图形的计算机软件,就叫做动态几何软件(*Dynamic Geometry Software*)。 动态几何的名称于上世纪 80 年代中期开始出现,经过了 20 多年的发展,动态几何吸引了越来越多人的关注,它在数学教育上的价值已经得到充分肯定。

目前具有动态几何功能的软件就有几十种,其中被中国数学教师所广泛了解的就有 *Geometer's Sketchpad、Hawgent* 皓骏、Z+Z 超级画板、*Cabri、AutoGraph、Derive、GeoGebra* 等。

由于计算机在动态几何方面的成功,也促使人们开始研究计算机在数学的其他领域的教育功能。

因此,计算机代数系统(Computer Algebra System)、动态统计软件(Dynamic Statistics System)、随机实验平台(Probability Simulation Platform)等相继出现并被广泛实验应用。

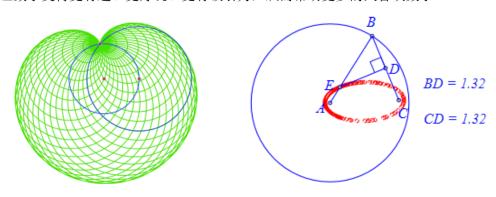
计算机代数系统最重要的应用就是数值计算与符号运算。

动态统计软件能方便地处理大量的数据并且按照人们所熟悉的形式展现。

随机实验平台能够模拟概率事件并且自动记录和计算实验结果。

动态几何软件、计算机代数系统、动态统计软件、随机实验平台等都属于动态数学软件(*Dynamic Mathematics Software*)。

可以说:动态数学的出现,让数学和数学教学的面貌焕然一新。因为,它能: 让数学变得更直观、更形象、更便于理解,从而帮助更多的人学好数学; 让数学变得更有趣、更好玩、更有吸引力,从而帮助更多的人喜欢数学。



## 1.2 了解 Hawgent 皓骏

Hawgent 皓骏动态数学软件能够处理几何、代数、三角、概率、统计、算法、微积分等数学知识内容。目前已经实现的核心功能主要包括以下几个方面:

#### ★动态数学作图

能够直接构造出几十种常见的几何图形,当图形被拖动、平移、旋转或放缩过程中,图形之间的几何性质始终保持不变。具体来说,包括:

点:能直接绘制直线、圆弧、曲线、轨迹等对象上的点,中点、垂足、坐标点、比例点、旋转放缩点、直线或圆锥曲线上的参数点,两条直线的交点、直线与圆锥曲线的交点、直线与圆弧的交点,等等。

直线:能直接绘制向量、射线、直线,平行线、垂线、角平分线、一个点和一个约束条件确定的直线、两个约束条件确定的直线、一次方程对应直线、两圆的公切线、圆锥曲线的切线。

多边形:能直接等腰三角形、直角三角形、等边三角形、平行四边形、等腰梯形、正方形、(圆内接/圆外切)正n边形等,n可以是字母或表达式。

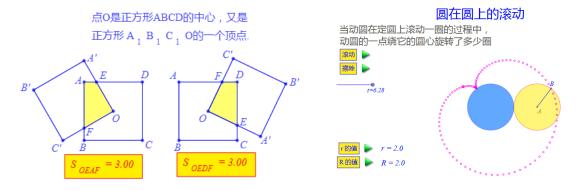
圆:能直接绘制已知圆心和半径的圆(3种方式)、已知圆心和切线的圆、已知直径的圆,经过三个点的圆。

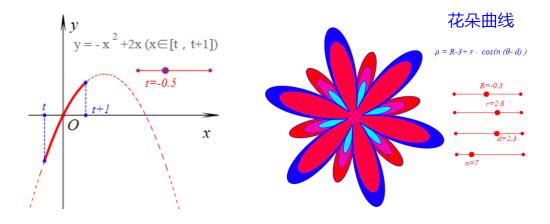
圆弧: 能直接绘制经过三个点的圆弧、角所对的圆弧、已知圆上的弧。

标注:能直接对线段和角进行标注,并可以设置标注符号的数量、间隔值、长度等,标注符号可以被单独选中。

- 一般函数/方程曲线:直接输入即可绘制对应的曲线,表达式可以是显性方程、参数方程或极坐标方程,表达式中的系数、自变量范围、曲线样点个数可以是字母或表达式,可以设置是否画样点、样点大小、样点的间断值、是否填充为矩形区域、是否绘制折线段等属性。
- 二次曲线:可直接绘制过三个点的标准抛物线、过五点的一般圆锥曲线、二次方程对应的二次曲线、已知焦点准线和离心率的圆锥曲线、已知焦点并过定点的椭圆/双曲线、已知焦点和长/实半轴的椭圆/双曲线。

迭代:包括几何迭代和变量迭代;几何迭代中可显示或隐藏迭代中的某个对象;变量迭代中可具有多个迭代变量,可设置迭代变量的初始值和表达式,迭代变量可作为点的坐标绘制对应的图像;可选择是否完整迭代,迭代次数可以是字母或表达式。





### ★动态图形变换

包括反射、平移、旋转、放缩、仿射、中心对称、点绕点旋转放缩等变换方式;选择要进行变换的对象和变换的条件,可以直接进行图形变换;若变换的条件是动态的或是参数,那么变换图形会根据变换条件的改变而改变。

反射变换: 选中要变换的对象和作为对称轴的直线即可直接进行反射变换。

平移变换: 指定要平移的对象和两个点确定的向量即可直接进行平移; 选中要变换的对象然后分别输入 x 轴平移量、y 轴平移量即可进行平移, 这里的平移量可以是字母或者表达式。

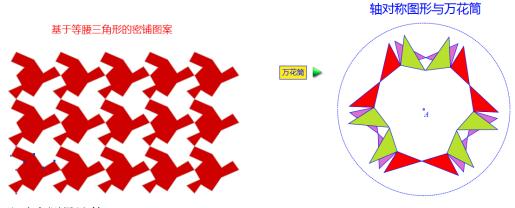
旋转变换:指定要旋转的对象、旋转中心、旋转角度即可进行旋转,其中旋转中心还可以通过输入 x 坐标、y 坐标的方式而实现,这里的 x 坐标、y 坐标、旋转角度均可以是字母或者表达式。

放缩变换: 指定要变换的对象、放缩中心、沿 x 轴放缩量、沿 y 轴放缩量即可进行放缩,其中放缩中心还可以通过输入 x 坐标、y 坐标的方式而实现,这里的 x 坐标、y 坐标、沿 x 轴放缩量、沿 y 轴放缩量均可以是字母或者表达式。

中心对称点: 指定要变换的点和对称中心即可做出中心对称点。

旋转放缩点:指定要变换的点、中心、旋转角度和放缩倍数即可做出旋转放缩点,这里的旋转角度、放缩倍数均可以是字母或者表达式。

仿射变换:可以进行坐标系到三角形的仿射变换,也可以进行三角形到三角形的放射变换。



#### ★动态测量计算

具体来说,包括:

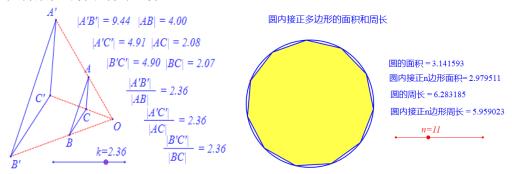
能够测量点的坐标、两点的距离、点到直线的距离、角度、两直线的夹角、多边形的面积、随机整数、随机浮点数等等。

测量结果能够自动以"字母+数字"的变量形式被记录,测量结果能够参与计算;测量和运算的结果允许利用指定的变量进行表示和记录。

具有嵌套测量功能,即支持形如利用 n 直接表示 f (n)的测量形式和结果。

当图形发生变动时,测量和计算结果会对应发生改变。

支持数值计算和符号运算。



### ★程序运行环境 它的功能包括:

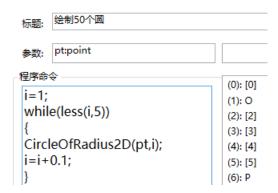
具有能够编写程序、运行程序的环境。

支持数值计算和符号运算。

支持赋值、判断、循环等语句;能够自定义函数,也可以调用系统内部的函数,函数的 参数支持通过鼠标选择和键盘输入等方式获得。

每个程序按钮可以有多个动作,按钮的动作可以自动连续播放,可以设置不同动作之间自动播放的等待时间。



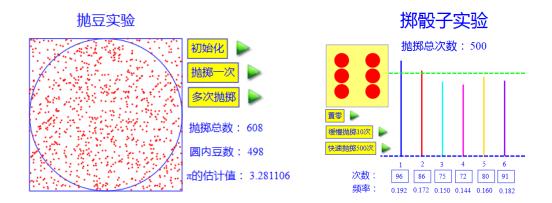


### ★随机过程模拟

具体来说,包括:

根据系统提供的随机数、符号函数、动画控制函数等函数可以进行随机过程实验,例如 模拟抛硬币、抛豆子和掷骰子等随机实验过程;

在随机实验进行过程中,能够即时统计实验过程中的实验的总次数、每种结果出现的次数和频率等数据。

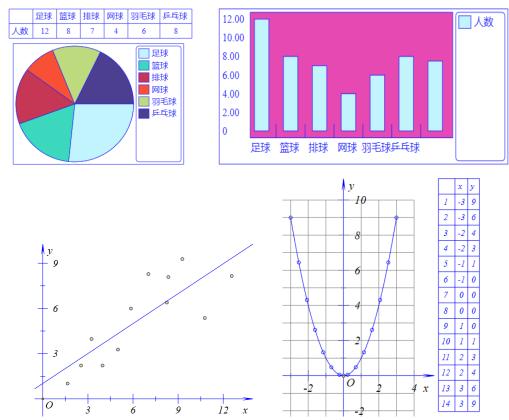


## ★动态数据处理

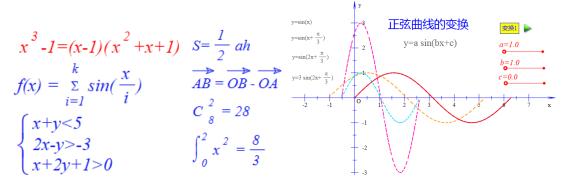
能够生成统计表格,可以同时显示统计表格对应的柱形图、折线图和饼形图。统计表格能够与曲线关联,显示绘制曲线样点的纵、横坐标。

能够自动计算、记录和显示变量迭代中的数据。

具有回归分析、函数拟合等功能。



★智能公式文本: 具有普通文本和公式文本,在普通文本中文本内容能够自适应文本的宽度,在公式文本中利用键盘输入的数学公式能够自动以传统格式显示,包括分数、根式、乘方、求和、求积、向量、上标、下标、上文本、下文本、方程组、不等式组、文本颜色、填充颜色等格式和内容。



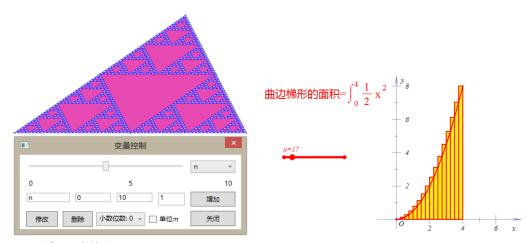
## ★动态参数控制

大致来说,包括:

变量控制框:可同时设置多个变量,为每个变量分别设置最小(大)值、变化间隔、小数位数、是否以 $\pi$ 为单位等属性。

变量拖动尺:每个变量尺可单独设置变化范围和小数位数,能拖到左右端点位置。

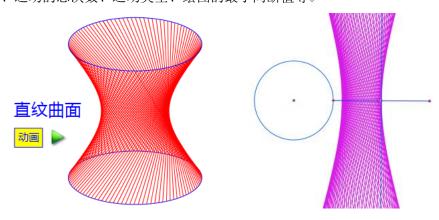
动画控制框:可对所有半自由点控制运动或停止,能分别设置每个对象的运动类型、运动步数、运动间隔时间等。



#### ★对象跟踪轨迹

跟踪: 所有数学对象均可以被跟踪,单击作图区空白处跟踪不会消失但可以被擦除并重新显示,可以设置跟踪样本的数量。

轨迹:主动点可以有单个或多个,可以针对每个主动点设置运动区间、每个运动区间的运动次数、运动的总次数、运动类型、绘图的最小间断值等。



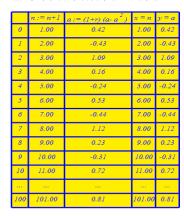
### ★自动推理计算

具体来说,包括:

能自动记录几何图形之间的逻辑关系,从而使得图形在变化过程中数学性质保持不变。 根据图形之间的逻辑关系,能够自动推导和计算图形之间的代数关系,从而能够实现图 形变换、动态测量、几何迭代、轨迹生成等过程。

具有智能推理和计算功能:测量得到的结果,能够继续进一步的运算;能够利用指定的 变量记录测量或运算的结果。

能够自动求解方程、方程组、不等式等常见代数问题。



## ★动态资源管理

媒体对象:具有普通文本和公式文本,左右拖动普通文本边界时内容能自动换行,公式文本包括分数、根式、乘方、积分、上下标、乘积、求和、文本颜色、填充背景、下文本、上文本、向量、上划线等格式和内容:能够插入图片、音频、视频和语音合成等对象。

对象编辑:对象能够前后移动;文本/图片能够与点进行关联;点/线/圆/曲线/文本/按钮被复制、粘贴后仍然是点/线/圆/曲线/文本/按钮,具有宏与迭代功能。

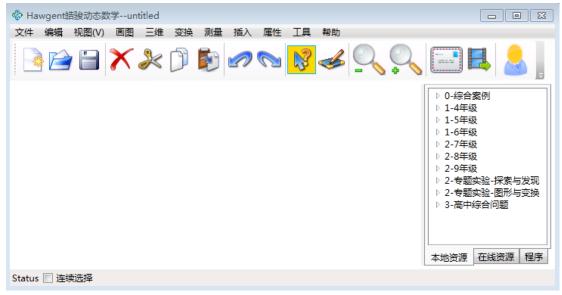
命令编辑:可以对现有的菜单、命令进行编辑,也可以自定义作图命令并添加到指定的位置。



## 1.3 认识软件界面

为了更加尽快熟悉 *Hawgent* 动态数学软件的上述功能,让我们首先了解一下它的工作界面吧。

Hawgent 动态数学软件启动后窗口如下:



利用 Hawgent 动态数学软件工作,主要就是通过菜单栏中的菜单命令来实现。菜单栏中包括缺省的【文件】【编辑】【视图】【画图】、【三维】、【变换】、【测量】、【插入】、【属性】、【工具】和【帮助】等菜单项。

菜单栏下是常用工具栏,就是将经常用的命令放在这里,提高我们的工作效率。



从左至右,依次是:

新建:建立一个新的 Hawgent 皓骏文档;

打开: 打开一个新的 Hawgent 皓骏文档;

保存:保存当前的 Hawgent 皓骏文档;

删除: 删除在工作区中所选择的对象;

剪切:将所选择的对象,移动到剪切板当中;

复制:将所选择的对象,复制到剪切板当中;

粘贴:将剪切板中的对象,粘贴到工作区当中;

上一页: 激活当前文档中的上一个案例, 如果存在的话;

下一页: 激活当前文档中的下一个案例, 如果存在的话;

选择: 让光标(由画图状态)转换到选择状态;

画图: 让光标(由选择状态)转换到画图状态;

缩小:缩小某个对象。如果什么都不选择,则缩小坐标系的单位长度;若选择的是点,则缩小点的大小;若选择的是线段,则缩小画线的宽度;所选择的是文本,则缩小文本的字号; ......

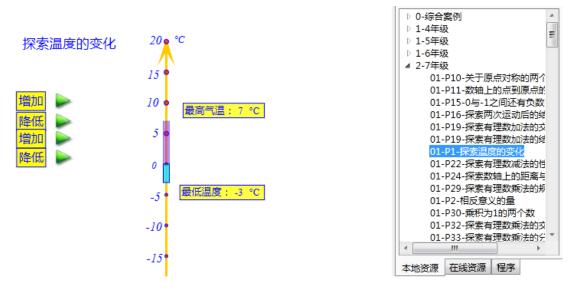
放大:放大某个对象。如果什么都不选择,则放大坐标系的单位长度;若选择的是点,则放大点的大小;若选择的是线段,则放大画线的宽度;所选择的是文本,则放大文本的字号; .....

工具栏下面的空白区域就是作图等工作的主要区域,简称为作图工作区。

作图工作区下方是状态栏,具有"连续选择"的开关,选择该选项后,不需要按住 Ctrl 键即可连续选择多个对象;被选择的对象,其序号会显示在状态栏的右侧部分。



作图工作区右侧是资源管理区,其中本地资源选项框中可以显示出本地已经加载的资源,如下图所示,单击某个目录下方的资源可以直接打开该文件.



在线资源选项卡,则可以更新和下载 Hawgent 皓骏服务器上所提供的最新资源,打开后,会自动保存到本地.

# 第二部分 几何图形与变换

## 2.1 经过一点的直线

在平面上,经过一个点的直线有多少条呢?我们可以在计算机上画出经过任意两个点的一条直线,然后其中一个点位置固定的情况下,拖动另外一个点,引导学生探索这个问题。

- (1) 在新的文档页面中, 勾选状态栏中的连续选择。
- (2) 单击【图画】工具,作出两个自由点 A 和点 B。
- (3) 单击【选择】工具依次选择点 A 和点 B,打开画图菜单,选择直线/直线作出一条直线。
  - (4) 打开画图菜单,选择跟踪。跟踪直线 AB。

拖动点 B,可以观察到经过点 A 的直线的运动轨迹,如图 2.1.1 所示:

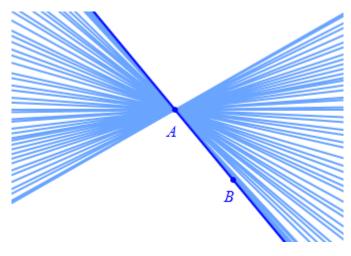


图 2.1.1

### 【思考与练习】

如上图所示,蓝色圆以点 A 为中心、经过点 B,点 C 是圆 A 上的任意点,红色圆以 C 为圆心经过点 B (图 2 )。当点 C 在圆周上运动时跟踪圆 C 留下的图形,如图 2.1.2 所示,这叫心脏线或者苹果曲线。请你自己动手展示生成苹果曲线的过程,结果如图 2.1.3 所示。

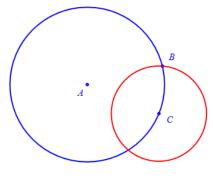


图 2.1.2

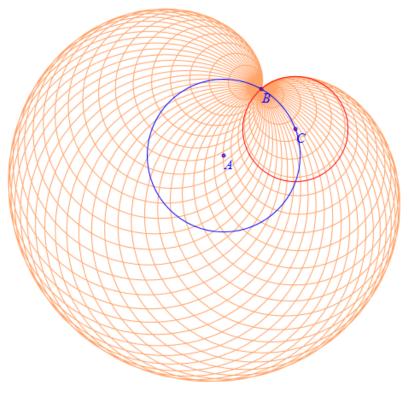
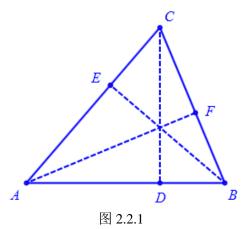


图 2.1.3

## 2.2 三角形的三条高线

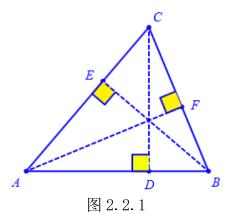
利用动态几何软件画出的图形,其形状可以被任意改变,而图形在改变过程中,所包含的几何关系始终保持不变,这可以帮助我们探索与发现图形中所隐含的内在规律。

- (1) 在新的文档页面中, 勾选状态栏中的连续选择。
- (2) 执行【画图】菜单下的【参数点】命令,输入横坐标-2,纵坐标 0,确定后作出 坐标为 (-3,0) 的参数点 A; 同理坐标为 (-3,0) 的参数点 B,单击【画图】工具在任意恰当 位置作出自由点 C,然后依次连接画任意三角形 ABC。
- (3) 依次选择点 A 和线段 BC,执行【画图】菜单下【约束点】命令的【垂足】命令作出点 A 到对边 BC 的垂足 D 和垂线段 AD,同理作出点 B 到对边 CA 的垂足 E 和垂线段 BE;作出点对 C 到边 AB 的垂足 E 和垂线段 CF。
  - (4) 单击【选择】工具,返回到选择状态。
- (5) 同时选择直线 AD、BE 和 CF,在属性菜单中设置【画线类型】为: 虚线。结果如图 2.2.1 所示:



对于垂直关系,我们往往需要标注出直角符号。例如要将 ∠ADC 的角标注出来,就首先要选择三个点,分别是角的始边上的点、角的顶点、角的终边上的点,然后执行标注角的命令。操作步骤如下:

- (6) 依次选择点 A、点 D 和点 C,单击【画图】菜单中的【标注角】命令。
- (7) 重复类似操作标注 ∠BEA 和 ∠CFB。
- (8) 选择角的标注符号,可以通过【属性】菜单中的【填充】工具将其内部填充。结果如图 2.2.2 所示:



#### 【思考与练习】

三角形的三条中线、三条角平分线也分别交于一点,请你动手完成展示和探索这两个性 质的内容。

要做三角形中的这类交点,可以通过【画图】菜单下【点】中的【内心】、【外心】和【垂心】等命令直接实现。

## 2.3 测量三角形的内角和

利用计算机作出的三角形,可以被任意改变形状,能够更加普遍地探究三角形三个内角测量值之和的规律。

- (1) 在新建文档中, 作点 C 可任意拖动三角形 ABC (参考 2.2 (1))。
- (2) 依次选择点 A、点 B 和点 C,执行【测量】菜单下的【角度】命令,结果得到 $\angle CAB$ 的测量值,如图 2.3.1 所示:

 $\angle ABC = 50.596023$ °

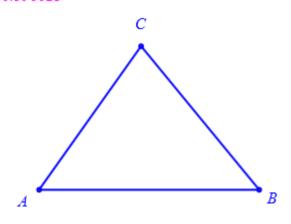


图 2.3.1

当图形比较简单时,我们一般习惯用角的顶点的名字表示这个角,例如在这里表示为 $\angle B$ ,所以我们可以将角的名称进行修改。操作步骤如下:

(3) 右击测量文本,如图 2.3.2 进入编辑状态,将字符 ABC 修改为 B,显示位数改为 2 位如图 2.3.3 所示,然后确认。



图 2.3.3

- (4) 重复类似操作测量角 A 和角 C 的值,并且按照上面的要求编辑测量结果名称。
- (5) 单击【测量】菜单下的【测量表达式】命令,打开测量表达式对话框。如图 2.3.4

所示,由于之前测量的值分别用 v001、v002、v003 记录,这是计算机系统自动产生的名称,而且这样做的好处是:之前测量得到的结果容易被以后所运用。

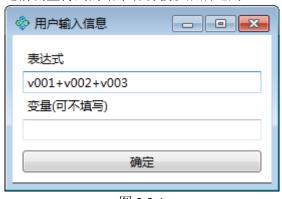


图 2.3.4

(6) 在【表达式】编辑框中输入: v001+v002+v003; 然后单击单击【确定】按钮得到测量结果。

#### v001+v002+v003 = 3.14

图 2.3.5 表达式显示结果

(7) 此时会发现结果为弧度制的 3.14, 我们右击表达式结果, 如图 2.3.6 修改参数:

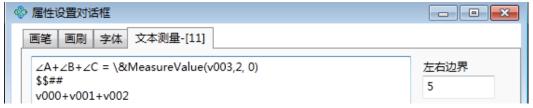


图 2.3.6

(7) 最后关闭参数对话框,结果如图 2.3.7 所示。拖动点 C,观察拖动过程中角度值得变化。

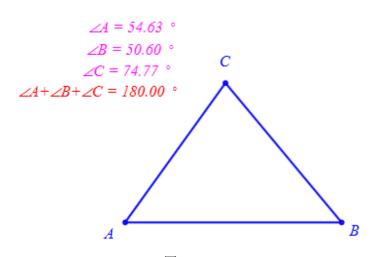


图 2.3.7

拖动点 C, 观察拖动过程中角度值的变化情况。

#### 【思考与练习】

如何测量三角形的外角和?同时实验探索或检验三角形外角和的规律。

## 2.4 三角形内角的直观说明

我们可以通过将一个角平移、将另一个角旋转直观地说明了三角形内角和等于平角的 道理,见下节的【思考与练习】。在这一节中,将通过同时两个角旋转来直观说明。

- (1) 画任意三角形 ABC。
- (2) 选择 A 点和 B 点,执行【画图】菜单下【约束点】中的【中点】命令作 AB 边的中点 D、同理作出 BC 边的中点 E、AC 边上的中点 F。
- (3) 同时选择点 A、点 B 和点 C,执行【画图】菜单下【约束点】中的【重心】命令,作出三角形 ABC 的重心 G。结果如图 2.4.1 所示:

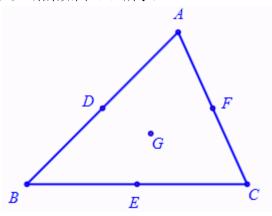


图 2.4.1

下面要完成以点 E 为中心、以 t 为旋转角,将  $\angle C$  旋转:

(4) 打开菜单中执行【插入】菜单下【变量】命令,此时,在弹出的输入框中填入变量名: *t*、最小值 0: 和最大值 180。如图 2.4.2 所示:

፟ 常用户輸入信息	
变量	
t	
最小值	
0	
最大值	
180	
确	定

图 2.4.2

确认后出现变量控制尺,用鼠标拖动线段上的点可以改变变量 t 的值。



图 2.4.3 变量控制线条

(5) 依次选择  $C \times E \times G \times F$  四点(注意 F 在最后),执行【变换】菜单中【旋转】命令,如图 2.4.4 所示:在弹出的用户输入对话框中输入:t,单击【确定】按钮完成。结果作

出点H、点I、点J。



图 2.4.4

(6) 依次选择点 E、点 H、点 I和点 J,单击鼠标右键,在快捷菜单中单击【多边形】 命令,作出多边形 FHIJ。结果如图 2.45 所示:

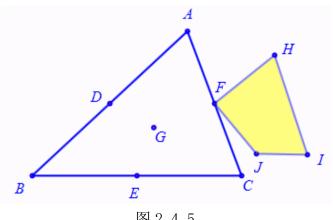


图 2.4.5

\*注:以下为方便起见,在不至于引起歧义的情况下,在描述执行多级菜单命令 的时候约定采用"执行【1】【2】…… 【n】"来代替"执行【1】菜单下的【2】子 菜单中的······  $\{n\}$ 命令。" 如此次的"作执行【画图】【直线】【垂线】命令"等 价于"执行【画图】菜单下【直线】子菜单中的【垂线】命令",

此外,有些简单常见的操作往后会不在说明具体操作过程而只是简单描述。

下面要完成以点 D 为中心、以-t 为旋转角,将  $\angle B$  旋转:

- (7) 依次选择 B、E、G、F 四点 (注意 F 在最后),执行菜单中【变换】【旋转】命 令,在弹出的用户输入对话框中输入:-t,单击【确定】按钮完成。结果作出点K、点L、 点M。
  - (8) 依次选择点 E、点 H、点 I 和点 J, 执行【作图】【多边形】命令, 作出多边形 DLMN。
- (9) 依次选择点 A、点 D 、点 G 和点 F,执行【画图】【多边形】命令,作出多边形 ADGF。结果如图 2.4.6 所示:

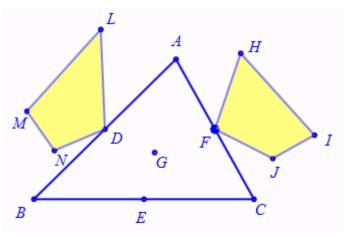


图 2.4.6

为了使得图像美观,可以给三个多边形填充不同的颜色:

(10) 选择一个多边形的,在【属性】菜单下【填充颜色】命令中选择一种颜色为多边 形设置填充颜色。按此操作给其他两个多边形填上恰当的颜色,如图 2.4.7 所示。

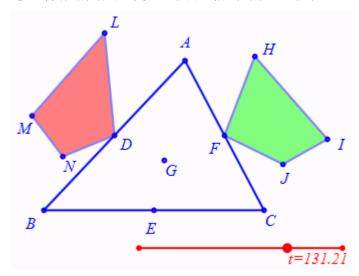


图 2.4.7

拖动变量控制尺上的控制点, 观察变化过程。

## 【思考与练习】

(1) 将三角形通过如图 2.4.8 所示变换,可以直观说明三角形的面积等于底边与高的乘 积的一半。请你在计算机上完成能动态展示这一过程的实验。

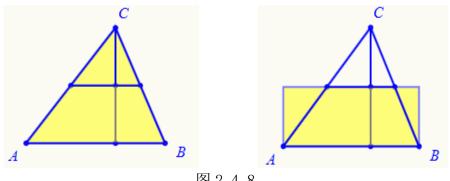
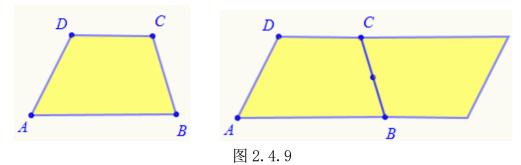


图 2.4.8

(2) 将梯形形通过如图 2.4.9 所示变换,可以直观说明梯形的面积等于上、下底边之和

与高的乘积的一半。请你在计算机上完成能动态展示这一过程的实验。



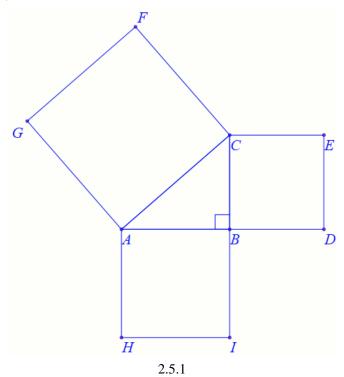
(3) 放缩操作和旋转操作的方法类似,请你动手设计一个放缩对象的变换。

## 2.5 勾股定理的直观说明

勾股定理是几何中最著名的定理之一,也叫毕达格拉斯定理。据说毕达格拉斯发现这个定理时认为这是神赐予的,于是杀了100头牛祭祀,大摆宴席。

到今天为止,证明勾股定理的方法有上百种之多。在本节中我们将通过计算机的动态演示直观地说明勾股定理。

- (1) 新建一个文档。
- (2) 执行【画图】【参数点】【坐标点】命令,作出坐标点A (-2 ,0),同理作出坐标点B (2,0)。
- (3) 依次选择点 B 和点 A,执行【画图】【多边形】【直角三角形(已知一条直角边的直角三角形)】命令,作出直角三角形 ABC。
  - (4) 依次选择点 C 和点 B,执行【画图】【多边形】【正方形】命令作出正方形 CBDF。
  - (5) 依次选择点 A 和点 C,执行【画图】【多边形】【正方形】命令作出正方形 ACFG。
- (6) 依次选择点 B 和点 A,执行【画图】【多边形】【正方形】命令作出正方形 BAHI。 结果如图 2.5.1 所示:



下面我们要将正方形 ABIH 分割成四部分。操作步骤如下:

- (6) 线段 AI、BH, 作线段 AI、BH 的交点 J。隐藏线段 AI、BH。
- (7) 依次选择点 J 和线段 AC,作过点 J 与 AC 垂直且的直线。作出该该直线与线段 HI 的交点 K。隐藏刚才所作的直线。结果如图 2.5.2 所示:

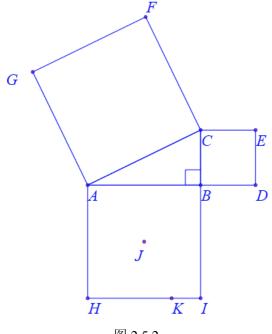


图 2.5.2

- (8) 依次选择点 J、K,执行【变换】【旋转】命令,输入旋转次数: 1次,旋转角度: 90,点击【确定】得出 K 绕 J 旋转 90 度得出的点 L。
  - (9) 依次选择点 J、K、I、L,执行【画图】【多边形】【多边形】命令,作出多边形 JKIL。
- (10) 依次选择多边形和点 J,执行【变换】【旋转】命令,设置旋转次数为 3 次,角度为: 90。点击【确定】完成该步操作。结果如图 2.5.3 所示。

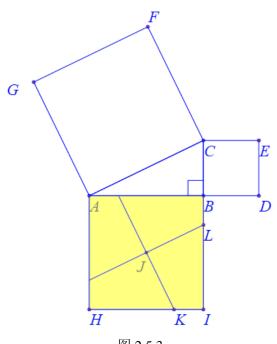
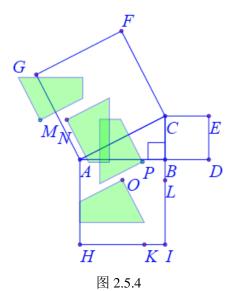


图 2.5.3

将正方形 ABIH 内的四部分分别以不同的向量平移而出,操作步骤如下:

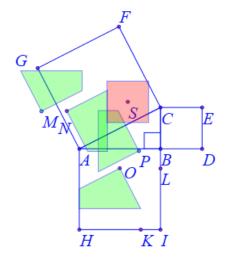
- (11) 作线段 JA、JG、JF、JC,并分别在这四条线段上作点 M、N、O、P。
- (12)依次选择四边形 AJ 和点 J、点 P,执行【变换】【平移】命令,作出四边形 AJ 按向量 JP 平移后的图形,隐藏四边形 AJ,隐藏线段 JC,把平移后的四边形的填充颜色改为绿色。

- (13)依次选择依次选择四边形 BJ 和点 J、点 M,执行【变换】【平移】命令,作出四边形 BJ 按向量 JP 平移后的图形,隐藏四边形 BJ,隐藏线段 JA,把平移后的四边形的填充颜色改为绿色。
- (14)依次选择依次选择四边形 IJ 和点 J、点 N,执行【变换】【平移】命令,作出四边形 BJ 按向量 JP 平移后的图形,隐藏四边形 IJ,隐藏线段 JG,把平移后的四边形的填充颜色改为绿色。
- (15) 依次选择依次选择四边形 HJ 和点 J、点 O,执行【变换】【平移】命令,作出四边形 BJ 按向量 JP 平移后的图形,隐藏四边形 HJ,隐藏线段 JF,隐藏点 J,把平移后的四边形的填充颜色改为绿色。完成后结果如图 2.5.4 所示



下面将正方形 CBDE 按照指定的向量平移而出,操作步骤如下:

- (16) 作线段 CD 和 BE, 作线段 CD 和线段 BE 的交点 O, 隐藏线段 CD 和线段 BE。
- (17) 作线段 AF 和 CG, 作线段 AF 和线段 CG 的交点 R, 隐藏线段 AF 和线段 CG。
- (18) 作线段 OR, 作线段 OR 上一点 S。
- (19)依次选择点  $C \times B \times D \times E$ ,执行【画图】【多边形】【多边形】命令,作出多边形 CBDE。
- (20) 依次选择依次选择四边形 CBDE 和点 Q、点 S,执行【变换】【平移】命令,作出四边形 CBDE 按向量 QS 平移后的图形,隐藏四边形 CBDE、线段 QR 和点 Q 把平移后的四边形的填充颜色改为红色。完成后结果如图 2.5.5 所示:



下面作出两个直角边对应的正方形内部剪切、平移到斜边对应的正方形内部的动画按钮。操作步骤如下:

(21) 执行【插入】【常用按钮】【对象一次运动】命令,如图 2.5.6 所示在弹出的对话窗口中把标题改为:"剪拼",的程序命令修改为:

ObjAnimation(42,50,3);

ObjAnimation(43,50,3);

ObjAnimation(44,50,3);

ObjAnimation(45,50,3);

ObjAnimation(57,50,3);

点击【修改动作】按钮再点击【确定】按钮完成该步操作。

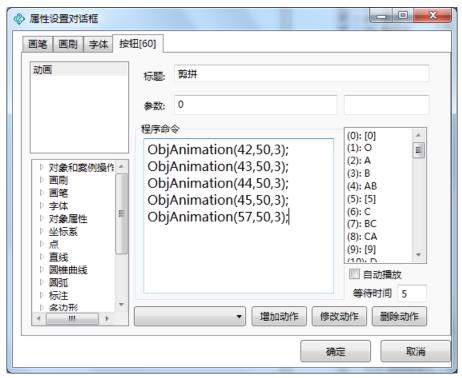


图 2.5.6

下面作出还原按钮。操作步骤如下:

(22) 执行【插入】【常用按钮】【对象一次运动】命令,在弹出的对话窗口中把标题改为:"还原",的程序命令修改为:

ObjAnimation(42,50,4);

ObjAnimation(43,50,4);

ObjAnimation(44,50,4);

ObjAnimation(45,50,4);

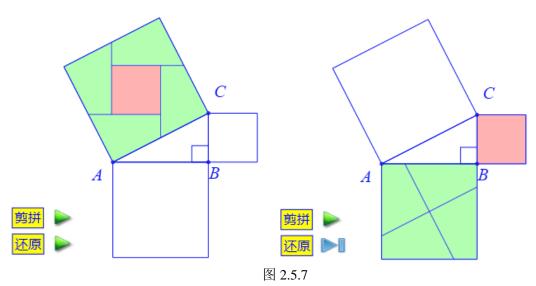
ObjAnimation(57,50,4);

点击【修改动作】按钮再点击【确定】按钮完成该步操作。

下面对作图区中不需要显示的对象隐藏,具体如下:

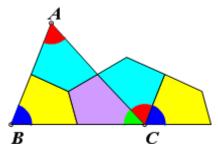
(23) 将点A、点B和点C之外的其他所有点隐藏。

到现在整个内容就制作完毕,单击【剪拼】按钮,结果如图 2.5.7 左所示;单击【还原】按钮,结果如图 2.5.7 右所示。

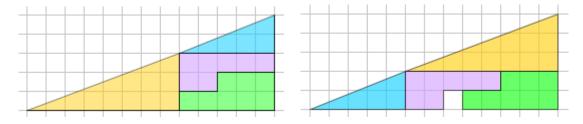


### 【思考与练习】

(1) 在上一节中我们通过旋转的方式直观地说明了三角形内角和为平角。我们还可以将角A旋转到角C的位置,然后将角B平移到角C的位置,同样能直观地说明三角形内角和为平角这个问题。请你动手完成它的直观展示过程。



(2)过去直接进入证明缺少发现的过程,因此几何定理仿佛是从天上掉下来的。新课程改革中则强调先从直观发现图形性质,积累了一定知识之后再进入逻辑论证。这时出现了一个新的问题:既然我们发现了图形的性质,为什么还要进行推理证明呢?似乎没有这个必要!怎么让学生感到不能光凭直觉研究图形性质?请你动手制作了下面的内容,将左边的图形剪开平移重新组合后得到了右边的图形。并增加一个标题:怎么少了一块?



## 2.6 在平面上滚动的车轮

圆形车轮在水平路面上无滑动地滚动的特点就是圆心在水平方向上经过的距离,即车轮行驶过的路面长度,等于车轮在路面上滚动所经过的弧长。根据这一原理,我们就可以模拟车轮在平面上滚动的过程。

首先作出一个路面,具体操作如下:

(1) 在空白文档中, 作直线 AB, 并在直线 AB 上取点 C。如图图 2.6.1 所示。

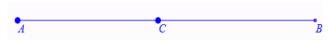


图 2.6.1

(2) 为后面的步骤操作简便起见,打开【视图】菜单把【对象框】勾选,这时我们可以在打开的对象框中看到各个对象的序号等信息。如图 2.6.2 所示:

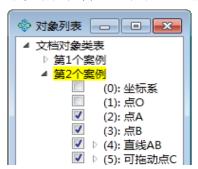


图 2.6.2

接着作出一个会旋转的车轮,操作步骤如下

- (3) 选择点 C 和直线 AB 作执行【画图】【直线】【垂线】\*命令。
- (4)在垂线上作出一点 D,依次选取 D、C 执行【画图】【圆和圆弧】【已知圆心和半径的圆】命令。
- (5) 选择点 A、C,执行【测量】【距离】得出 AC 的距离,接着用同样的方法测量 DC 的距离,再执行【测量】【表达式】按图 2.6.3 输入测量表达式,单击确认。(注:其中 v000 是第一次测量的 AC 的距离,v001 是第二次测量 DC 的距离,本次测量的结果保存在变量 v002 中。)



图 2.6.3

(6) 依次选择点 C 和圆心 D,执行【变换】【旋转】,在弹出的输入框中输入旋转角度 v002 (测量表达式的计算结果) 如图 2.6.4 所示,点击确认得到点 E



图 2.6.4

(8) 依次选择点 E 和圆心 D,执行【变换】【旋转】命令,在弹出的输入框中输入旋转次数 3,旋转角度 90,点击确认得到与 E 一起等分圆周的另外三个点,用直线如下图所示把它们连接起来。

接下来作点C在直线AB上运动的动画

(9) 执行【插入】【常用按钮】【对象重复运动】命令,在弹出的窗口中按图 2.6.5、图 2.6.6 分别修改"动画"和"停止"的参数。



图 2.6.5

(注:修改完要按"修改动作"按钮进行保存才能继续下一项操作)



图 2.6.6

确认后得到如下图 2.6.7 所示的动画按钮。



2.6.7 生成的动画按钮

- (10) 在对象列表框中点击去掉(6)直线和(5)可拖动点 C 前的选钩使得这两个对象隐藏起来。
- (13)分别把余下点的名字隐藏。单击动画按钮,模拟轮子在水平路面上滚动的过程,如图 2.6.8 所示:

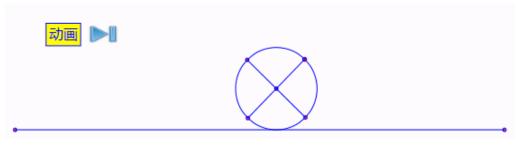


图 2.6.8

#### 【思考与练习】

(1) 跟踪车轮上的点 C, 然后再次单击动画按钮, 结果图 2.6.9 生成点 C 的跟踪图形:

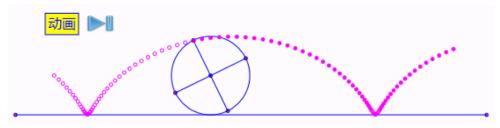
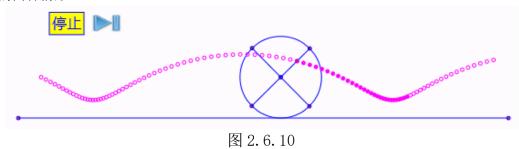


图 2.6.9

这种由旋转的轮子在无滑动的滚动中而生成的曲线,就叫做旋轮线。请你研究旋轮线的性质。

(2) 在轮子的半径上任意取一点,然后跟踪该点,如图 2.6.10 和图 2.6.11 所示,是其中的两种情形:



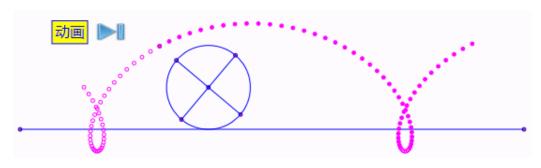


图 2.6.11

这种图形也是无滑动滚动的轮子边沿上的点而生成的曲线吗? 它们被称为广义旋轮线。

- (3)请你探索得出(1)中轨迹图形对应点 C的轨迹方程,也进一步尝试得到(2)中对应的轨迹曲线方程。
  - (4) 将轮子的半径大小改变, 重新制作上面的内容。

## 2.7 圆在圆上滚动

在固定圆上滚动的圆,跟踪其上面的一点所生成的曲线也是摆线。下面制作半径为 r 的动圆在半径为 R 的定圆上滚动的实验。

为了方便讨论问题,在这里我们将圆的半径取为整数。在计算机和数学语言中取整数的函数是 floor(a),结果表示比 a 小的最大整数。

首先作出半径为 floor(a)的定圆和半径为 floor(b)的动圆。

(1) 在新建文档中,勾选【视图】菜单中的【对象框】,如图 2.7.1 所示:在对象框中勾选(1)点 O 使坐标原点处于可见状态。

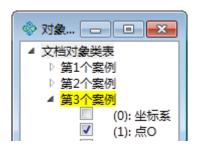


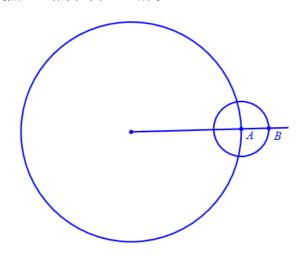
图 2.7.1

(2)执行【画图】【参数点】【坐标点】命令,参考图 2.7.2 作出坐标为 (floor(a)\*cos(t), floor(a)\*sin(t)),的点 A。



图 2.7.2

- (3) 以点 O 为圆心、OA 长为半径作圆。
- (4) 以点 A 为圆心、作半径长为 floor(b)的圆。执行【画图【直线】【射线】作射线 OA 与圆 A 的另一个交点 B。结果如图 2.7.3 所示:



- (4) 隐藏圆 A 和射线 OA。
- (5) 作出以点B为圆心经过点A的圆。

假定初始位置时,圆 B 与圆 O 相切于点 A,若点 B 绕 O 旋转了 t 弧度,则点 A 绕点 B 旋转的角度是: t\*floor(a)/floor(b)弧度。下面作出点 A 绕点 B 旋转的对应点:

(6) 依次选择点 *A* 和点 *B* 执行【变换】【旋转】命令,设置旋转次数: 1,旋转角: 180/*pi\*t\*floor(a)/floor(b)*,将点 *A* 绕点 *B* 旋转,得到点 *C* 和线段 *BC*。。结果如图 2.7.4 所示:

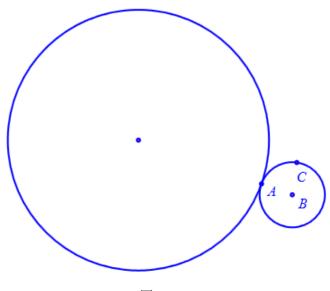


图 2.7.4

#### (8) 跟踪点 C。

下面增加点A的动画按钮。

设 k=floor(a)/floor(b)。当 k 为整数时,曲线由 k 支组成,圆 B 上的点描完 k 支后(即圆 B 绕圆 A 一周),返回到起始位置;当 k 为分数(k=g/h,g、h 为互素的整数)时,曲线由 g 支组成,圆 P 上的动点描完 g 支后(即圆 P 绕圆 O 旋转 h 周),返回到起始位置。

在这里,当 k 为整数时,让圆 P 绕圆 O 滚动一周;当 k 为分数时,让圆 P 绕圆 O 旋转 floor(b)周(若用 m 表示 floor(a)与 floor(b)的最大公约数,则旋转 floor(b)周的过程中,是 m 次的循环)。

"floor(a)被 floor(b)整除"可以用"floor(a)/floor(b)的小数部分小于任何一个正数"这句话解释;表示成计算机的数学语言就变为: (k-floor(k))<0.01(在这里用 0.01 表示一个小的正数)。

"floor(a)不被 floor(b)整除"可以用"k 的小数部分(的绝对值)大于 0"这句话解释;表示成计算机的数学语言就变为: (k-floor(k))>0。

上面的表达式,当floor(b)为负数时也成立,请自己验证。

- (9)打开测量表达式对话框, 计算 *floor(a)*的值, 计算 *floor(b)*的值, 计算 *floor(a)/floor(b)*的值。其中 *floor(a)/floor(b)*用变量 *v*002 记录。
- (10) 执行【插入】【常用按钮】【变量一次运动】,参考图 2.7.5 增加点参数 t 的动画按钮,标题改为"画图",命令改为:

ClearAllTrace();

 $\label{localization} Var Animation(t, 0, sign(v002-floor(v002))*floor(b)*2*pi+(1-sign(v002-floor(v002)))*2*pi, 100, 3);$ 

确定后作出变量 t 的按钮。

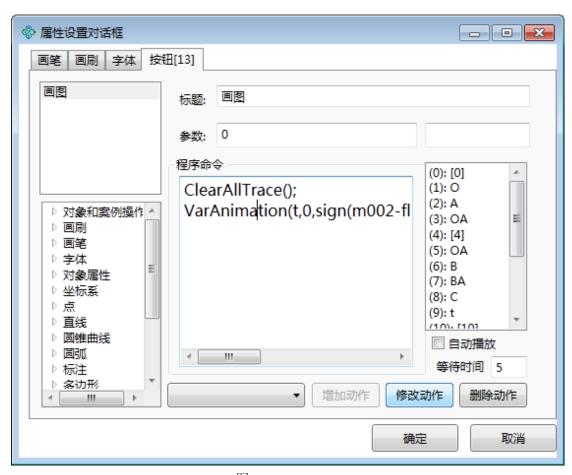


图 2.7.5

说明: ClearAllTrace()的作用是清除一起所画的轨迹。

增加可以控制定圆、动圆半径的按钮。操作如下:

(11) 仿照 (11) 增加改变参数 a 的按钮。如图 2.7.6 所示:标题改为"圆 O 的半径",命令改为:

VarAnimation(a,0,k,1,3);

第二个参数框填入: n:请输入参圆 O 的半径值 确定后作出变量 b 的按钮。

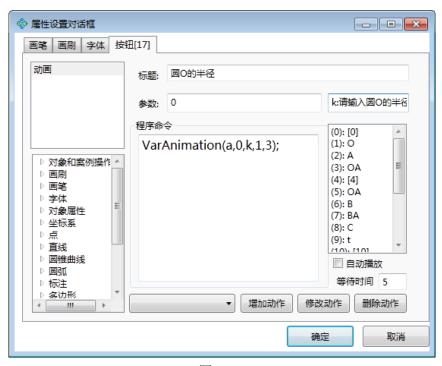


图 2.7.6

(12)类似地增加改变参数 a 的按钮。标题改为"圆 B 的半径",命令改为: VarAnimation(b,0,k,1,3);

第二个参数框填入: n:请输入参圆 B 的半径值

确定后作出变量 b 的按钮。

对作图区中的内容进行整理,操作如下:

- (13) 隐藏点 A,隐藏 floor(a)/floor(b)的测量文本。
- (14)将 floor(a)的测量文本的名称修改为"a=",将 floor(b)的测量文本的名称修改为"b="。 调整位置,结果如图 2.7.7 所示:

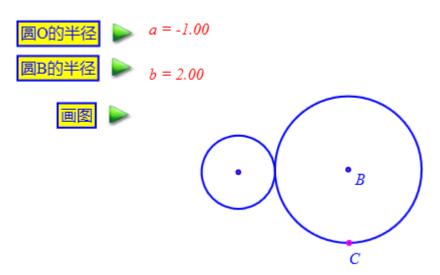


图 2.7.7

(15) 设置点 C 的跟踪对象的颜色,并将其移动到最后。

单击【动画】按钮,输入圆O半径2,圆B半径2,结果如图2.7.8所示:

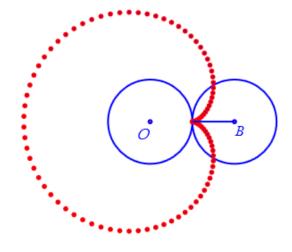
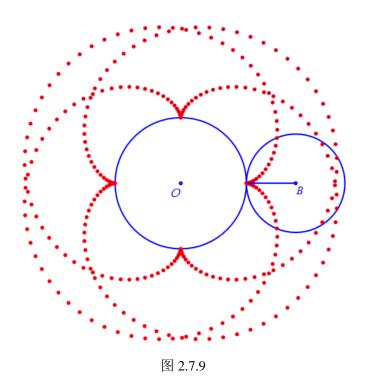


图 2.7.8

多次改变动圆的半径,再次单击【动画】按钮,其中之一结果如图 2.7.9 所示:



## 【思考与练习】

- (1) 在这里,我们所设计的实验中,a 和 b 的值能不能为负数,若为负数会有怎样的结果?
  - (2) 请你设计一个动圆在定圆内部滚动的实验。

## 2.8 翻折得到轴对称图形

轴对称图形有什么特点?如何设计轴对称图形?利用仿射变换和计算机的直观动态展示,可以加深学生轴对称图形相关概念和性质的理解。

首先作出以坐标原点 O 为中心的椭圆,并作出椭圆的左顶点和上顶点。操作如下:

(1) 执行菜单【画图】【圆锥曲线】【二次方程曲线】命令,如图 2.8.1 所示,在弹出的圆锥曲线作图对话框中,修改表达式,设置长半轴为 3,设置短半轴为: 2,单击【确定】按钮完成。打开对象列表窗口,使坐标系和原点 0 处于可见状态,结果如 2.8.2 图所示:

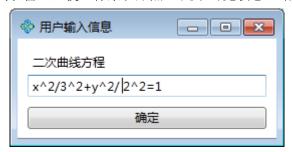


图 2.8.1

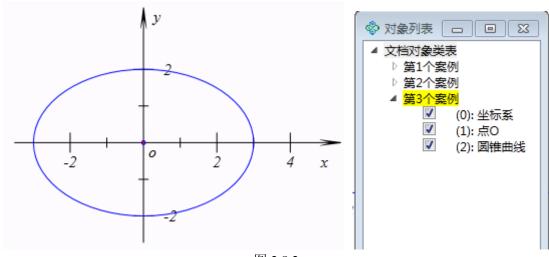


图 2.8.2

(2) 执行【画图】【参数点】【坐标点】命令,在如图 2.8.3 (左)的输入框中,输入左顶点的坐标 (-3,0)作出左顶点 A。类似操作,作出椭圆的上顶点 B (0,2)。然后在椭圆上任取一点 C。结果如图 2.8.3 (右)所示:

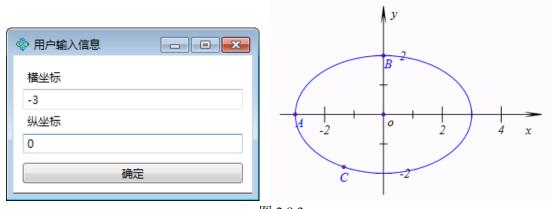


图 2.8.3

作出以 $\angle AOB$  为一角的平行四边形和以 $\angle COB$  为一角的平行四边形,操作如下:

(3) 依次选择点 A、点 O 和点 B,执行【画图】【多边形】【平行四边形】命令,作出平行四边形 AOBD;同理,依次选择点 C、点 O 和点 B,作出平行四边形 COBE。如图 2.8.4 所示:

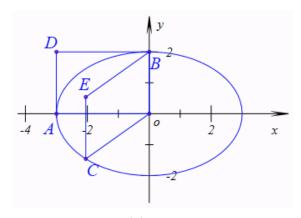


图 2.8.4

拖动点 C,四边形 COBE 就像一扇门,以 OB 直线为轴转动着。而四边形 AOBD 就像一扇静止的门。若将一个图形"画"在这扇转动的门上,那么就可以动态、直观地展示图形的翻折过程了。

那就是建立一个仿射变换,从点O、点A、点B确定的坐标系到点O、点C、点B确定的坐标系之间的一个变换。然后将在当前坐标系中画出的图形进行仿射变换即可。具体操作步骤如下:

下面我们任意画一个图形,然后按照指定的仿射变换进行变换。操作如下:

- (4) 任意画三角形 FGH。
- (5) 如图 2.8.5 所示: 在窗口右侧的程序标签栏中输入:

#### AffineTriangle(19,3,1,4,5,1,4);

然后按 F8 执行,即可作出三角形 FGH 从三角形 AOB 到三角形 COB 的仿射变换图像。



图 2.8.5

(6)给变换后的图像填充上黄色,拖动点 C观察图形的运动。也可作出点 C点动作按钮来进行控制。如图 2.8.6 所示:

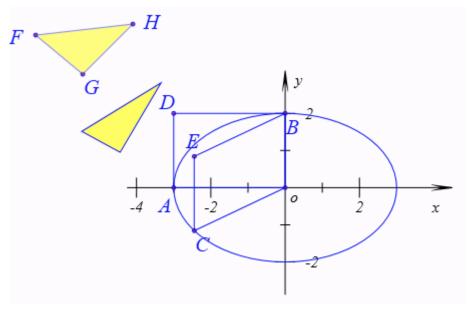


图 2.8.6

## 【思考与练习】

- (1)利用当前建立的仿射变换,你还可以继续对其他对象进行变换。请你利用【插入】 菜单中的【普通文本】命令插入一个段文本文字,按后设置你自己喜欢的字体和填充颜色, 然后将该文本进行仿射变换。
- (2)建立一个从三角形 COB 到三角形 AOB 的仿射变换,然后将多边形 IJK 进行变换,点 C 动画过程中,变换得到的多边形有什么特点?
- (3)将不需要显示的对象隐藏,例如上图中的坐标系、椭圆、平行四边形 AOBD 和平行四边形 COBE。

## 2.9 等分圆周设计的图案

在动态几何工具平台中,利用简单的几何图形和设置就可以设计许多漂亮的图案。下面我们设计一个漂亮的五角星。

- (1) 在新建文档中,作两个点 OA,依次选择 O、A 执行【画图】【圆和圆弧】【已知圆心和半径的圆】作出一个圆。
- (2) 在圆上作一点 C,依次选中 C和圆心 O,执行【变换】【旋转】在弹出的对话框中输入旋转次数: 4,旋转角度: 360/5。如图 2.9.1 所示,然后点击确定得出与 B 一起五等分圆周的点 C、D、E、F。

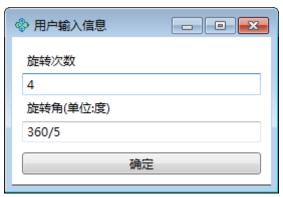


图 2.9.1

(3) 连接线段 AC、CE、EB、BD、DA。如图 2.9.2 所示:

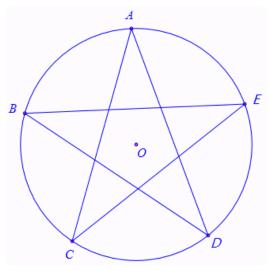


图 2.9.2

(4)如图 2.9.3 (左)所示,作出线段之间的交点。然后隐藏线段,结果如图 2.9.3 (右)所示:

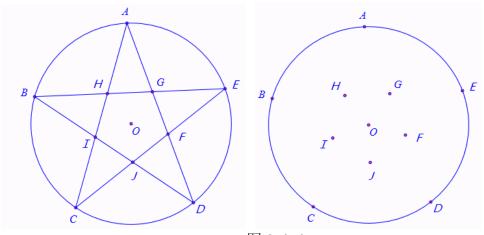
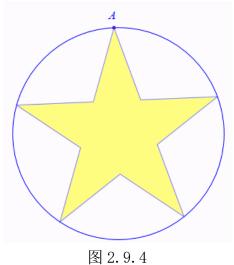


图 2.9.3

(8) 依次选择 B、G、C、H、D、I、E、J、F、K 共十个点,作多边形,; 隐藏除点 A和点O之外其他点,结果如图2.9.4所示:



(9) 右击多边形的边界, 打开其属性对话框, 如图 2.9.5 所示: 选择【径向填充】选项 卡,选择合适的渐变颜色,勾选【填充】,单击【确定】按钮完成。

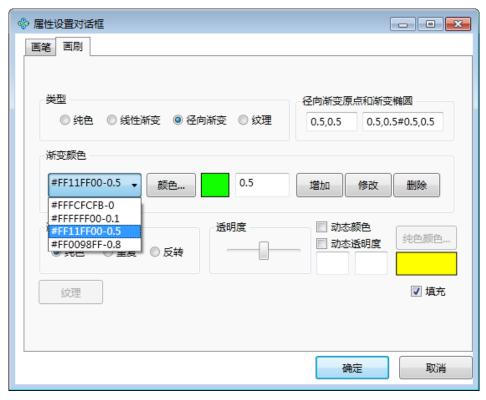


图 2.9.5

结果如图 2.9.6 所示,拖动点 B,可以让五角星绕其中心旋转。

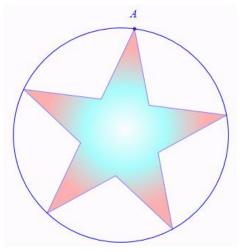
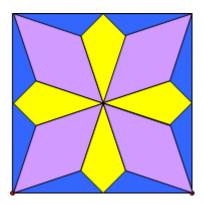


图 2.9.6 最终效果图

## 【思考与练习】

- (1)请你利用旋转的方式,通过点A作出圆内接正五边形的其他顶点。
- (2)请你设计出如图 2.9.7 两幅图案。



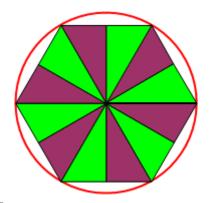


图 2.9.7

## 2.10 直线与圆的位置关系

首先作出一个圆和一条直线。具体操作如下:

- (1) 在新建文档中,作出一个圆 O。
- (2) 在圆上取一点A,测量AO的距离。
- (3) 插入参数 *m o*,取值范围为-2 到 2,如图 2.10.1 所示:



图 2.10.1

- (3) 依次选择 OA 两点, 作过 OA 的直线。
- (4) 选择该直线,执行【画图】【约束点】【直线上的参数点】,填入参数数为mo,如图 2.10.2 所示,确定后得到新点B。

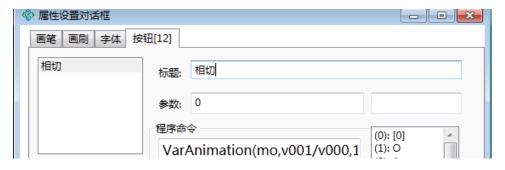


图 2.10.2

- (5) 测量 OB 的距离。
- (6)依次选择点 B 和直线 OA,执行【画图】【直线】【垂线】作出一条以 B 为垂足垂直于 OA 的直线。

增加直线 m 移动到与圆 O 相切的位置的动画按钮,操作如下:

(7) 执行【插入】【常用按钮】【变量一次运动】命令,按图 2.10.3 修改命令: *VarAnimation(mo,v001/v000*,1,20,3);



(8)确定后移动变量控制条的控制点可以控制直线的移动,见 2.10.4: 按相切按钮,直线会移到相切的位置,请注意过程中 OA、OB 之间的距离关系。

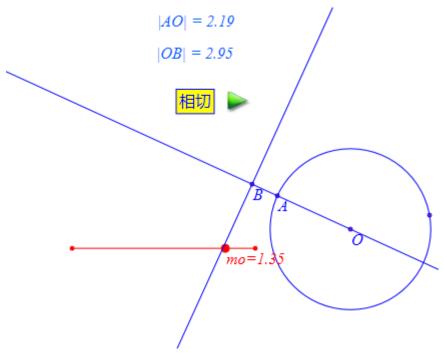


图 2.10.4 最终效果

## 【思考与练习】

请你利用另外一种思路设计一个探索直线与圆的位置关系的实验,能够方便地改变和表现直线与圆的三种位置关系。

# 2.11 圆与圆的位置关系

作出两个半径固定、而圆心可以被随意拖动的圆,具体操作如下:

- (1) 在新建文档中,插入变量 a,和 b,作坐标点 A (a, 0)、B (b, 0)。
- (2) 拖动变量可知点使得A点和B点分开到合适的位置。以点A为圆心作半径为2的圆,以点B为圆心作半径为3的圆。如图2.11.1所示:

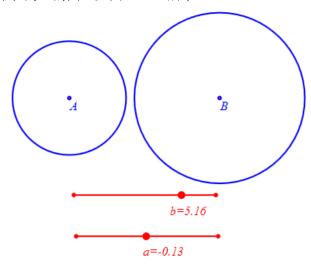


图 2.11.1

增加圆 A 移动到分别与圆 B 外切和内切的动画按钮,操作步骤如下:

(3) 作点 A 的动画按钮,如图 2.11.2 所示,设置动画频率为:300,设置参数 a 的范围为:a 到 b-5,选择运动类型为:3(一次运动),设置按钮的名称为:外切。



图 2.11.2 外切按钮参数

(4) 作点 A 的动画按钮,如图 2.11.3 所示,设置动画频率为:300,设置参数 a 的范围为:a 到 b-1,选择运动类型为:3 (一次运动),设置按钮的名称为:内切。



作出两圆的交点和公切线,具体操作如下:

(5)选择两个圆,通过【画图】菜单中【直线】子菜单下的【两圆的公切线】命令即可作出两圆的公切线段。

增加控制四条公切线显示或隐藏的开关按钮,操作如下:

(6)插入一个按钮,按图 2.11.4 修改参数(注意这里的 12、15、18、21 分别对应着对像列表中对应对象的序号。),按【增加动作】添加显示公切线的动作。接着按图 2.11.5 给该按钮增加隐藏公切线的功能,按【增加动作】再按确定完成该按钮。

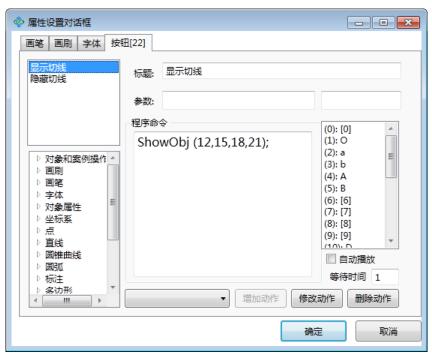


图 2.11.4

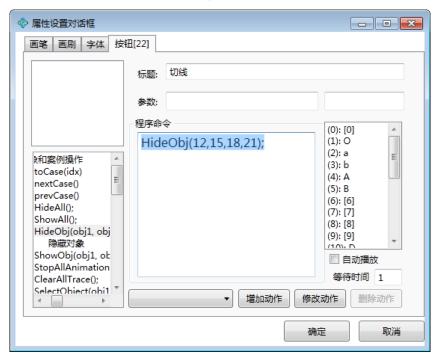


图 2.11.5

通过拖动点 a 或者拖动点 b 的控制条可以改变两圆的位置关系,单击【外切】按钮则圆 A 移动到与圆 B 外切的位置,单击【内切】按钮则圆 A 移动到与圆 B 内切的位置。最终效果如图 2.11.6 所示:

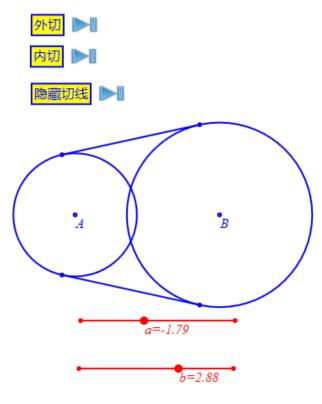


图 2.11.6

单击【公切线】按钮,则隐藏公切线;再次单击【公切线】按钮,则公切线被重新显示。 【思考与练习】

- (1)增加控制两圆的交点开关按钮【交点】。
- (2) 可以根据需要增加两圆半径大小的说明文本,并测量两圆的圆心距。
- (3) 与圆 B 外切(内切)时,圆 A 可能有两个位置,请增加圆 A 移动到另外一个位置的动画按钥。
- (4) 若点 A 和点 B 是平面上的完全自由点,以点 A 为圆心的圆的半径为 2,以点 B 为圆心的圆的半径为 3。如何设计点 B 的运动,使得圆 B 与圆 A 分别外切和内切?

# 2.12 研究圆幂定理

相交弦定理、切割线定理、割线定理、切线定理统称为圆幂定理。计算机可以直观、连续地将一种形式转化为另一种形式,从而反映圆幂定理几种形式之间的联系与统一。

- (1) 在新建文档中,作出任意点A,并以点A为圆心作半径为2的圆。
- (2) 在圆 A 外任意取一点 B, 在圆 A 的圆周上任一取一点 C, 连接线段 BC, 作出线段 BC 与圆 A 的另外一个交点 D, 结果如图 2.12.1 所示:

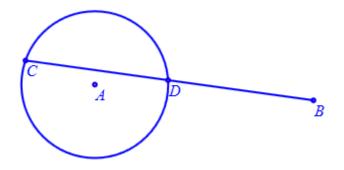


图 2.12.1

(3) 选择点 B 和圆,执行【画图】【直线】【过圆锥曲线外一点的切线】,得到过点 B 的圆 A 的两条切线段和两个切点,删除其中一条切线段和切点,标记剩下的切点为 E。结果如图 2.12.2 所示:

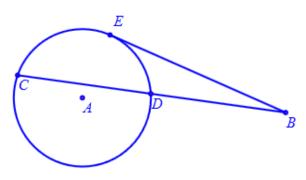


图 2.12.2

拖动圆上的点 C,可以发现,当点 C"越过"点 E 到达靠近点 B 的圆弧上时,如图图 2.12.3 所示,点 C 和点 D 重合在一起。

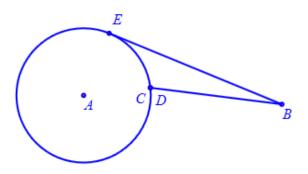


图 2.12.3

因为直线与圆有两个交点,而计算机并不能识别我们在什么时候需要显示两个交点中的哪一个。所以,就出现了这种情形。下面我们设计一种方法解决这个问题:

- (4)将点C拖动到左边位置,删除点D。
- (5) 作出点 A 到直线 BC 的垂足 F 和垂线段 AF (自动产生)。
- (6) 作出点 C 关于直线 AF 的轴对称点 G,结果如图 2.12.4 所示:

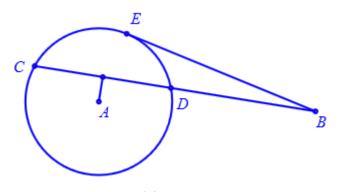


图 2.12.4

- (7) 隐藏线段 AF 和点 F。将点 G 的名字修改为 D,将点 B 的名字修改为 P,连接线段 PD。
  - (8) 按照以上的方法,作出另外一条割线 PMN,结果如图 2.12.5 所示:

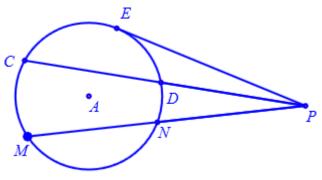
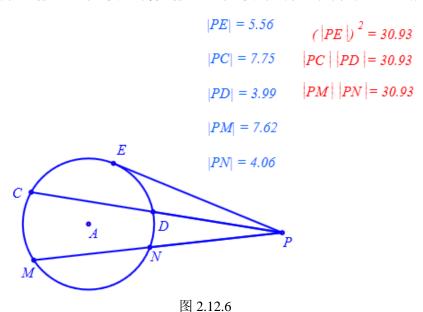


图 2.12.5

- (9) 测量线段 PC、PD、PM、PN 的长度,计算线段 PC 和 PD 的长度之积,计算线段 PM 和线段 PN 的长度之积。
  - (10) 测量线段 PE 的长度, 计算线段 PE 的长度的平方。结果如图 2.12.6 所示:



若将点 P 拖动到圆 A 之内,则有如图 2.12.7 的情形:

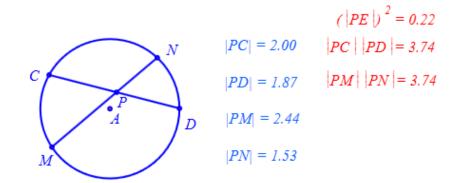


图 2.12.7

其中点C和点M是圆上的任意点可以被拖动。

## 2.13 毕达哥拉斯树

画出的直角三角形,操作如下:

- (1) 在新建文档中画出点 A 和 B。
- (2) 以 AB 为直径作一个圆。
- (3) 依次选取圆 B点、A 点执行【画图】【圆和圆弧】【圆周上的圆弧】,然后隐藏之前作的圆,在圆弧上取一点 C,隐藏圆弧。

画出直角三角形各个边对应的正方形:

(4) 依次选择 A、C 执行【作图】【多边形】【正方形】作出正方形,修改属性让新作的正方形另外两点的字母处于显示状态,同样方法作出 CB 和 BA 所对应的正方形。分分别给三个正方形填充不同的颜色。结果如图 2.13.1 所示:

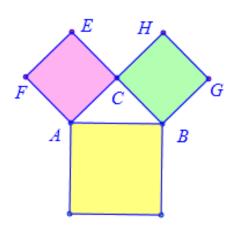


图 2.13.1

(5) 依次选择点 A 和点 B,执行【画图】【几何迭代】命令,在输入框中按照图 2.13.2 填写迭代信息:



图 2.13.2

(6)给迭代对象填充适当的颜色即可完成,拖动点 C,观察勾股数的动态变化。如图 2.13.3 所示:

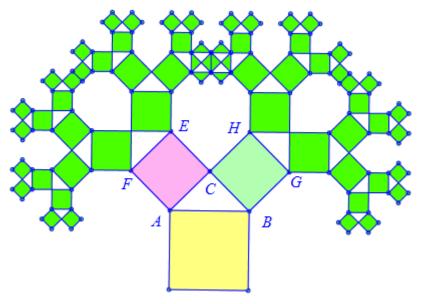


图 2.13.3

### 【思考与练习】

- (1)利用【工具】菜单中的【迭代】子菜单下的命令可以快速完成迭代作图。例如本节的勾股树就可以选择两个点后执行【工具】【迭代】【勾股树】命令作出试试通过该方法作出一个勾股树图案,和本节的结果对比一下异同之处。
  - (2) 通过本节学习的内容,完成如图 2.13.4 所示图案。

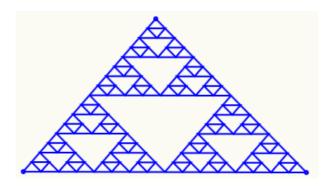


图 2.13.4 谢尔品斯基三角形

## 2.14 二叉树

首先增加参数的变量控制尺,并将对应的参数设置在一定的范围之内:

(1) 增加参数 a、b、c、d 的变量控制尺,如图 2.14.1 所示,设置参数 a 的改变范围为: -180 到 180,设置参数 b 的改变范围为: 0 到 1,设置参数 c 的改变范围为: -180 到 180,设置参数 d 的改变范围为: 0 到 1。

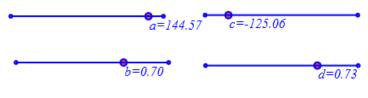


图 2.14.1

生成一个迭代对象:

- (2) 画任意线段 AB。
- (3) 依次选择点 A、点 B,执行【画图】【参数点】【旋转缩放点】命令,设置旋转角 参数(度数)为: a 放缩参数为 b。做出点 A 绕点 B 的旋转放缩点 C.,连接 BC
- (4) 与 (3) 类似,做出点 A 绕点 B 的旋转放缩点 D,设置旋转角参数(度数)为: C、缩放参数为 d,连接 BD,结果如图 2.14.2 所示:

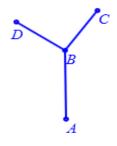


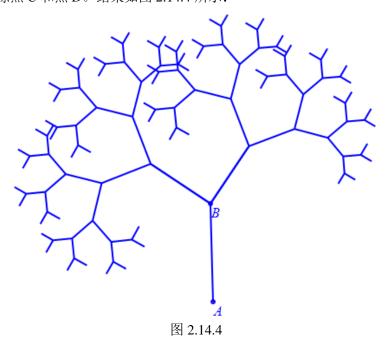
图 2.14.2

(5) 选择点 A、点 B,执行【画图】【几何迭代】命令,如图 2.14.3 设置迭代参数: 迭代规则中勾选对象 AB、BC 和 BD,设置两组迭代点,分别把 A 点迭代为点 B 和点 C,点 B 和点 D; 设置迭代的次数为: 5。



图 2.14.3

- (6) 单击【确定】按钮完成。
- (7) 删除点 *C* 和点 *D*。结果如图 2.14.4 所示:



## 2.15 多边形的密铺

### (一) 任意三角形的密铺

- (1) 在新建文档中,作出任意三个点 $A \times B \times C$ 。
- (2)作出点A和点C之间的中点(若作出了线段AC,则将它删除)。
- (3)作出以点 A、点 B 和点 C 为顶点的多边形。选择多边形 ABC,将其内部填充,根据需要填充合适的颜色。结果如图 2.15.1 所示:

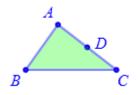


图 2.15.1

(4)选择三角形 *ABC* 和点 *D*,执行【变换】【旋转】命令,如图 2.15.2 所示设置旋转角度为 180 度得到点 *E*:选择 *ACE* 三点按前法作出三角形 *ACE* 并填充合适颜色。



图 2.15.2

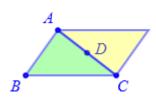


图 2.15.3 旋转结果

(5) 选择两个三角形和  $B \times C$  两点,执行【变换】【平移】,平移次数设为 4 次。将有向线段 BC 设置为平移向量,结果如图 2.15.4 所示:

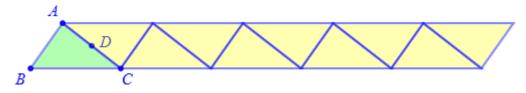


图 2.15.4

(6) 选择所有三角形和 B、A 两点,执行【变换】【平移】,平移次数设为 4 次。将有向线段 BA 设置为平移向量,结果如图 2.15.5 所示:

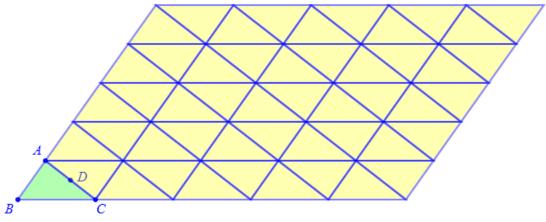


图 2.15.5

(7) 隐藏点 D。给各个三角形填上合适的颜色,这样就得到了任意三角形的密铺图案。

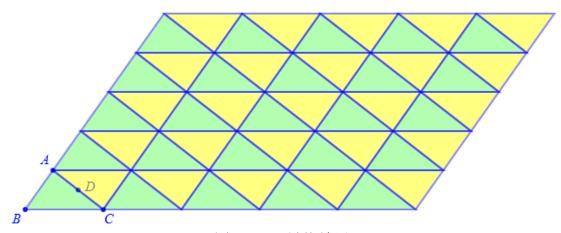


图 2.15.6 最终结果

## (二) 四边形的密铺

- (1) 在新建文档中,任意画四个点 $A \times B \times C \times D$ 。
- (2)作出点A和点D之间的中点E,点C和点D之间的中点F。
- (3)以点A、点B、点C和点D为顶点,作多边形,将其内部填充适当颜色。
- (4) 作出多边形 ABCD 和点 C 关于点 E 的中心对称图形,将旋转得到的多边形内部填充为另外一种颜色。
- (5) 作出多边形 ABCD 和点 B 关于点 F 的中心对称图形,将旋转得到的多边形内部填充为另外一种颜色。
- (6)以有向线段 *BD* 为平移向量,将多边形 *ABCD* 平移,将平移得到的多边形内部填充为重新选择一种填充颜色。结果如图 2.15.7 所示:

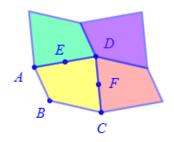


图 2.15.7

(7) 将 B 点绕 F 点旋转 180 度得到 H 点,以 A H为平移向量,将四个多边形对象平移 3 次。 结果如图 2.15.8 所示:

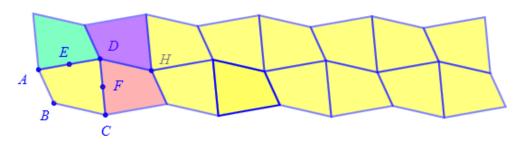
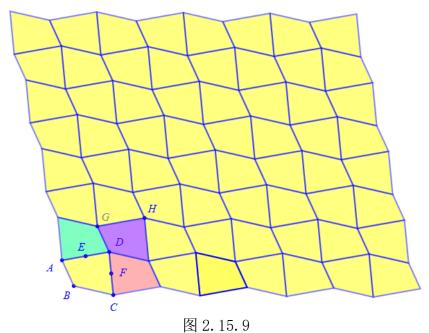


图 2.15.8

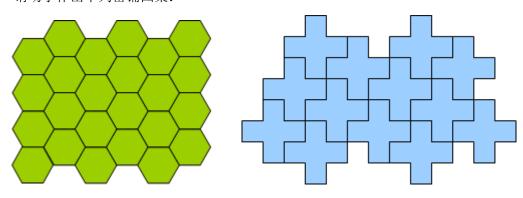
- (8)将 B 点绕 E 点旋转 180 度得到 G 点,选定所有多边形,以 BG 标记为平移向量,平移三次。
  - (9) 将点E、点F、点G和点H隐藏。

这样就做出了任意四边形的密铺图案。拖动四边形 ABCD 的顶点,可以改变其形状,如图 2.15.9 所示,四边形 ABCD 也可以是凹多边形。



## 【思考与练习】

请动手作出下列密铺图案:



# 2.16 密铺曲线

### (一) 基于等腰三角形的密铺曲线

- (1) 在新建页面中, 画任意两点 A 和 B。
- (2) 依次选择点 B、A,执行菜单【画图】下子菜单【多边形】【直角三角形(已知一直角边的直角三角形)】。把三角形另一点标记为 C,如图 2.16.1 所示:

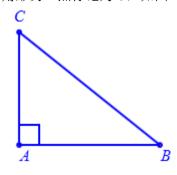


图 2.16.1

- (3) 在线段 AC 两侧各取两个任意点 D、E。
- (4) 在点 C、点 C之间任取三个任意点 F、G、H。
- (5) 依次选择点 A、点 D、点 E、点 B; 依次选择点 A、点 F、点 G、点 H、点 C,结果如图 2.16.2 所示:

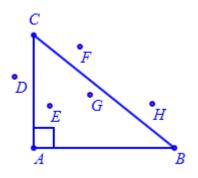


图 2.16.2

(6) 依次选择 *C、D、E、A* 四个点,执行【工具】菜单的子菜单【样条曲线】下的【过 4 点的样条曲线】命令,作出过四点的曲线;类似地,依次选择 *A、F、G、H、C* 五个点,执行【工具】菜单的子菜单【样条曲线】下的【过 4 点的样条曲线】命令,作出过五点的曲线;结果如图 2.16.3 所示:

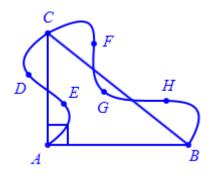


图 2.16.3 作出样条曲线

- (7) 作出样条曲线 ADEB 关于直线 AC 的发射变换图形,轴对称得到的曲线的序号为 17, 将该曲线按照向量 CA 进行平移变换。
  - (8) 将点 C 绕点 A 旋转 180 度,得到的点标记为 I。
- (9) 作出样条曲线 AFGHC 关于直线 AB 的反射变换图形,将该曲线按照向量 CB 进行 平移变换,现在得到的结果图 2.16.4 所示:

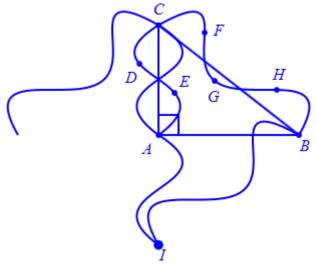


图 2.16.4

(10) 隐藏反射变换得到的两条曲线,隐藏线段  $AB \times BC \times CA$  和角 A 的标记。结果如图 2.16.5 所示:

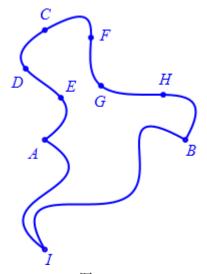


图 2.16.5

(11) 选择显示的 4 条曲线,将有向线段 AB 设定为平移向量,执行平移命令,平移次数设为 3,结果如图 2.16.6 所示:

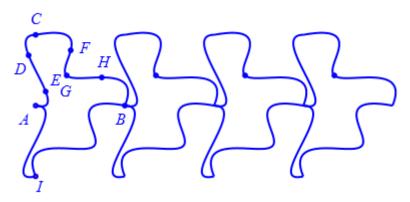


图 2.16.6

(12)选择显示的 4 条曲线和上一步平移得到的所有曲线,将有向线段 IC 设定为平移向量,执行平移命令,平移次数设为 3,结果如图 2.16.7 所示:

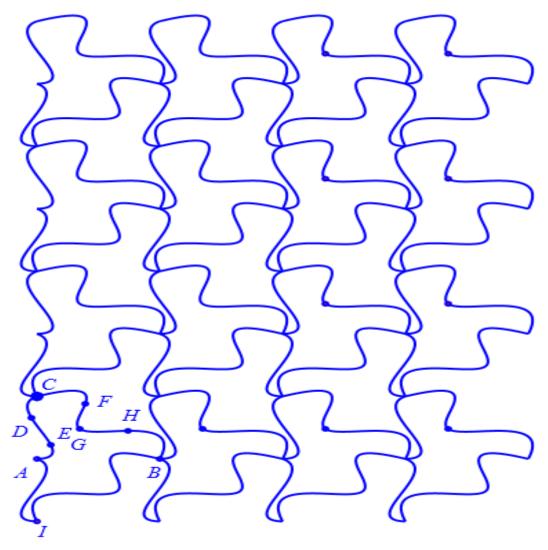


图 2.16.7 完成效果图

这就形成了以四条曲线为基础的平面密铺图案。

## 【思考与练习】

(1) 点 D、点 E、点 F、点 G 和点 H,可以被任意拖动,从而改变曲线的形状,从而可以设计出丰富多彩的图案,下面就是设计的另外一种图案,请你自己动手再设计一些有趣的图案。

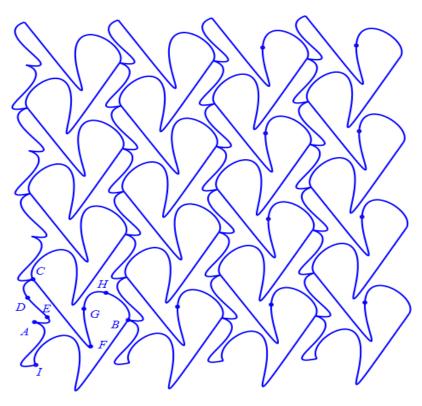


图 2.17.8

(2) 将曲线 ADEB 和曲线 AFGHC 的类型修改为: 折线段,结果如下图所示,得到了密铺多边形。

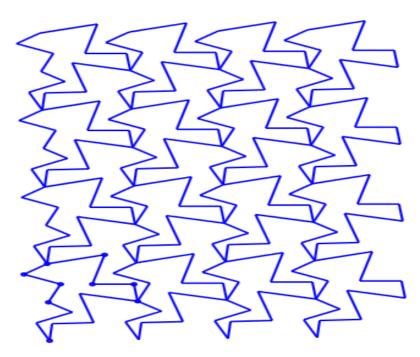


图 2.16.9

## (二)基于平行四边形的密铺图案

- (1) 在新建文档中,画任意三点  $A \times B \times C$ 。
- (2) 如下图 2.16.10 所示,作出曲线 ADEFB 和曲线 BGHIC。

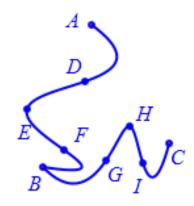


图 2.16.10

(3) 以有向线段 BA 为平移向量,将曲线 BGHIC 平移,得到平移后的曲线;结果如图 2.16.11 所示:

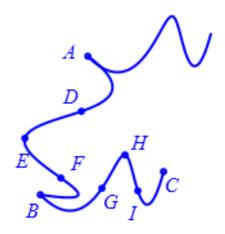


图 2.16.11

(4) 以有向线段 BC 为平移向量,将曲线 ADEFB 平移。到平移后的曲线;结果如图 2.16.12 所示:

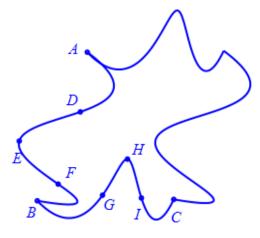


图 2.16.12

(5) 选择以上所作的四条曲线,以 BC 为平移向量,作四次平移变换,结果如图 2.16.13 所示:

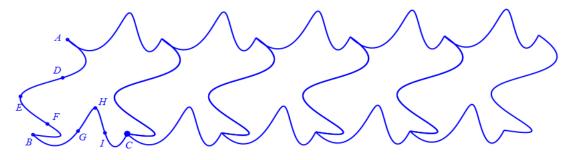


图 2.16.13

(6)以有向线段 *BA* 为平移向量,选择除最下方的一行曲线之外的其他曲线,进行平移,将平移得到的曲线继续平移,几次平移之后,结果如图 2.16.14 所示:

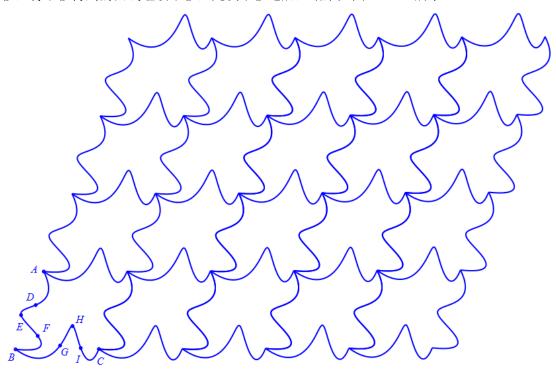


图 2.16.14 完成效果图

这就形成了以两条曲线组成的曲边平行四边形为基础的平面密铺图案。

## 【思考与练习】

- (1) 点 D、点 E、点 F、点 G、点 H 和点 I,可以被任意拖动,从而改变曲线的形状,从而可以设计出丰富多彩的图案,请你自己动手设计一些有趣的图案。
- (2) 将曲线 ADEB 和曲线 BGHIC 的类型修改为: 折线段,结果如图 2.16.15 所示,得到了密铺多边形。

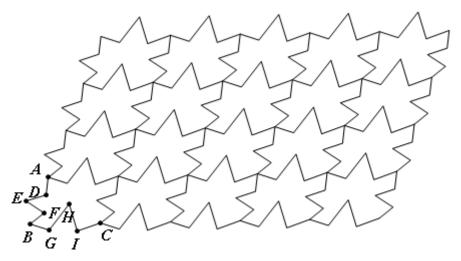


图 2.16.15

(3) 下图 2.16.16 是基于曲边梯形的平面密铺图案,请你动手设计出这种图案。

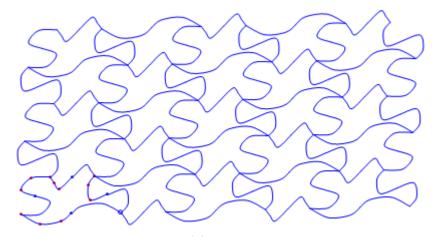


图 2.16.16

## 2.17 平面中的变换

平面上我们所熟悉的变换,大多数是可以用变换公式表示。利用变换公式研究对应的变换,能帮助我们更加透彻地研究变换的特点性质。

### (一) 旋转变换

平面上的一个点绕坐标原点 O 逆时针旋转 t 弧度的坐标变换公式是:

x'=x\*cos(t)-y\*sin(t); y'=x\*sin(t)+y\*cos(t)

下面我们利用这个变换公式对执行的对象进行旋转:

(1) 在【视图】勾选【对象框】,然后在对象框中勾选坐标系和点O,使得坐标系处于可见状态。如图 2.17.1 所示:

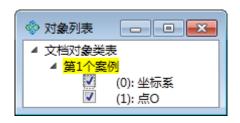


图 2.17.1

- (2) 在平面上任取一点A, 连接OA。
- (3)选择点 A 和线段 OA,执行【变换】菜单下的【通过变换公式进行仿射变换】命令,如图 2.17.2 所示,在弹出的坐标变换公式输入对话框中输入对应的参数值: a1=cos(t),a2=sin(t),b1=-sin(t),b2=cos(t),x0=0,y0=0。

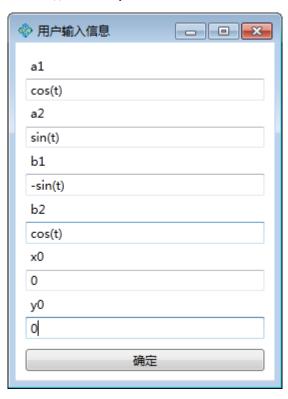
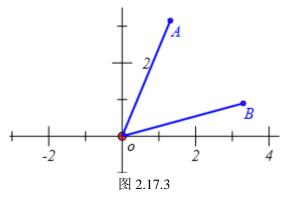


图 2.17.2

(3) 单击【确定】按钮完成,得到点 B 和线段 OB。结果如图 2.17.3 所示:



(4) 执行【插入】【常用按钮】【变量一次运动】命令增加参数 t 的动画按钮,如图 2.17.4 设置按钮参数:

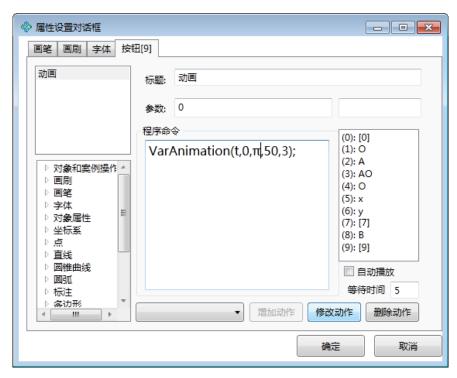


图 2.17.4

单击参数 t 的动画按钮,可以观察到点 B 从点 A 的位置出发,绕坐标原点 O 旋转了半周。

#### 【思考与练习】

- (1) 作任意三角形 ABC,通过坐标变换公式作出其关于坐标原点 O 的旋转图形 DEF。
- (2)展示 DEF 绕坐标原点 O 从三角形 ABC 的初始位置旋转到 ABC 关于点 O 中心对称图形的位置。
  - (3) 任作平面上一点 M,利用旋转变换公式,应该如何作出该点关于 v 轴的对称点?
- (4) 你能求出关于平面上任意点 P(x0, y0) 旋转的坐标变换公式吗? 然后通过实验检验你的结论。

#### (二) 反射变换

平面上任意点 A(x, y) 关于 x 轴的对称点 A'的坐标为 (x, -y),相应的坐标变换公式为: x'=x; y'=-y

下面我们利用这个变换公式进行几何对象的反射变换:

- (1) 任意作一个圆 A。
- (2)选择圆 A 和点 A,执行【变换】菜单下的【通过变换公式进行仿射变换】命令,如图 2.17.5 所示,在弹出的坐标变换公式输入对话框中输入对应的参数值: a1=1, a2=0, b1=0, b2=-1, x0=0, y0=0。

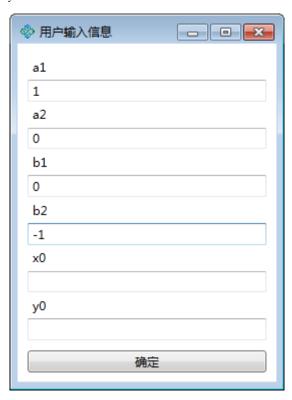
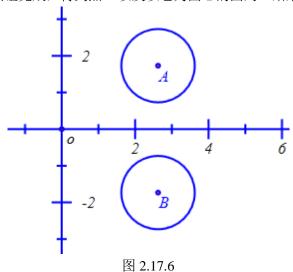


图 2.17.5

(3) 单击【确定】按钮完成,得到点 B 以及以它为圆心的圆周。结果如图 2.17.6 所示:



### 【思考练习】

- (1) 根据关于y轴对称的坐标变换公式,作出圆A关于y轴对称的图形。
- (2) 先作出函数  $y=2^x$  的图像,根据关于直线 y=x 对称的坐标变换公式,再作出它的反函数的图像。
  - (3) 你能求出平面上的点 A(x, y) 关于任意直线 y=k\*x+b 对称的坐标变换公式吗?

然后通过实验检验你的结论。

## (三) 伸缩变换

在直角坐标系中,将点的 x 坐标变为原来的 k1 倍、将点的 y 坐标变为原来的 k2 倍的几何变换叫做伸缩变换。对应的坐标变换公式为:

x'=k1\*x; y'=k2\*y

下面我们利用伸缩的坐标变换公式将函数 y=sin(x)的图像上点的横坐标变为原来的 2 倍:

(1) 执行【画图】【一般曲线】【三角函数: y=f(x)】命令,在弹出窗口中按图 2.17.7 填写作函数 y=sin(x)在[-2\*pi, 2\*pi]上的图像。

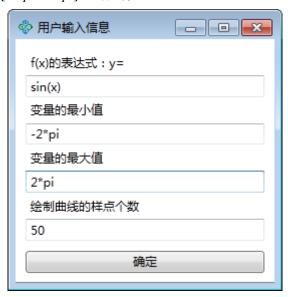
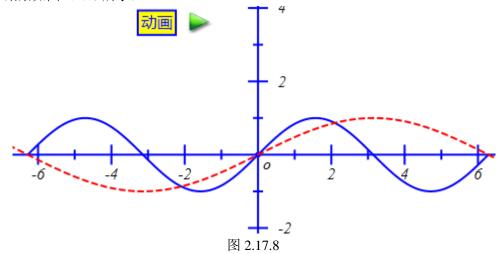


图 2.17.7

- (2)选择该函数图像,执行【变换】菜单下的【通过变换公式进行仿射变换】命令,在弹出的坐标变换公式输入对话框中输入对应的参数值: a1=k, a2=0, b1=0, b2=1, x0=0, y0=0。
  - (3) 将变换后的图像以红色虚线显示。
- (4) 增加参数 k 的动画按钮,设置参数 k 的范围为: 1 到 2,选择运动类型为: 一次运动。结果如图 2.17.8 所示:



单击参数 k 的动画按钮,红色曲线从 y=sin(x)的位置出发,在 x 方向上伸长为原来的 2 倍。

### 【思考练习】

- (1) 在 x 方向上伸长为原来 2 倍后的函数曲线的方程表达式是?
- (2)通过坐标变换公式,由函数 y=sin(x)的图像得到 y=sin(2x)的图像,并展示动态变换过程。
- (3)通过坐标变换公式,由函数 y=sin(x)的图像得到 y=3sin(x)的图像,并展示动态变换过程。
- (4) 你能求出平面上的点 A(x, y) 关于点 P(x0, y0) 为中心伸缩旋转的坐标变换公式吗? 然后通过实验检验你的结论。

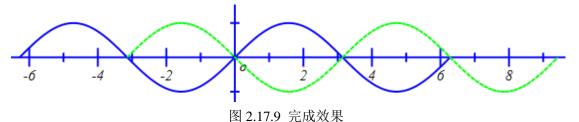
#### (四) 平移变换

在直角坐标系中,将点 P(x, y) 在 x 方向、y 方向分别平移 x0、y0 个单位的几何变换 叫做平移变换。对应的坐标变换公式为:

x'=x+x0; y'=y+y0

下面我们利用平移变换的坐标变换公式将函数 y=sin(x)的图像平移得到 y=sin(x+pi)的图像:

- (1) 作函数 y=sin(x)在[-2\*pi, 2\*pi]上的图像。
- (2)选择该函数图像,执行【变换】菜单下的【通过变换公式进行仿射变换】命令,在弹出的坐标变换公式输入对话框中输入对应的参数值: a1=1, a2=0, b1=0, a2=0, b2=1, x0=pi, y0=0。
  - (3) 将变换后的图像以绿色虚线显示。如图 2.17.9 所示:



#### 【思考练习】

- (1)通过坐标变换公式,由函数 y=sin(x)的图像得到 y=sin(x-pi/2)的图像,并展示动态变换过程。
- (2) 通过坐标变换公式,由函数 y=sin(x)的图像得到 y=2+sin(x)的图像,并展示动态变换过程。
- (3) 你能求出平面上的点 A(x, y) 按照指定的向量 PQ 进行平移的坐标变换公式吗? 其中点 P 的坐标为(x1, y1),点 Q 的坐标为(x2, y2)。然后通过实验检验你的结论。

## (五) 切变变换

对于下列坐标变换公式对应的变换叫作切变变换:

x'=x+s\*y; y'=y 或者 x'=x; y'=t\*x+y

下面我们研究切变变换的性质:

- (1) 作出以原点为圆心、半径为2的圆,作出圆的外切正方形(方法任意)。
- (2) 执行【画图】【一般曲线】【三角函数: y=f(x)】命令,如图 2.17.10 (左) 所示,设置样点的个数为: 500,参数 t 的范围: 0 到 2\*pi。作出参数方程 {x=2\*sin(2\*t); y=2\*sin(3\*t)\*sin(5\*t)}对应的曲线; 如图 10 设置样点的个数为: 500,参数 t 的范围: 0 到 2\*pi。结果如图 2.17.10 (右) 所示:

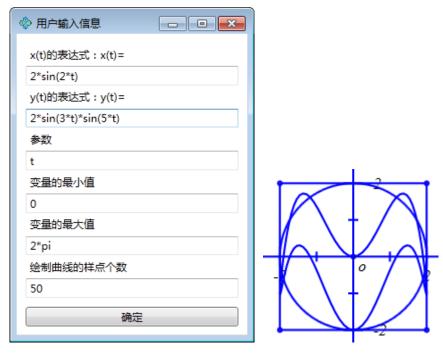
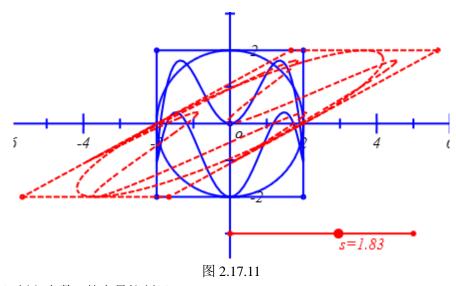


图 2.17.10

- (3)选择曲线、圆和正方形,执行【变换】菜单下的【通过变换公式进行仿射变换】命令,在弹出的坐标变换公式输入对话框中输入对应的参数值: a1=1, a2=0, b1=s, b2=1, x0=0, y0=0。
  - (4) 将变换得到的对象以红色虚线显示。结果如图 2.17.11 所示:



(5) 插入参数 s 的变量控制尺。

操作变量控制点可以发现,原来的正方形变成了平行四边形,原来的圆变换成了椭圆, 原来参数曲线经过变换后形状也发生了改变。

除了这些改变的性质外,在这个切变变换中,有哪些性质是保持不变的?

### 【思考练习】

(1) 将原来的图形,利用变换公式 x'=x,y'=t\*x+y 进行变换。总结变换 x'=x,y'=t\*x+y

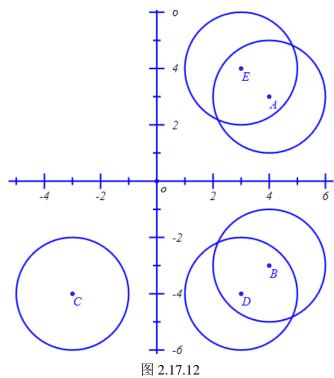
的特点。

(2) 将原来的图形,利用变换公式 x'=1.1\*x+0.3\*y,y'=0.2\*x+0.9\*y 进行变换。观察变换后的图形,有哪些性质仍然保持?

### (六) 变换的复合与性质

实数的乘法满足交换律,即 a\*b=b\*a。变换的乘法是否也满足交换律呢?即 AB=BA 是否成立?

- (1) 作出一个任意点 A 以及以点 A 为圆心的圆。
- (2) 将点 A 和圆 A 按照变换公式{x'=x; y'=-y}进行变换,得到点 B 和圆 B。
- (3)将点B和圆B按照变换公式{x'=cos(-pi/2)\*x-sin(-pi/2)\*y; y'=sin(-pi/2)\*x+cos(-pi/2)\*y} 进行变换,得到点C和圆C。
- (4)将点A和圆A按照变换公式{x'=cos(-pi/2)\*x-sin(-pi/2)\*y; y'=sin(-pi/2)\*x+cos(-pi/2)\*y} 进行变换,得到点D和圆D。
- (5) 将点 D 和圆 D 按照变换公式 $\{x'=x; y'=-y\}$ 进行变换,得到点 E 和圆 E。结果如图 2.17.12 所示:



可以发现,点C和点E不在同一位置上,圆C和圆E也不重合。

## 【思考与练习】

请你设计两个满足交换律的变换。并进行实验、验证你的结果。

### (七) 仿射变换

线性变换不能保持两点的距离,不能保持图形的形状。但是对于,可逆的线性变换将直线仍变成直线,平行直线经过变换后仍保持平行关系。正方形变成了平行四边形,形状改变了,但平行四边形还是变成平行四边形,这就是不变的性质;圆可以变成椭圆,形状改变了,然而若将圆看成是长轴和短轴相等的椭圆,则还是椭圆变成椭圆,这也是不变的性质。

可逆的线性变换、平移变换以及他们的复合变换,构成一个变换群,称为仿射变换群。

研究在这个群的作用下不变的性质的学科,就是仿射几何。

下面我们举一个熟悉的例子:

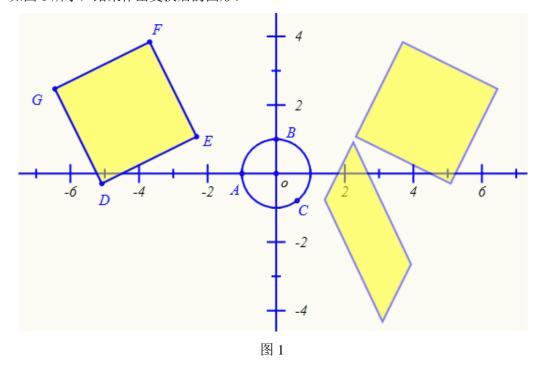
- (1) 作以原点 O 为中心、半径为 1 圆。
- (2) 作出圆与x轴负半轴的交点A,作出圆与y轴正半轴的交点B。
- (3) 作任意正方形 DEFG。
- (7) 选择点  $D \setminus E \setminus F \setminus G$  执行【画图】【多边形】【多边形】命令,得出正方形 DEFG。
- (3) 依次选择正方形内部和点 O、点 A、点 B,执行【变换】菜单下的【坐标系到三角形的仿射变换】,实现了坐标即原三角形(0,0)(1,0)(0,1)->三角形 OAB(0,0)(-1,0)(0,1)的变换。你能写出这个仿射变换的变换公式吗?这是一个什么变换?

#### 继续操作:

- (4) 在圆上任取一点 C。
- (5) 依次选择上一步生成的正方形内部和点 O、点 C、点 B,执行【变换】菜单下的【坐标系到三角形的仿射变换】。实现了坐标即原三角形(0,0)(1,0)(0,1)->三角形 OAB 的变换。

对正方形 DEFG 来说,这是实现了 坐标系(0,0)(1,0)(0,1)->(0,0)(-1,0)(0,1)->三角形 OAB的两次变换。

如图 1 所示,结果作出变换后的图形。



可以发现变换后的图形为一个平行四边形。原来的正方形 DEFG 相比较,有什么变化?有哪些不变的关系?

拖动点 C,可以使得平行四边形以  $\gamma$  轴所在直线为轴进行翻折。

#### 【思考与练习】

- (1) 增加点 C 的动画按钮,设置参数范围为: pi 到 2\*pi,运动类型为: 一次运动。单击动画按钮,运动结束后,变换后的图形与正方形 ABCD 之间有什么关系?
  - (2) 先后两个仿射变换的复合变换的变换公式是什么?

#### (八)射影变换

- (1) 作坐标点 A (4, 0)、 B (2, 0)、 C (1, 0)、 D (-1, 0)、 E (1, 2)。
- (2) 连接线段 AE、BE、CE、DE。
- (3) 在线段 AE、BE、CE、DE 分别取点 F、G、H、I。结果如图 2 所示:

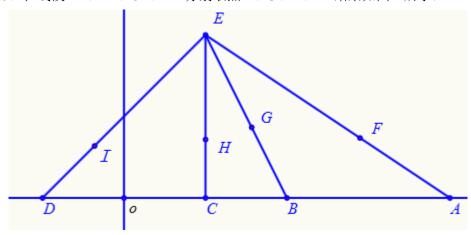
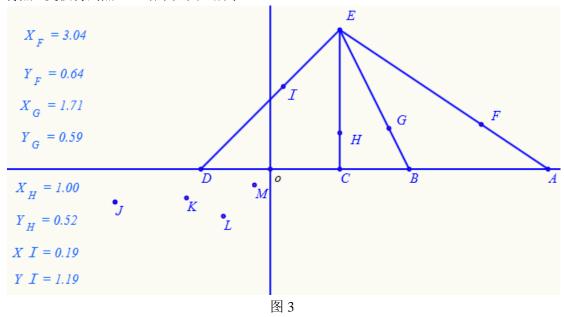


图 2

下面将线段上的点按照下列变换公式进行变换:

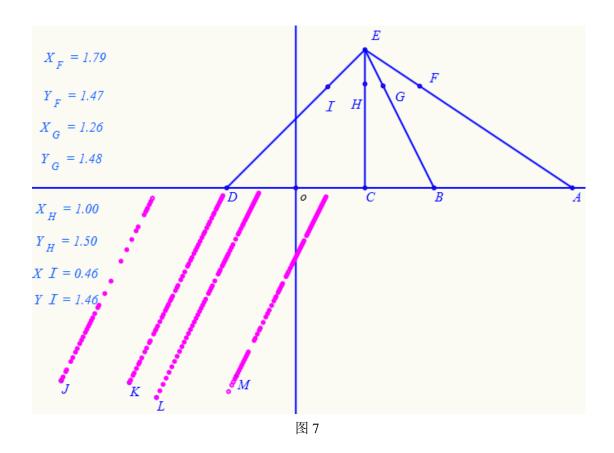
x'=x/(y-2); y'=y/(y-2)

- (4) 分别测量点 F 的横坐标和纵坐标,执行【画图】【参数点】【坐标点】命令,设置新生成点的 x 坐标为 v000/(v001-2),y 坐标为 v001/(v001-2)。其中(v000,v001)是所选择的点 F 的坐标。结果得到点 J。
- (5) 重复类似操作,利用上面的变换公式将点 G 变换得到点 K,将点 H 变换得到点 L,将点 I 变换得到点 M。结果如图 3 所示:



(6) 跟踪点 J、点 K、点 L、点 M。并设置跟踪颜色为红色。

同时选择点 F、G、H、I,通过键盘中的向上方向键 $\uparrow$ 或向下方向键 $\downarrow$ ,让它们同时运动,得到点 J、K、L、M 运动的跟踪图形,结果如图 7 所示:



#### 【思考练习】

- (1) 点 J、K、L、M 运动过的轨迹分别是什么图形?与变换前的线段 AE、BE、CE、DE 相比较,有什么变化?
- (2) 这个变换是可逆变换吗?若是可逆变换求出它的变换公式,并通过实验检验你的结论。
  - (3) 为了更加直观地观察 J、K、L、M 运动过的图形,请构造出他们的轨迹图形。
- (4) 以坐标原点 Q 为圆心、作半径为 r 的圆,在圆 Q 上任取一点 Q ,利用上面的变换公式将点 Q 变换到点 Q,构造点 Q 驱动下点 Q 的轨迹。改变半径 r 的大小,观察点 Q 的轨迹图形。点 Q 的轨迹图形是什么形状?半径 r 对它有什么影响?
- (5) 与线性变换相比较,上面的变换有哪些不同?它改变了哪些关系?而哪些性质又保持不变?

# 第三部分 函数、方程及图像

### 3.1 二次函数的图像

作函数  $y=x^2$  的图像,具体操作步骤如下:

- (1) 在新建文档,执行【画图】【一般曲线】【幂函数: y=f(x)】命令,弹出函数作图对话框。
- (2) 如图 3.1.1 所示,在"y="对应的编辑框中输入: x^2,单击【确定】按钮完成,作出 y=x2 的图像。

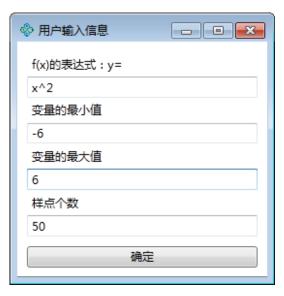


图 3.1.1

- (3) 执行【视图】【系统坐标系】命令,使坐标系处于可见状态。
- (4) 鼠标移到坐标系上,单击右键,即可打开它的属性对话框。
- (5)如左下图所示,在属性对话框中选择"显示刻度"选项和"画坐标网格"选项,取消"全屏显示",用鼠标调整好坐标轴,单击【确定】按钮完成。结果如图 3.1.2 所示:

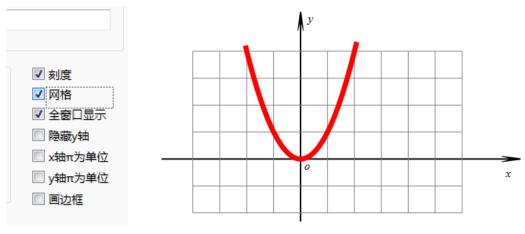


图 3.1.2

若我们希望探索函数  $y=1/2*(x-a)^2-1$  在区间[m,n]上最值的情况,则可以对作出的曲线的表达式进行修改。操作步骤如下:

(6) 鼠标指向曲线,右击,打开其属性对话框。如图 3.1.3 所示,将曲线表达式修改为: $1/2*(x-a)^2-1$ ,设置参数(变量)的范围为:m 到 n。

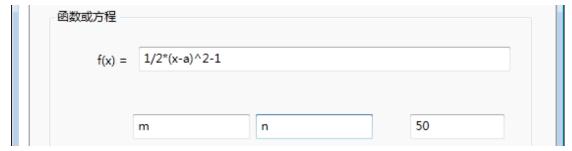


图 3.1.3

(7) 单击【确定】按钮完成,结果如图 3.1.4 所示,作图区中只出现出现一小段函数曲线,原因何在呢?请继续下面的操作。

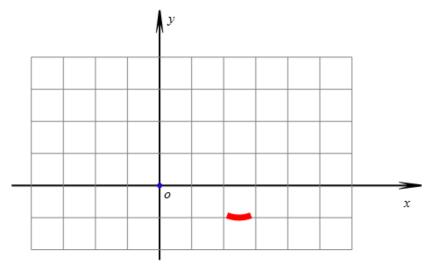


图 3.1.4

(8) 如图 3.1.5 所示,执行【插入】【变量】命令,作出变量 a、的变量控制条。同理作出 m、n 的变量控制条。



图 3.1.5

(9) 结果如下图 3.1.6 所示,可以观察到参数 m 和参数 n 的初始值大小相差很小,那么函数的图像只有很小的一段。

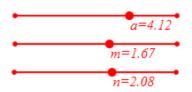


图 3.1.6

(10) 向右拖动变量 n 控制条上的控制点,增大参数 n 的值;往左拖动点 m 控制条上的控制点,减小参数 m 的值,结果如图 3.1.7 所示:

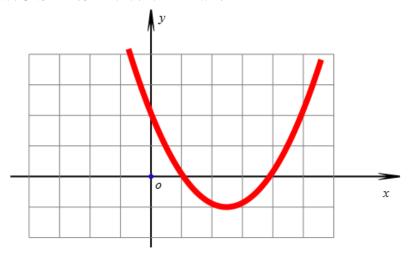


图 3.1.7

- (11) 利用【画图】【参数点】【坐标点】命令,依次作出坐标点A(m, 0)、B(n, 0)  $D(m, 1/2*(m-a)^2-1)$ 、 $E(n, 1/2*(n-a)^2-1)$ 。
- (13)连接线段 AD、BE,并在右键菜单中将线段 AD 和 BE 的画笔"线型"设置为:虚线。
- (14)将点 A 的名字修改为: m,将点 B 的名字修改为: n,将点 C 的名字修改为: a。 结果如图 3.1.8 所示:

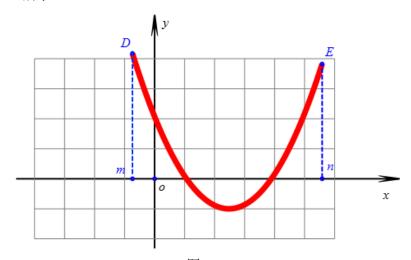
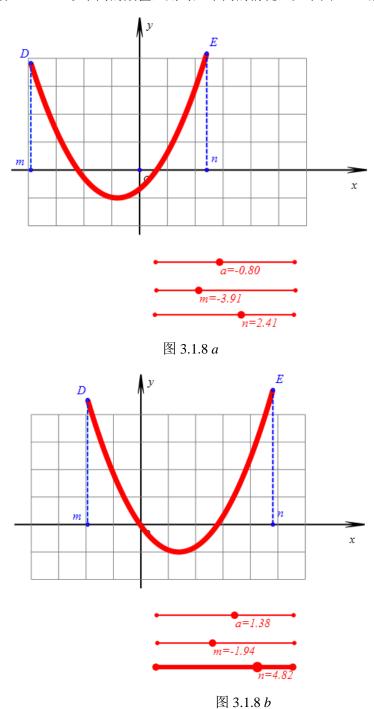


图 3.1.8

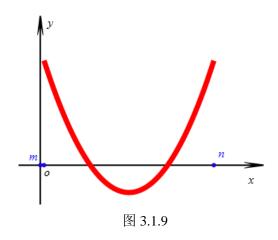
(15)选择曲线,单击工具栏中【放大】工具,增加曲线的画线宽度;设置曲线的画线颜色为红色。

可以让参数 m、n、a 取不同的数值,则对应不同的情况,如下图 3.1.8 所示:



#### 【思考与练习】

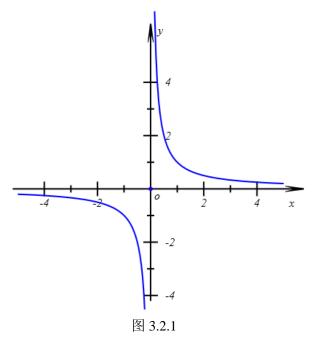
- (1) 请设法作出顶点坐标,并得出能够直接读出函数的最值,以及对应的x取值。
- (2)作出函数  $y=a*(x-k)^2+h$ ,要求当 a、k、h 改变过程中,曲线总是能够显示对称轴两侧等距离的图像。如图 3.1.9 所示:



### 3.2 反比例函数的图像

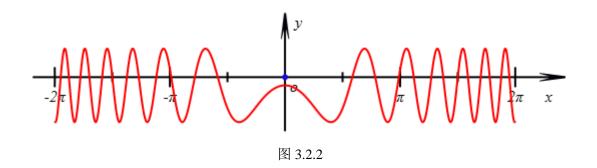
画 y=1/x 的图像,具体操作如下:

- (1) 在新建文档,使坐标系可见(参考 3.2 (3)),执行【画图】【一般曲线】【幂函数: y=f(x)】命令,弹出函数作图对话框。
- (2) 在"y="对应的编辑框中输入: 1/x,作图范围-10 到 10,曲线点数: 50,数单击【确定】按钮完成,作出 y=1/x 的图像,增加曲线的宽度,并设置画线颜色为红色,适当调整坐标系,结果如图 3.2.1 所示。



#### 【思考与练习】

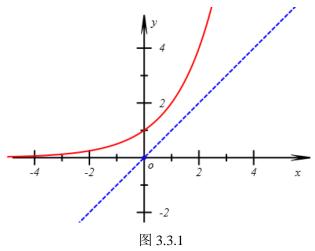
如图 3.2.2 所示,为函数  $y=sin(x^2)$ 在区间[-2\*pi,2\*pi]上的图像,请你在计算机中将它画出来。



# 3.3 指数函数及其反函数的图像

画函数  $y=a^x$  的图像,操作如下:

- (1) 在新建文档,使坐标系可见,执行【画图】【一般曲线】【幂函数: y=f(x)】命令,弹出函数作图对话框: 在"y="对应的编辑框中输入:  $a^x$ , 作图范围: -10 到 10 "曲线的点数"为: 5000,单击【确定】按钮完成,作出  $y=a^x$  的图像,增加曲线的宽度,并设置画线颜色为红色。
- (2) 插入变量 a, 设置其范围为 0 到 3, 并把参数 a 调整到正数。结果如图 3.3.1 所示。



作出直线 y=x, 操作如下:

(2) 选择坐标原点 O,执行【作图】【直线】【点斜式方程】命令,弹出构造直线对话框,如图 3.3.2 所示,在"直线的斜率"编辑框中输入: 1。

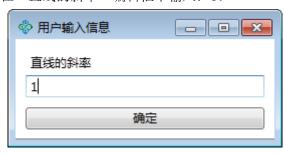
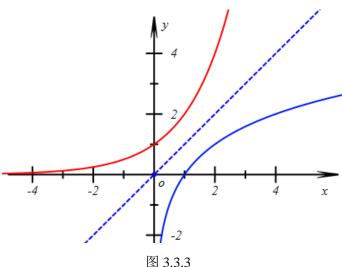


图 3.3.2

(3) 右击直线,在对话框中的【画笔】选项卡,设置画笔"类型"为:虚线,单击【确定】按钮完成。

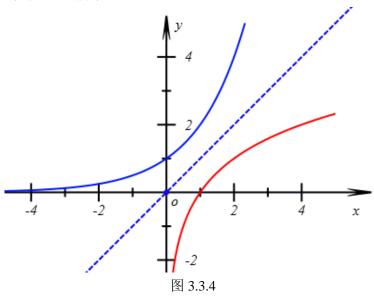
作出曲线  $y=a^x$  关于直线 y=x 的轴对称图像,操作如下:

- (4) 依次选择曲线  $y=a^x$  和直线 y=x,执行【变换】【反射】命令,得到  $y=a^x$  关于直 线 v=x 的对称曲线。
  - (5) 将对称得到的曲线的画线颜色设置为: 蓝色。结果如图 3.3.3 所示:



将红色曲线的方程改为对数方程,操作如下:

(6) 打开曲线  $y=a^x$  的属性对话框,如左下图所示,将函数的表达式修改为: log(a,x), 单击确定按钮,如图 3.3.4 所示:



### 【思考与练习】

- (1) 移动数 a 的变量控制尺观察图形变化。
- (2) 观察当 a 取不同情况的数值时,探讨对数函数 y=log(a,x)与其反函数图像交点的情 况。

# 3.4 参数方程曲线

半径为 1 的圆 A 沿 x 轴无滑动的滚动, 在起始位置点 B 与坐标原点 O 重合。当圆滚动 一段时间距离后,点B绕点旋转过的角度为a弧度,则圆心A在水平位置上平移过的距离 00'也为 a。

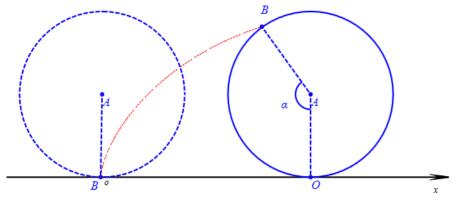


图 3.4.1

跟踪点 B 得到的轨迹图形就是旋轮线。

容易推导,点 B 的轨迹所在旋轮线的方程可表示为:

$$\begin{cases} x=a+cos(-\frac{pi}{2}-a) \\ y=1+sin(-\frac{pi}{2}-a) \end{cases} (a 为参数)$$

化简后得到:

由此,我们可以直接画出半径为 1 的圆在 x 轴上无滑动地滚动时,圆周上一点对应的旋轮线,操作如下:

(1) 执行【画图】【一般曲线】【参数方程:x=x (t) y=y(t)】命令,如图 3.4.2 所示,选择函数的类型为:参数方程,在"x="对应的编辑框中输入:a-sin(a),在"y="对应的编辑框中输入:1-cos(a)。

	×	
x(t)的表达式:x(t)=		
a-sin(a)		
y(t)的表达式:y(t)=		
1-cos(a)		
参数		
a		
变量的最小值		
-4*pi		
变量的最大值		
4*pi		
绘制曲线的样点个数		
500		
确定		

图 3.4.2

(2) 在"参数范围"一行的中间编辑框中输入: a,将 a 设置为参数方程的参数。设置参

数范围为: -4\*pi 到 4\*pi, 单击【确定】按钮, 结果如图 3.4.3 所示:

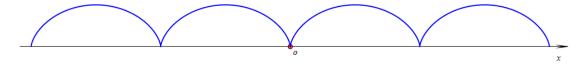


图 3.4.3

下面我们作出一个圆,在该圆上任意取一点跟踪后得到的图形都是旋轮线。首先作出圆心及圆周上的点,操作如下:

(3)隐藏上部所作曲线,作坐标点 A(a, 1)、 B(a-sin(a), 1-cos(a)),执行【工具】 【变量】命令作出变量 a 的控制条。



图 3.4.4 点 B 参数

(4) 跟踪点 B。如图 3.4.5 所示,拖动变量控制条上的控制点,则点 B 留下的踪迹就是旋轮线。

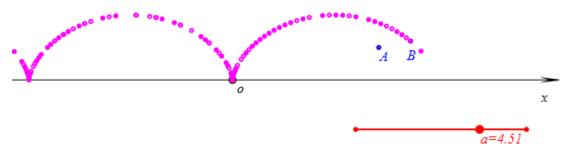


图 3.4.5

作出从点B出发、以点A为圆心的圆形曲线,操作如下:

(5) 执行【画图】【一般曲线】【参数方程:x=x(t) y=y(t)】命令,如图 3.4.1 所示,如图 3.4.6 所示,选择函数的类型为:参数方程,在"x="对应的编辑框中输入:a-sin(a+u),在"y="对应的编辑框中输入:1-cos(a+u)。

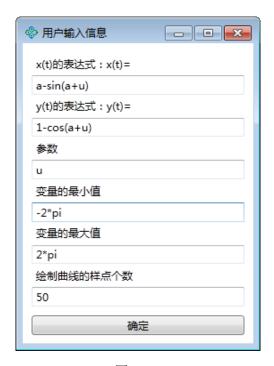


图 3.4.6

(6) 在"参数范围"一行的中间编辑框中输入: u,将u设置为参数方程的参数。设置参数范围为: -2\*pi到2\*pi,单击【确定】按钮,结果如图3.4.7所示:

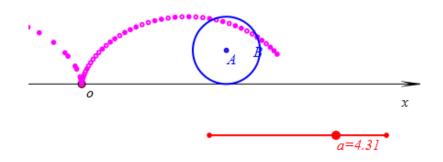
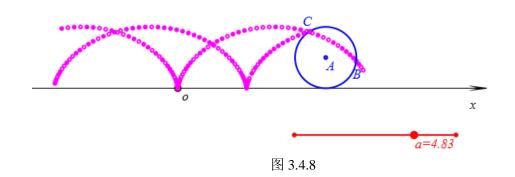


图 3.4.7

(7) 进入画笔状态,在圆形上任意取一点 C,跟踪点 C。

在拖动点变量 a 控制条上的控制点的过程中可以观察到 $\triangle C$  的轨迹图形,如图 3.4.8 所示,



点 C 可以在圆周上被任意拖动,但结果跟踪点 C 留下的图形始终是旋轮线,只是相位会有区别。

#### 【思考与练习】

- (1) 通过上面的步骤(5)、(6),我们可以明白圆形是从点B 出发绕点A 旋转一周的曲线。
- (2) 在上面的作图过程中,点 *B* 只是帮助我们理解作出的圆形曲线。你可以将它和它的跟踪对象隐藏或者删除,可以刚开始不作出这个点。
  - (3) 当圆的半径为 R 时,请你重新设计上面的实验。
- (4) 若半径为r的圆沿半径为R的圆无滑动地滚动时,动圆周上一点M所描成的轨迹也是旋轮线,它的轨迹方程可表示为:

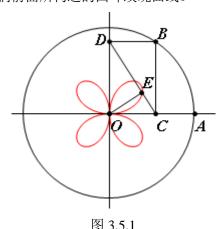
$$\begin{cases} & x = (R+r) \cdot cos(t) - r \cdot cos(\frac{R+r}{r} \cdot t) \\ \\ & y = (R+r) \cdot sin(t) - r \cdot sin(\frac{R+r}{r} \cdot t) \end{cases}$$

t 为参数,表示动圆的圆心从 x 轴正方向上出发旋转过的角度。

请你设计一个圆在圆上滚动的实验。使得在滚动的圆上任意取一点,跟踪得到的图形都是旋轮线。

### 3.5 极坐标方程曲线

如图 3.5.1 所示,是我们前面所构造的四叶玫瑰曲线。



四叶玫瑰曲线可以用极坐标方程标示为:  $\rho$ = $sin(2\theta)$ 。

下面我们通过函数方程直接作出四叶玫瑰曲线,操作如下:

(1) 执行【画图】【一般曲线】【参数方程:x=x (t) y=y(t)】命令,如图 3.5.2 所示,选择函数的类型为: 极坐标: $\rho=f(\theta)$ ,在" $\rho=$ "对应的编辑框中输入: sin(2\*thet)(其中 thet 是希腊字母  $\theta$  的单词的第 4 个字母,在这里表示  $\theta$ ),

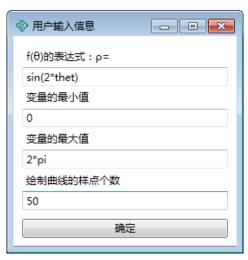
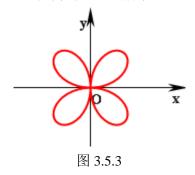


图 3.5.2

(2) 在"参数范围"一行设置参数 thet 的范围为: 0 到 2\*pi。单击【确定】按钮,右击曲线设置设置画笔颜色为: 红色。结果如图 3.5.3 所示:



#### 【思考与练习】

- (1) 方程  $\rho$ =3 $sin(2\theta)$ 表示半径为 3 的四叶玫瑰曲线。请你作出半径为 R 的四叶玫瑰曲线。
- (2) 方程  $\rho$ =5 $sin(n\theta)$ 也是玫瑰曲线。做出 n 为整数时的半径为 5 的玫瑰曲线,并改变 参数 n 的值,观察玫瑰曲线的变化特点。
- (3) 方程  $\rho$ =5 $cos(2\theta)$ 对应的曲线也是四叶玫瑰线,请你作出该方程对应的曲线,并与方程  $\rho$ =5 $sin(2\theta)$ 对应的四叶玫瑰曲线进行比较,说出他们的联系与区别。

## 3.6 分段函数的图像

如图 3.6.1 所示,阶梯函数是一个分段函数,对于任何一个 x,对应函数的值是比 x 小的最大整数。

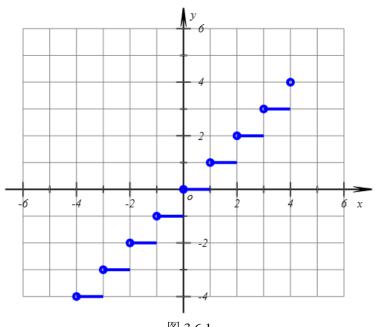


图 3.6.1

下面我们作出阶梯函数的图像,操作如下:

- (1) 在新建文档中,显示坐标网格。
- (2) 执行【画图】【一般曲线】【幂函数: y=f(x)】命令,在"y="对应的编辑框中输入: floor(x), 设置"曲线的点数"为: 500, 设置变量 x 的范围为: -4 到 4, 单击【确定】按钮完 成, 结果如图 3.6.2 所示:

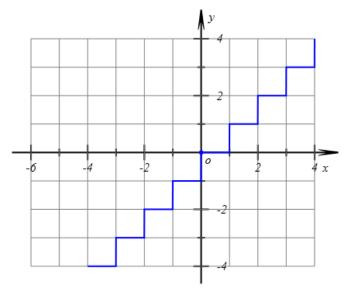


图 3.6.2

这里我们做出的图形是不符合要求的曲线,原因在于它将上下两段连接起来了。 通过前面的学习,我们知道只要在曲线的属性对话框中将"间断点的最小值"设置为小于 1的值即可。操作如下:

(3) 右击曲线, 打开其属性对话框, 如图 3.6.3 所示, 将"间断点的最小值"设置为: 0.9。

间断值 0.9 □ 画样本点 样本点大小 4 □ 埴充为矩形区域 □ 折线段 (4) 单击【确定】按钮完成,结果如图 3.6.4 所示:

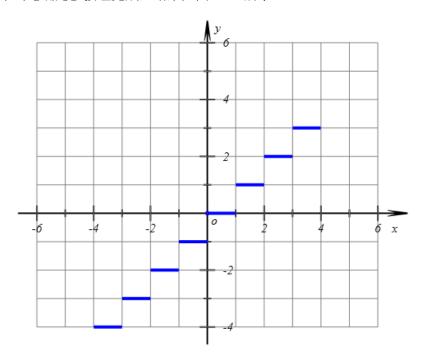


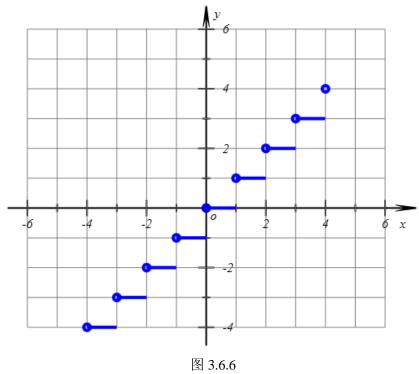
图 3.6.4

对于阶梯函数的每一段,应该是包含左端点而不包含右端点。我们通过画曲线的方式一次性地作出每一段的左端点。操作如下:

- (5) 执行【画图】【一般曲线】【幂函数: y=f(x)】命令,在"y="对应的编辑框中输入: 按【确定】生成直线。
- (6) 右击曲线,如图 3.6.5 所示:设置画线的宽度为: 4,设置"间断点的最小值"为: 0.5,设置"曲线的点数"为: 2,设置变量 x 的范围为: -4 到 4,选择"画点"选项,设置"点的大小"为: 4,单击【确定】按钮完成。完成后如图 3.6.6 所示:



图 3.6.5



分段函数 g(x)的表达式如下:

$$\mathbf{y} = \begin{cases} \mathbf{x}^2 & \mathbf{x} < \mathbf{2} \\ (4-\mathbf{x})^2 & \mathbf{x} \ge \mathbf{2} \end{cases}$$

这是常见的分段函数的表达形式。为了画出这种形式的分段函数的图像,我们需要了解符号函数 sign(a,b)。

a 和 b 是符号函数的两个参数,当 a 大于 b 时,sign(a,b)的结果是 1;当 a 不大于 b 时,sign(a,b)的结果是 0。

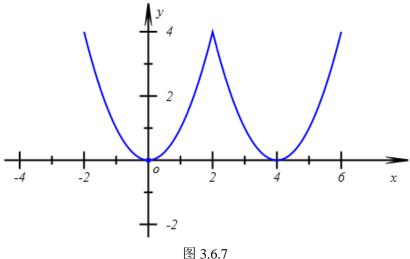
所以上面 g(x)的表达式可以表示为:  $y=sign(2,x)*x^2+(1-sign(2,x))*(4-x)^2$ 。

当 x < 2 时,sign(2,x)为 1,则  $y=1*x^2+(1-1)*(4-x)^2=x^2$ ;

当  $x \ge 2$  时,sign(2,x)为 0,则  $y=0*x^2+(1-0)*(4-x)^2=(4-x)^2$ 。

下面我们画出 g(x)在区间[-2, 6]上的图像, 操作如下:

(7) 在新建文档中,执行【画图】【一般曲线】【幂函数】命令,如下图所示,在"y=" 对应的编辑框中输入:  $sign(2,x)*x^2+(1-sign(2,x))*(4-x)^2$ ,设置"曲线的点数"为: 500,设置变量x的范围为: -2 到 6,单击【确定】按钮完成,结果如图 3.6.7 所示:



#### 【思考与练习】

(1) 画出下列函数的图像:

$$y = \begin{cases} 2 & 0 < x < 5 \\ 3 & 5 \le x < 10 \\ 4 & 10 \le < 15 \\ 5 & 15 \le x \le 20 \end{cases}$$

(2) 画出下列函数的图像:

$$\mathbf{y} = \left\{ \begin{array}{ccc} 11 & 0 {<} x {\leq} 3 \\ \\ 2.1x {+} 4.7 & 3 {<} x {\leq} 10 \\ \\ 3.15x {-} 5.8 & x {>} 10 \end{array} \right.$$

## 3.7 描点连线画函数图像

通过先列表后描点再连线的方式,是一般数学教材向学生阐述画函数图像通用方法。例 如刚开始学习画函数  $y=x^2$  的图像时,其过程基本上都是:

- (1) x 取-3, -2, -1, 0, 1, 2, 3, 然后 y 取对应的数值。
- (2) 将对应的点在坐标系中描出来。
- (3) 最后"用光滑的曲线顺次连接各点"。

于是就得到函数  $y=x^2$  的图像。

也有的教材将第(3)步中的"光滑"叙述成"平滑"。

而无论是"平滑"或者"光滑",对于中学生尤其是初中阶段的学生是无法理解的,而且老 师根据学生知识的掌握情况, 也是难以说清楚的。

但是学生对直线形(例如线段、多边形)是熟悉的。所以若将相邻两点用直线段连接起

来,则是他们易于理解并且是易于想到的。

而用直线段连接起来的折线段并不是函数的图像。

不过,利用计算机则可以方便地增加曲线上样点的个数,随着样点的个数增加,让学生观察函数的图像性质,从而抽象出函数图像的形状。下面我们完成这个过程:

- (1) 在坐标系的属性对话框中,选择"画坐标网格"和"显示刻度"选项。
- (2) 在新建文档中,执行【画图】【一般曲线】【幂函数】命令,在弹出函数作图对话框中: "y="对应的编辑框中输入: x^2,然后在下面的曲线属性中,如图 3.6.8 所示: 设置"曲线的点数"为: 6,按【确定】按钮完成后,右击曲线弹出属性面板,选择曲线以"折线段"方式显示,设置变量 x 的参数范围为: -3 到 3,选择在折线段上"画点",设置点的大小为: 2; 最后单击【确定】按钮完成。

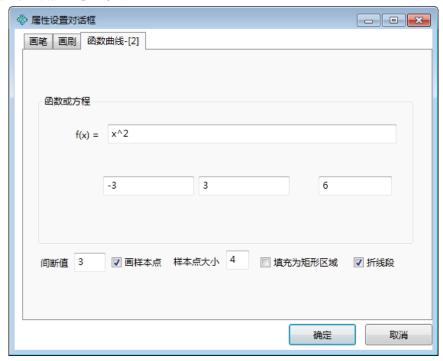


图 3.6.8

结果如图 3.6.9 所示:

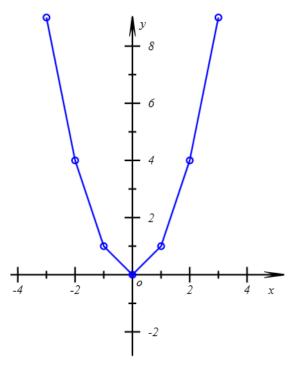


图 3.6.9

打开曲线的属性对话框,若将"曲线的点数"改为: 12,结果如图 3.6.10 所示:

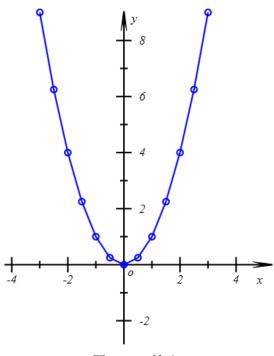
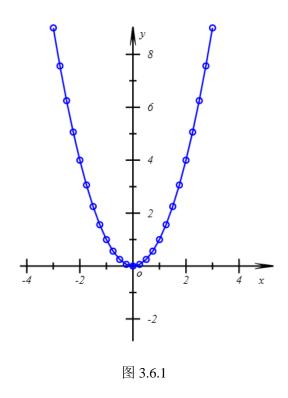


图 3.6.10 所示

打开曲线的属性对话框,若将"曲线的点数"改为: 24,结果如图 3.6.11 所示,



### 【思考与练习】

- (1插入参数n, 作坐标点A (floor(n),-2), 测量点A的x坐标。
- (2) 拖动点参数 n 的控制点,使得其 x 坐标为正数。
- (3)打开曲线的属性对话框,设置"曲线的点数"为: 6\*v000+1。其中 v000 为点 A 的 x 坐标测量值。
  - (4) 向右拖动点 n 的控制点,观察函数图像的变化趋势。

# 3.8 美丽的玫瑰线

(1) 在新建文档中,执行【画图】【一般曲线】【极坐标方程:  $\rho=f(\theta)$ 】命令,如图 3.9.1 所示,在" $\rho=$ "对应的编辑框中输入: 3\*cos(2\*thet),设置曲线的点数为: 500,设置参数 *thet* 的范围为: 0 到 2\*pi。



图 3.9.1

(2) 单击【确定】按钮完成,打开【视图】菜单勾选【系统坐标系】,使坐标系可见,结果如图 3.9.10 所示。

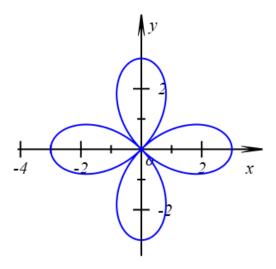


图 3.9.10

下面将表达式 cos(2\*thet)中的数值 2 修改为参数 n,操作如下:

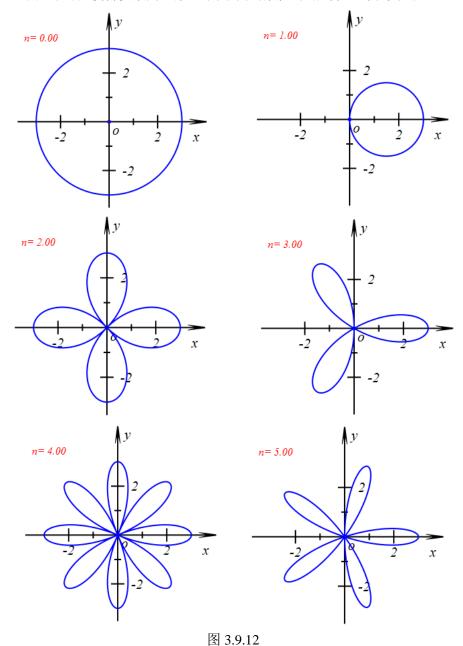
- (3) 执行【插入】【参数】命令,插入参数n。
- (4) 执行【测量】【表达式】命令,输入测量表达式:floor(n),系统自动用参数 v000 记录。
- (5) 右击测量表达式,在弹出的属性菜单中,图 3.9.11 所示,将测量文本的名称修改为"n=",并且以整数格式显示。



图 3.9.11

- (6) 打开曲线的属性对话框,将方程表达式修改为: cos(v000\*thet)。
- (7) 设置曲线的颜色为:红色。

将点n向右拖动,使得参数为正数。n为不同的数值时,所得曲线形状如图3.9.12所示:



#### 【思考与练习】

- (1)继续改变参数 n 的值,观察曲线的变化情况,你能发现什么规律?
- (2) 将曲线的方程修改为:  $\rho=3*cos(thet/v000)$ ,改变参数 n 的值,观察曲线的变化规律。
- (3) 将方程  $\rho$ =3\*cos(thet/v000)对应的曲线的参数 thet 的范围设置为 0 到 2\*v000\*pi,再次观察曲线的性质,你能观察到曲线的哪些特点?你能解释这种现象吗?

# 3.9 根据通项公式画数列的图像

数列,就是一种数集到数集的对应。只不过与学生之前所熟悉的函数概念相比较来说,它是定义在正整数集上的函数。在学生对函数概念有了一定的理解和基础之后,从函数的观点研究数列的性质,可以起到事半功倍的效果。

在研究数列的过程中,若知道它的通项公式,像之前定义域为连续的函数一样可以在 计算机上轻松地画出它的图像。通过观察数列的直观图像,学生可以更容易了解数列的规 律和性质,有助于学生对数列的认识和理解。

#### (一) 利用函数曲线命令画数列的图像

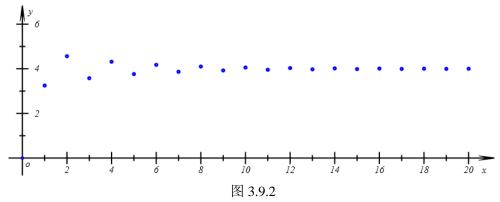
画出通项公式为 的数列前 20 项的函数图像:

- (1) 在新建文档中,在坐标系的属性对话框中选择"显示刻度"和"坐标网格"选项。
- (2) 打开函数作图对话框,在"y="对应的编辑框中输入: 4+4\*(-3/4)^x,单击【确定】生成函数曲线;右击曲线,弹出曲线属性按钮,如图 3.9.1 所示:设置间断点的间断值为: 1,设置曲线的点数为: 19,设置参数范围为: 1到 20,设置点的大小为: 2。



图 3.9.1

(3) 在【画笔】属性页面中,设置画线宽度为: 3,单击【确定】按钮,结果如图 3.9.2 所示:



这是巧妙地利用了计算机画图的特点,根据数列的的通项公式(函数表达式)作出了数列的图像。

#### 【思考与练习】

- (1) 请作出上面的数列前 50 项的图像。
- (2) 请作出数列 an=(-1)^n 前 20 项的图像。

#### (二) 利用编程实现画通项公式对应数列的图像

- (1) 在新建文档中,显示坐标的刻度和网格。
- (2) 执行【插入】【按钮】命令,如图 3.9.3 所示:在弹出的对话框中填入标题:画数列的图像,第二次参数框中输入: *n*:请输入项数 *n*:,程序命令中输入:

```
for(i=1;greaterEql(n, i);i=i+1) \\ \{ \\ CoordPoint(i,4+4*(-3/4)^i, ); \\ \}
```

点击【修改动作】按钮,接着按【确定】按钮完成按钮的制作。

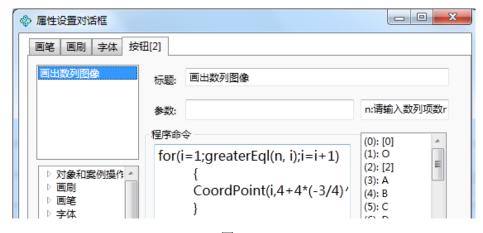
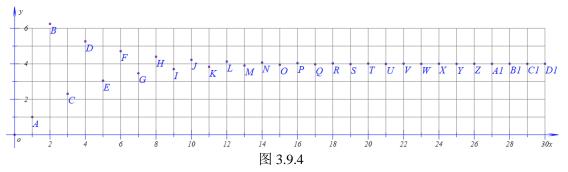


图 3.9.3

(3) 点击【画数列的图像】按钮,执行后,输入n的数 30,结果如图 3.9.4 所示:



(4) 可以将所点的名字隐藏。

#### 【思考与练习】

请你定义一个函数画出数列从 k 项开始的 m 项的图像, 并画出数列 m=10 到 25 的图像。

## 3.10 根据递推公式画数列的图像

限于传统教学手段的约束,一般教材上关于数列的递推公式讲得相当简单,往往通过 一个例题就给出递推公式的概念,然后就是几个根据递推公式求数列前几项的练习。

利用传统的纸笔,根据数列的递推公式计算数列时间非常烦琐的事情,尤其是越往后计算就越复杂和困难。在这种情况下,教学过程中或在作业练习中一般也只要求学生算出数列的前几项,就草草了事。这样,学生学习起来就相当乏味,而且教学的意义不大。

我们知道,根据一个递推公式给出的数列,它的性质敏感地依赖于初值和公式中的参数,它能直观、形象地表现出自然界广泛存在的混沌现象。关于混沌的研究,是现代数学研究前沿一个十分活跃的分支。而且现在在诸多领域对混沌已经有广泛的应用。

利用计算机计算速度快、作图精确等特点和功能,在计算机上通过递推公式计算出更多的数列项并准确画出对应的点是轻松、容易的工作。这样就为学生提供了利用数列的递推公式方便地观察、了解、学习和研究混沌现象的实验环境。

尽管混沌不是中学数学范围内的主要内容,并且也不能期望学生对它有多么深刻的理解和认识。但是,这可以激发学生对混沌这一数学现象的兴趣,可以了解到现代数学的气息,可以对学生在计算机上进行数学实验和研究的认识产生深刻的影响。更重要的是,在学习递推公式的内容时,也不会那么孤立和枯燥乏味。

下面,我们就通过一个具体的例子说明在计算上研究递推公式数列和混沌现象的实验过程。

递推公式
$$a_{n+1} = (1+r)(a_n - a_n^2)$$
, 首项为 0.2

(1)执行【画图】菜单下的【变量迭代】命令,打开输入框。按图 3.10.1 所示填写对话框:其中 n 作为点列的 x 坐标,初始值为 1,为正整数数列。在"变量迭代"的输入框中的"初值"框中输入 1,在"迭代变量"框中分别输入 n,n+1。然后单击左上角的【增加】按钮,这样就增加了一个数列。



图 3.10.1

构造一个数列:  $a_{n+1} = (1+r)(a_n - a_n^2)$ , 首项为 0.2, 作为点列的 y 坐标。

(2) 在"变量迭代"的输入框中的"初值"框中输入 0.2,在"迭代变量"框中分别输入 an,(1+r)\*(an-an^2)。然后单击左上角的"增加"按钮,这样就增加了一个新数列。如图 3.10.2 所示

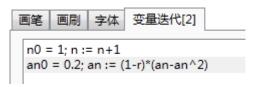


图 3.10.2

使用两个数列构造一个点列。

(3) 如图 3.10.3 所示: 在 "x" 框中输入 0.5\*n,在 "y" 框中输入 50\*an,(x 坐标缩小为 0.5,y 轴坐标放大 50,想想为什么要这样?),点击下面的"增加"按钮,则增加一个点列。

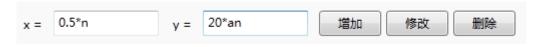


图 3.10.3

设置迭代层数为 200, 勾选"连成折线段", 去掉勾选"画迭代表格"单击"确定"完成。

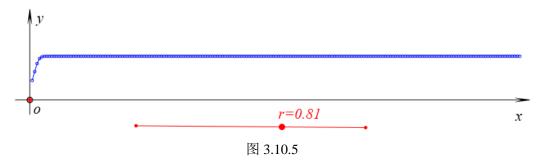
这时可能还看不到,不用着急,这是因为r的值不在合理范围内造成的,下面添加r的控制条,把它改到合适的范围。

(3) 点击"插入"下的"变量",如图 3.10.4 输入 r,范围为-3~3。单击"确定"完成。

变量		
r		
最小值		
-3		
最大值		
3		
确定		

图 3.10.4

(4)为方便观察,右击坐标系,在属性窗口中把坐标系的范围调整到 0 到 200。 结果如图 3.10.5 所示,坐标系中是递推公式  $a_{n+1}=(1+r)(a_n-a_n^{\ 2})$  ,  $a_0=0.2$  对应数列的 前 200 项对应的图像,其中相邻两点之间用直线段连接。当前图像是参数 *r*=0.81 时的情形,可以观察到这个数列趋向于某一个极值。



通过参数r的变量控制对象,改变r的值,可以发现数列会发生对应变化。

如图 3.10.6 所示,当 r=1.52 时,这个数列仍是一个收敛数列,不过是趋向于另一个不同的数值。

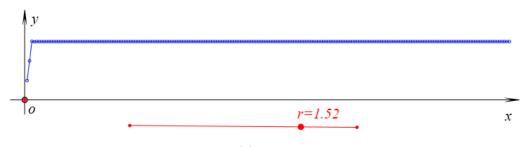


图 3.10.6

如图 3.10.7 所示,当 r=2.45 时,这个数列好像并不是一个收敛的数列,而是从某一项起数列的值发生周期性的变化。若这时的数列真是具有这种周期变化的性质,则可以预则项数很大时数列的值。这个性质,可以通过观察更多地项,加以验证。具体如何观察更多项的操作,请参照后面【**请你动手**】中的内容。

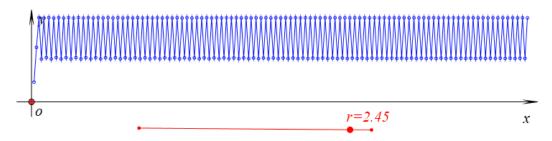


图 3.10.7

如图 3.10.8 所示,当 r=2.63 时,这个数列好像按照另一种情况发生周期性的变化。在上图中,每经过两个项就重复一次,这种情况成为 2-循环;在这里是每经过四项就重复一次,这种情况成为 4-循环。继续改变 r,是否会出现 6-循环 8-循环、16-循环呢?而是否对应任意 r,这个数列从某一项开始都会出现循环呢?这就引发了一系列有趣的同时也具有挑战性的问题。

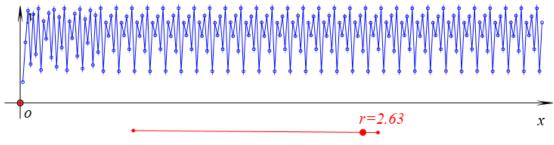


图 3.10.8

果然, 当r为 2.63 时, 出现了 6-循环。

也可以使得r取其他数值,继续进行观察和研究。另外,若首项 $a_0$ 的初始值发生改变,会对数列的性质产生怎样的影响呢?

#### 【思考与练习】

- (1) 请你动手研究r为其他数值时,数列的性质。
- (2)对于上面的各种情况下,让r保持不变,改变数列的首项,即点A的y坐标,观察数列图像的性质。
- (3)若数列的递推公式是:  $a_{n+1}=(a_{n-1}-a_n^{\ 2})$ ,  $a_1=a$ ,  $a_2=b$ 。 试在坐标系中作出它的前 100 项对应的图像。并研究在 a、b 分别取不同数值时,对应数列的变化趋势。

# 第四部分 解析几何问题

### 4.1 固定长度的线段在滑动

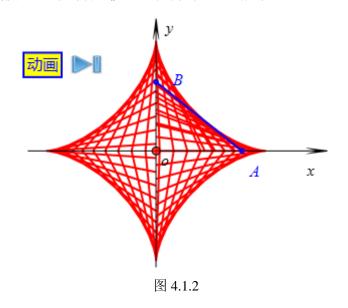
现在我们通过解析几何的方法解决固定长度的线段其两个端点分别在两条线段上滑动的问题。

- (1) 在新建文档中作坐标点 *A* (3\*cos(t), 0)、*B* (0, 3\*sin(t))。
- (2) 连接线段 AB, 颜色修改为红色。
- (3) 执行【插入】【常用按钮】【变量重复运动】命令作点 A 的动画按钮,按图 4.1.1 填写"动画"项的程序命令:设置运动范围为: 0 到 2\*pi,运动频率为 50,选择运动类型为: 0 (重复运动)。



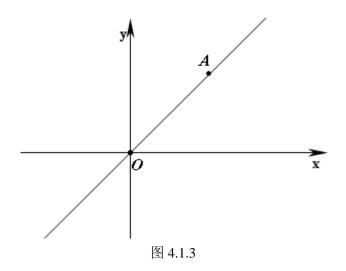
图 4.1.1

(4) 跟踪线段 AB, 单击动画按钮, 结果如图 4.1.2 所示:



#### 【思考与练习】

如图 4.1.3 所示,假设直线 OA 的斜率为 k,如何构造出两端点分别在 x 轴、直线 OA 上滑动的固定长度的线段?



### 4.2 斜率之积为定值的两直线交点的轨迹

问题: 已知经过点 A 的直线与经过点 B 的直线的斜率之积为 1,探索两直线交点 C 的轨迹。

下面我们作出满足上述要求的点 C:

- (1) 在新建文档中,作任意点A和点B。
- (2) 选择点 A,执行【作图】【直线】【点斜式直线】命令,如图 4.2.1 所示,在弹出的直线构造对话框"斜率"编辑栏中输入: k,单击【确定】按钮,作出经过点 A 斜率为 k 的直线。



图 4.2.1

- (3) 类似地:选择点 B,作出斜率为 1/k 的直线。
- (4) 作出两条直线直线的交点 C,跟踪点 C。
- (5) 增加参数 k 的变量控制尺,取值范围为: -10 到 10。

通过参数 k 的变量控制尺改变参数 k 的值,可以发现点 C 的轨迹是两条分别以点 A、点 B 为顶点的双曲线。结果如图 4.2.2 所示:

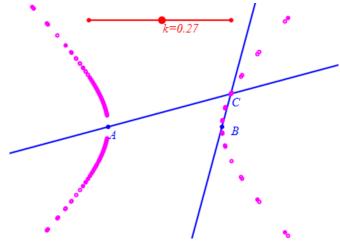


图 4.2.2

设置经过点 A 的直线的斜率为 k,有一点不能让人满意的地方就是,斜率 k 不能取所有 数值。结果也使得点C的轨迹曲线在点A附近留有一个小空隙。

为了改善这个结果,我们将经过点 A 的直线的倾斜角设置为 k。操作如下:

(6) 右键经过点 A 的直线,在弹出的属性对话框中,如图 4.2.3 所示,在斜率编辑框 中输入: tan(k), 单击【确定】按钮完成。



图 4.2.3

- (7) 右键经过点 B 的直线,在弹出的属性对话框中,将其斜率修改为: 1/tan(k),单击 【确定】按钮完成。
  - (8) 将参数 k 的变量控制尺的参数可拖动范围修改为: 0 到 3.141593。 再次通过参数 k 的变量控制尺改变参数 k 的值,结果如图 4.2.4 所示。

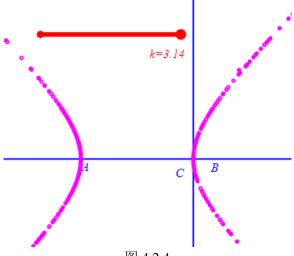


图 4.2.4

若两直线的斜率之积为定值 a,则点 C 的轨迹又可能是什么呢? 为了构造点 C 的轨迹

图形,可以进行如下的构造:

- (9) 在新建文档中,作任意点A、B和,插入参数k,设置范围为-Pi/2到Pi/2。
- (11) 经过点 A 作斜率为 tan(k) 的直线。
- (12) 作出经过点 B、斜率为 a/tan(k) 的直线。
- (14)作出两直线交点 E。
- (15) 依次选择变量 k (控制条) 和点 E,执行【画图】【轨迹】命令,右键生成的轨迹,如图 4.2.3 所示,在弹出的【属性设置对话框】中设置的【轨迹】选项页面中设置合适的参数:

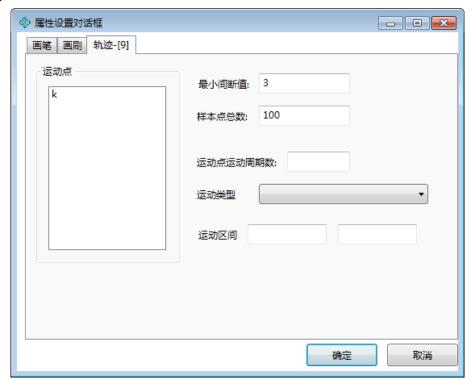
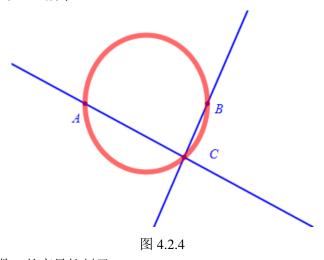


图 4.2.3

(16) 在【画笔】选项页面中,选择画线的类型为:虚线,画线颜色为:红色。单击【确定】按钮,结果如图 4.2.4 所示:



(17) 增加参数 a 的变量控制尺。

通过参数 a 的变量控制尺改变参数的值,可以观察到当 a 大于 0 时,点 E 的轨迹是双曲线; 当 a 小于 0 时,点 E 的轨迹是椭圆。

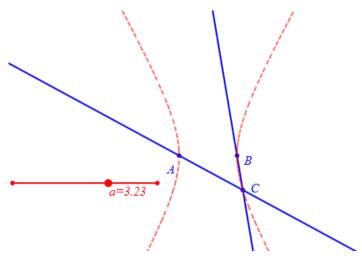


图 4.2.4 a 大于 0 时轨迹为一双曲线

### 【思考与练习】

- (1) 当 a<-1 与-1<a<0 时,点 E 的轨迹都是椭圆,这两种情况下的椭圆有什么不同之处?
  - (2) 增加 a=-1 的动作按钮,观察 a=-1 时,点 E 的轨迹曲线。
  - (3) 增加 a=0 的动作按钮,观察 a=0 时,点 E 的轨迹曲线。
- (4)为了更加透彻地研究点 E 的轨迹,通过建立适当的坐标系,求出点 E 的轨迹方程,验证上面的结论。

# 4.3 根据定义构造椭圆和双曲线

- (1) 任意画一个点 A, 以点 A 为圆心作半径为 3 的圆。
- (2) 在圆A上任意取一点B, 在圆A内部任意取一点C。
- (3) 连接 AB、CB。
- (4) 作线段 CB 的垂直平分线且交 AB 于点 E, 作线段 DE, 结果如图 4.3.1 所示:

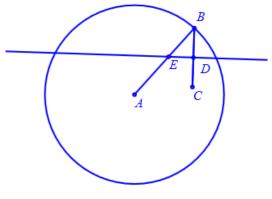


图 4.3.1

- (5) 跟踪点 E。在左边对象工作区中选择 E 的跟踪对象,设置其跟踪对象的画线颜色为:红色,执行【编辑】【对象管理】【移动对象到最后面】命令,把轨迹移动到最后面。
- (6) 执行【插入】【常用按钮】【对象重复运动】命令,如图 4.3.2 所示:增加点 B 的重复动画按钮。

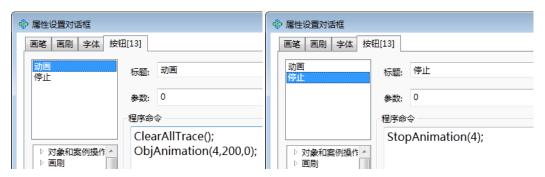


图 4.3.2

单击【动画】按钮,结果如图 4.3.3 所示:

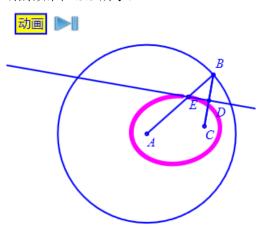


图 4.3.3

将点 C 拖动到圆 A 之外,再次单击【动画】按钮,结果如图 4.3.4 所示:

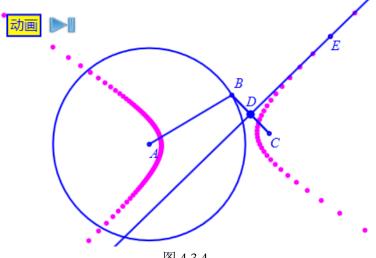
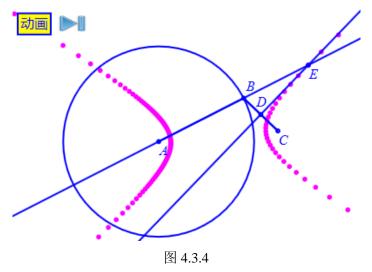


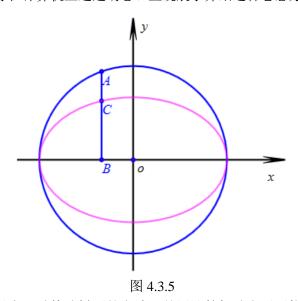
图 4.3.4

为了更加清晰地说明点 E 为直线 DE 和 AB 的交点,可以作出过点 A 和点 B 的直线。结 果如图 4.3.4 所示:



### 【思考与练习】

(1)圆是椭圆的特殊情形。也可以把椭圆看作是将圆沿着某一方向伸缩而得到的图形,如图 4.3.5 所示,请你在计算机上通过动态、直观展示介绍这种思想方法。



(2)请你自己思考一种构造椭圆的方法,并用计算机动态展示描绘椭圆的过程。

## 4.4 根据定义构造抛物线

(1) 任意画点 A 和一直线。

下面我们构造以点A为焦点、以线段BC所在直线为准线的抛物线。继续操作如下:

- (2) 在直线上任意取一点 B, 连接 AB, 取 AB 的中点 C。
- (3) 过点 C 作直线线 AB 的垂线。
- (4) 过点 B 作原直线的垂线。
- (5) 作出前两部作出的直线的点 D。结果如图 4.4.1 所示:

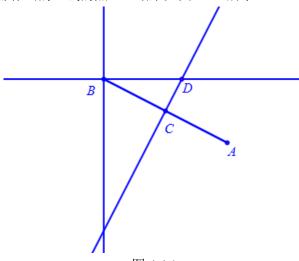
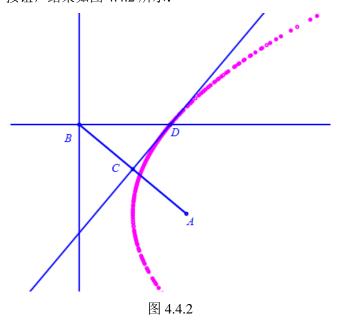


图 4.4.1

- (6) 跟踪点 D。选择 D 的跟踪对象,设置其跟踪对象的画线颜色为:红色,执行【编辑】【对象管理】【移动对象到最后】命令。
  - (7) 增加点 B 的动画按钮,设置"动画的运动频率"为: 300,保留其他属性不变。 单击【动画】按钮,结果如图 4.4.2 所示:



### 【思考与练习】

(1) 如果 C 点不是线段 AB 的中点,而是线段 AB 上的某定比分点,则轨迹会变成什

### 么样子呢?

(2) 如果直线 CD 点不是垂直于线段 AB,而是与线段 AB 成某个夹角,则轨迹会变成什么样子呢?

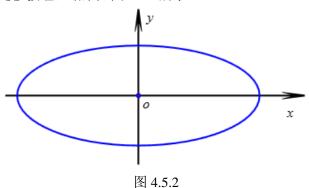
### 4.5 构造标准圆锥曲线

下面作出以坐标原点O为中心的标准椭圆,操作如下:

- (1) 选择点 O,执行【作图】【圆锥曲线】【标准椭圆】命令,弹出构造标准圆锥曲线的属性对话框。
  - (2) 如图 4.5.1 所示,分别填入需要的长轴与短轴长即可。



(3) 单击【确定】按钮,结果如图 4.5.2 所示:



我们还可以作出长半轴和短半轴为参数的椭圆,例如作出长半轴和短半轴之和为3的椭圆,操作如下:

(4) 双击椭圆,打开其属性对话框,如图 4.5.3 所示,将长半轴长修改为: a,将短半轴长修改为: 3-a。

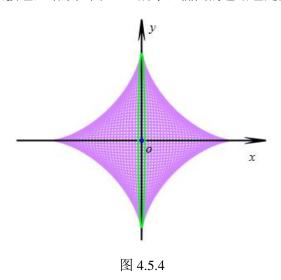


图 4.5.3

(5) 增加参数 a 的动画按钮,设置运动的频率为: 30,参数 a 的范围为: 0.01 到 2.99。

(6) 跟踪椭圆,并设置跟踪颜色。

单击参数 a 的动画按钮,结果如图 4.5.4 所示,椭圆的包络也是星形线。



4.6 研究圆锥曲线的光学性质

首先画任意椭圆:

- (1) 作两个指标点 A 和 B, 坐标分别为 (-3,0) 和 (3,0)。
- (2) 选择这两点,执行【画图】【圆锥曲线】【已知焦点和长半轴的椭圆】命令,如图 4.6.1 所示:在弹出的用户输入对话框中输入长半轴长:5,单击【确定】按钮作出对应的椭圆。



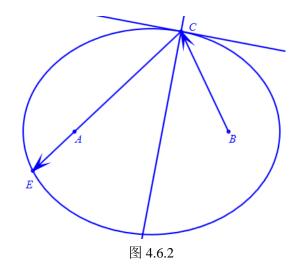
图 4.6.1

- (3) 在椭圆上任取一点 C。
- (4)选择点 C 和椭圆,执行【画图】【直线】子菜单中的【过圆锥曲线上一点的切线】命令,作出椭圆上经过点 C 的切线。
  - (7) 作出经过点 C 与切线垂直的直线,即椭圆在点 C 处的法线。

继续构造入射光线和反射光线:

下面构造从椭圆的焦点 A 处发出的光线,入射到椭圆的内部点 C 处并被椭圆内部反射出去的光线。

- (8) 作出点A 关于法线的对称点D。
- (9) 作出射线 CD, 并作出射线 CD 与椭圆的交点 E; 隐藏射线 CD, 连接线段 CE, 隐藏点 E。
  - (10) 依次选择点 A 和点 C,执行【画图】【直线】【向量】命令,作出向量 AC。
  - (11) 同理作出向量 CE, 结果如图 4.6.2 所示:



### 最后进行修饰:

(12) 你可以将经过点 C 的切线和法线设置为虚线显示。如图 4.6.3 所示:

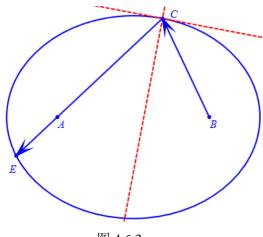


图 4.6.3

拖动点 C, 改变入射光线的位置, 可以发现从右焦点 A 发出的光线点经过椭圆内部被反 射出去后总是经过椭圆的左焦点 B。

### 【思考与练习】

- (1) 请你动作设计探索抛物线光学性质的实验。
- (2) 请你动作设计探索双曲线光学性质的实验。

# 4.7 多驱动点下的轨迹

### (一) 单个驱动点产生的轨迹曲线

如图 4.7.1 所示: 点 A 是平面上的任意点,点 B 是圆 O 上的任意点,点 C 是线段 AB 的 中点。根据你所掌握的知识,你是否知道当点B在圆上运动时点C的轨迹是什么图形?

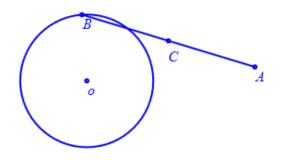


图 4.7.1

- (1)选择坐标原点 O,执行【画图】【圆和圆弧】【已知圆心和半径的圆】命令,在弹出的输入对话框中输入: 2,单击【确定】按钮完成,作出以点 O 为中心、半径为 2 的圆。
- (2)在平面上任取一点 A、在圆 O 上任取一点 B,连接线段 AB,作出线段 AB 的中点 C。
  - (3) 跟踪点 *C*。

拖动点B,可以观察到点C的轨迹是一个圆。

(4) 依次选择点 B 和点 C,执行【画图】【轨迹】命令,即可生成点 B 驱动下点 C 的轨迹。结果如图 4.7.2 所示:

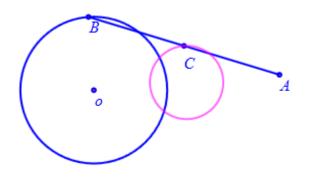


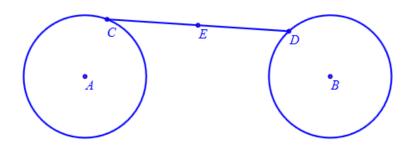
图 4.7.2

### 【思考与练习】

- (1) 你能求得点 C 的轨迹方程吗?
- (2) 若点 C 不是线段 AB 的中点,而是直线 AB 上的任意一点,那么点 B 在圆周上运动时点 C 的轨迹是什么图形呢?请你重新设计类似的实验,探索点 C 的轨迹。

### (二) 两个点驱动下的轨迹曲线

如图 4.7.3 所示: 圆 O 和圆 B 的半径大小相等,点 C 和点 D 分别是圆 A 和圆 B 上的点,点 E 是线段 CD 的中点。当点 C、点 D 分别在圆 A、圆 B 上以不同的速度运动时,点 E 的轨迹是什么图形?



下面我们在计算机进行实验,研究点C的轨迹图形的性质。

- (1) 在平面上任取两点  $A \times B$ ; 分别作出以点  $A \times$  点 B 为圆心, 半径均为 2 的两个圆。
- (2) 在圆 A 上任取一点 C, 在圆 B 上任取一点 D, 作出线段 CD 的中点 E。
- (3) 执行【插入】【常用按钮】【对象反复运动】命令,如图 4.7.3 所示修改参数,作出点 A 和 C 一起运动的动画按钮。



图 4.7.3

#### (4) 跟踪点 E。

当点 C 和点 D 同时分别在圆 A 和圆 B 上运动的过程中,点 E 的轨迹是什么图形?我们可以进行实验观察,单击动画按钮结果如图 4.7.4 所示:

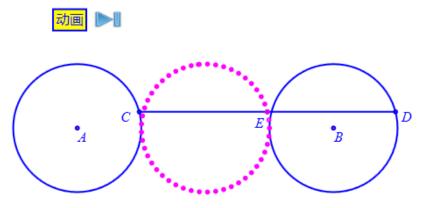


图 4.7.4

可以发现,在上述实验过程中,点 C 从有向线段 AC 与 x 轴正方向的夹角为 0 度的位置绕点 A 按照逆时针方向旋转了一周;与此同时,点 D 从有向线段 BD 与 x 轴正方向的夹角为 0 度的位置绕点 B 按照逆时针方向旋转了一周。

若点 C 绕点 A 逆时针方向旋转一周的同时,点 D 绕点 B 顺时针旋转了一周,那么点 E 的轨迹图形是什么形状的呢?

(5) 右击动画按钮弹出属性对话框,如图 4.7.5 所示:修改动画的程序命令,运动类型分别为:3(正向一次运动),4 反向一次运动。

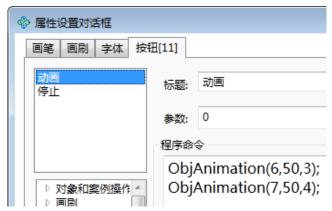


图 4.7.5

点击【确定】后按下个动画按钮,结果如下图 4.7.6 所示:

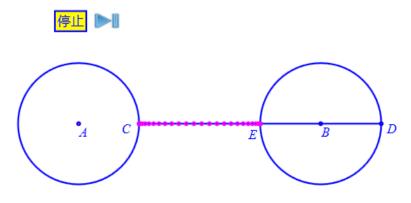


图 4.7.6

若点 C 绕点 A 逆时针方向旋转 a 周的同时,点 D 绕点 B 逆时针旋转了 b 周,那么点 E 的轨迹图形是什么形状的呢?

为了更加方便地研究点 E 在点 C 和点 D 驱动下的轨迹图形,我们可以直接作出点 E 的轨迹曲线,操作如下:

- (6) 为了不影响观察, 先删除前述步骤所作的轨迹。
- (7) 依次选择点 C、点 D 和点 E,执行【画图】【轨迹】命令,右键生成的曲线,如图 4.7.7 所示,在弹出的轨迹属性对话框中:设置间断点的最小值为:0.5;样本点总数为:5000;设置点 C 运动区间的最大值为:a\*2\*pi,设置点 D 运动范围的最大值为:b\*2\*pi,单击【确定】按钮完成,即可生成点 E 的轨迹。



图 4.7.7a 点 C 的运动参数图

4.7.7a 点 C 的运动参数

(8) 执行【插入】【常用按钮】【变量一次运动】,增加参数 a 的动画按钮,如图 4.7.8

所示设置程序命令: 生成变量 a 的值复位到 1 的按钮

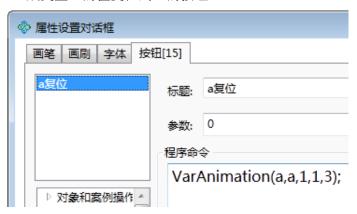


图 4.7.8

(9) 执行【插入】【常用按钮】【变量一次运动】,增加参数 a 的动画按钮,如图 4.7.9 所示设置程序命令:生成变量 a 的值增加 1 的按钮。

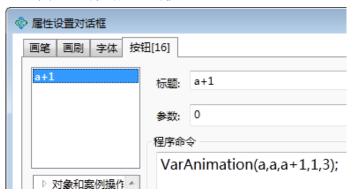


图 4.7.9

(10) 执行【插入】【常用按钮】【变量一次运动】,增加参数 a 的动画按钮,如图 4.7.10 所示设置程序命令:生成变量 a 的值减少 1 的按钮。

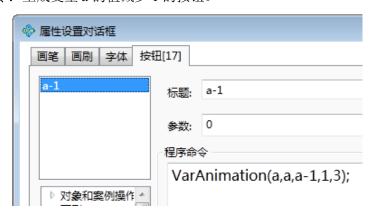


图 4.7.10

- (11) 按照前述方法,分别作出变量 b 的 "b 复位" "b+1" 及 "b-1" 三个按钮。
- (14) 分别测量参数 a 和参数 b 的值。

图 4.7.11 和图 4.7.12 是其中两种轨迹曲线:

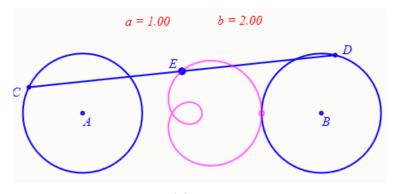


图 4.7.11

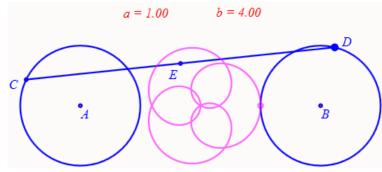


图 4.7.12

为了更加一般地研究点 E 的轨迹曲线,我们可以插入参数 a 和参数 b 的变量控制尺,让参数 a 和参数 b 可以任意的实数。

图 4.7.13 是其中一种情况下的轨迹曲线:

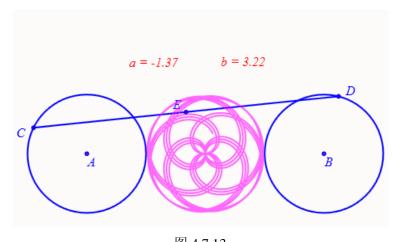


图 4.7.13

#### 【思考与练习】

- (1) 当 a=1、b=2 与 a=3、b=6 时,点 E 的轨迹曲线有什么特征?你能发现哪些规律?
- (2) 通过建立适当的坐标系, 你能写出点 E 的轨迹曲线对应的方程吗?
- (3) 若点 E 不是线段 CD 的中点,而是直线 CD 上的任意一点,请你重新设计类似的实验,研究点 E 的轨迹曲线,能产生什么新的结论?
  - (4) 若圆 A 和圆 B 的半径不同,请你重新研究上面的问题,是否有不同的结论?
- (5) 若点 C 和点 D 在同一个圆上,请重新研究点 E 的轨迹曲线,与上面的实验相比能有哪些共同点和不同点?

### (三) 三点驱动下的轨迹曲线

- (1) 在平面上任取三个点A、B、C。
- (2)作出以点 A 为圆心、半径为 1 的圆;作出以点 B 为圆心、半径为 2 的圆;作出以点 C 为圆心、半径为 3 的圆。
  - (3) 在圆A上任取一点D、在圆B上任取一点E、在圆C上任取一点F。
- (4) 连接线段 DE,在 DE 上任取一点 G,连接 GF,在 GF 上任取一点 H。结果如图 4.7.14 所示:

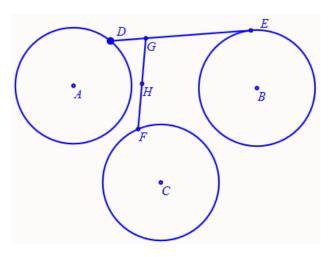


图 4.7.14

(5) 依次选择点 D、点 E、点 F 和点 H,执行【画图】【轨迹】命令,右键作出的轨迹,在弹出的轨迹属性对话框中,设置运动点的样本点总数为:5000,设置点的最小间断值为:0.5,设置点 D 运动范围为:0 到 a\*2\*pi,设置点 E 运动范围为:0 到 b\*2\*pi,设置点 E 运动范围为:0 到 c\*2\*pi。它们的运动类型均设为【向前】。如图 4.7.15 所示为点 D 的运动参数:

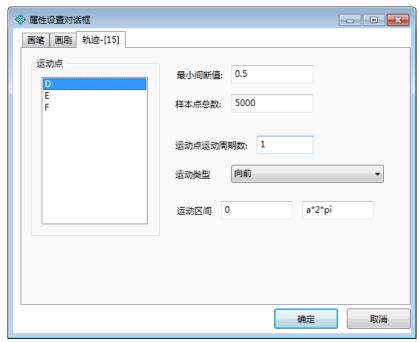


图 4.7.15

(6) 在【属性】菜单中,设置轨迹的画线颜色为:红色;在画刷属性页面中,选择填充类型为:路径渐变填充。

- (7) 增加参数 a、b、c 的变量控制尺,设置可改变范围为: -100 到 100。
- (8) 测量变量 a、b、c 的值。

图 4.7.16 是几种不同情况下的轨迹曲线:

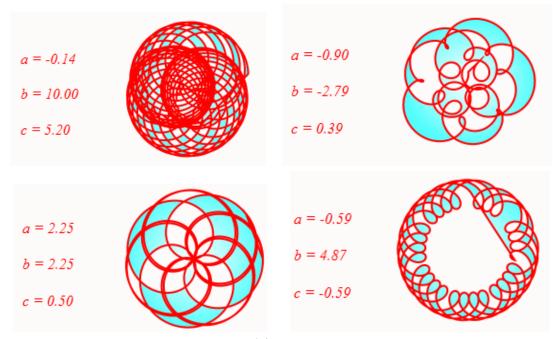


图 4.7.16

### 【思考与练习】

- (1) 自己继续动手操作、观察、探索和实验, 你能发现多少规律?
- (2) 通过建立适当的坐标系, 你能写出点 H 的轨迹曲线对应的方程吗?
- (3) 当点 G 在直线 DE 上的不同位置时, 曲线变化有什么规律?
- (4) 当点 H 在直线 GF 上的不同位置时, 曲线变化有什么规律?
- (5) 若圆 A、圆 B 和圆 C 的半径之比为其他值时,请你重新研究上面的问题,会有新的结果出现吗?
  - (6) 若三个点 D、E、F 在同一个的圆上, 会得到不同的结论吗?

# 第五部分 三角函数关系

### 5.1 利用正弦线画正弦函数的图像

为了研究正弦函数 y=sin(x)的性质,需要先画出正弦函数的图像。在单位圆上,正弦线就代表了对应角的正弦值,所以可以利用正弦线画出正弦函数 y=sin(x)的图像。

相对于传统的作图手段,利用计算机作图可以在圆周上取任意多条正弦线,使得到的函数曲线更精细、更光滑,也更快捷、省时省力。而且利用计算机画图还可以动态地展示正弦线连续变化生成正弦函数图像的过程,让学生通过观察详细的函数图像生成过程,对曲线的升降起伏变化有形象而直观的了解,对正弦曲线其关键作用的 5 个点(最高点、最低点、与 x 轴的交点)产生深刻的认识。

下面是具体的操作过程。

首先以 x 负半轴上的点为圆心作单位圆:

- (1) 执行【视图】【系统坐标系】命令使得系统坐标系可见,单击【画图】【参数点】 【坐标点】命令,作出坐标点(-2,0),将其名字修改为: *O*'。
- (2)选择点O',执行【作图】【圆和圆弧】【已知圆心和半径的圆】命令,在弹出的用户输入对话框中,输入半径: 1,单击【确定】按钮完成。

取角的终边在单位圆上的点,并作出对应正弦线:

- (3) 作出圆上的自由点 B; 右击点 B, 将其名字修改为: P; 连接 O'P。
- (4) 执行【画图】【直线】【一次方程直线】,在弹出输入框中输入: y=0,在 x 轴上的直线; 过点 P 作到该直线的垂足 C,作圆与直线的交点 D,依次选择圆和点 D、点 P,执行【画图】【圆和圆弧】【圆周上的圆弧】命令,作出圆弧 DP,结果如图 5.1.1 所示。

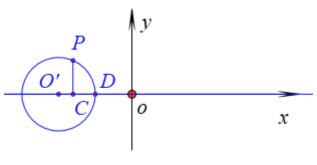


图 5.1.1

作正弦线对应函数曲线上的点:

(5) 测量点 P 纵坐标,测量弧 DP 的的角度,右击该测量值,在弹出菜单中如图 5.1.2 所示把测量结果该为弧度:。作坐标点 E(v001,000)、F(v001.0)。



图 5.1.2

v000 是储存度量点 P 的纵坐标值的变量,由计算机系统自动生成。v001 在这里表示以 O '为顶点、射线 O 'O 为始边、O 'P 为终边的角的大小。

(6) 连接 PE, 连接 EF。结果如图 5.1.2 所示。

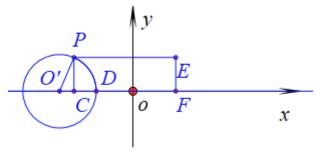


图 5.1.2

增加控制点P的动作按钮:

(7) 执行【插入】【常用按钮】【对象一次运动】命令,如图 5.1.2 所示,修改动画程序命令为: *ObjAnimation*(4,50,3);单击【确定】按钮退出。



### 跟踪E点:

(9) 选择点 E、执行【画图】【跟踪】命令。

增加与正弦线同步的函数曲线:

(10) 执行【画图】【一般曲线】【三角函数: y=f(x)】,按照图 5.1.3 填写参数,单击【确定】按钮退出。

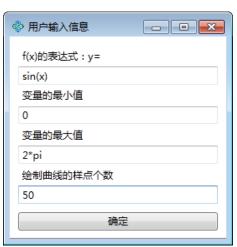


图 5.1.3

显示坐标系的坐标刻度:

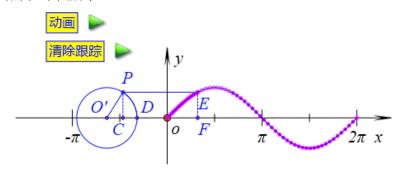
(11) 右击坐标系,打开其属性对话框;如图 5.1.4 所示,勾选【x轴  $\pi$ 为单位】,单击【确定】按钮退出。



图 5.1.4

### 最后进行修饰:

- (12) 设置线段 DE、线段 PC 为虚线。隐藏直线 y=0。
- (13) 执行【插入】【常用按钮】【清除跟踪】命令,点击【确定】生产清除跟踪轨迹的按钮。完成结果如下图所示:



单点【清除跟踪】,将清除跟踪E的跟踪轨迹。

点击【动画】,则生成0到2\*pi之间的图像。

右击【动画】按钮打开其属性对话框,可以通过修改其运动频率,改变正弦线及其端点(函数曲线上的点)的跟踪数目。

### 【思考与练习】

- (1) 根据定义得到余弦函数曲线
- (2) 根据定义得到正切函数曲线

## 5.2 y = sin(x) -> 3sin(2x + pi/4)的动态变换

对于函数  $y=Asin(\omega x+\varphi)$ ( $x\in R$ , A>0,  $\omega>0$ )的图像,可以通过列表、描点、连线的方式得到,但通过下面的三个步骤得到它的图像,更能体现函数  $y=Asin(\omega x+\varphi)$ 与函数 y=sin(x)之间周期、振幅以及相位等特征量之间的关系。

- (1) 将正弦曲线 v=sin(x)向左(当  $\varphi>0$ )或向右(当  $\varphi<0$ )平行移动 $|\varphi|$ 个单位长度。
- (2) 再将所得曲线上每一点的横坐标伸长  $(0<\omega<1)$  或缩短  $(\omega>1)$  为原来的  $1/\omega$  倍。
- (3)进一步再将曲线上每一点的纵坐标扩大(A>1)或缩小(0<A<1)为原来的A倍。但是,在传统教学手段的约束下,对于上面三个步骤所述的过程,大部分时候也只能让学生开动脑筋去想象一下这些过程是怎么样的,不能产生形象的认识,更是难以开展师生之间、学生之间的交流和探讨。

在计算机的帮助下,能够动态、直观、轻松地展示上面三个步骤中所述的变换过程,通过让学生观察生动、形象的变换过程,可以加深学生对上面三个步骤中所述问题的理解。同时,更清晰的认识到参数 A、 $\omega$ 、 $\omega$  对 v= $Asin(\omega x+\omega)$ 的影响。

让学习过程变得更有趣, 让学习知识变得也更容易。

例如在 Hawgent 动态数学软件中作图,可以直接作出系数带参数的函数方程曲线。计算机通过控制参数的变化来使得函数的曲线对应变化,就完成了动态变换的过程。控制参数的变化方向可以是从 a 到 b,也可以是从 b 到 a,所以动态变换过程也可以是逆向的,让学生从不同过程理解图形变换的内涵。

利用计算机展示动态变换过程,可以多次重复展示,能提高效率并节约时间。

下面是具体的操作步骤。

显示坐标系的坐标刻度:

作 y=sin(x)的图像:

(2)执行【画图】【一般曲线】【三角函数: y=f(x)】,如图 5.2.1 所示,在"y="对应编辑框中输入: sin(x),设置变量 x 的范围为: 0 到 2\*pi,点击【确认】作出图像; 然后右击曲线,在弹出的属性窗口中,"画笔"属性页面中选择画线类型为: 虚线; 单击【确定】按钮退出。

፟ 用户输入信息	
f(x)的表达式:y=	
sin(x)	
变量的最小值	
0	
变量的最大值	
2*pi	
绘制曲线的样点个数	
50	
确定	

图 5.2.1

作  $y=Asin(\omega x+\varphi)$ 的图像:

(3) 执行【画图】【一般曲线】【三角函数: y=f(x)】,如图 5.2.2 所示,在"y="对应编辑框中输入:  $A*sin(\omega*x+\varphi)$ ,设置变量 x 的范围为:  $-\varphi/\omega$  到( $2*pi-\varphi$ )/ $\omega$ 。

፟	
f(x)的表达式: y=	
A*sin(ω*x+φ)	
变量的最小值	
-φ/ω	
变量的最大值	
(2*pi-φ)/ω	
绘制曲线的样点个数	
50	
确定	

图 5.2.1

单击【确定】按钮完成,设置曲线的颜色为红色结果如图 5.2.3 所示:

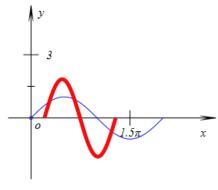


图 5.2.3

关于输入希腊字母  $\omega$  和  $\varphi$  的方法:在输入法工具条中右键单击右侧的软键盘图标 在弹出的符号列表中选择"希腊字母",打开希腊字母的软键盘。在软键盘中,如图 5.2.4 所示鼠标分别单击希腊字母  $\omega$ 、 $\varphi$  所在的键位即可在编辑框中输入。

PC键盘	标点符号	
希腊字母	数字序号	
俄文字母	数学符号	
注音符号	单位符号	
拼音	制表符	
日文平假名	特殊符号	
日文片假名		
<b>劉</b> 全拼 ▶ □ ■		

图 5.2.4

再次单击软键盘图标型,即可退出软键盘窗口。

在上面为什么要将变量 x 的范围设置为:  $-\varphi/\omega$  到(2\* $pi-\varphi$ )/ $\omega$ ? 想想看。增加控制参数变化的动作按钮:

(4) 执行【插入】【常用按钮】【变量一次运动】命令,按图 5.2.5 填写,单击【确定】按钮退出得到  $\varphi$  的变化按钮。

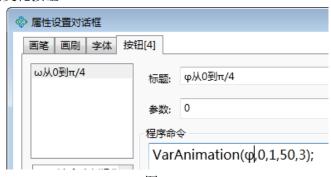


图 5.2.5

(5) 执行【插入】【常用按钮】【变量一次运动】命令,按图 5.2.6 填写,单击【确定】按钮退出得到  $\omega$  的变化按钮。

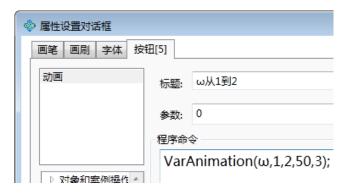


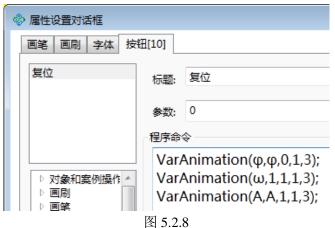
图 5.2.6

(6) 执行【插入】【常用按钮】【变量一次运动】命令,按图 5.2.7 填写,单击【确定】按钮退出得到 A 的变化按钮。



图 5.2.7

(7) 执行【插入】【常用按钮】【变量一次运动】命令,按图 5.2.8 填写,单击【确定】按钮退出得到三个参数复位的按钮。



详细的使用方法如下:

如图 5.2.9 所示,按复位键,在坐标系上, $y=Asin(\omega x+\varphi)$ (红色曲线)的初始位置与 y=sin(x) 重合,此时对应的 A=1、 $\omega=1$ 、 $\varphi=0$ 。

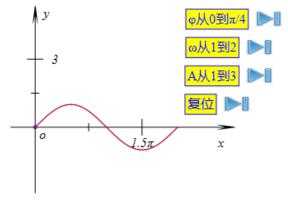


图 5.2.9

(1) 单击【 $\varphi$  从 0 到  $\pi/4$ 】按钮,参数  $\varphi$  从 0 变化到 pi/4,对应的红色曲线也动态向左 平移了 pi/4 个单位,结果如图 5.2.10 所示。

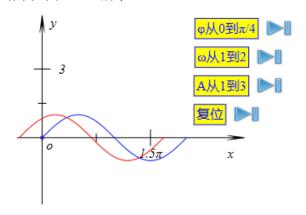


图 5.2.10

(2) 单击【 $\omega$  从 1 到 2】按钮,参数  $\omega$  从 1 变化到 2,对应的红色曲线中每一个点的横坐标动态缩短为原来的 1/2,纵坐标保持不变,结果如图 5.2.10 所示。

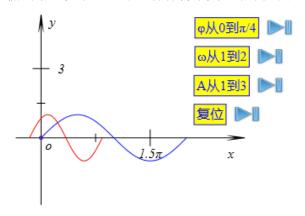


图 5.2.10

(3) 单击 【A 从 1 到 3】按钮,参数 A 从 1 变化到 3,对应的红色曲线中每一个点的纵坐标动态扩大为原来的 3 倍,横坐标保持不变,结果如图 5.2.11 所示。

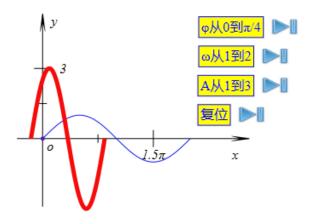


图 5.2.11

单击【动画: *A*[1,3]】按钮的中间部分,则参数 *A* 从 3 变化到 1,对应的红色曲线中每一个点的纵坐标缩小为原来的 1/3 倍,横坐标保持不变。与上面的过程(3)互为逆向的。

单击【动画:  $\omega[1,2]$ 】按钮的中间部分,则参数  $\omega$  从 2 变化到 1,对应的红色曲线中每一个点的横坐标伸长为原来的 2 倍,纵坐标保持不变。与上面的过程(2)互为逆向的。

单击【动画:  $\varphi[0,pi/4]$ 】按钮的中间部分,参数  $\varphi$  从 pi/4 变化到 0,对应的红色曲线整体向右平移了 pi/4 个单位。与上面的过程(1)互为逆向的。

在曲线变换过程中,可以通过按钮上方的文本观察到红色曲线的动态方程表达式。

将曲线  $y=Asin(\omega x+\varphi)$ 变量 x 的范围设置为:  $-\varphi/\omega$  到(2\* $pi-\varphi$ )/ $\omega$ ,是为了使得作出的图像 始终在一个周期内变动,可以要求  $\omega x+\varphi$  的范围为: 0 到 2\*pi,容易解得变量 x 的范围为:  $-\varphi/\omega$  到(2\* $pi-\varphi$ )/ $\omega$ ,这样无论  $\varphi$  和  $\omega$  怎样变化,都显示变量  $\omega x+\varphi$  从 0 到 2\*pi 之间的图像。

在文本中,如果希望直接显示插入变量的值,则可以使用函数\$*bl*{,},它有两个参数用逗号","隔开。其中前一个参数是变量的名称,后一个参数是要显示变量的浮点数的位数。

例如上面的 $Sbl\{A,2\}$ ,就是要插入变量 A,显示其浮点数之后的 2 位小数。

### 【练习作业】

根据本节学习到的设计思路和操作步骤,请你制作从 y=sin(x)变换到 y=1/2sin(x/3-pi)的 实验。

# 5.3 由两边及一边对角解三角形

"已知两边和其中一边的对角解三角形"是解三角形一章内容的难点。问题的关键在于如何找到一种关系展开讨论,通过这种关系将抽象的问题具体化、形象化,并通过考虑这种关系的各种情形,找到解的各种情况。

例如,在这个问题中,已知两边的大小关系就是解决问题的一种途径。在讨论问题的过程中,画出对应的图形,能帮助学生加深对问题的理解。

而若利用计算机作图,则可以动态、连续地改变两条边的大小关系,通过图形直观地观察和理解各种情形下解的各种情况,对讨论不同情况下解的情况可以起到事半功倍的作用。

作出任意射线 AB:

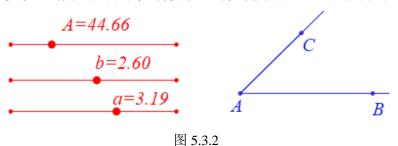
- (1) 作出两坐标点点: 点A (-2, 0) 和点B (2, 0)。
- (2) 依次选择点 A 和点 B,执行【画图】【直线】【射线】命令,作出射线 AB。作出以点 A 为顶点、与射线 AB 夹角为 A 度的射线:

- (3) 依次选择点 B 和点 A, 执行【变换】【旋转缩放点】命令。
- (4) 在弹出的用户输入对话框中输入: 旋转角度: *A*, 放缩倍数: 0.25\**b*, 单击【确定】按钮完成。
  - (5) 依次选择点 A 和点 C: 执行【画图】【直线】【射线】命令,作出射线 AC。插入参数 A 的变量控制对象:
- (6) 执行【插入】【变量】命令,如图 5.3.1 所示,在弹出的用户输入对话框中,输入参数 A,设置范围为: 0 到 180,单击【确定】按钮完成。同理插入变量 a、b 控制条,范围分别设置为 0 到 5。



图 5.3.1

(7)通过参数 A 的变量控制对象,将参数 A 的值改变到 60 左右。结果如图 5.3.2 所示。



作出以点 C 为中心半径为 a 的圆:

(8)选择点 C,单击【已知圆心和半径的圆】命令,在弹出的用户输入对话框中输入:a,单击【确定】按钮完成。

作出点 C 到射线 AB 的垂线段:

(9) 依次选择点 C 和射线 AB,执行【画图】【约束点】【垂足】,作出点 C 到射线 AB 的垂线段 CD,垂足为 D。结果如图 5.3.3 所示。

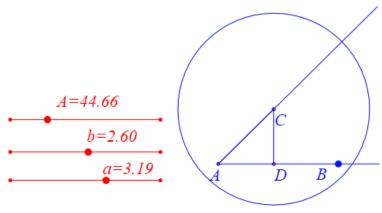


图 5.3.3

增加参数 a 的动画按钮:

- (14) 选择线段 CD,单击菜单命令【测量】【距离】,得到线段 CD 长度的测量值,系统把该值储存在变量 v000 中。
- (15) 执行【插入】【常用按钮】【变量一次运动】命令,如图 5.3.4 所示,在弹出的用户输入对话框中:标题改为:动画: *a*[*a*,*v*000]。程序命令改为: *VarAnimation*(*a*,*a*,*v*000,50,3); 单击【修改动作】和单击【确定】完成插入按钮的操作。

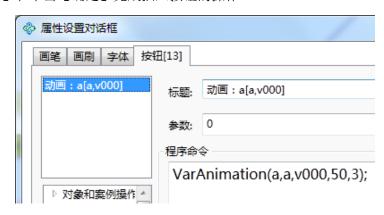


图 5.3.4

(15) 执行【插入】【常用按钮】【变量一次运动】命令,如图 5.3.5 所示,在弹出的用户输入对话框中:标题改为:动画: *a*[*a*,*b*]。程序命令改为: *VarAnimation*(*a*,*a*,*b*,50,3); 单击【修改动作】和单击【确定】完成插入按钮的操作。完成后如图 5.3.5 所示:

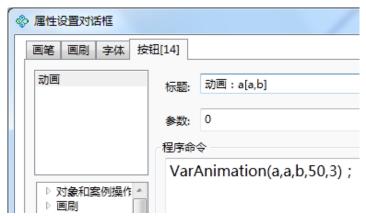


图 5.3.5

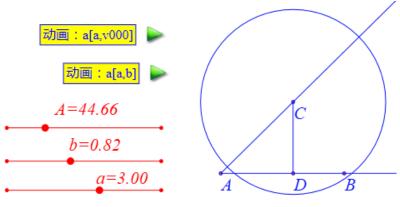


图 5.3.5

修饰课件:

- (17) 隐藏射线 AC; 连接线段 AC。
- (18) 隐藏线段 CD 的长度测量文本,隐藏点 B,将线段 CD 设置为虚线。

结果如图 5.3.7 所示,线段 AC 的长度为 b,圆 C 的半径为 a,虚线段 CD 为点 C 到射线 AD 的距离。

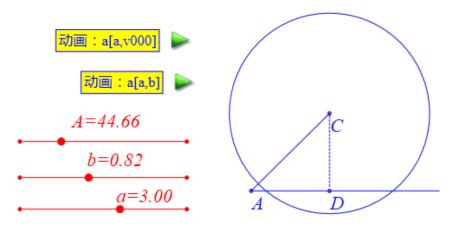


图 5.3.7

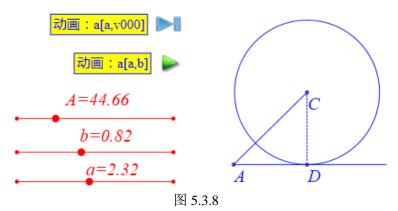
通过参数 A 的变量控制对象,可以改变角 A 的大小;通过参数 a 的变量控制对象,可以改变圆 C 的半径大小;通过参数 b 的变量控制对象,可以改变线段 AC 的长度。

对于角 A 为锐角的情况,可以从以下几个方面考察研究:

(1) 容易观察到当圆 C 的半径 a 大于线段 CD 的长度而且小于线段 AC 的长度时,即 b\*sin(A)<a<b,圆 C 与经过点 A 的水平射线有两个交点。这时角 B 有两个解,继续探讨或者在图形上标示出来之后容易知道这两个解为互补的两个角。

点 B 的位置就是圆 C 与射线 AD 的交点处(以下同)。

(2) 单击按钮【动画: a[a,v000]】,如图 5.3.8 所示,使得圆 C 与射线 AD 相切。此时圆 C 的半径等于线段 CD 的长度,即 a=b\*sin(A),这时角 B 有一个唯一解: 90°。



单击按钮【动画: a[a,b]】,如图 5.3.9 所示,使得圆 C 经过点 A,此时圆 C 的半径等于 线段 AC 的长度,即 a=b,这时角 B 有一个唯一的解: B=A。

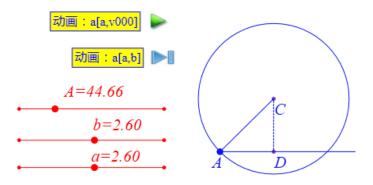


图 5.3.9

通过参数 a 的变量控制对象,如图 5.3.10 所示,使得圆 C 的半径大于线段 AC 的长度,即 a>b,这时角 B 有一个唯一的解,点 B 的位置在圆 C 与射线 AD 的交点处。

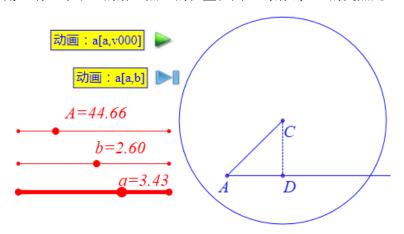


图 5.3.10

(3)通过参数 a 的变量控制对象,如图 5.3.11 所示,使得圆 C 的半径小于线段 CD 的长度,即 a < b \* sin(A),这时圆 C 与射线 AD 没有交点,即角 B 无解。

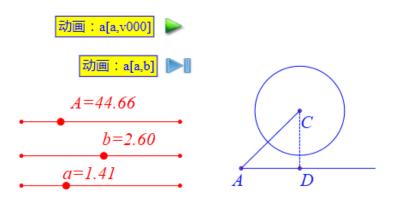


图 5.3.11

根据正弦定理知道: sin(B)=b\*sin(A)/a, 在这种情况下因为 a<b\*sin(A), 所以有 sin(B)>1, 显然此时角 B 无解。

角 A 为锐角时,通过以上的讨论,非常清楚了根据 a 与 b 和 A 的大小关系不同,对应 角 B 的解的情况。

当角 A 为直角或钝角时,解的情况较为简单,可以让学生利用该课件自主探索、发现和解决。

# 5.4 三角函数曲线的变换

(1) 在新建文档页面中,把【视图】菜单中的【系统坐标系】勾选,使系统坐标系可 见,右击坐标系,打开坐标系的属性对话框,如图 5.4.1 所示,勾选:"刻度"、"网格"和"x 轴以π为单位"。



图 5.4.1

(2) 执行【画图】【一般曲线】【三角函数】命令,打开函数作图属性对话框,如图 5.4.2 所示,在"y="对应的编辑框中输入: sin(x),设置变量的范围: 0 到 2\*pi。单击【确定】按钮 得到函数图像。

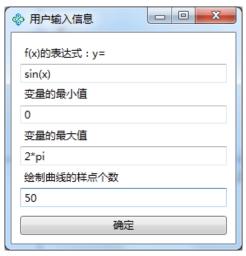


图 5.4.2

(3) 右击生成的图像,在【画笔】属性页面中,设置画线线型为:虚线,单击【确定】 按钮,结果如图 5.4.3 所示。

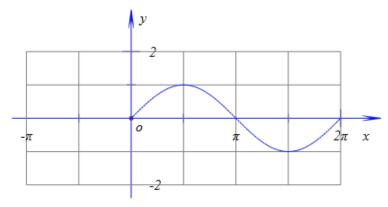


图 5.4.3

(4) 执行【画图】【一般曲线】【三角函数】命令,打开函数作图对话框,如图 5.4.4

所示,作函数 y=a\*sin(b\*x+c)的图像,设置"曲线的点数"为: 500,设置参数范围为: -c/b 到 (2\*pi-c)/b,单击【确定】按钮得到函数图像;右击作出的曲线,在【画笔】属性页面中,设置画线宽度为: 2,设置画线颜色为: 红色,单击【确定】按钮完成。

\$	) 用户输入信息 □ □ X
	f(x)的表达式: y=
	a*sin(b*x+c)
	变量的最小值
	-c/b
	变量的最大值
	(2*pi-c)/b
	绘制曲线的样点个数
	50
(	确定

图 5.4.4

(5) 执行【插入】【变量】命令,输入变量名: a,点击确认得到变量 a 的控制条,同理作出变量 b 和变量 c 的控制条。

分别拖动点 a、点 b、点 c,控制条上的控制点,观察参数 a、b、c 对曲线的影响。下面增加一个动态显示曲线的表达式的文本,操作如下:

(7) 执行【测量】【字符串】命令,在什么都不输人的情况下点击【确定】。右击得到的结果。打开编辑对话框,如图 5.4.5 所示,输入:

y=&MeasureValue(a,)sin(&MeasureValue(b,)x+&MeasureValue(c,)), 适当地调整【画笔】和【画刷】选项卡中颜色,单击【确定】按钮完成。结果如图 5.4.5 所示。



图 5.4.5

$$y = -1.80 \sin(-4.25 x + 3.84)$$

图 5.4.5

下面展示将 y=sin(x)动态变换为 y=2\*sin(x/3+pi)的过程,操作如下:

- (8)增加点 a 运动的动画按钮,设置参数范围为: 1 到 2,选择运动类型为:一次运动。
- (9) 增加点 b 运动的动画按钮,设置参数范围为: 1 到 1/3,选择运动类型为: 一次运动。
- (10)增加点 c 运动的动画按钮,设置参数范围为: 0 到 pi,选择运动类型为: 一次运动。

将曲线 y=a\*sin(b\*x+c)的初始化为曲线 y=sin(x), 操作如下:

单击点 a 的动画按钮的附按钮(按钮的中间部分),变量 a 从 2 变化到 1;单击点 b 的 动画按钮的附按钮,变量 b 从 1/3 变化到 1;单击点 c 的动画按钮的附按钮,变量 c 从 pi 变化到 0。

将曲线 y=a\*sin(b\*x+c)由 y=sin(x)逐步转换为 y=2\*sin(x/3+pi),操作如下: 单击点 a 的动画按钮,变量 a 从 1 变化到 2 ,结果如:

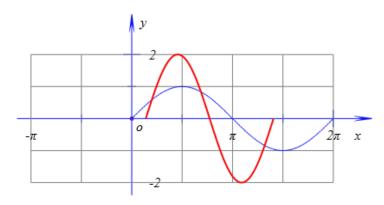


图 5.4.7

单击点 b 的动画按钮,变量 b 从 1/3 变化到 1,结果如图 5.4.8 所示:

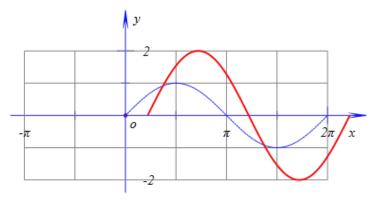
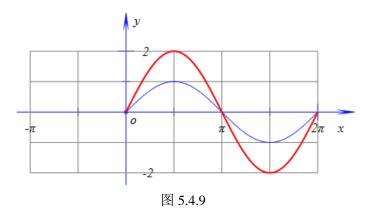


图 5.4.8

单击点 c 的动画按钮,变量 c 从 pi 变化到 0,结果如图 5.4.9 所示:



【思考与练习】

动态展示将  $y=x^2$  变换为  $y=2*(x-3)^2-1$  的过程。

### 5.5 正弦波的叠加

f(x)=sin(x)的图像是正弦波曲线,最小正周期是  $2\pi$ 。

f(x)=sin(x)+sin(2x)/2 的图像是什么形状?它的最小正周期是多少?我们可以计算一下,也可以利用计算机作出图形后观察一下。

f(x)=sin(x)+sin(2x)/2+....+sin(nx)/n 的图像呢?若在这里 n 分别取 10、20、30、40、50...。他们图像都会是什么形状的呢?如果 <math>n 不断依次递增,曲线会会如何变化?

利用传统教学的手段画这些函数图像,是异常困难的。

若利用计算机上,直接根据函数方程的表达式画对应函数曲线,也会过于繁杂和辛苦,且不能观察当 n 依次递增时,曲线的变化规律。

这样,我们的要求就变成了这样一件事:希望有一个参数 (例如 n),曲线的表达式 f(x)=sin(x)+sin(2x)/2+....+sin(nx)/n+...总是取多项式的前 n 项之和。例如当 n=3 时, f(x)=sin(x)+sin(2x)/2+sin(3x)/3;当 n=5 时, f(x)=sin(x)+sin(2x)/2+sin(3x)/3+sin(4x)/4+sin(5x)/5。

利用 Hawgent 动态数学软件中的符号函数功能,我们可以将方程的表达式写成:

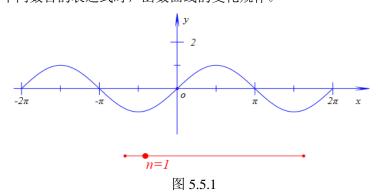
f(x) = sign(n,1) \* sin(x) + sign(n,2) \* sin(2x)/2 + sign(n,3) \* sin(3x)/3 + sign(n,4) \* sin(4x)/4 + sign(n,5) \* sin(5x)/5 + sign(n,5) \* sin(5x)/5 + sign(n,7) \* sin(7x)/7 + ... + sign(n,k) \* sin(kx)/k ...

具体操作步骤如下:

(1) 执行【画图】【一般曲线】【三角函数: y=f(x)】命令,所示在"y="对应的编辑框中输入:

sign(n,1)\*sin(x) + sign(n,2)\*sin(2\*x)/2 + sign(n,3)\*sin(3\*x)/3 + sign(n,4)\*sin(4\*x)/4 + sign(n,5)\*sin(5\*x)/5 + sign(n,5)\*sin(5\*x)/5 + sign(n,7)\*sin(7\*x)/7 + sign(n,8)\*sin(8\*x)/8 + sign(n,9)\*sin(9\*x)/9 + sign(n,10)\*sin(10\*x)/10

- (2) 将"曲线样点个数"设置为: 500, 设置变量 x 的范围为: -2\*pi 到 2\*pi, 单击【确定】按钮退出。
- (3) 单击【插入】菜单中的【变量】命令,在弹出的对话框中输入变量 n,设置 n 的范围为 0 到 10,单击【确定】按钮退出。
- (4) 如图图 5.5.1 和 5.5.2 所示,拖动参数 n 的控制对象,改变参数 n 的值,可以观察 到当函数叠加不同数目的表达式时,函数曲线的变化规律。



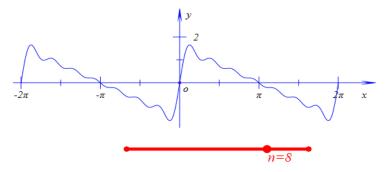


图 5.5.2

对于这种方法,若要观察 n 对应更大的数值对应的函数曲线时,需要输入更长的表达式。例如当 n 取 50,则需要输入 f(x)的前 50 项之和,这时表达式将会变得非常之长。若要求 n 取 100,将会花费大量的时间来书写这个表达式。

有没有一个更好地解决办法呢?让计算机代替我们书写这个冗长的表达式。

利用循环或者迭代,是解决问题的一种方法。

这样我们只需要定义一个函数,能自动计算表达式中的 f(x)=sin(x)+1/2 sin(2x)+....+1/n sin(nx)前 k 项之和即可。

具体操作如下:

- (5) 在新建文档中,选择右侧的程序工作区。
- (6) 在程序工作区中输入:

f=0:

for(i=1;greater(20,i);i=i+1)

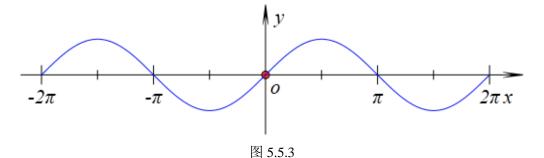
{

f=f+sign(k,i)\*sin(i\*x)/i;

**}**:

Curve2Dy(f,-2\*pi,2\*pi,100);

(7) 将光标放在最后一行的大括弧"}"之后,按住 F8 键,得到返回结果,向计算机定义了函数 f,并且作出了 f 在- $2\pi$  到  $2\pi$  的图像,如图 5.2.3 所示。



函数 Curve2Dy(f,-2\*pi,2\*pi,100)的作用是,作函数 f 对应的曲线,其中参数 x 的范围是从-2\*pi 到 2\*pi,曲线取 100个样点。

插入控制参数 k 变化的变量控制尺:

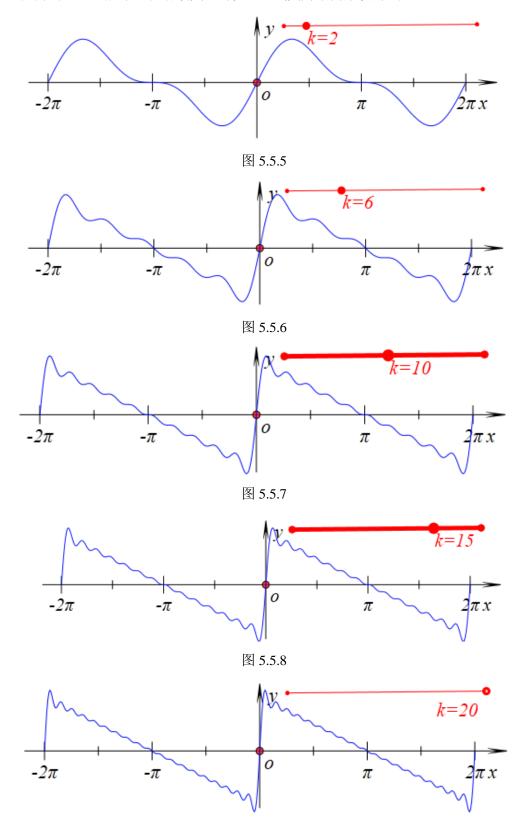
- (10) 只想【插入】菜单中的【变量】命令,在弹出的对话框中输入变量 k,设置 k 的范围为 1 到 20,单击【确定】按钮退出。
- (11) 右击变量控制尺子,打开属性窗口,如图 5.2.4 所示,在下方将"显示符点数的精度"的属性值修改为: 0,然后单击属性工作区中其他空白处,之后退出。

### 显示小数的位数

0

### 图 5.5.4

结果如图 5.2.5 所示,拖动改变 k 的值,曲线变为不同形状的"锯齿"。



#### 增加课件的标题:

(12)执行【测量】菜单中的【字符串】命令,在什么都不输入的情况下点击【确定】退出,接着右击生成的对象,如图 5.2.6 所示,在弹出窗口对话框中输入: f(x) = & Product((sin(i\*x))/i, i=1, & Measure Value(k,0)),单击【确定】按钮完成,结果如图 5.2.7 所示。

$$f(x) = \sum_{i=1}^{10} \frac{\sin(ix)}{i}$$

图 5.5.11

### 【思考与练习】

- (1) 如果将上面 f(x)中的通项 sin(k\*x)/k 改变为 sin((2k-1)\*x)/(2k-1),则结果变为什么样的形式呢。请你动手完成这个实验。
- (2)将上面 f(x)中的通项 sin(kx)/k 改变为 cos(kx)/k ,重新进行试验,看看叠加之后的 图形是什么形状的。
- (3)将上面 f(x)中的通项 sin(kx)/k 改变为 cos((2k-1)x)/(2k-1),重新进行试验,看看叠加之后的图形是什么形状的。
- (4) 将上面 f(x)中的通项 sin(kx)/k 改变为(-1) $^{\wedge}k*sin(kx)/k$ ,重新进行试验,看看叠加之后的图形是什么形状的。
- (5) 将上面 f(x)中的通项 sin(kx)/k 改变为其他三角形式,重新进行试验,观察叠加之后的图形性质。

# 第六部分 算法与编程

### 6.1 Hawgent 皓骏的编程环境

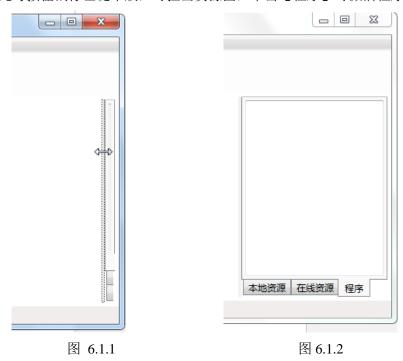
相对于很多计算机高级程序设计语言来说,利用皓骏动态数学平台的程序工作区执行程序具有无须编译直接执行命令并且即时得到运算结果的优点,也无须嵌入头文件、各种类文件即可进行对应的运算。

学习算法,关键在于编写出让计算机执行的对应程序。看到计算机执行自己的程序,并且快速准确地给出问题的解答,会让学习者体会到学习算法的成就感,另一方面,若通过计算机执行程序而并未得到自己预期的答案,则可以回过头来,对自己的算法或程序进行检查错漏,通过实践中的问题学习算法,可以加深对算法及程序的认识和理解。

另一方面,在平时研究的一些问题中,若能将算法和程序有机地融入其中,作为研究这些问题的途径和工具,很多时候可以达到仅通过菜单命令难以实现和完成的效果。通过在研究一些实际问题中对算法的应用和程序的学习,可以对算法理解得更深刻、对程序设计语言也运用得更熟练。

接下来,我们简要介绍一下皓骏动态数学软件中的程序设计语言。

如下图所示,当鼠标靠近皓骏动态数学平台的工作区右侧边界,其选择状态会变成左右 箭头状态,此时按住鼠标左键不放,可拉出资源区,单击【程序】可激活程序选项卡。



### (一) 赋值语句、表达式求值和运算

皓骏动态数学平台中的赋值语句和数学中常用的格式一样,用等号"="。例如要将 5 赋值给变量 a,操作是,输入:

a=5;

将鼠标的光标放在分号之后,按 F8 键,结果执行程序,在下面一行显示:

5 #

如下图所示。



图 6.1.3

### 解释一下:

- (1)按 F8 键,是皓骏动态数学平台程序工作区中执行程序的操作方法,以下简称为"执行命令"。而且执行命令前,需要将光标处于所执行程序最后一行的结尾。
- (2) 计算机执行的结果,从执行后的第一行开始阅读,到#表示结束。这是计算机对所 执行程序的回答,叫做"返回"。
- (3) 当计算机再次执行下面的程序时,从最后一个#后开始阅读,所以这个#是有作用的。这时,你可以继续输入:

a+3;

执行命令,结果为:

8#

这表示计算机已经知道 a 的当前值为 5。如下图所示:

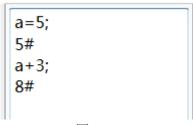


图 6.1.4

再例如计算  $a^2$ -3ab-2b<sup>2</sup> 的值,其中 a=3.45、b=-2.75,其操作步骤是:输入:

a=3.45;

b=-2.75;

a^2-3\*a\*b-2\*b^2;

执行命令后,结果为:

>> (631)/(25) #

如下图所示。

```
a=5;
5#
a+3;
8#
a=3.45;
b=-2.75;
a^2-3*a*b-2*b^2;
631/25#
```

图 6.1.5

执行命令时,计算机从" 8 #"之后开始执行程序,对 a 重新赋值为 3.45,对 b 赋值为-2.75,并且计算了  $a^2-3*a*b-2*b^2$  的值返回结果为(631)/(25)。

在这里"^2"表示平方,"\*"表示乘号,另外通过返回结果也可以看到"/"表示分数线或者除号。这是一般程序设计语言通用的符号,并且乘号不能省略。

### 【思考与练习】

(1) 按照下面的要求操作,并回答问题。

首先执行语句:

a=1;

再执行语句:

b=a+1;

然后再执行语句:

a=b;

然后重复执行后面的两个语句 5 次, 最后 a、b 的值是多少?

(2) 按照下面的要求操作,并回答问题。

首先执行语句:

m=1;

再执行语句:

m=m+1;

重复执行后面的语句 5 次,最后的返回结果是什么?

(3) 通过上面的操作, 你能理解 m=m+1 表示的意义吗?

### (二)按照需求定义函数

若希望计算当 a = -3.75、b = 4.60 时  $a^2$ -3ab-2 $b^2$  的值,则将上面的命令语句中,a 的值修改为: -3.75,b 的值修改为 4.60,然后将光标放至" $a^2$ -3\*a\*b-2\* $b^2$ -2;"之后,执行命令即可。结果如下图所示。

```
a=-3.75;
b=4.60;
a^2-3*a*b-2*b^2;
9397/400#
```

返回结果为: (9397)/(400)#。

它的好处在于,若要计算 a、b 对应其他值时表达式 a<sup>2</sup>-3ab-2b<sup>2</sup> 的结果时,只需要将前面对 a、b 赋值的语句修改一下就可以重新执行,而不必重新输入整个表达式。在所用公式比较复杂的情况下,这样先赋值再用公式计算的方式的优越性就比较明显了。

若希望再方便一些,可以将上面的语句定义成一个函数,而这就需要用到皓骏动态数 学平台的按钮。单击【插入】菜单的【按钮】,此时弹出按钮的属性对话框。如下图所示

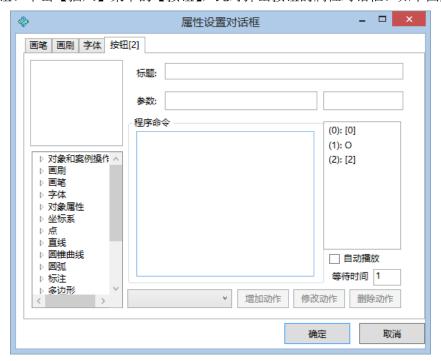


图 6.1.7

皓骏动态数学平台的按钮也是一个程序区,在里面定义的函数可以再工作区执行,也可以放至菜单执行。一个按钮可以定义很多函数,称为"动作"。每一个动作都必须有一个标题,而且都要求互不相同。"参数"一行有两个框,左边是"选择参数框",是确定"动作"执行的前提条件。例如,定义一个函数,用来构造两个点的线段。则"选择参数框"需要输入两个参数代表点,如若不然,则必须在"程序命令框"中指明哪两个点,具体操作步骤下面会详细介绍。右边是"输入参数框",是执行"动作"前需要用户输入的参数。一些函数的参数,有时候需要用户输入,这时候就需要"用户输入框"来指定哪些参数需要输入。按钮的属性对话框的左下角是"函数库",里面有系统的大部分内部函数。在右边则是工作区的对象列表。下面先初步了解使用按钮定义函数的步骤。

我们希望定义一个函数,使得每一次输入 a 与 b 两个数值,返回 a^2-3\*a\*b-2\*b^2 的数值。首先在"程序命令框"中输入

#### $s = a^2-3*a*b-2*b^2$ ;

这个命令表示计算式子 a^2-3\*a\*b-2\*b^2,并将结果赋值给 s。在皓骏动态数学平台的按钮中,所有命令执行完毕都必须以英文";"结束,这样计算机才能够区分出哪些是完整的一条命令。

为了将计算的结果在工作区中显示出来,需要将计算的结果"测量"。在"函数库"找到测量的函数,首先下拉滚动条,找到"文本和测量"栏目,点击左边的小三角,展开函数列表;找到函数 Measure Expression。如下图所示:



图 6.1.8

双击找到的函数,此时"程序命令框"中出现这个函数 MeasureExpression(,,);将括号中的逗号删除,并在括号中写入 s,如下图所示

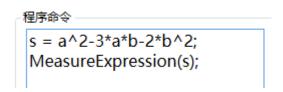


图 6.1.9

在"输入参数框"中输入: "a:请输入一个数值##b:请输入一个数值", 如下图所示



图 6.1.10

在"标题框"中输入"计算",为这个"动作"添加一个名字,此时按钮的属性对话框的"增加动作"按钮处于激活状态,单击该按钮,然后单击【确定】关闭属性对话框,如下图所示



图 6.1.11

单击"计算"按钮的绿色部分,弹出输入对话框,在第一个输入框中输入-3.75,在第二个输入框中输入 4.60,这就相当于 a=-3.75,b=4.60,单击【确定】,如下图所示



图 6.1.12



图 6.1.13

再次"计算"按钮的绿色部分,分别输入-5,7,结果如下图所示



图 6.1.14

若希望 a 与 b 有两个缺省的值,可以将"输入对话框"的命令修改为"a=1:请输入一个数值##b=2:请输入一个数值",此时每一次执行按钮,其输入框都有两个缺省值,这在某些情况为用户提供一个方便。



图 6.1.15

再例如编写一个有三角形的三边 a、b、c 计算其面积 s 的函数, 函数程序为:

t=(a+b+c)/2;

 $s=(t*(t-a)*(t-b)*(t-c))^{(1/2)};$ 

MeasureExpression(s);

输入参数为:

a:请输入三角形的第一条边的边长##b:请输入三角形的第二条边的边长##c:请输入三角形的第三条边的边长

标题为"计算三角形的周长",单击"增加动作"和【确定】。

若要计算三边长分别为 7、8、9 的三角形的面积,只需要单击按钮的绿色部分,分别输入 7.8.9,如下图所示



图 6.1.16

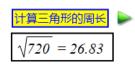


图 6.1.17

## 【思考与练习】

定义一个函数,根据给定的上低长、下底长和高度,计算梯形的面积。

## (三)条件判断语句

我们所熟悉的符号函数是:

$$sign(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases}$$

用自然语言表示为: 当x大于0时,函数值为1; 当x等于0时,函数值为0; 当x小于0时,函数值为-1。

在计算机中处理类似这种分情况讨论的问题,要用到条件判断语句。 在皓骏动态数学平台中,条件语句的一般格式是:

 $if(A){B}else{C};$ 

if(A){B}elseif(C){D}else{E};

第一个命令表示: 若条件 A 成立,则执行语句 B; 否则若条件 A 不成立,执行语句 C。

第二个命令表示: 若条件 A 成立,则执行语句 B; 否则若条件 A 不成立且条件 C 成立,执行语句 D, 否则执行语句 E。 elseif 的命令可以有多个。

在这里,条件 A 一般是一个逻辑运算表达式,例如大于(>)、小于(<)、等于(==)、不小于(>=)、不大于(<=)、不等于(!=)等; 花括弧内的语句 B 和 C,是可能被执行的一个语句或一组语句。

在皓骏动态数学平台中 a>b 对应的函数命令为 greater(a,b);a<b 对应的命令为 less(a,b);a≥b 对应的函数命令为 greaterEql(a,b);a≤b 对应的函数命令为 lessEql(a,b);a==b 对应的函数命令为 eql(a,b);另外的一些逻辑表达式如下所示:

and(x, y): x 或 y=0 时为 0, 否则为 1

or(x, y): x=y=0 时为 0, 否则为 1

xor(x, y): x=y 时为 0, 否则为 1

not(x): x=0 时为 1, 否则为 0

例如,编写一个从两个数 a,b 中返回一个较大的数的函数,程序内容如下:

if(greater(a,b))

{MeasureExpression(a);}

else{MeasureExpression(b);};

输入参数内容为:

a:请输入一个数值##b:请输入一个数值

"标题"可为"比较两个数的大小"

再例如,编写我们上面提到的符号函数的程序,由符号函数 sgn(x)的定义可得程序内容为:

if(eql(x,0))

{MeasureExpression(0);}

elseif(greater(x,0))

{MeasureExpression(1);}

else{MeasureExpression(-1);};

输入参数内容为:

x:请输入一个数值

"标题"可为"sgn(x)"

请你自己执行一些运算,观察其结果是否与预期的相符。

#### 【思考与练习】

- 1、请动手定义一个从三个实数 a、b、c 中求最大者的函数 Max(a,b,c),并利用定义的函数找出  $0.5^{\circ}(1/3)$  、2/3、 $0.72^{\circ}(2/3)$  中最大的一个数。
- 2、在按钮的"输入参数框"中输入"a=1:请输入一个数值##b=2:请输入一个数值",此时每一次执行按钮时 a 与 b 都有两个缺省值,这与在程序命令的开头中输入"a=1;b=2"有什么异同,哪一个在实际中更好用?

#### (四) 循环执行语句

一张普通的报纸被对折多少次之后,它的高度就超过了珠穆朗玛峰的高度?若一张普通报纸的厚度按照 0.5mm 计算,则折叠 n 次之后它的厚度为:  $2^n*0.5/1000$ ,单位 m。

我们可以将 n 分别取 1、2、3、4、5、...,让  $2^n*0.5/1000$  与 8848 比较。直到找到一个最小的正整数 N,使得  $2^n*0.5/1000>=8848$ 。

在计算机中,可以用循环语句处理这种重复操作的问题。

一种循环语句是 while 语句:

While(A) $\{B;\}$ 

它表示的意义和过程是:

判断条件 A 是否成立, 若条件 A 成立, 执行大括弧中的语句(被称为循环体);

继续判断条件 A 是否成立, 若条件 A 成立, 执行大括弧中的语句;

... ...

若条件 A 不成立,则退出循环。

对于上面的问题,我们可以编写如下的程序:

n=1:

while(less(2^n\*0.5/1000,8848))

 ${n=n+1;};$ 

MeasureExpression(n);

标题可为"计算对折次数"

其中 n=n+1 表示 n 加 1 得到的结果重新赋值给 n,即将 n 增加 1。执行按钮,结果得到 25,如下图所示:

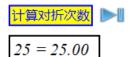


图 6.1.18

像这种由 while 引导的循环结构,可以事先不知道循环的次数。若是事先知道循环的次数,也可以使用由 for 引导的循环结构。例如要计算前 n 项自然数的倒数之和,可以编写如下的程序:

s=0;

for(i=1;lessEql(i,n);i=i+1)

 $\{s=s+1/i;\};$ 

MeasureExpression(s);

输入参数为:

n:请输入一个项数

标题可为"计算前 n 项自然数的倒数之和"

其中 for(i=1;i<=n;i=i+1)就是由 for 所引导的一个循环结构。在这里定义了一个临时的变量 i,并且将其初始值设定为 1;然后判断 i<=n 是否成立,若成立则执行循环体 $\{s=s+1/i;\}$ 一次;执行完循环体之后,执行 i=i+1 语句,将 i 增加 1。

然后,继续判断  $i \le n$  是否成立,若成立则执行循环体 $\{s = s + 1/i;\}$ 一次; 执行完循环体之后,执行 i = i + 1 语句,将 i 增加 1 。

... ..

然后,继续判断 i<=n 是否成立,若不成立则退出循环结构。

上面的循环结构式的循环执行了 n 次。

#### 【思考与练习】

- (1) 编写一个函数, 计算前 n 个正奇数的倒数之和。
- (2) 在皓骏动态数学平台中,由 while 引导的循环结构和 for 引导的循环结构是完全等价的。

请你利用 while 循环结构重新定义下面的程序:

s=0;

for(i=1;lessEql(i,n);i=i+1)

 $\{s=s+1/i;\};$ 

MeasureExpression(s);

并且利用 for 循环结构重新定义下面的语句:

n=1:

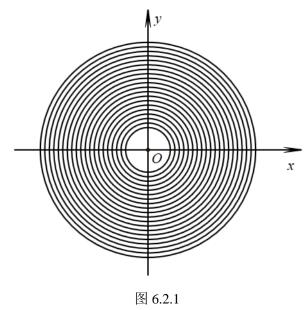
while  $(less(2^n*0.5/1000,8848))\{n=n+1;\};$ 

MeasureExpression(n);

# 6.2 了解系统中的函数

(一) 画一组同心圆

如下图所示,是一组以原点 O 为圆心的同心圆,其中最小的圆的半径为 0.5,相邻两圆的半径之差为 0.1。



假定这一组圆为 20 个,若通过【作图】菜单中【圆和圆弧】子菜单中的【已知圆心和半径的圆】命令作图,则需要将"选择原点 O、单击菜单命令、输入圆的半径"这一系列动作重复 20 次。

皓骏动态数学平台对使用者开放系统内部的作图函数命令,这样我们只要在按钮中输入 2 行程序,执行按钮后,瞬间即可作出上面的图形。

这两行程序的内容是:

```
for(i=0.5;less(i,2.5);i=i+0.1)
{
  CircleOfRadius2D(1,i);
}.
```

请你首先在计算机上运行这段程序,检验在作图区中生成的结果。

其中函数 CircleOfRadius2D(1,i,)就是系统内部已知圆心和半径画圆的函数:

CircleOfRadius2D(O,r,[,Text])

该函数有三个参数:圆心、半径和圆的名称。参数的属性分别:点、数值或参数表达式、 文本。被中括弧"[]"括起来的参数可以省略。

同一个文档中,不同的对象(点、线、圆、曲线等)可能有相同的名字,而它们的序号却是唯一的。所以当函数的参数为系统生成的对象时,需要使用该对象的序号。例如坐标原点 O 的序号就是 1。

如下图所示,点击【视图】菜单的【对象框】子菜单,打开对象工作区。在对象工作区中每个对象前的数字就是该对象的系统编号,即序号。另外,在作图区中选择一个对象,在软件窗口下方的状态栏中也会显示出该对象的序号。



图 6.2.2

在按钮的函数库中,列出了大量的系统函数,其中绝大部分是作图函数,如下图所示



图 6.2.3

向下拉动函数列表窗口的滚动条,上面使用的函数 CircleOfRadius2D(O,r, [Text])就在"圆锥曲线"栏目中。单击栏目前面的小三角,可以展开栏目中的函数。

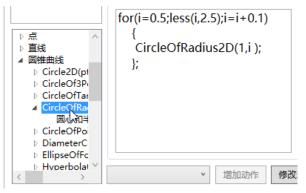


图 6.2.4

你可以打开函数列表查看一下,系统已经定义了哪些函数,这些函数具体有什么作用,它们对应的参数是什么。不需要去记忆,当使用的时候找到它们,然后双击,即可自动输入到程序工作区中。

## 【思考与练习】

画坐标点的函数是 CoordPoint(,,),函数的参数依次表示: x 坐标、y 坐标、点的名字。其中后面一个参数可以省略。

数列  $a_n$  的通项公式是:  $a_n=(3*n-2)/(2*n+1)$ 。 请你利用程序命令在直角坐标系中画出数列前 20 项的图像。

## (二) 线段构成的美丽图案

- (1) 在新建文档中, 画任意线段 AB、BC。
- (2) 在线段 AB 上任取一点 D、在线段 BC 上任取一点 E, 连接线段 DE。
- (3)依次选择点 D、点 E 和线段 DE,单击【画图】菜单中的【轨迹】命令,结果如下所示

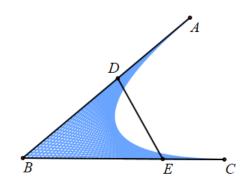
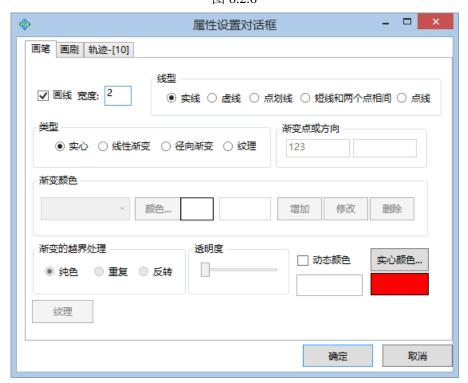


图 6.2.5

(4) 右击轨迹打开其属性设置对话框,将"样本点总数"修改为: 30,设置轨迹的画线 颜色为:红色,单击【确定】按钮完成。如下图所示:



图 6.2.6



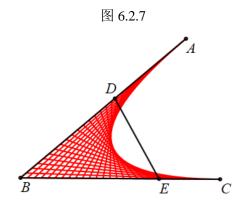


图 6.2.8

- (4) 将点 D、点 E 和线段 DE 隐藏即可。要作出上面的图形,也可以用程序实现:
- (5) 在新建文档中画出一个角 ABC。其中点 A、点 B、点 C 线段 AB 和线段 BC 的序号分别为 2、3、5、4、6。
  - (6) 在程序工作区中输入下面的程序:

for(i=1;less(i,30);i=i+1)

 $\{M=RatioPoint(2,4,i/30,);$ 

N=RatioPoint(3,6,i/30,);

S=Segment(M,N);

HideObj(M,N);;

(7) 执行命令后,结果如下图所示:

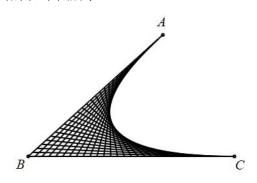


图 6.2.9

函数 RatioPoint(2,4,i/30,)的作用是作出以点 A 为起点,作与有向线段 AB 的比例为 i/30 并且平行于 AB 的线段的端点。

RatioPoint(,,,,)像其他函数命令一样,返回运行结果(一个点)的序号。

M=RatioPoint(3,6,i/30, )是将 RatioPoint(3,6,i/30, )返回的结果赋值给变量 M。 函数 Segment(M,N)的作用是以点 M 和点 N 为端点画一条线段。

HideObj(M,N)的作用是隐藏点 M 和点 N,函数 HideObj()的参数可以是多个。

#### 【思考与练习】

设计一段程序,画出下面的图案,其中每一条弦所对应的劣弧弧度是 2\*pi/3:

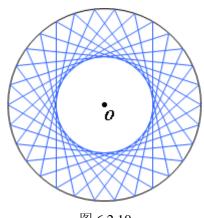


图 6.2.10

## (三) 检验角谷猜想

日本数学家角谷提出了一个猜想:若一个自然数是奇数则将它乘以三再加一,若它是偶数则将它除以二,这样反复进行运算下去,对于任何一个数运算的最后结果都是1。

下面我们编写一段程序,给出一些自然数,检验角谷猜想。

(1) 在按钮的程序命令区中输入下面的程序:

x=1;

while(greater(n,1))

 $\{x=x+1;$ 

if(eql(mod(n,2),0))

 $\{n=n/2;\}$ 

else $\{n=3*n+1;\};$ 

## MeasureExpression(n);};

- (2) 在输入参数框中输入: "n:请输入一个奇数"。将按钮的标题设为"角谷猜想"
- (3) 对 n=7 的情形进行检验。执行按钮,输入 7。
- (4) 执行命令后,在工作区中得到的每个步骤计算的结果,如下图所示:

图 6.2.11

在这里我们使用了函数 MeasureExpression(n)来观察计算的过程,MeasureExpression()的结果会按照先后顺序进行排列,这样便于我们查看每一个计算结果。

mod(n,2)的结果是 n 整除 2 所得到的余数。

## 6.3 求最大公约数的程序

## (一) 求最大公约数的更相减损术

我国古代的数学专著《九章算术》中提到了求两个数的最大公约数的方法--"更相减损术":

可半者半之;不可半者,副置分母、分子之数,以少减多,更相减损,求其等也,以等数约之。

用现代语言来表示就是:

若给定的两个数是偶数,则用 2 约简它们。否则,以较大的数减去较小的数,接着把所得的差与较小的数比较,并以大数减去小数;继续这个操作,直到所得的数相等,则这个数 (等数)就是所求的最大公约数。

利用计算机实现的算法思路是:

首先检查给定的两个数是否相等,若相等则返回这个数(等数);否则,

用大的数减去小的数,所得的差与小的数进行比较,并以大数减去小数;

重复这一过程...

直到所得的两个数相等时,退出循环过程。

对应的程序:

if(eql(a,b)){MeasureExpression(a);};

while(xor(a, b))

{if(greater(a,b)){a=a-b;}

else{b=b-a;};};

MeasureExpression(a);

输入参数为: "a:请输入一个数值##b:请输入一个数值"。

标题可为"更相减损法"。

语句 if(eql(a,b)){MeasureExpression(a);}的作用是判断 a==b 是否成立,若成立则测量参数 a 的值,并结束程序;若不成立,则跳转到后面的程序语句。

语句 while(xor(a, b))以及后面"{}"中的内容,是由 while 引导的一个循环结构。当 a 不等于 b 时,xor(a, b)返回 1,条件成立,此时执行后面的循环体,否则退出循环。

语句 a=a-b 就是将 a-b 的结果赋值给 a。类似地, b=b-a 就是将 b-a 的结果赋值给 b。

#### 【思考与练习】

(1)下面是一个分段函数,编程定义这个分段函数。对于任意给定的变量 x,返回对应的函数值。

$$y{=} \left\{ \begin{array}{l} 0 \text{ , } x{<}0 \\ \\ 1 \text{ , } 0{\le}x{<}1 \\ \\ x \text{ , } 1{\le}x \end{array} \right.$$

(2)下面是一个分段函数,编程定义这个分段函数。对于任意给定的变量 x,返回对应的函数值。

$$\mathbf{y} = \left\{ \begin{array}{cccc} (\mathbf{x} + \mathbf{2})^2 & , & \mathbf{x} < \mathbf{0} \\ \\ 4 & , & \mathbf{x} = \mathbf{0} \\ \\ (\mathbf{x} - \mathbf{2})^2 & , & \mathbf{x} > \mathbf{0} \end{array} \right.$$

#### (二) 求最大公约数的辗转相除法

因为999=185\*5+74,所以,

185 与 74 的最大公约数也是 999 与 185 的最大公约数。

因为 185=74\*2+37, 所以,

74 与 37 的最大公约数也是 185 与 74 的最大公约数。

因为 74=37\*2, 所以 74 与 37 的最大公约数是 37。

所以,999与185的最大公约数是37。

这就是另外一种求两个数最大公约数古老而有效的方法----辗转相除法。

辗转相除法的算法思想就是:

若求 a 与 b 的最大公约数(假定 a>b),则需要求 b 与 c 的最大公约数,其中 c 是 b 除 a 的余数。

若求b与c的最大公约数,则需要求c与d的最大公约数,其中d是c除b的余数。

....

若求j与k的的最大公约数,则需要求k与m的最大公约数,其中m是k除j的余数。若k能被m整除,则k与m的最大公约数就是m,并且返回m的值。

利用辗转相除法求两个数的最大公约数的对应程序为:

```
gcd(a,b)
{
    t=Mod(a,b);
if(t==0){return b;}
else{gcd(b,t);}
}
```

在这里,定义函数 gcd 的过程调用了它自身。这种由后向前递推的编程思想,叫作递归。

#### 【思考与练习】

意大利数学家斐波那契(Leonardo Fibonacci)在他出版的《算盘全书(Liber Abacci)》中提出了一个关于兔子繁殖的问题:

如果一对兔子每月能生 1 对小兔子(一雌一雄),而每 1 对小兔子在它出生后的第三个月里,又能生 1 对小兔子。假定在不发生死亡的情况下,有 1 对初生的小兔子开始,50 个月后会有多少对兔子?

- (1) 通过编写程序定义一个函数,对于任意给定的月数 n,能求出该月的兔子对数。
- (2) 求 50 个月后兔子的对数。

# 6.4 判断是否为闰年的程序

翻开 2008 的日历, 大家都会发现: 阳历 2 月份比往年的 28 天多了 1 天, 变成了 29 天, 这在历法上叫阳历闰年。为什么阳历有闰年呢?

我们知道,以春分点为参照点:地球绕太阳公转一周叫做一回归年,一回归年长 365 日 5 时 48 分 46 秒,现行的公历即阳历就是按回归年的长度制定的。为便于计算,一年取整数 365 天。365 天又分为 12 个月: 1、3、5、7、8、10、12 月为大月,每月 31 天;其余为小月,4、6、9、11 月每月 30 天,2 月为 28 天。这样,每年比回归年短 0.242199 日,四年

共短 0.968796 日,故每四年增加一日,这一年有 366 日,就是闰年,由此规定年份是 4 的 倍数的年份为闰年。但四年增加一日比四个回归年又多 0.0312037 日,400 年后将多 3.12037 日,故在 400 年中少设 3 个闰年,也就是在 400 年中只设 97 个闰年,这样公历年的平均长度与回归年就相近似了。因此再规定:年份是整百数的必须是 400 的倍数才是闰年,例如 1900年、2100年就不是闰年。

熟悉了上面的规定和简单的乘除法之后,任意给定一个年份不难在短时间内判断是否为 闰年。但这也不妨碍我们编写一段判断任意给定的年份 y 是否闰年的程序,体验一下计算机 的速度之快。

```
判断年份 v 是否为闰年,下面给出了两个函数,这两个函数均可以独立运行和使用。
Rn1(y)
{
if(Mod(y,4)>0)\{No;\}
else\{if(Mod(y,100)>0)\} Yes;}
   else{if(Mod(y,400)>0){No;}}
      else{Yes;}
          }
      }
}
函数 Rn1(y)的算法是:
对于任意给定的一个年份 y, 若它不能被 4 整除, 则输出结果 No; 否则,
若可以被 4 整除而又不能被 100 整除,则输出结果 Yes;否则,
若可以被 4 整除且能被 100 整除,但不能被 400 整除,则输出结果 No;否则,
若可以被 4 整除且能被 100 整除,同时能被 400 整除,则输出结果 Yes。
Rn2(y)
if(Mod(y,400) = =0){return Yes;}
if(Mod(y,100) = =0) \{ return No; \}
if(Mod(y,4)==0)\{return Yes;\}
   No:
}
函数 Rn2(y)的算法是:
对于任意给定的一个年份 y, 若它能被 400 整除, 则输出结果 Yes; 否则,
若不能被 400 整除, 而能被 100 整除, 则输出结果 No; 否则,
```

#### 【思考与练习】

- (1) 上面两个函数的程序各有什么特点?
- (2)编写程序定义一个函数 IfTriangle(a,b,c),对于任意给定的 3 个实数 a、b、c,判断它们能否构成三角形的三条边。

若不能被 100 ( 当然也不能被 400 整除 ), 而能被 4 整除, 则若输出结果 Yes; 否则,

若不能被 4 整除(当然也不能被 100 和 400 整除),则输出结果 No。

(3)继续阅读关于闰年的资料。

也许有人还一定要问:那为什么二月的天数反倒最少了呢?这里还有一段故事: 传说在公元前46年,罗马皇帝恺撒在修改历法时,规定每年为12个月,一、三、五、 七、十、十二月定为大月,每月 31 天;其他月份定为小月,每月 30 天。这样,大小各 6 个月,使人很容易记住,应用起来也很方便。但是照这样规定,一年就不是 365 天,而是 366 天了,因此得找出一个月扣去一天。扣哪个月合适呢?那个时候被判处死刑的犯人都在 二月份处死,所以人们都希望二月能快点过去。于是,就把二月扣去了一天。这样,二月就 剩下 29 天了。后来,有一个叫奥古斯特的人做了罗马皇帝。他发现恺撒是七月份生的,七月是大月,而他自己是八月份生的,八月却是小月。他为了显示自己和前一位皇帝有同样尊严,就蛮横地把八月也定为大月,改为 31 天。而八月多出的这一天仍然从二月份扣除,这样,二月只剩下 28 天了。只是每过 4 年,也就是闰年,二月才是 29 天。这就是二月份天数少的来历。

# 第七部分 概率与统计

# 7.1 随机得到一个 100 以内的自然数

- (1) 单击【测量】菜单中的【表达式...】命令,打开测量表达式对话框。
- (2) 如下图所示,在表达式编辑框中输入: sign(t)\*rand(1,100), sign()函数的定义为

$$sign(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \text{ 或者 } sign(x, y) = \begin{cases} 1, & x \ge y \\ 0, & x < y \end{cases}$$
,这里使用第一种用法,其目的仅仅是为

了让测量随机数的表达式与 t 有关, 当 t>1 时对测量的结果没有什么影响, 其作用在后面会作说明。单击【确定】按钮得到测量结果。如下图所示:



图 7.1.1

## sign(t) rand(1,100) = 15.59

图 7.1.2

(3) 单击【插入】菜单的【变量...】命令,输入变量 t,最小值为 0,最大值为 10,单击【确定】按钮得到变量尺,如下图所示:



图 7.1.3 sign(t) rand(1,100) = 44.14

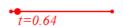


图 7.1.4

使用鼠标拖动变量尺,使得 t 的值发生改变,或者改变变量尺、测量文本的位置,可以观察到 sign(t)\*rand(1,100)的结果就会变化一次。改变变量尺的值,也可以选定变量尺之后,通过键盘中的方向键( $\rightarrow$ 、 $\leftarrow$ ),这样改变 t 的值的过程中,能更加清楚地辨别出数据每次的变化。

测量结果不断发生改变是因为,当t的值或者变量尺、测量文本本身的位置发生改变时,计算机会重新计算 sign(t)\*rand(1,100)的值,也就得到一个新的结果。同时,拖动其他对象,除非拖动的对象会影响 t 的值,否则 sign(t)\*rand(1,100)的测量结果都是不会发生改变。这避免了随机数的变化的过程中不受控制。

当然,sign(t)\*rand(1,100)的结果的变化过程会与t有关是因为表达式中含有t这个变量,如果表达式除了t之外,还含有其他变量,则改变这些变量的值,测量结果也会不断发生改变。

若我们希望得到的结果是一个自然数,那就通过 floor()函数对测量结果取整数部分。具体操作如下:

(4)选择测量文本,单击鼠标右键,打开文本属性设置对话框。其中\$\$##之上的部分是显示的格式符,v000 是测量变量,一般都对应测量文本的测量结果(可以手动修改为其他的测量变量,此时显示的结果与该测量文本测量的结果不相符)。测量文本后面可以输入一个整数值,表示测量结果显示的小数位数,不输入任何数值则默认为2,因此测量的结果默认都显示2位小数;\$\$##之下的部分就是我们之前在测量表达式对话框中输入的内容。

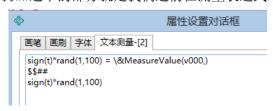


图 7.1.5

(5) 如下图所示,将 sign(t)\*rand(1,100)修改为 sign(t)\*floor(rand(1,100))。



(6) 单击【确定】按钮,结果如下图所示:

$$sign(t) \ rand(1,100) = 4.00$$

图 7.1.7

若希望以整数形式显示,则改变控制小数点后显示位数的整数即可。操作如下:

(7) 如下图所示,在 v000 的逗号后面输入 0。

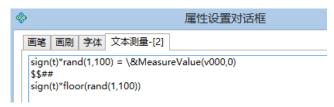


图 7.1.8

(8) 单击【确定】按钮,结果如下图所示:

$$sign(t) \ rand(1,100) = 3$$

图 7.1.9

#### 【思考与练习】

随机得到0到10000之间的一个完全平方数。

## 7.2 模拟随机抛硬币实验

## (一)参数变量的系统初始值和重新赋值

对于测量得到的第一个结果,系统会自动用变量 v000 表示。这样做的好处是便于后面利用这个测量结果参加更复杂的运算。就像我们习惯用 $\Delta$ 表示  $b^2$ -4ac,只要将  $ax^2$ +bx+c=0 的根表示为:

$$\frac{b\pm\sqrt{\Delta}}{2a}$$

然后第二个、第三个、第四个...测量结果分别用 v001、v002、v003 ...表示。

实际上对于每一个参数变量,例如 v000、v001、...,系统内部都有一个初始值,只不过我们在进行测量操作的过程中,将这些测量结果依次赋值给了变量 v000、v001、...。这就像前面在程序工作区中对一个参数变量赋值的操作一样:例如在程序工作区中输入"a=1;b=2;",然后执行命令。

为了验证这一点,你可以一个新建文档中,没有进行任何测量操作之前,通过【插入】菜单中的【变量...】插入参数变量 v000 的变量控制对象,如下图所示,可以观察它当前的系统初始值。



图 7.2.1

然后作一个任意点 A,通过【测量】菜单中的【x-坐标】命令,测量点 A 的 x 坐标,得

到测量文本之后拖动一下测量文本的位置,你会发现在参数 v000 的变量控制尺中对应的数值也对应改变。

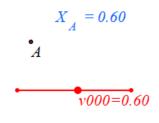


图 7.2.2

这个过程就类似于在程序工作区中对一个参数变量重新赋值。

## (二) 系统更新与执行命令

从前面可以看出,对随机数 rand(1,100)进行测量,在表达式中关联其他变量(比如 t),则改变 t 的值,随机数的结果都会发生改变。也就是说,t 的值每改变一次,系统内部就更新一次,也会对 rand(1,100)重新运算一次取一个新的结果。

在作图区中,执行一个动作,例如拖动一下坐标原点 O,系统内部也会自动更新,从而在屏幕上重新画出坐标系的图像。

下面我们通过测量得到 sign(t,0)\*rand(-1,1)的返回结果,操作如下:

(1) 打开测量表达式对话框,测量 sign(t,0)\*rand(-1,1)的值,并打开其属性对话框如下图所示:

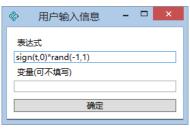


图 7.2.3

 $sign(t,0) \ rand(-1,1) = 0.06$ 

图 7.2.4



图 7.2.5

从测量文本系统把测量得到的第一个结果用变量 v000 表示。然后第二、第三...个测量结果分别用 v001、v002、...表示。

在程序工作区中我们可以通过执行一次语句命令"a=a+1;", 让 a 的值增加 1。同样,我们也可以在测量表达中使用类似这样的命令。系统会用 v001 记录下一个测量结果,而我们就将"v001+1"作为下一个测量表达式,操作如下:

(2) 在测量表达式对话框中输入: sign(t)\*(v001+1), 单击【确定】按钮,如下图所示,

得到测量结果。



图 7.2.6

## sign(t) (v001+1) = 3.00

图 7.2.7

(3) 单击【插入】菜单的【变量...】命令,输入变量 t,最小值为 0,最大值为 10,单击【确定】按钥得到变量尺。

拖动 sign(t)\*(v001+1)的测量文本,或者拖动变量尺改变 t 的值。你会发现它的测量结果不断地增加 1。这就像在程序工作区中不断执行命令"v001+1;",道理是一样的,当测量文本、 t 的变量尺位置发生改变,或者变量 t 的值发生改变时,计算机系统内部就需要更新,同时命令测量表达式"sign(t)\*(v001+1)"也重新运算一次,得到一个新的结果。这就是,拖动 sign(t)\*(v001+1)的测量文本,改变 t 的值或者改变 t 的变量尺位置时,测量结果会不断增加 1 的原因。

我们还可以增加一个动画按钮, 让 sign(t)\*(v001+1)的测量结果自动增加。操作如下:

(4) 单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",如下图所示:

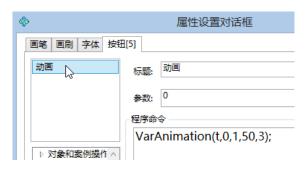


图 7.2.8

(5) 修改参数为 VarAnimation(t,0,10,99,3), 这个命令表示 t 从 0 变到 1, 共需要 99 个步骤(即运动频率为 99), 运动类型为正向一次运动(0 代表正向重复运动,1 代表负向重复运动,2 代表往返运动,3 代表正向一次运动,4 代表负向一次运动)。将按钮的标题修改为:实验 100 次,单击【确定】按钮。

将拖动变量尺使 t 为 0,此时两个测量结果都变为 0,然后单击【实验 100 次】按钮 "实验 100 次"。可以发现,sign(t)\*(v001+1)的结果改变 100 次,每次都是增加 1;sign(t)\*rand(-1,1)的结果也同时改变了 100 次。之所以结果的次数会比 t 的运动频率多一次,是因为单击按钮的时候系统会刷新一次,也就使得结果多了一次。

(6) 我们也可以再增加 t 的动画按钮来控制两个测量结果为 0。再次单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,修改参数为 VarAnimation(t,0,0,1,3),将按钮的标题修改为: 初始化。单击【修改动作】按钮,然后再单击【确定】按钮,如下图所示:

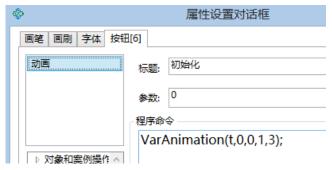


图 7.2.9

单击【初始化】按钮,结果如下图所示,sign(t)\*(v001+1)(下面我们用 n 表述)的测量结果自动变为 0。

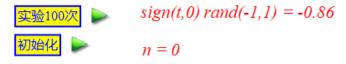


图 7.2.10

然后单击【实验 100 次】按钮,则 n的值从 0变化到 100。

通过【初始化】按钮使得 n 的值初始化为 0,是因为单击该按钮则 t 的值为 0,那么表达式 sign(t)\*(v001+1)的结果为 0。

而按钮【实验 100 次】使得 t 的值大于 0,那么 sign(t,0)就为 1,v000 的值就从 0 开始 递增。在该按钮中,设置的运动频率为 99,也就是 t 的值从 0 变化到 1 需要 99 步,再加上 单击该按钮后系统将 t 的值重新初始化为 0 的 1 个步,一共 100 步,所以 n 的值从 0 变化到 100。

## (三)模拟多次随机抛硬币实验

当 rand(-1,1)的结果即变量 v000 的值为正数时,可以认为所抛出的硬币正面朝上。下面 我们计算实验 100 次的过程中,计算正面朝上的次数,操作如下:

(1) 在测量表达式对话框中,在表达式输入框中输入: sign(t)\*(v002+sign(v000,0)),单击【确定】按钮,结果如下图所示:



图 7.2.11

## sign(t) (sign(v000,0)+v002) = 0.00

图 7.2.12

(2) 如下图所示,将 v002 的测量文本的标题修改为"k=",设置显示格式为整数,即在 v002 的逗号后面输入 0。



图 7.2.13

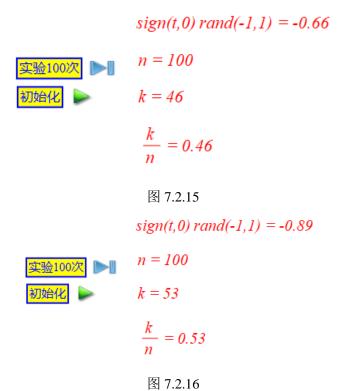
计算硬币正面朝上的次数,操作如下:

(3) 在测量表达式输入框中,测量 v002/v001 的值。结果如下:

$$sign(t,0) \ rand(-1,1) = 0.79$$

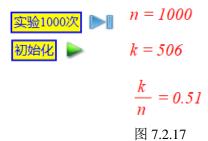
图 7.2.14

单击【实验 100 次】按钮一次,则进行 100 次实验,单击【初始化】按钮之后再次单击 该按钮则重新实验 100 次。下面是两组实验的结果:



在【实验 100 次】按钮的属性对话框中 t 运动的频率修改为: 999, 运动范围改为 0~1,则对应的按钮标题修改为: 实验 1000 次; 然后将 k/n 的测量结果中小数点显示的位数设置为: 3。则再次单击该按钮,每组实验的次数变为 1000。如下图所示:

## $sign(t,0) \ rand(-1,1) = 0.70$



在【实验 1000 次】按钮的属性对话框中,如下图所示,可以增加函数 AnimationInterval(0.001),这样每一次运行按钮的速度会大大加快。AnimationInterval()的参数是一个浮点数,其值越小,运动的速度越快。

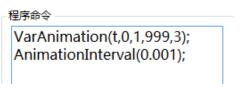


图 7.2.18

皓骏动态数学平台的运动按钮的频率有时候不一定等于实验结果的次数加 1, 这是一个小漏洞,有时候修改一下变量的运动范围就能解决。

## (四) 抛硬币实验的可视化显示

上面我们设定 rand(-1,1)的结果大于 0 时,所抛出的硬币正面朝上。现在我们作出对应的硬币图案模型:

(1) 首先准备两张没有白色背景色的 png 格式的硬币图片,如下图所示:



图 7.2.19

- (2) 在测量表达式输入框中,测量 sign(v000,0)的值,此时其对应的测量变量为 v004。
- (3) 单击【插入】菜单中的【图片...】命令,如下图所示,在弹出的属性对话框中,单击【选择文件】按钮,找到正面硬币的图片;单击【画笔】选项卡,去除"画线"的勾选,单击【确定】按钮完成。



图 7.2.20

(4)选择插入的图片,鼠标移至图片边缘,当鼠标变成上下箭头或者左右箭头时按住左键移动鼠标调整图片的大小,如下图所示:





图 7.2.21

另外,当 v000 不为正数时需要显示为"反"面,下面我们增加一张反面硬币并设置它的属性。其操作步骤同步骤(12)、(13)、(14),只是第(12)步骤的测量表达式为 1-sign(v000,0),其对应的测量变量为 v005。其最终效果如下图所示:

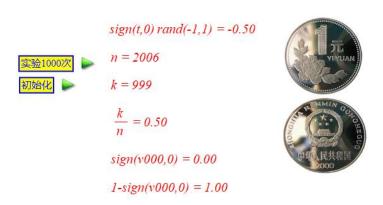


图 7.2.22

(5)将两张硬币图片移到同一个位置,并适当调整图片的大小,使得两张图片基本重合,如下图所示:

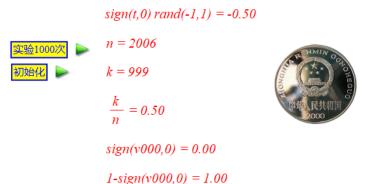


图 7.2.23

(6)单击【视图】菜单的【对象框】命令,打开对象列表,找到第一张图片(即正面硬币图片)并单击鼠标左键选中,然后右击打开其属性对话框。单击【画刷】选项卡,勾选"动态透明度"的选项框,在下面的右侧输入框中输入 v004; 对第二张图片(即反面硬币图片)执行相同的步骤,只是动态透明度输入框输入的是 v005。如下图所示:



最后将制作的内容进行修饰:

- (7)可将 sign(t)\*rand(-1,1)、sign(v000,0)、1-sign(v000,0)的测量文本以及变量尺 t 隐藏。
- (8) 将 v001 的测量文本的标题修改为: "实验的总次数:"; 将 v002 的测量文本的标题修改为: "出现正面的频数:"; 将 v002/v001 的测量文本的标题修改为: "出现正面的频率:"。如下图所示:



图 7.2.25

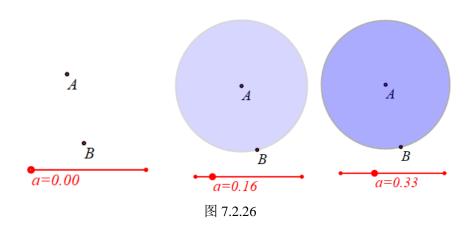
如果没有找到合适的硬币图片,可以单击【插入】菜单的【公式文本...】,输入"正"或"负"来代替相应的图片,其他步骤都是相同的。

#### (五) 理解动态透明度

上面我们通过巧妙地设置对象的动态透明度,实现了不同两个对象在不同情况下分别单独显示的方式。

下面我们通过一个例子理解对象的动态透明度。

- (1) 在新建文档中,任意画一个圆,将圆内部任意填充一种颜色,例如:蓝色。
- (2) 设置这个圆的动态透明度的参数为 a。
- (3)插入参数 a 的变量尺,设置其可改变范围为: 0 到 1。 通过参数 a 的变量尺改变参数 a 的值,可以观察对应圆的内部填充情况:



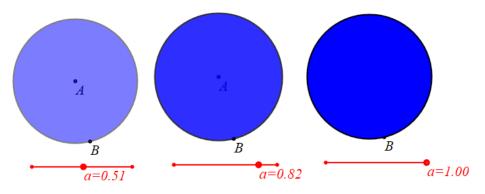


图 7.2.27

可见动态透明度的值越小,该对象的透明度越大,当动态透明度的值为 0 时该对象完全透明,当动态透明度的值为 255 时该对象完全不透明。

## 【思考与练习】

- (1)请制作一个直观展示模拟随机抛硬币的实验,每组实验为500次,记录出现反面的频数并计算出现反面的频率。
- (2)请制作一个直观展示模拟随机掷骰子的实验,每组实验为1200次,记录出现点数为1的频数并计算出现点数为1的频率。

## 7.3 掷骰子实验

(1) 在新建文档中,单击【测量】命令的【表达式…】,打开测量表达式输入框,输入 sign(t)\*floor(rand(1,7)),单击【确定】按钮得到测量结果,系统自动用 v000 记录该测量结果。如下图所示;

## sign(t) floor(rand(1,7)) = 4.00

#### 图 7.3.1

- (2) 打开测量表达式输入框,测量 sign(t)\*(v001+1)的值,系统自动用 v001 记录该测量结果。
- (3) 如下图所示,将变量 v000 对应的测量文本显示格式修改为"&MeasureValue(v000,0)",将变量 v001 对应的测量文本显示格式修改为"抛掷的总次数: &MeasureValue(v001,0)"。



图 7.3.2

由于在实数区间[1,7]上,取到右边界 7 的可能性为 0,所以 rand(1,7)的结果为 7 的可能性为 0。所以 rand(1,7)的结果为 7 的可能性为 0。所以 rand(1,7)的结果只可能是: rand(1,7)的结果只可能是: rand(1,7)的结果为 7 的可能性为 0。

对于出现各点数的情况可以表述为:

若出现的点数为 1,则有 v000<1.5;

若出现的点数为 2,则有 1.5<v000<2.5;

若出现的点数为 3,则有 2.5<v000<3.5;

若出现的点数为 4,则有 3.5<v000<4.5;

若出现的点数为 5,则有 4.5<v000<5.5;

若出现的点数为 6,则有 v000>5.5。

下面我们统计出现对应点数的频数。操作如下:

- (4) 测量 sign(t)\*(v002+sign(1.5,v000))的值,系统自动用 v002 记录该测量结果。
- (5) 测量 sign(t)\*(v003+sign(v000,1.5)\*sign(2.5,v000))的值,系统自动用 v003 记录该测量结果。
- (6) 测量 sign(t)\*(v004+sign(v000,2.5)\*sign(3.5,v000))的值,系统自动用 v004 记录该测量结果。
- (7) 测量 sign(t)\*(v005+sign(v000,3.5)\*sign(4.5,v000))的值,系统自动用 v005 记录该测量结果。
- (8) 测量 sign(t)\*(v006+sign(v000,4.5)\*sign(5.5,v000))的值,系统自动用 v006 记录该测量结果。
  - (9) 测量 sign(t)\*(v007+sign(v000,5.5))的值,系统自动用 v007 记录该测量结果。



图 7.3.3

下面计算出现对应点数的频率。操作如下:

- (13) 测量 v002/v001 的值,系统自动用 v008 记录该测量结果。
- (14) 测量 v003/v001 的值,系统自动用 v009 记录该测量结果。
- (15) 测量 v004/v001 的值,系统自动用 v010 记录该测量结果。
- (16) 测量 v005/v001 的值,系统自动用 v011 记录该测量结果。
- (17) 测量 v006/v001 的值,系统自动用 v012 记录该测量结果。
- (18) 测量 v007/v001 的值,系统自动用 v013 记录该测量结果。
- (19) 将变量 v008 对应的测量文本显示格式修改为"出现 1 点的频率:\&MeasureValue(v008,4)";将变量 v009 对应的测量文本显示格式修改为"出现 2 点的频率:\&MeasureValue(v009,4)";将变量 v010 对应的测量文本显示格式修改为"出现 3 点的频率:\&MeasureValue(v010,4)";将变量 v011 对应的测量文本显示格式修改为"出现 4 点的频率:\&MeasureValue(v011,4)";将变量 v012 对应的测量文本显示格式修改为"出现 5 点的频率:\&MeasureValue(v012,4)";将变量 v013 对应的测量文本显示格式修改为"出现 6 点的频率:\&MeasureValue(v013,4)"。

## 结果如下图所示:

抛掷的总次数:73 3

出现1点的频数: 31 出现1点的频率: 0.4247

出现2点的频数: 18 出现2点的频率: 0.2466

出现3点的频数: 15 出现3点的频率: 0.2055

出现4点的频数: 19 出现4点的频率: 0.2603

出现5点的频数: 143 出现5点的频率: 1.9589

出现6点的频数: 18 出现6点的频率: 0.2466

图 7.3.4

增加控制实验进程的动作按钮,操作如下:

(20)单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",如下图所示:

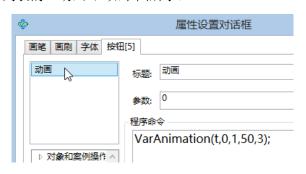


图 7.3.5

- (21) 修改参数为 VarAnimation(t,0,0,1,3),将按钮的标题修改为:初始化,单击【确定】按钮。
- (22)单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,在弹出属性对话框中,修改参数为 VarAnimation(t,0,1,1199,3),增加参数为 AnimationInterval(0.001)。将按钮的标题修改为:实验 1200次,单击【修改动作】按钮,然后再单击【确定】按钮。可以对文本的内容进行修饰,操作如下:
- (23) 将变量 v000 对应的测量文本放大,然后设置对应的字体颜色和填充颜色。将其他文本对象也对应设置字体颜色和填充颜色。
- (24)插入一些数学公式,内容分别为 1、2、3、4、5、6,设置其动态透明度以及字体颜色和填充颜色。

单击【实验 1200 次】按钮,实验完成后,结果如下:



出现1点的频数: 190 出现2点的频数: 196

出现3点的频数: 199

出现4点的频数: 211 出现5点的频数: 212

出现6点的频数: 192

出现1点的频率: 0.1583

出现2点的频率: 0.1633

出现3点的频率: 0.1658

出现4点的频率: 0.1758

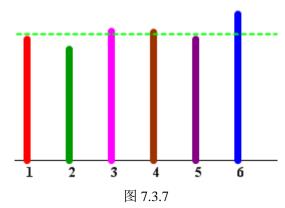
出现5点的频率: 0.1767

出现6点的频率: 0.1600

图 7.3.6

## 【思考与练习】

(1)如下图所示,绿色虚线与下面的黑色直线平行,且他它们之间的距离 1/6。数字上方的彩色柱子表示对应点数出现的频率大小,这种用直方图的形式表示统计数据更加直观。请你动手在上面的实验中,这种随着实验的进行会随时改变的"彩柱"直方图。



(2) 若同时投掷两枚骰子,则能出现的点数为: 2、3、4、5、6、7、8、9、10、11、12。请你设计一个模拟随机投掷两枚骰子的实验,并且利用"彩柱"直方图动态记录每个点数出现的频率。

# 7.4 利用抛豆估计 π

作一个半径为1的圆及其外切正方形,操作如下:

- (1) 单击【视图】菜单的【系统坐标系】,显示工作区的坐标系。
- (2) 单击【画图】菜单的【参数点】的【坐标点(参数)...】子菜单,分别作坐标点 *A* (-1, -1)、*B* (1, -1)、*C* (1, 1)、*D* (-1, 1),并且使用画笔工具 ✓ 连接线段 *AB*、*BC*、*CD*、*DA*。隐藏点 *A*、点 *B*、点 *C*、点 *D*。
- (4)在不选择任何对象的情况下,单击工具条中的【放大】工具<sup>≤</sup>、增加坐标系的单位长度。

作一个在正方形 ABCD 内随机出现的点,操作如下:

(5) 作坐标点 *E* (sign(t)\*rand(-1,1), sign(t)\*rand(-1,1))。

(6)选择点 *E*,单击【画图】菜单的【跟踪】子菜单跟踪点 E,选择跟踪对象并右击打开属性对话框,修改"跟踪最大数量 (1~2000)"为:2000。

插入 t 的变量尺,范围为 0~10,拖动变量尺 t,则点 E 会在正方形内随机出现,如下图 所示:

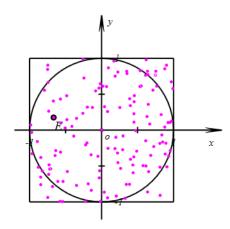


图 7.4.1

- (7) 为了不影响后续操作, 先隐藏跟踪对象。
- (8) 选择点 O 和点 E,单击【测量】菜单的【距离】子菜单测量点 E 到点 O 的距离,系统自动用 v000 记录该测量结果。

统计实验的总次数、点落在圆内的次数并计算圆周率的近似值。操作如下:

- (9) 单击【测量】菜单的【表达式…】子菜单打开测量表达输入框,测量表达式 sign(t)\*(v001+1),系统自动用 v001 记录该测量结果。
  - (10) 测量表达式 sign(t)\*(v002+sign(1,v000)),系统自动用 v002 记录该测量结果。
  - (11) 测量表达式 4\*v002/v001, 系统自动用 v003 记录该测量结果。
- (12) 将变量 v001 对应的测量文本显示格式修改为"抛出豆子的总数量:\&MeasureValue(v001,0)";将变量 v002 对应的测量文本显示格式修改为"落在圆内的数量:\&MeasureValue(v002,0)";将变量 v003 对应的测量文本显示格式修改为"圆周率的估计值:\&MeasureValue(v003,10)"。

增加控制实验进程的动作按钮,操作如下:

(13) 单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",修改参数为 VarAnimation(t,0,0,1,3),增加参数为 ClearAllTrace(),这个参数表示清除所有跟踪,将按钮的标题修改为:初始化,单击【修改动作】按钮,然后再单击【确定】按钮。如下图所示:

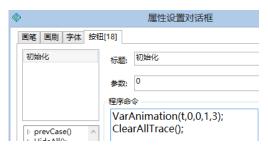


图 7.4.2

单击【插入】菜单的【常用按钮】的子菜单【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",修改参数为 VarAnimation(t,0,1,999,3),增加参数为AnimationInterval(0.001)。将按钮的标题修改为: 抛出 1000 粒豆,单击【修改动作】按钮,

然后再单击【确定】按钮。如下图所示:

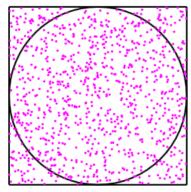


图 7.4.3

对实验界面进行修饰,操作如下:

(14) 隐藏坐标系;选择点 E,单击工具条中的【缩小】 工具,将点 E 的大小设置为: 1;隐藏点 O 与点 E 之间距离的测量文本、变量尺 t;隐藏点 E 的名字;设置对象的画线和填充颜色;打开对象列表,显示跟踪对象。

单击【抛出 1000 粒豆】按钮,实验完成后,结果如下:





抛出豆子的总数量: 1000

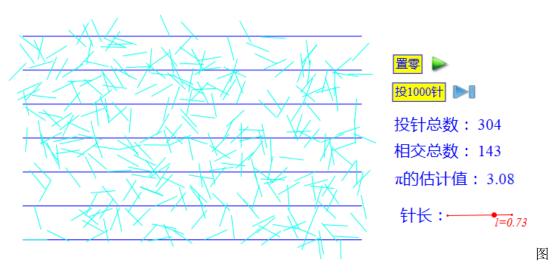
落在圆内的数量: 794

圆周率的估计值: 3.1760000000

图 7.4.5

## 【思考与练习】

先阅读下面的短文,然后再动手制作一个通过 Buffon 投针实验估计圆周率  $\pi$  的实验,要求平行线之间的距离为 d,针的长度 a 不大于 d 且可以改变。如下图所示:



7.4.6

公元 1777 年的一天, 法国科学家 D•布丰(D.Buffon 1707~1788)的家里宾客满堂, 原来他们是应主人的邀请前来观看一次奇特试验的。

试验开始,但见年已古稀的布丰先生兴致勃勃地拿出一张纸来,纸上预先画好了一条条等距离的平行线。接着他又抓出一大把原先准备好的小针,这些小针的长度都是平行线间距离的一半。然后布丰先生宣布:"请诸位把这些小针一根一根往纸上扔吧!不过,请大家务必把扔下的针是否与纸上的平行线相交告诉我。"

客人们不知布丰先生要玩什么把戏,只好客随主意,一个个加入了试验的行列。一把小针扔完了,把它捡起来又扔,而布丰先生本人则不停地在一旁数着、记着,如此这般地忙碌了将近一个钟头。最后,布丰先生高声宣布:"先生们,我这里记录了诸位刚才的投针结果, 共投针 2212 次,其中与平行线相交的 704 次。总数 2212 与相交数 704 的比值为 3.142。"说到这里,布丰先生故意停了停,并对大家报以神秘的一笑,接着有意提高声调说:"先生们,这就是圆周率π的近似值!"

众客哗然,一时疑议纷纷,大家全部感到莫名期妙:"圆周率  $\pi$ ? 这可是与圆半点也不沾边的呀!"

布丰先生似乎猜透了大家的心思,得意洋洋地解释道:"诸位,这里用的是概率的原理,如果大家有耐心的话,再增加投针的次数,还能得到 $\pi$ 的更精确的近似值。不过,要想弄清其间的道理,只好请大家去看敝人的新作了。"随着布丰先生扬了扬自己手上的一本《或然算术试验》的书。

π 在这种纷纭杂乱的场合出现,实在是出乎人们的意料,然而它却是千真万确的事实。由于投针试验的问题,是布丰先生最先提出的,所以数学史上就称它为布丰问题,布丰得出的一般结果是:如果纸上两平行线间相距为 d,小针长为 l,投针的次数为 n,所以投的针当中与平行线相交的次数的 m,那么当 n 相当大时有:π  $\approx$  (2ln)/(dm)

在上面故事中,针长 1 恰等于平行线间距离 d 的一半,所以代入上面公式简化得:  $\pi \approx n/m$  值得一提的是,后来有不少人步布丰先生的后尘,用同样的方法来计算  $\pi$  值。其中最为神奇的要算意大利数学家拉兹瑞尼(Lazzerini)。他在 1901 年宣称进行了多次的投针试验,每次投针数为 3408 次,平均相交数为 2169 次,代入布丰公式求得  $\pi \approx 3.1415929$ 。这与 $\pi$  的精确值相比,一直到小数点后第七位才出现不同!用如此轻巧的办法,求得如此高精度的 $\pi$  值,这真是天工造物!倘若祖冲之再世,也会为之惊讶得瞠目结舌!

不过,对于拉兹瑞尼的结果,人们一向非议甚多,究其原因,也不能说都没有道理,因 为在数学中可以证明,最接近 π 真值的,分母较小的几个分数是:

- (1)(22)/7≈3.14(疏率)
- $(2) (333)/(106) \approx 3.1415$
- (3) (355)/(113)≈3.1415929 (密率)
- $(4) (103993)/(33102) \approx 3.141592653$

而拉兹瑞尼居然投出了密率,对于万次之内的投掷,不可能有更好的结果了。难怪有不少人提出怀疑:"有这么巧吗?"但多数人鉴于拉兹瑞尼一生勤勉谨慎,认为他确实是"碰上了好运气"。事实究竟如何,现在也无从考查了!

现在也许你一定迫不及待地想知道布丰先生投针试验的原理, 其实这也没什么神秘, 下面就是一个简单而巧妙的证明。

找一根铁丝弯成一个圆圈,使其直径恰恰等于平行线间的距离 d。可以想象得到,对于这样的圆圈来说,不管怎么扔下,都将和平行线有两个交点。因此,如果圆圈扔下的次数为 n 次,那么相交的交点总数必为 2n。

现在设想把圆圈拉直,变成一条长为 $\pi$ d 的铁丝。显然,这样的铁丝扔下时与平行线相交的情形要比圆圈复杂些,可能有4个交点,3个交点,2个交点,1个交点,甚至于都不相交。

由于圆圈和直线的长度同为 πd, 根据机会均等的原理, 当它们投掷次数较多, 且相等

时,两者与平行线组交点的总数可望也是一样的。这就是说,当长为 $\pi$ d 的铁丝扔下n次时,与平行线相交的交点总数应大致为2n。

现在转而讨论铁丝长为1的情形。当投掷次数 n 增大的时候,这种铁丝跟平行线相交的交点总数 m 应当与长度1成正比,因而有: m=kl,式中 k 是比例系数。

为了求出 k 来, 只需注意到, 对于  $l=\pi d$  的特殊情形, 有 m=2n。于是求得  $k=(2n)/(\pi d)$ 。 代入前式就有:  $m\approx(2ln)/(\pi d)$ 从而  $\pi\approx(2ln)/(dm)$ 

这,就是著名的布丰公式!

利用布丰公式, 我们还可设计出求: 根 2, 根 3, 根 5 等数的近似值的投针试验呢! 这只需把 1/d 选得等于你那个数就行. 不过这时的  $\pi$  要当成知道的。

# 7.5 模拟转盘游戏

## (一) 落在一个区域内的随机实验

如下图所示,将圆盘 8 等分,指针绕着中心旋转,通过实验验证落在区域 2 内的可能性 大小。

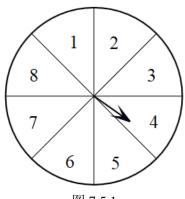
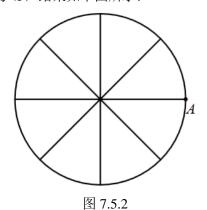


图 7.5.1

- (1) 单击【视图】菜单的【系统坐标系】,显示工作区的坐标系。以坐标原点 O 为圆心作半径为 4 的圆。
- (2) 单击【画图】菜单的【参数点】的【坐标点(参数)...】子菜单,作出圆 O 与 x 轴 正方向的交点 A (4,0),使用画笔工具连接线段 OA。单击【视图】的【对象框】打开对象 列表,去除第一个对象的勾选,隐藏坐标系的 x 轴和 y 轴,勾选第二个对象,保留显示坐标 原点 O。
- (3) 按住 Ctrl 键, 依次选择线段 OA、原点 O,单击【变换】菜单的【旋转】命令,设置旋转次数为 7,旋转角为 45,结果如下图所示:



169

- (4) 测量表达式 sign(t)\*rand(0,2\*pi),系统自动用参数 v000 记录测量结果。
- (5) 作坐标点 B(2\*cos(v000), 2\*sin(v000)), 按住 Ctrl 键, 依次选择原点 O, 点 B, 单击【画图】菜单的【直线】的【向量】子菜单作出向量 OB。
  - (6) 测量表达式 sign(t)\*(v001+1), 系统自动用参数 v001 记录计算结果。
- (7) 测量表达式 sign(t)\*(v002+sign(v000,pi/4)\*sign(pi/2,v000)), 系统自动用参数 v002 记录计算结果。
- (8)增加参数 t 的动画按钮:单击【插入】菜单的【常用按钮】子菜单下的【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",修改参数为 VarAnimation(t,0,0,1,3),将按钮的标题修改为:初始化,单击【修改动作】按钮,然后再单击【确定】按钮。
- (9) 增加参数 t 的动画按钮:单击【插入】菜单的【常用按钮】子菜单下的【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",修改参数为 VarAnimation(t,0,1,79,3),增加参数为 AnimationInterval(0.001)。将按钮的标题修改为:指针转动 80 次,单击【修改动作】按钮,然后再单击【确定】按钮。
- (10) 单击【插入】菜单的【公式文本...】命令,分别为每个区域增加文本标识;隐藏 参数 v000 对应的测量文本隐藏;隐藏点 A 和点 B。
- (11)将 v001 的测量文本的名称修改为: "指针转动总次数: \&MeasureValue(v001,0)"; 将 v002 的测量文本的名称修改为: "落在区域 2 内的次数: \&MeasureValue(v002,0)"。结果如下图所示:

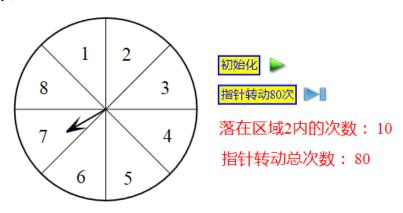


图 7.5.3

## 【思考与练习】

- (1) 如何统计落在区域 1 或区域 2 内的频率?
- (2)实际上,我们上面完成的内容是转动的指针,而不是转动的圆盘。请你动手制作一个让圆盘转动的实验。

#### (二) 落在红色区域内的转盘游戏

如下图所示,圆盘等分成 7 块,其中有三块红色区域,三块蓝色区域,一块黄色区域。 指针绕着中心旋转,通过实验研究指针落在红色区域内的可能性大小。

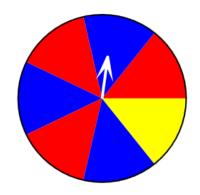
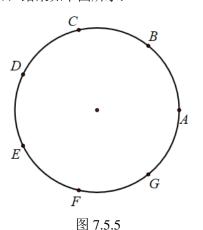


图 7.5.4

- (1) 新建一个文档,单击【视图】菜单的【系统坐标系】命令,显示工作区的坐标系。 以坐标原点 O 为圆心作半径为 4 的圆。
- (2) 单击【画图】菜单的【参数点】子菜单下的【坐标点(参数)...】命令,作出圆 O与 x 轴正方向的交点 A (4,0),使用画笔工具连接线段 OA。单击【视图】的【对象框】命令打开对象列表,去除第一个对象的勾选,隐藏坐标系的 x 轴和 y 轴,勾选第二个对象,保留显示坐标原点 O。
- (3) 按住 Ctrl 键,依次选择点 A、原点 O,单击【变换】菜单的【旋转】命令,设置旋转次数为 6,旋转角为 360/7,结果如下图所示:



(4) 依次选择圆 O、点 A 和点 B, 单击【作图】菜单下【圆和圆弧】子菜单中【圆周上的圆弧】命令,作出圆弧。选择圆弧并右击打开属性对话框,点击【画刷】激活画刷的选项卡,勾选填充,如下图所示:



图 7.5.6

(5) 单击【纯色颜色】按钮打开颜色管理器。如下图所示,在"ScR"输入框中修改参数为1。然后单击【确定】按钮关闭颜色管理器。点击【画笔】激活画笔选项卡,去除【画线】的勾选,单击【确定】按钮关闭圆弧的属性对话框,此时圆弧设置为无画线且红色填充。

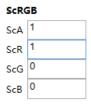


图 7.5.7

(6) 重复类似操作,作出圆弧 BC,填充为蓝色;重复类似操作,作出圆弧 CD,填充为红色;重复类似操作,作出圆弧 DE,填充为蓝色;重复类似操作,作出圆弧 EF,填充为红色;重复类似操作,作出圆弧 FG,填充为蓝色;重复类似操作,作出圆弧 GA,填充为黄色。单击【视图】的【对象框】命令打开对象列表,选择圆 O、原点 O,单击【编辑】菜单下的【对象管理】子菜单中【移动对象至最前面】,结果如下图所示:

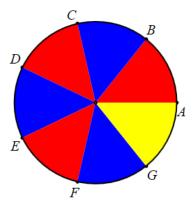


图 7.5.8

- (7) 测量表达式 sign(t)\*rand(0,2\*pi), 系统自动用参数 v000 记录测量结果。
- (8) 作坐标点 H(2\*cos(v000), 2\*sin(v000)), 作出向量 OH, 设置向量 OH 的画线和填充颜色为: 白色; 线宽为: 4。
  - (9) 测量表达式 sign(t)\*(v001+1), 系统自动用参数 v001 记录计算结果。
  - (10) 测量表达式:

sign(t)\*(v002+sign(2\*pi/7,v000)+sign(v000,4\*pi/7)\*sign(6\*pi/7,v000)+sign(v000,8\*pi/7)\*sign(10\*pi/7,v000)),系统自动用参数 v002 记录计算结果。

- (11)增加参数 t 的动画按钮:单击【插入】菜单的【常用按钮】子菜单下的【变量一次运动】命令,在弹出属性对话框中,单击动画列表的"动画",修改参数为 VarAnimation(t,0,0,1,3),将按钮的标题修改为:初始化,单击【修改动作】按钮,然后再单击【确定】按钮。
- (12)增加参数 t 的动画按钮:单击【插入】菜单的【常用按钮】子菜单下的【变量一次运动】命令,在弹出属性对话框中,单击动作列表的"动画",修改参数为 VarAnimation(t,0,1,69,3),增加参数为 AnimationInterval(0.001)。将按钮的标题修改为:指针转动70次,单击【修改动作】按钮,然后再单击【确定】按钮。
- (13) 隐藏参数 v000 对应的测量文本隐藏;设置点 H 的画线、填充颜色为白色,点的大小为 1;隐藏其余所有的点。
- (14)将 v001 的测量文本的名称修改为: "指针转动总次数: \&MeasureValue(v001,0)"; 将 v002 的测量文本的名称修改为: "落在红色区域内的次数: \&MeasureValue(v002,0)",。结果如下图所示:



图 7.5.9

# 【思考与练习】

请统计和计算不落在蓝色区域内的频率?

# 第八部分 专题介绍

# 8.1 变量尺

变量尺是经常使用到的功能,功能是对特定的变量进行数值上的控制,对应的函数是 *Variable*(,,)。

点击"插入"菜单的"变量"命令,可以打开变量尺的输入框(这里输入一个n,即生成n的变量尺):





图 8.1.1

变量尺其实本质上就是线段的点,但又有所不同,这也是为什么在使用轨迹时可以把变量尺当动点来看。

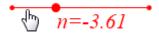


图 8.1.3

当把鼠标移到变量尺上时,鼠标变成一个手型,此时按住鼠标左键可以拖动变量尺,改变n的值。

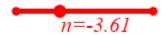


图 8.1.4

如果变量尺处于选择状态,此时鼠标移到变量尺上,鼠标会变成一个十字箭头。此时按住鼠标左键可以拖动变量尺,改变变量尺的位置。



图 8.1.5

变量尺处于选择状态时,如果鼠标移到变量尺的两端,鼠标会变成一个十字箭头。此时 按住鼠标左键可以拖动变量尺,改变变量尺的长度。



图 8.1.6

变量尺处于选择状态时,如果鼠标移到变量尺的文本位置,鼠标会变成一个十字箭头。 此时按住鼠标左键可以拖动变量尺文本的位置。



图 8.1.7

# 8.2 属性对话框

每一个对象都有相应的属性对话框,通过属性对话框可以设置对象的字体、颜色、大小、透明度等等。首先,先了解文档的属性对话框,在作图区的空白处右击,可以打开文档的属性对话框。

文档的"属性对话框"分别有"画笔"、"画刷"、"字体"和"其他"四个选项卡。其中, "画笔"和"画刷"是所有对象都有的内容。

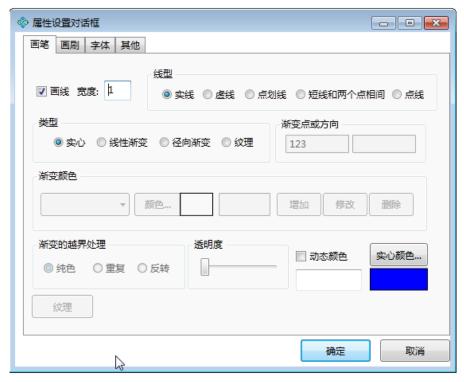


图 8.2.1

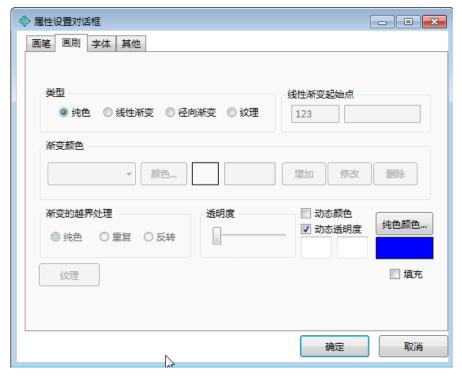


图 8.2.2

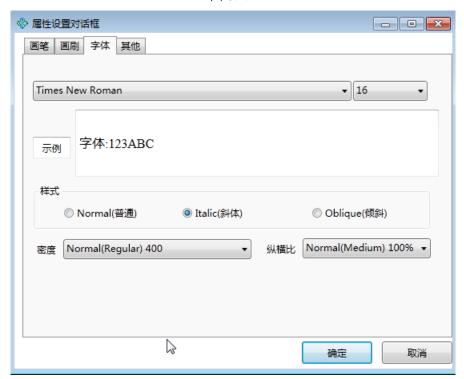


图 8.2.3

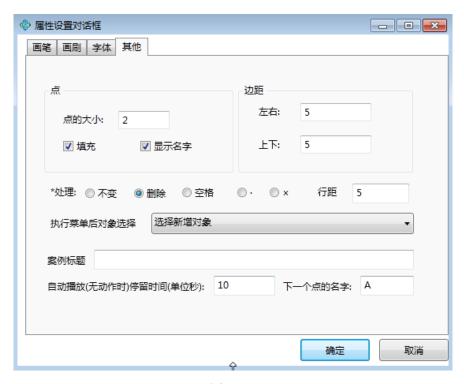


图 8.2.3

设置完文档的"属性对话框",则以后生成的对象都默认应用这些属性。例如,设置"其他"中点的大小为"2","填充"以及"显示名称",则以后生成点都是默认大小为2,填充颜色以及显示点的名称。当然,点的名称的属性需要在"字体"中进行设置。

后面我们主要了解对象的颜色设置以及动态颜色与动态透明度。

首先,使用"画笔"工具绘制一个圆 AB,选择圆并且右击,打开属性对话框。

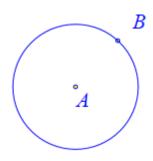


图 8.2.4

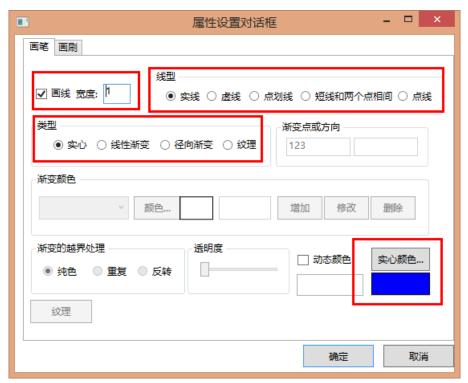


图 8.2.5

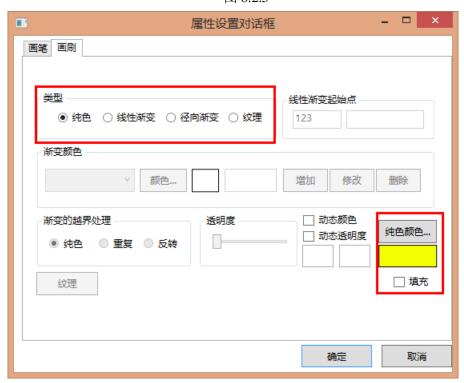


图 8.2.6

其中,"画笔"的属性设置是对圆的线条设置,而"画刷"是对填充颜色的设置。由于填充颜色默认是不填充的,所以这里没有具体看出来。将画刷的填充颜色的勾选框勾选起来,然后确定。

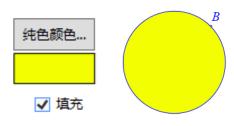


图 8.2.7

填充的颜色默认是纯色填充("画笔"里面是"实心"),也可以是其他类型,例如改为"径向渐变"。



图 8.2.8

修改两个渐变颜色为白色和黑色。 首先选择第一个颜色(中心颜色):



### 然后点击颜色:



图 8.2.10

### 选择白色, 然后确定:

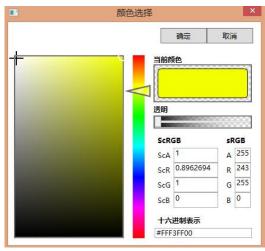


图 8.2.11

最后单击"修改":



图 8.2.12

第二个渐变颜色修改为黑色,步骤同上。 修改两个颜色之后,设置一下中心颜色的起始位置为 0.4,0.4。

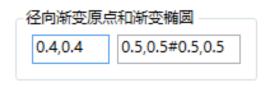


图 8.2.13

最后结果如下:

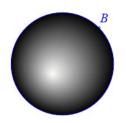


图 8.2.14

现在介绍一下如何设置对象的"动态透明度"和"动态颜色"。仍以圆 AB 为例,打开圆 AB 的属性对话框,查看"画笔"和"画刷"两个栏目,都发现有"动态颜色"的字样。而"画刷"还有一个"动态透明度"。

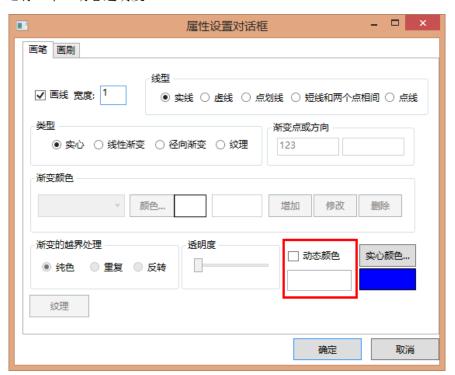


图 8.2.15

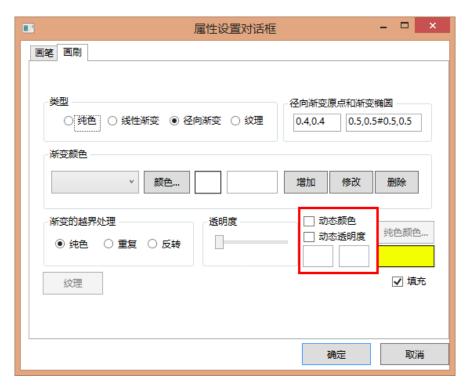


图 8.2.16

下面以一个例子说明动态透明度的用法。如下图,勾选"画刷"的"动态透明度",并在右边框中输入一个变量 t。



图 8.2.17

点击 "插入" 菜单的 "变量" 命令,输入 t,范围为-1~2。点击 "确定" 生成 t 的变量 尺。

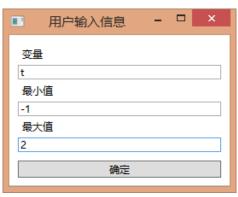
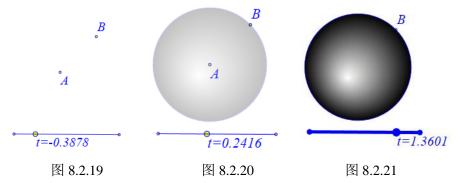


图 8.2.18

拖动 t, 可以发现当 t 的值小于等于 0 时,对象是透明的;当 t 大于等于 1 时对象不透明,而当 t 在 0~1 之间变动时,对象处于透明到不透明的渐变过程。

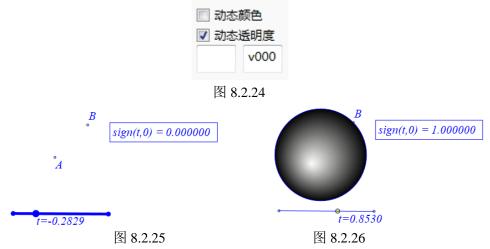


需要注意的是, 动态透明度输入的是一个变量, 不能是一个数值, 也不能是一个式子。如果想要输入一个式子, 可以先测量一下所要输入的式子, 再在动态透明度对应的框中输入测量变量名称即可。

例如,点击"测量"菜单下的"表达式",输入"sign(t,0)\*1",测量变量不输入,使用系统默认的变量名称。(注 sign(a,b)是系统的一个函数,当 a 大于等于 b 时,sign(a,b)=1,否则等于 0)



将"画刷"的t换为v000(注:默认测量变量具体视情况而定,后面会介绍到。想要查看填入哪个测量变量,可以右击生成的测量文本,打开属性对话框,在右下角可以看到。具体请参见《初识测量》一节)。就可以实现当t大于0时,对象不透明的效果。

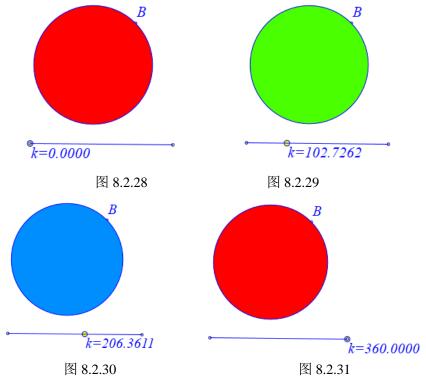


动态颜色与动态有明度的操作方法类似。注意画笔的"动态颜色"控制线条和文本的动态效果,而画刷的"动态颜色"控制填充颜色的动态效果。而且"动态颜色"只对纯色填充有效果。下面以画刷的"动态颜色"为例。如下图,勾选"画刷"的"动态颜色",在对应的左框中输入一个变量 k,并且将填充类似改为"纯色"。



图 8.2.27

同样添加 k 的变量尺, 范围是 0~360。拖动 k, 观察 k 的值与填充颜色的对应关系。



Hawgent 皓骏的动态颜色使用的是HLS的颜色空间,且只对H的动态,其范围是0~360。

# 8.3 案例的操作

单击菜单栏的"文件",可以看到四个关于"案例"的四个命令。



图 8.3.1

- "案例"指的就是课件的每一页。Hawgent 皓骏的每一个案例有相应的"页码",起始页就是第0页,后面接着就是第1页,以此类推。
  - "新建案例": 通过该功能可以新建一个按能力,对应的函数是 newCase()。
- "删除案例": 删除某一个不需要的案例,这时候 Hawgent 皓骏自动跳转至下一案例。 对应的函数是 *deleteCase()*。
- (注: 当删除的案例是最后一个案例的时候,由于不能跳转至下一个案例,因此 Hawgent 皓骏并没发生什么变化,此时需要手动单击"激活上一案例"或者单击工具栏的┛。)
  - "保存案例":保存需要重复用到的案例,对应的函数是 saveCase()。
  - (注:保存的案例的后缀是 sec,与课件不是一回事。)
  - "加载案例": 加载已经保存的案例,对应的函数是 loadCase();。

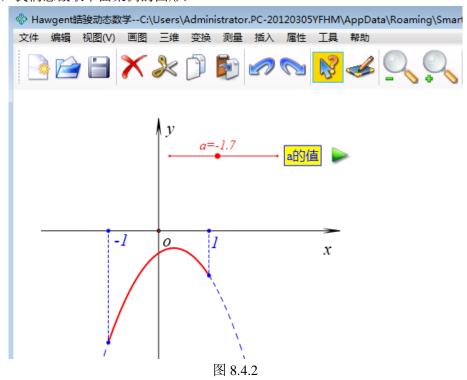
## 8.4 截图工具

对于一个绘制好的图形,如果我们想要把图像保存到 word 里面,可以单击"编辑"菜单栏"的"复制图形"(注:对应的函数是 copyBmp()):

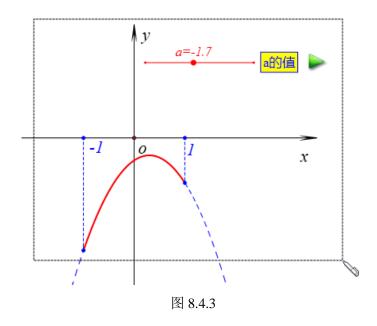


图 8.4.1

例如,我们想截取下面案例的图形:



单击"编辑"菜单栏"的"复制图形",此时出现一只画笔。使用画笔选定一个区域,然后 word 里面按粘贴的快捷键 Ctrl+V 即可:



### 8.5 对象间的关联

在"编辑"菜单下有一个"对象关联"子菜单。需要注意的是选择对象的顺序要与菜单的顺序相同。例如,"文本和点关联"就必须先选择文本,再选择点,这样才能激活菜单。

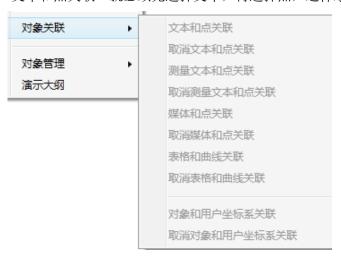


图 8.5.1

"文本和点关联": 选中一个"数学文本"(或者"普通文本")以及一个点,激活该命令并使用,可以使得点与文本进行关联。对应的函数是 *TextAssoc*(,)。

"测量文本和点关联": 选中一个"测量文本"以及一个点,激活该命令并使用,可以使得点与测量文本进行关联。对应的函数是 *TextAssoc*(,)。

(注:"测量文本"是指利用测量功能形成的文本,特点是文本基本都有一个测量变量 ("字符串"除外)。)

"媒体和点关联":选中一个"媒体"以及一个点,激活该命令并使用,可以使得点与媒体进行关联。对应的函数是 *MediaAssoc*(,)。

(注: Hawgent 皓骏的"媒体"是指图片、音频、视频与语音合成。)

上面三个关联的作用是让被关联的对象随着关联的点的位置变化而变化。让原本不能"动"的对象动起来。

"表格和曲线对象关联": 选中一个统计表格以及函数曲线,激活该命令并使用,可以使得函数曲线与统计进行关联。对应的函数是 *StatAssoc(、)*。

该关联的作用是统计表格现在曲线上的一系列点,例子如下。

点击"画图"菜单下的"一般曲线"子菜单,选择"y=f(x)",绘制函数曲线:  $y=x^2$ ,范围是-2~2,样本点是 n。





图 8.5.2

点击"插入"菜单下的"变量",输入n,做n的变量尺。



图 8.5.4

图 8.5.5

点击"插入"菜单下的"表格",生成一个统计表格。



图 8.5.6

选择统计表格与函数曲线进行关联,拖动n,观察n的值与统计表格的关系。

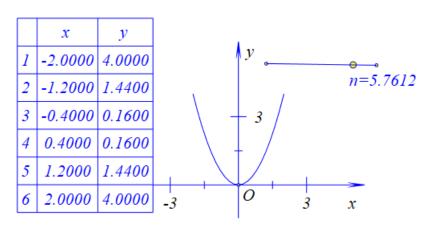
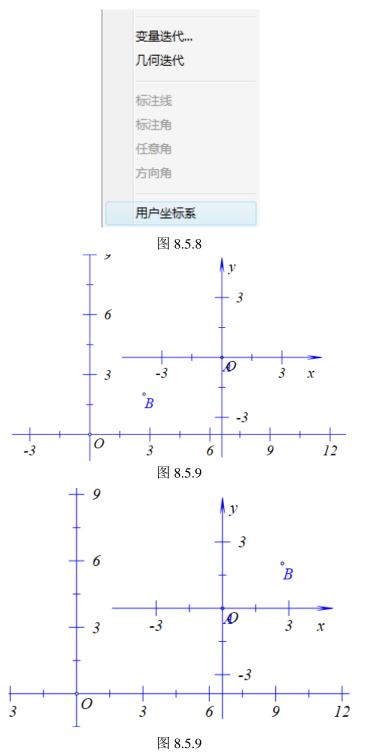


图 8.5.7

(注: 右击表格, 打开属性对话框, 可以设置"统计表格"的数值的小数点位数。)

"对象和用户坐标系关联":选中任意一个对象以及一个用户坐标系,激活该命令并使用,可以使得对象脱离系统坐标系,成为用户坐标系的一个对象。对应的函数是*UserCoordAssoc(*,)。具体例子如下。

首先建立一个"用户坐标系": 选择一个点 A,点击"画图"菜单下的"用户坐标系"命令,生成一个用户坐标系。使用画笔绘制一个点 B,选择点 B,并且选择用户坐标系,使用"对象和用户坐标系关联",此时点 B 进入用户坐标系。



若要使得点 B 回到原来的系统坐标系,可以选择点 B,然后单击"取消对象的用户坐标系关联"(对应的函数是 UnUserCoordAssoc())。

# 8.6 视口的作用

Hawgent 皓骏的视口有非常多的用途。

首先,我们单击"插入"菜单下的"矩形视口",然后再绘制一个圆:

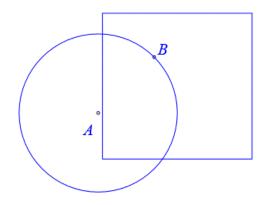


图 8.6.1

然后先选择圆,再选择矩形视口,最后单击"编辑"菜单中的"裁剪对象"命令。

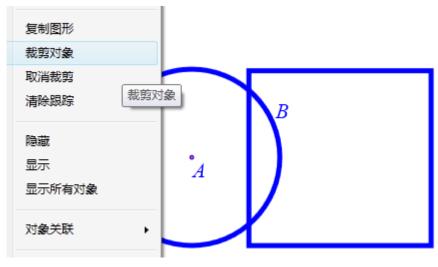


图 8.6.2

效果如下:

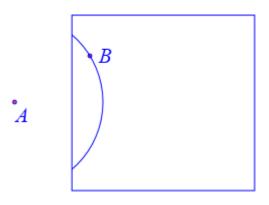


图 8.6.3

圆 AB 已经被矩形视口"裁剪"了,只有在视口内才会显示圆。将矩形视口删除,只需

要手动刷新一下界面即可显示原来的圆 *AB*。注意裁剪的顺序是先选择被裁剪的对象,再选择窗口。对于"椭圆视口"有相同的操作。如果需要取消裁剪,按照相同的步骤选择对象,然后单击"取消对象的裁剪"。

事实上,"矩形视口"与"椭圆视口"只是提供了两个现成的窗口,任何一个对象都可以作为一个窗口。而这些窗口因为都是对象,可以使用动画命令,因此在 Hawgent 皓骏中非常容易制作一个遮罩动画:

新建一个案例,绘制线段 AB,取线段 AB 上的一点 C,然后以 C 为圆心做一个半径为 3 的圆。使用"数学公式"输入一个文本"遮罩动画"。

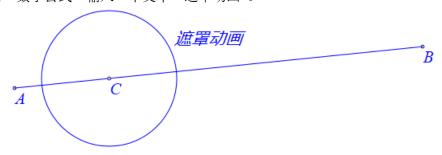


图 8.6.4

先选择文本,然后选择圆,最后单击"裁剪对象":

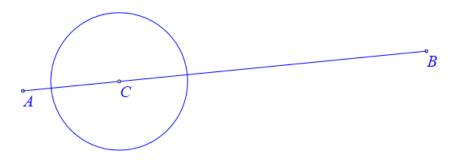


图 8.6.5

点击"视图"菜单栏的"动画控制对话框":



图 8.6.6

选择点 C, 单击播放按钮 $\longrightarrow$ , 效果如下:

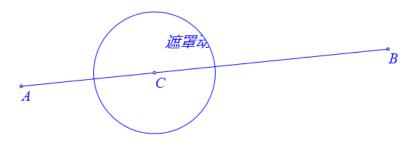


图 8.6.7

需要注意的是,作为窗口的对象如果被另一个窗口裁剪,并不会影响对原来对象的显示。 同一个对象只能被裁剪一次,多个窗口剪切同一个对象,默认最后一次有效。

## 8.7 对象组

对象组的使用包括新建"生成对象组","加至对象组"与"脱离对象组",分别对应的函数分别是 ObjGroup(,), AddToGroup(,), MoveFromGroup(,)。将某些对象设置为统一的对象组之后,可以通过对象组进行全部选中,这样可以方便对这些对象做统一的操作。比如,可以让对象组与视窗进行裁剪。

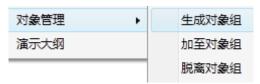


图 8.7.1

下面以一个例子做一下说明: 使用"画笔"工具绘制一个圆。

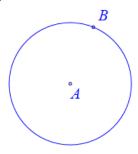


图 8.7.2

选择圆以及圆上的点 B, 然后单击"生成对象组", 输入标题——"圆"。



图 8.7.3

点击"视图"菜单下的"对象框",打开对象列表。



图 8.7.4

我们可以看到圆与点 B 已经成为同一个对象组了。

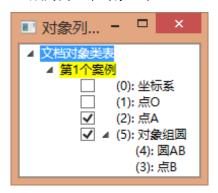


图 8.7.5

如果希望点 A 也成为"圆"这一个对象组,则可以在对象列表中选中点 A 和对象组圆,然后点击"加至对象组"。



图 8.7.6

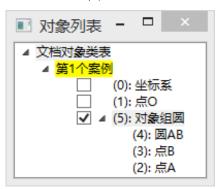


图 8.7.7

如果想将里面的对象移除,操作步骤与增加对象到对象组是一样的,只不过点击的是"脱离对象组"。对象组是一个非常重要的功能,它把一系列对象当成一个对象来看待,可以大大方便用户操作。例如,我们现在增加一个矩形视口,选中对象组和矩形视口可以进行裁剪。

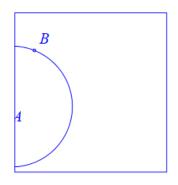


图 8.7.8

这样的方法比起一个一个对象的进行裁剪,效率大大提高了。

## 8.8 作图规则

Hawgent 皓骏中的"画图"菜单有许多个功能,这里也不一一介绍,只介绍几个较难用或者需要注意的地方。其他功能读者可以自行探索。

(1)"画笔"工具绘制的自由点,是作图菜单里面没有的。鼠标左击点并且按住不放拖动,可以改变自由点的位置。点处于选中状态时,当鼠标靠近点的名称时,鼠标会变成十字箭头,此时按住鼠标左键不放并进行拖动,可以改变点的名称的位置。



图 8.8.1

- (2)除了点之外,其他对象是没有名称的。想要给其他对象,只能使用文本并且关联。
- (3)除了"平行线上的点"和"垂直线上的点",其他所有的"自由点"的命令都可以使用"画笔"实现。



图 8.8.2

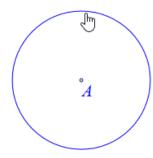


图 8.8.3

(4)"平行线上的点"和"垂直线上的点"命令,必须先选择一个点,再选择一条线段(射线、直线),才能激活菜单。



图 8.8.4

#### (5)"中点"

使用"画笔"工具绘制两个点,然后依次选择两个点,单击"画图"菜单的"约束点"的"中点"命令。



图 8.8.5

- (6)"点在直线上的垂足",必须先选择一个点,再选择一条线段(射线、直线),才能激活菜单。
- (7)"比例点"必须依次选择四个点才能激活菜单,点击菜单之后还需要输入一个比值。例如,依次选择点 A、点 B、点 C 和点 D,则做出的点 E 是  $\overline{AB}$  上的一点(有方向,由输入的比值的符号决定),且满足  $\frac{AE}{CD}$  = 输入的比值。



图 8.8.6

- (8)"直线上的参数点"与"圆上的参数"输入的可以是一个数值、变量,也可以是一个式子。
- (9)"交点"的一系列命令在使用的时候,要按照菜单的说法选取对象。例如"平行直线和圆弧的交点",必须先选择一条线段(射线、直线),再选择圆弧,才能激活菜单。



图 8.8.7

- (10)"正多边形"、"圆内接正多边形"与"圆外切正多边形"边数可以是一个数值,也可以是一个变量,甚至是一个式子。
- (11) Hawgent 皓骏一般以顺时针方向为正方向,因此许多构图方式是顺时针的。例如,依次选择点 A、点 B,点击"正方形",则生成的正方向的另外两个点与点 A、点 B 构成逆时针方向。



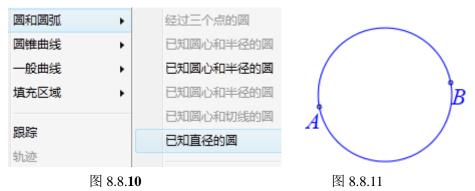
图 8.8.8

(12)使用"画笔"工具画出的圆就是"圆心和圆上一点",因此这个菜单几乎不使用,可以取消掉。

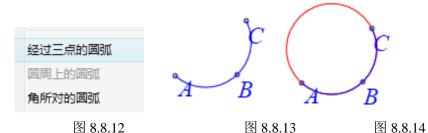


图 8.8.9

(13)"直径的圆"只需要选择两个点就可以激活菜单。



(14) 依次选择三点,可以激活"过三点的圆弧"。其中,所构造的圆弧对应的圆,是过所选择三点的圆。



(15)函数曲线除了(三次)Bezier曲线和二次Bezier曲线不能在上面绘制点,其余函数曲线都可以在上面绘制可拖动点。

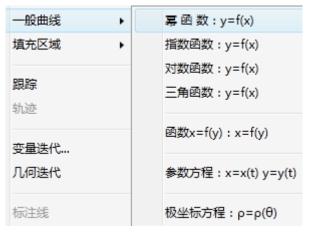
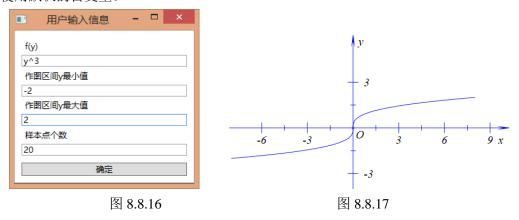
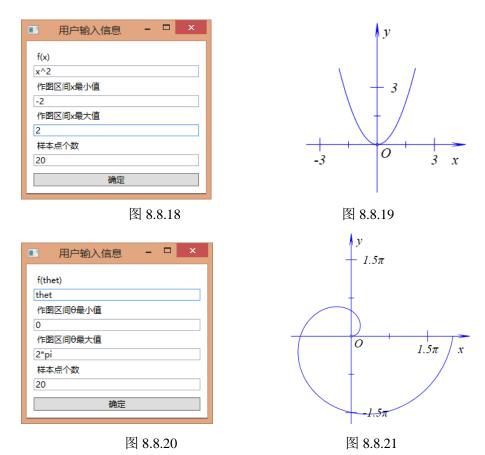


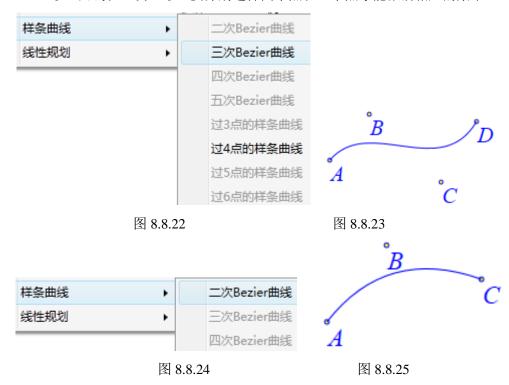
图 8.8.15

(16)函数曲线除了参数曲线,其余曲线的自变量都不能更改。因此在绘制特定的曲面时也要使用默认的自变量。





(17) Bezier 曲线和二次 Bezier 要分别选择四个点和三个点才能激活相应的菜单。



## 8.9 对象的轨迹

Hawgent 皓骏的轨迹是非常强大的,按照用法可以分为几何轨迹,测量轨迹,坐标点轨迹。不过这是不严格的分法,这里只是为了说明轨迹的三种用法。后面使用几个例子来说明。

#### (1) 几何轨迹

使用"画笔"绘制一个圆 AB 与一条线段 CD。

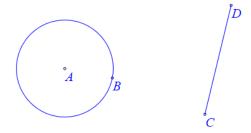


图 8.9.1

在圆AB上取一点E, 在线段CD取一点F, 连接EF, 在线段EF取一点G。

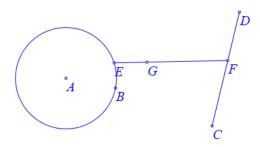


图 8.9.2

依次选择点E、点F和点G,点击"画图"菜单的"轨迹"。



图 8.9.3

拖动点G,可以看到轨迹的变化。

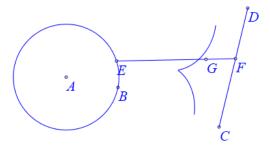


图 8.9.4

右击点 G 的轨迹,打开属性对话框,左上角是动点的列表,点击"点 E",打开"点 E" 的属性设置。将"样本点数"设置为 100,"运动点运动周期数"设置为 2。注意,"运动类型"中的"一次向前"和"一次向后"会让"运动点运动周期数"默认为 1,因此这里不需要修改。打开"点 F"的属性设置,将运动区间设置为 0~2。

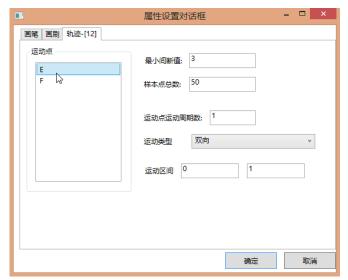


图 8.9.5

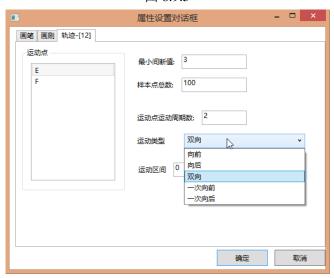
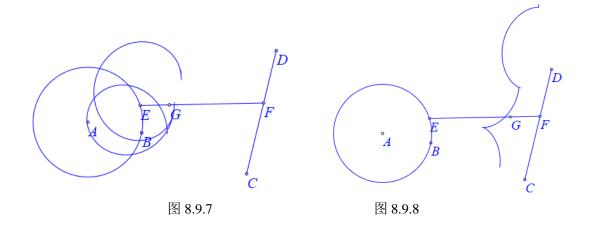


图 8.9.6



### (2) 测量轨迹

使用"画笔"工具绘制一条线段AB,在线段AB上取一点C。

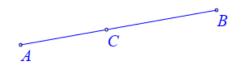
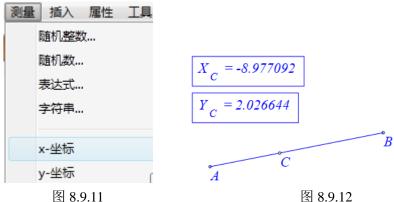


图 8.9.10

选择点 C,点击"插入"菜单下的"测量"的"点的 x-坐标"命令,测量点 C 的 x 坐标,其测量变量为 v000 (具体参见《初识测量》一节)。使用相同的步骤测量点 C 的 y 坐标,其测量变量为 v001。

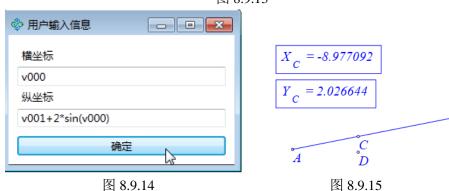


点击"画图"菜单下的"参数点"的"坐标点"命令。输入: v000, v001+2\*sin(v000)。

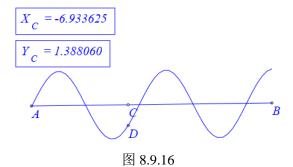
 $^{\circ}B$ 



图 8.9.13

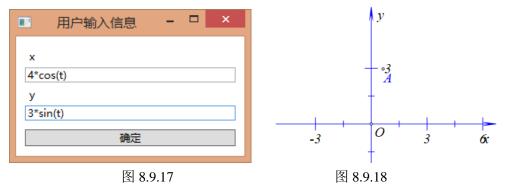


依次选择点 C, 点 D, 点击"画图"菜单的"轨迹"。



### (3) 坐标点轨迹

点击"画图"菜单下的"参数点"的"坐标点"命令,输入: 4\*cos(t), 3\*sin(t)。



点击"插入"菜单的"变量"命令,输入 t,范围为 0~3,单击"确定"生成一个 t 的变量尺。重复操作生成一个 k 的变量尺,范围为-10~10。



图 8.9.19

图 8.9.20

依次选择t的变量尺,坐标点A,点击"画图"菜单的"轨迹"。

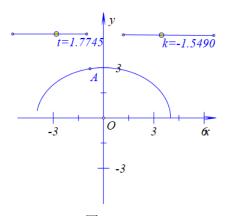


图 8.9.21

右击生成的轨迹,打开属性对话框,修改t的范围为0~k。

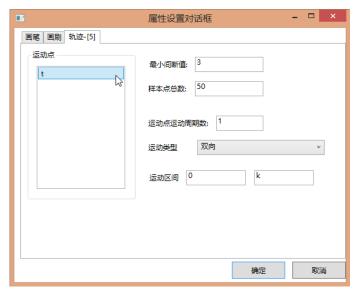
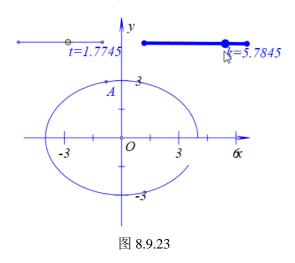


图 8.9.22



# 8.10 对象的跟踪

Hawgent 皓骏的跟踪不会因为界面的刷新而消失,因此跟踪的数目是有限定的,默认为 200。具体操作如下:

新建一个案例,使用画笔工具  $\checkmark$  绘制一个点 A。选择点 A,单击"画图"菜单下的"跟踪"命令:

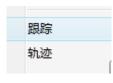


图 8.10.1

此时选择点A并拖动它,可以看到拖动的踪迹:

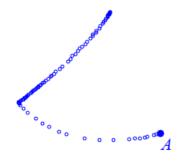


图 8.10.2

如果需要清除跟踪,可以单击"编辑"菜单下的"清除跟踪"命令:



图 8.10.3

当然,"清除跟踪"与删除跟踪是不一样的,拖动点 A,跟踪仍旧会出现。如果需要删除跟踪,可直接选择跟踪,然后单击删除工具 $\times$ .

# 8.11 两类迭代

Hawgent 皓骏提供了两种迭代,一种是常见的几何迭代,一种是常用的变量迭代。



图 8.11.1

首先介绍变量迭代。变量迭代主要的用途是构造递推数列并且绘制相应的点列。首先, 单击"画图"菜单下的"变量迭代",打开输入框。

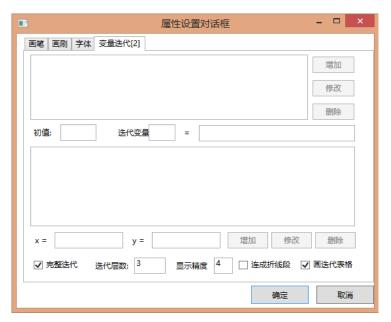


图 8.11.2

首先,我们构造一个数列,作为点列的x坐标——初始值为1,递推公式为 $x_n = x_{n-1} + 1$ 。在"变量迭代"的输入框中的"初值"框中输入1,在"迭代变量"框中分别输入xn,xn+1。然后单击左上角的"增加"按钮,这样就增加了一个数列。

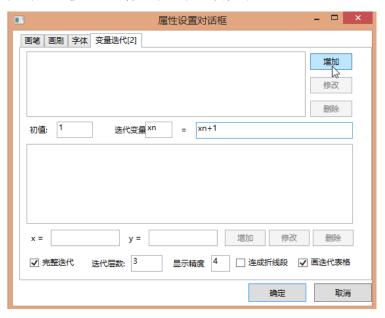


图 8.11.3

接着,我们构造一个数列  $y_n = \sum_{i=1}^n \frac{1}{i}$ ,作为点列的 y 坐标。则该数列的初值为 1,递推

公式为  $y_n = y_{n-1} + \frac{1}{x_n}$  , 其中  $x_n$  是上面构造的数列,因此只需要在"变量迭代"的输入框中

的"初值"框中输入 1,在"迭代变量"框中分别输入 yn, yn+1/xn。然后单击左上角的"增加"按钮,这样就增加了一个新数列。

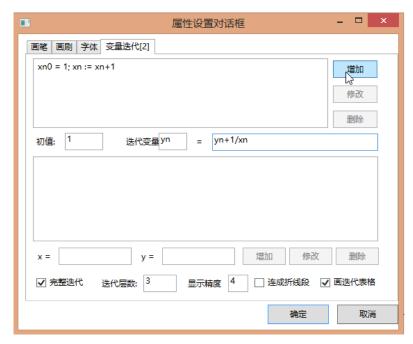


图 8.11.4

最后,使用两个数列构造一个点列。在 "x" 框中输入 xn,在 "y" 框中输入 yn,点击下面的 "增加" 按钮,则增加一个点列。

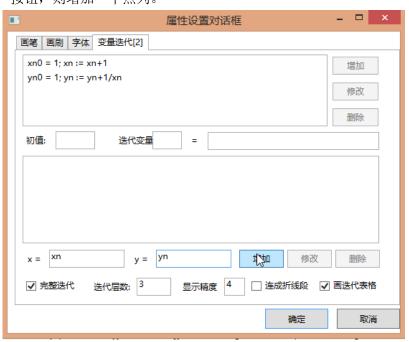


图 8.11.5

设置迭代层数为20,勾选"连成折线段",单击"确定"完成。

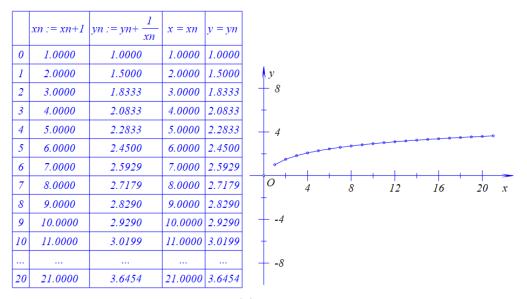


图 8.11.6

同一个变量迭代,可以构造多个点列(如果是连线则会进行交互连线,要单独连线必须再做一个"迭代变量")。同时,构造的数列输入的"初值"、"递推公式"和"迭代层数"可以是一个数值、或者是变量、甚至是一个式子。这样就能构造各种动态的点列,可以用来探究一些有趣的东西。例如,右击生成的表格或者对象列表中的"变量迭代"对象,打开"变量迭代"的输入框中,在"初值"框中输入 0.5,在"迭代变量"框中分别输入 zn, $(1+r)*(zn-zn^2)$ ,然后单击左上角的"增加"按钮(即增加数列  $z_n$ ,初值为 0.5,递推公式为  $z_n=(1+r)*(z_{n-1}-z_{n-1}^{-2})$ )。删除原来的点列,增加新的点列(xn,zn)。单击"确定"完成。

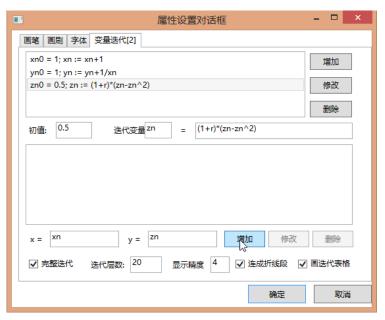


图 8.11.7

点击"插入"下的"变量",输入 r,范围为-3~3。单击"确定"完成。

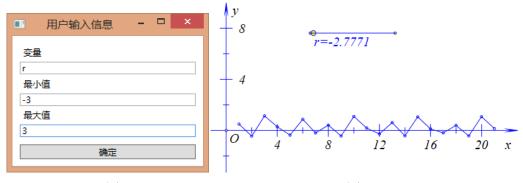


图 8.11.8

图 8.11.9

几何迭代是比较熟悉的迭代,这里主要介绍一下一般的操作步骤。

单击"画图"菜单的"参数点"的"坐标点"命令,分别使用相同的步骤添加 A (1,1), B (5,1), C (5,5), D (1,5)。

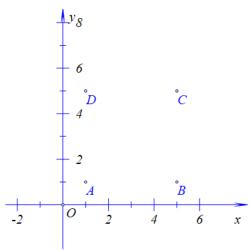


图 8.11.10

点击"插入"的"变量",添加t的变量尺,范围是0~1。

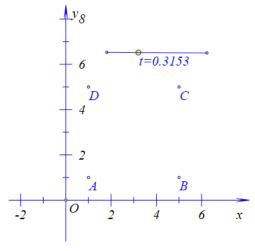


图 8.11.11

依次选择点 A, 点 B, 点击 "画图"菜单的"参数点"的"旋转缩放点"。旋转角度为 0, 放缩倍数为 t, 单击确定生成点 E。对点 B 与点 C, 点 C 与点 D, 点 D 与点 A 分别作相同的操作,生成点 F, 点 G 和点 H。

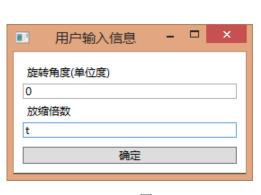


图 8.11.12

连接线段 DE、EF、FG 和 GD。

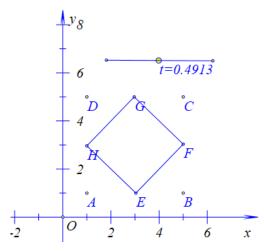


图 8.11.14

依次选择点 A、点 B、点 C 和点 D。单击"作图"菜单的"几何迭代"。



变量迭代...

图 8.11.15

图 8.11.16

勾选"对象列表"的全部对象(即在迭代中显示所有这样的对象)。



图 8.11.17

分别双击"迭代列表"的点 E、点 F、点 G 和点 H。此时这些点就会自动进行"映射框"中。如果双击到错误的点,只需要在"映射框"中再次双击错误的点即可取消。



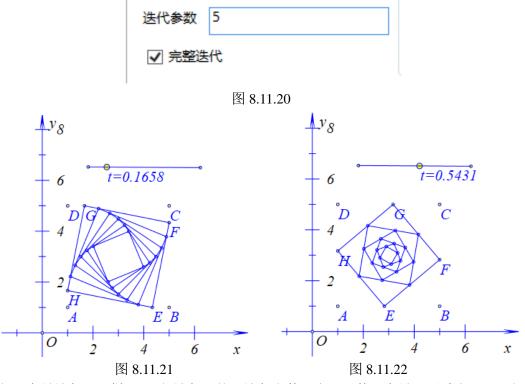
图 8.11.18

选择完毕之后,点击"增加映射",此时"映射表"出现一个映射。一个迭代可以建立 多个映射,这里只需要一个映射。



图 8.11.19

设置"迭代参数"为5,单击确定完成



与"变量迭代"一样,"几何迭代"的"迭代参数"也可以使用变量,不过由于"几何迭代"对电脑的负荷较大,因此建议设置的值不要过大。

"变量迭代"与"几何迭代"都有一个"完整迭代"的勾选框。一般是默认勾选的,如果不勾选,则结果只显示最后一次迭代的结果。

## 8.12 文本与公式

Hawgent 皓骏提供两种"文本"的输入,一种是"普通文本",另一种是"数学公式"。





图 8.12.1

图 8.12.2

点击"插入"的"普通文本",粘贴一个文本(这里粘贴是《蜀道难》),单击确定完成。

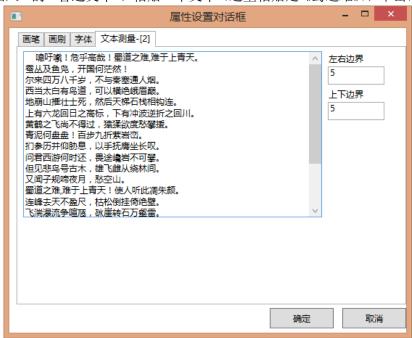


图 8.12.3

"普通文本"有一个特点,就是可以首字空格,并且可以拖动边框,使文本自动换行。

```
噫吁喊!危乎高哉!蜀道之难,难于上青天。

蚕丛及鱼凫,开国何茫然!

尔来四万八千岁,不与秦塞通人烟。

西当太白有鸟道,可以横绝峨眉巅。

地崩山摧壮士死,然后天梯石栈相钩连。

上有六龙回日之高标,下有冲波逆折之回川。

黄鹤之飞尚不得过,猿猱欲度愁攀援。

青泥何盘盘!百步九折萦岩峦。

扪参历井仰胁息,以手抚膺坐长叹。

问君西游何时还,畏途巉岩不可攀。

但见悲鸟号古木,雄飞雌从绕林间。

又闻子规啼夜月,愁空山。

蜀道之难,难于上青天!使人听此凋朱颜。
```

噫吁嚱!危乎高哉!蜀道之 难难于上青天。 蚕丛及鱼凫,开国何茫然! 尔来四万八千岁,不与秦塞 通人烟。 西当太白有鸟道,可以横绝 峨眉巅。 地崩山摧壮士死,然后天梯节 石栈相钩连。 上有六龙回日之高标,下有 冲波逆折之回川。 黄鹤之飞尚不得过,猿猱欲 度愁攀援。 青泥何盘盘!百步九折萦…

图 8.12.4

图 8.12.5

"数学公式"没有"普通文本"的优点,但是它是最常用的文本。它主要在显示数学公式有独特的优势。

点击"插入"的"数学公式"。打开输入框,点击下拉框,选择"不定积分"。

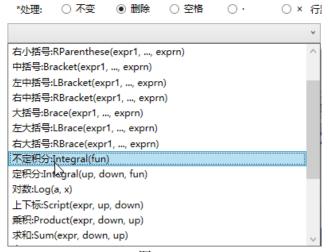


图 8.12.6

修改 fun 为  $x^2$ , 并且再后面加上 dx。单击确定按钮完成。

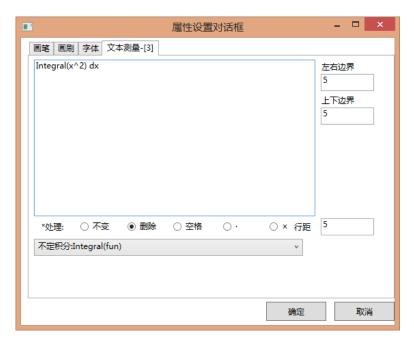


图 8.12.7

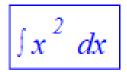


图 8.12.8

## 8.13 初识测量

Hawgent 皓骏的"测量"非常丰富,这里重点介绍测量的一些使用方法。

#### (1) 测量表达式

在"测量"功能中,最常用到的是"表达式"的功能。点击"测量"菜单下的 "表达式"命令。





图 8.13.1

图 8.13.2

弹出的输入框中需要输入两个参数——"表达式"和"测量变量"。"表达式"就是需要计算的式子,这里输入一个 *a*。而"测量变量"不输入任何数据,使用系统默认生成的变量。单击确定完成,右击生成测量文本,打开属性对话框,如下图所示:

a = 1.960631

图 8.13.3

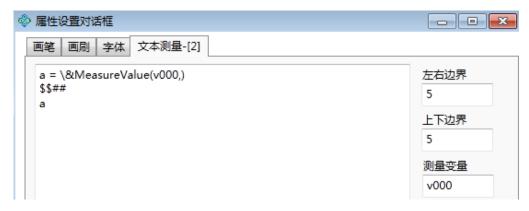
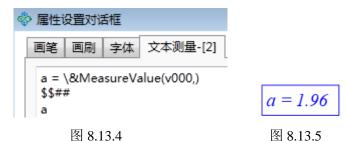


图 8.13.4

从图中可以看出,默认的测量变量,Hawgent 皓骏使用的是"v000"、"v001"、"v002"……将 a = & Measure Value(v000,)修改为 a = & Measure Value(v000,2)可以修改默认的小数点位数为 2 位。



如果需要显示的结果以度作为单位,则可将 a = & Measure Value(v000,2)改为 a = & Measure Value(v000,2,0)。



"测量变量"一般使用系统默认的,除了特殊情况很少自定义。那么,"变量"有什么作用呢?

将 a = & Measure Value(v000,2,0)重新改为 a = & Measure Value(v000,2)并确定。再次点击"测量"菜单下的"表达式"命令。输入: v000+1,单击"确定"完成。插入 a 的变量尺,拖动 a 观察结果。

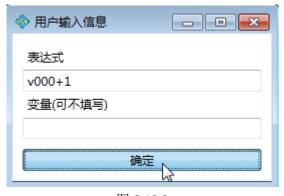


图 8.13.8

$$a = -3.17$$
  $a = 1.53$   
 $1+v000 = -2.17$   $1+v000 = 2.53$   
 $a = -3.17$   $a = 1.53$   
 $\bigcirc$  8.13.9  $\bigcirc$  8.13.10

从结果可以大致理解"测量变量"的作用——对应的测量表达式的计算结果,可以使用 在其他方面,比如重新用于计算。

### 2, 测量随机数

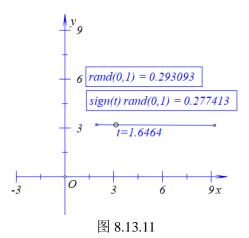
Hawgent 皓骏中随机数使用的函数是 *rand*(,), 函数中需要输入两个参数, 分别是最小值和最大值。Hawgent 皓骏的随机数有自己独特的刷新机制——"关联刷新机制"。在介绍这个刷新机制的用法,首先来认识两个常用到的函数——*sign*()和 *greater*()。

sign 函数在 Hawgent 皓骏中的定义具体如下:

上面这两者方法都可以使用,主要是看输入的参数的个数。 greater 定义如下:

$$greater(a,b)$$
  $\begin{cases} 1, & \text{id} a > b \\ 0, & \text{id} a \leq b \end{cases}$ 

了解这两个函数之后,我们利用"表达式"分别计算两个式子:"rand(0,1)"和"sign(t)\*(rand(0,1))"然后插入t的变量尺,范围为0~10。



拖动坐标系,两个随机数均没有改变数值。

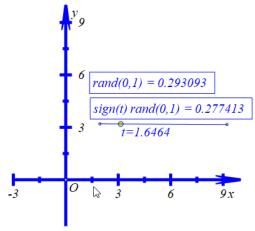


图 8.13.12

拖动变量尺 t, 第二个随机数数值发生改变。

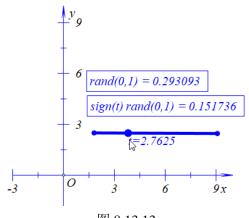


图 8.13.13

从上面可以知道,随机数的刷新是跟着所关联的变量的刷新而刷新的。第一个随机数的数值想要改变,只有拖动自身才行,不过这没有什么意义。而且,不仅仅只是t的值改变随机数才会改变,当t的变量尺的位置改变的时候,随机数的值也会改变,这就是所谓的"关联刷新机制"。

### 3, 嵌套测量

在学习本节时,需要知道按钮的初步使用方法,具体可以参见《初识按钮》一节。 点击"测量"菜单下"表达式"命令。,输入:"greater(t,0)\*(d+1)",测量变量为 d。



图 8.13.14

所谓的"嵌套测量",即测量变量在表达式中出现,这样计算的结果不断反馈给自己,进行不断计算,形成一个"嵌套"。

插入 t 的变量尺,范围为  $0\sim10$ 。单击"插入"菜单的"按钮",添加变量 t 的动画按钮——标题为 t,范围为  $0\sim10$ ,频率为 t0,运动类型为 t3(即一次运动),单击"确定"完成。

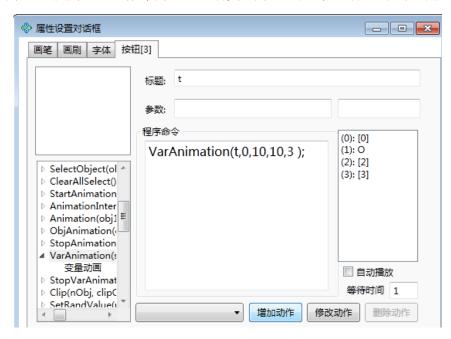


图 8.13.15

单击按钮的绿色部分启动按钮,查看结果,如下图:

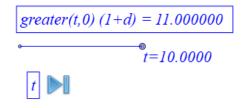


图 8.13.16

测量的结果比频率多 1,即 t 运动了 11 次。如果定义变量 t 每一次走过的长度为"步长",则步长=区间长度/(频率-1)。按钮中定义了 t 的范围为 0~10,即区间长度为 10,如果希望 t 每一次运动走过 1,则需要定义频率为 11。这就好比在 10 米长的公路上种一排树,使得每一棵树的间距为 1 米,则求得的结果为 11 棵树。

那为什么测量的结果会比频率多 1 呢?在 Hawgent 皓骏中其实没有严格的说法, 当频

率为 20 的时候测量的结果为 20, 而不是 21。总的来说,测量结果要么比频率多 1, 要么和 频率相等, 这是由 Hawgent 皓骏代数系统的缺陷造成的, 现在暂时没办法修改。

还有一个地方需要注意,如果我们再次单击按钮(需要先单击按钮使其恢复绿色状态), 结果如下。

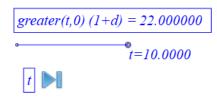


图 8.13.17

本来按照 greater()函数的定义,当 t=0 时,greater(t,0)\*(d+1)的计算结果应该为 0 才对,这也是 Hawgent 皓骏的代数系统的缺陷造成的,现在暂时无法修改,使用时需要注意。如果想把结果置零,可以拖动变量尺,让 t 的值为 0,或者再定义一个按钮。

#### 3, 一次测量

Hawgent 皓骏的"一次测量"是改进了嵌套测量而形成的一个非常重要的功能,主要用在随机数方面,功能是控制随机数的刷新。由于"一次测量"是来源于"嵌套测量"的思路,故也可以控制"测量"的"嵌套"次数。现在通过对"嵌套测量"的一步步改进,来看看"一次测量"的原理。

点击"测量"菜单下的"表达式"命令。,输入:"sign(t)\*rand(0,1)",测量变量为空(使用系统默认的测量变量)。插入t的变量尺,范围为0~10。

接着在测量表达式中输入: sign(t)\*(v001+v000)。结果如下图所示:

$$sign(t) \ rand(0,1) = 0.91$$
  
 $sign(t) \ (v000+v001) = 5.75$   
 $t=1.63$   
 $8.13.18$ 

这种方法就是上面介绍的"嵌套测量"。当我们拖动 t 的时候,v001 的作用就是不断累加随机数 v000; 而当 t=0 的时候,v001 就置零。现在我们想要当 t 超过 1 的时候,v001 不再累加 v000。

右击第二个测量文本(即 v001 对应的文本)将表达式修改为: sign(t)\*(v001+(1-greater(t,1))\*v000)。

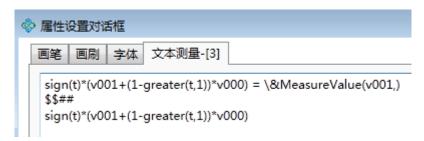


图 8.13.19

再次拖动 t, 我们可以发现 v001 的值在当 t 超过 1 的时候就不再增加,已经得到我们想

要的效果。

$$sign(t) \ rand(0,1) = 0.95$$
  
 $sign(t) \ (v001+(1-greater(t,1)) \ v000) = -6.21$   
 $t=3.25$ 

图 8.13.20

我们进一步提高要求,v001 在每一次累加 v000 的值,能否清空掉自身原来的值变为 0; 而当 v001 不再累加的时候,v000 就不会置零。受到前面 v001 置零方法的启示,我们把 v001 的表达式修改为:

 $sign(t)*((1-greater(abs(v01),0)+greater(t,1))*v001+(1-greater(t,1))*v000)\circ\\$ 

我们在表达式中的v001添加了一个式子: (1-greater(abs(v01),0)+greater(t,1))。这个式子解读起来并不难,就是当t小于等于1的时候,由于v001累加了v000,所以1-greater(abs(v01),0)返回的值是0,因此(1-greater(abs(v01),0)+greater(t,1))整个式子就等于0,这样就不断把v001置零。

而当 t 超过了 1 的时候,v001 不再累加,也不再置零。幸运的是,整个式子的执行过程是先将 v001 置零之后,再做一次累加,因此,在 t 超过 1 的一瞬间,v001 就获得了一个不变的随机数。

 $sign(t) \ rand(0,1) = 0.90$ 

sign(t) ((1-greater(abs(v01), 0) + greater(t, 1)) v001 + (1-greater(t, 1)) v000) = 0.01 t=1.48

### 图 8.13.21

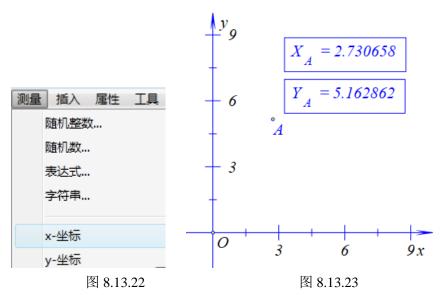
上面的思想方法就叫做"一次测量",利用这个方法,可以控制随机数的刷新(第二次修改)和嵌套测量的嵌套次数(第一次修改)。这个方法的用途非常广泛,比如,朝一个靶子射箭,需要演示射箭的过程,但是箭从哪里出现确实随机的,这就要求箭在出现的时候(设置动态透明度)在出现的时候是随机的,而出现之后位置就固定不变了,这时候"一次测量"就可以派上用处了。

(注:在测量表达式中输入: "greater(t,0)\*rand(0,1)",结果为 0,这也是 Hawgent 皓骏的代数系统造成的,现在不能修改,在测量表达式中,能不利用 greater(),就尽量不要使用,因为这个函数本来设计时是给按钮的程序使用的。如果非要使用,可以先随便输入一个式子,确定之后再修改为 greater(t,0)\*rand(0,1),或者输入: "{greater(t,0)\*rand(0,1)}"。不过按钮的程序由于"输入参数"的影响,又要求尽量少用大括号,则是一个矛盾。)

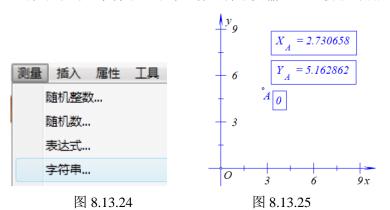
### 4,测量字符串

Hawgent 皓骏的"字符串"可以显示测量计算的结果。由于 Hawgent 皓骏的代数系统有一些问题,现在介绍简单的使用方法。

使用"画笔"绘制点 A,选择点 A,点击"测量"菜单下"x-坐标"命令,测量点 A 的 x 坐标。同理,测量点 A 的 y 坐标。



点击"测量"菜单下的"字符串"命令,什么都不要输入,直接单击确定按钮。



右击点 A 的 x 坐标的测量文文本, 打开输入框, 复制如下内容:



图 8.13.26

右击"字符串"文本,粘贴刚刚复制的文本。再一次粘贴并修改 v000 为 v0001,最终结果为: A(&MeasureValue(v000,2),&MeasureValue(v001,2))



图 8.13.27

$$X_A = 2.58$$
 $Y_A = 4.02$ 
 $^{\bullet}A$ 
 $A ( 2.58 , 4.02 )$ 

图 8.13.28

最终的结果可以将点 A 的名称隐藏,将"字符串"与点 A 进行关联。

### 5, 随机测量数

在学习本节时,需要知道按钮的初步使用方法,具体可以参见《初识按钮》一节。 点击"测量"菜单下"随机整数..."命令,弹出如下的窗口:



图 8.13.29

单击"确定"按钮,此时出现一个整数。

"随机浮点数"与"随机整数"的操作一样,只是生成一个浮点数。

"随机整数"与"随机浮点数"函数分别为 MeasureRandInt(min,max)和 MeasureRandFloat(min,max),它们是两个特殊的测量变量,其特殊性主要表现在两个变量不会轻易更新。要刷新"随机整数"与"随机浮点数"的数值,只有使用按钮输入相应的命令才行,具体操作如下。

点击"插入"菜单的"按钮"命令。添加一个按钮,标题为"随机整数",程序为"NextValue(2); (或者 SetRandValue(2);)"。(注: 2 是刚刚生成的"随机整数"文本的编号)。此时单击按钮即可不断改变"随机整数"的值。



# 8.14 插入多媒体

Hawgent 皓骏的多媒体包括图片、音频、视频和语音合成。其中由于语音合成的用法比较特殊,因此作为一节独立讲解。而图片、音频和视频的操作方法类似,因此只介绍插入图片的方法。事实上,Hawgent 皓骏在插入多媒体的时候,是把多媒体合并到课件中,原来的文件删除不会影响到课件中的多媒体。但是这也使得课件变得较大,不便于网上传输,因此在插入多媒体方面要慎用。

首先,点击"插入"菜单的"图片"命令。



图 8.14.1

点击"选择文件"按钮,打开图片所在的位置并且选择一张图片,单击"打开",最后单击"确定"完成。



图 8.14.2

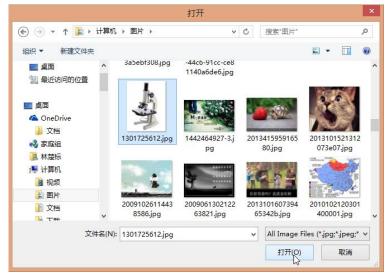


图 8.14.3

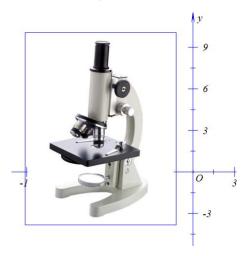
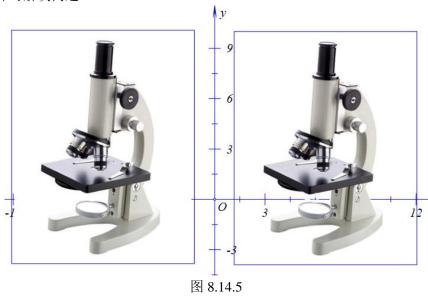


图 8.14.4

插入的图片有一个问题,就是白色的背景挡住了后面的坐标系,不过这不是 Hawgent 皓骏的问题。使用 *PhotoShop* 处理一下图片,将原来的图片改为无背景的 *png* 格式的图片。再次插入即可解决问题。



# 8.15 语音合成

使用 Hawgent 皓骏的"语音合成",可以将文字朗读出来。具体操作如下:新建一个案例,单击菜单栏的"插入"的命令"语音合成":



图 8.15.1

弹出如下窗口:

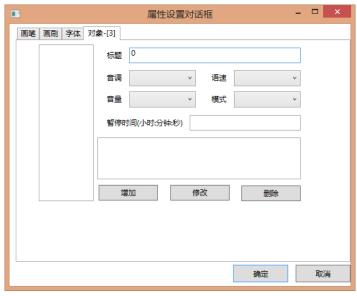


图 8.15.2

在标题栏输入"静夜思",文本框内输入如下内容:

静夜思。李白。

床前明月光, 疑是地上霜。

举头望明月, 低头思故乡。

这些内容是即将朗读的内容,当程序遇到逗号和句号时,会停顿一下,因此注意在需要停顿的地方加一下标点符号。对于其他设置我们先缺省,此时单击"增加",然后单击"确定"即可。



图 8.15.3

播放时的效果图:



图 8.15.4

中间的按钮式"暂停",右边的按钮是"停止"。 如果对效果不满意,可以右击"语音合成"按钮,打开属性对话框:

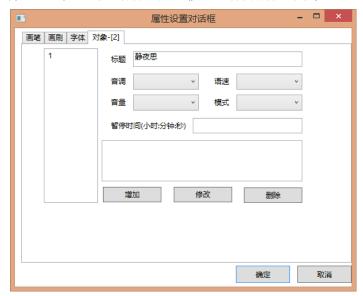


图 8.15.5

点击一下左边列表框的"1",显示朗读的内容,然后对音调、语速等进行设置,暂停时间的设置是对标点符号停顿的时间的设置。

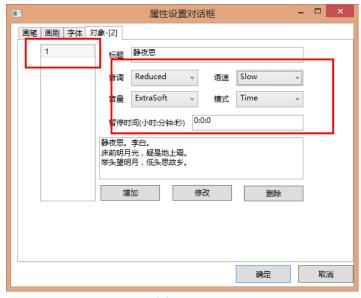


图 8.15.5

语音合成也是属于多媒体的一种,因此可以通过前面介绍的方法,利用按钮控制播放、暂停和停止。"语音合成"在实际使用时主要是用来朗读题目的。现阶段语音合成只能识别中文、英语单词和数字,对于希腊字母还不行。

# 8.16 统计表格

点击"插入"菜单的"表格"命令。

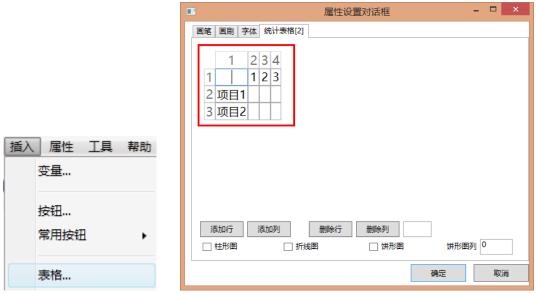


图 8.16.1

在"统计表格"的属性对话框中,红色区域是可以进行输入的区域,但是只能输入一些中文和数字,不能输入变量。下面输入如图所示的一些数据:

	1	2	3	4
1		2011	2012	2013
2	某地降水量	299	255	250
3	某地GDP	300	320	400

图 8.16.3

确定之后,结果如下图:

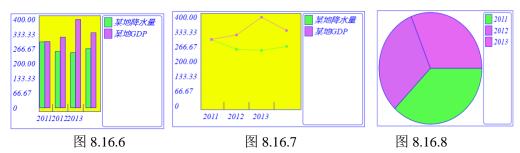
	2011	2012	2013
某地降水量	299	255	250
某地GDP	300	320	400

图 8.16.4

如果需要显示柱形图、折线图与饼形图,可以在属性对话框中勾选起来。



图 8.16.5



其中, 柱形图与折线图的第四个数据是前三个数据的平均值。

## 8.17 动作按钮

按钮是 Hawgent 皓骏最重要的功能。任何使用菜单命令实现的功能都可以使用按钮实现。在使用中,最多的是使用按钮实现对象动画或者变量动画。首先,介绍如何使用按钮制作一个变量动画按钮。

添加t的变量尺,范围是-10~10,点击"插入"菜单的"按钮"。

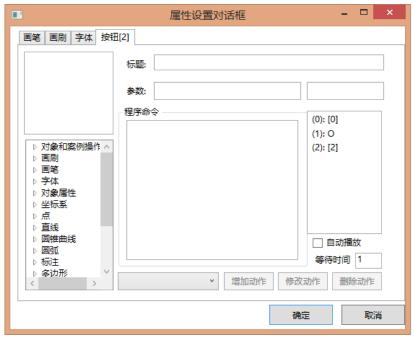


图 8.17.1

在标题栏输入"t动画"。

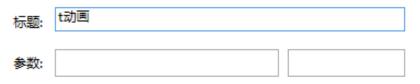


图 8.17.2

在"函数库"中寻找"变量动画"函数: VarAnimation(,,,,);, 当然能记住可以直接在程序区中输入。



图 8.17.3

图 8.17.4

双击寻找到的函数,此时程序区出现该函数。



图 8.17.5

修改函数里面的参数为 VarAnimation(t,0,10,50,3);这个命令表示 t 的运动范围是 0~10,运动的频率为 50,运动类型为一次运动(注:重复运动为 0,逆向重复运动为 1,往返运动为 2,一次运动为 3。更多函数的使用请参见《皓骏动态数学资源软件函数使用指南》)。修改完毕之后,点击"增加动作",此时"标题列表"出现刚才的标题。



点击该标题,可以对内容作修改。修改之后只要点击"修改动作"即可,这里我们不做 修改。



图 8.17.7

点击"确定"完成。单击按钮的绿色部分,观察变量尺的变化。



图 8.17.8

Hawgent 皓骏按钮的还提供了动态选择的功能。下面使用一个例子来说明。使用"画笔"工具绘制两个点。

 $^{\circ}B$ 

 $^{\circ}_{A}$ 

#### 图 8.17.8

现在希望做一个按钮,单击按钮可以让点 B 绕点 A 做一个旋转缩放点 C,旋转角度为 180 度,缩放比例为 1。

点击"插入"菜单的"按钮",新建一个按钮,在函数库中找到"旋转缩放点"的命令。



双击一下这个命令(注:如果不知道怎么用这个命令,可以先拖动函数库的左拉条,查看相关的提示,如上图所示,PointFlexRotate2D(,,,)需要填入四个参数,前两个是点(pt),后面两个分别是角度和比例(angle 与 scale))。

查看按钮的对象列表,可以知道点 B 和点 A 的编号分别是 3 和 2。或者单击"视图"菜单的"对象框"也可以查看点 A 与点 B 的编号。(注:按钮的对象列表是实时更新的,而主对象列表,当删除某些对象之后,对象编号可能会出现偏差。因此,查看按钮的对象列表比较可靠一点,或者直接选中对象在下边栏进行查看)

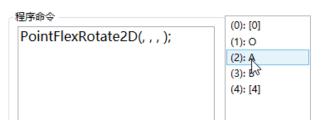


图 8.17.11



图 8.17.12

图 8.17.13

因为旋转度数是 180, 比例是 1, 因此按钮的主程序应该为"*PointFlexRotate2D*(3,2,180, 1);",按钮标题为"旋转缩放点",单击"增加动作",然后再一次单击"确定"。



图 8.17.14

现在我们就可以实现单击按钮快速做一个旋转缩放点,结果如下图所示:



 $^{\circ}B$ 

 $^{\circ}_{A}$ 

 ${}^{\circ}C$ 

#### 图 8.17.15

如果希望按钮的功能是让某个点(不一定是点 B)绕点 A 旋转缩放。由于我们不知道这个点是哪个参数,因此需要利用按钮的"对象参数"。

右击按钮打开属性对话框,点击标题列表的"旋转缩放点"。

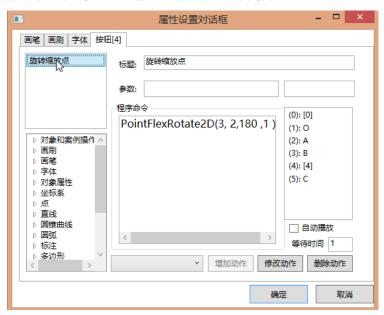
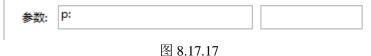


图 8.17.16

修改程序命令为: "PointFlexRotate2D(p,2,180,1);"。这时命令的意义就变为有某个点,它的编号为p,绕着点A生成一个旋转缩放点。

为了让程序知道 p 是某个点的编号,我们必须在对象参数框中输入 "p:",然后点击下拉框,选择 "点类型",如下图所示:



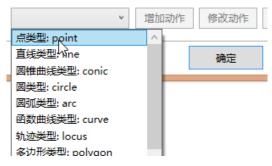


图 8.17.18

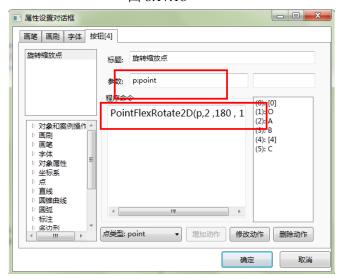


图 8.17.19

单击"修改动作",然后再单击"确定"。这样,每一次在单击"旋转缩放点"按钮时,必须事先选择一个点,为 p 赋值,这样才能执行按钮的命令,快速生成某个点绕点 A 的旋转缩放点。例如,使用"画笔"绘制一个点 D,选中点 D,由于点 D 的编号是 5,此时按钮程序 p 的值就是 5,这样单击按钮就可以成点 D 绕点 A 的旋转缩放点。(注:由于限制了 p 是点类型,所以选择其他类型的对象,p 是不会有值的,因此按钮也是不会执行的。)



图 8.17.20

如果甚至不知道谁绕谁旋转缩放,可以再修改按钮的命令,将程序命令改为: "PointFlexRotate2D(p,p1,180,1);",在对象参数框输入一个英文逗号隔开原来的代码,再添一个值——"p1:point",如下图所示:

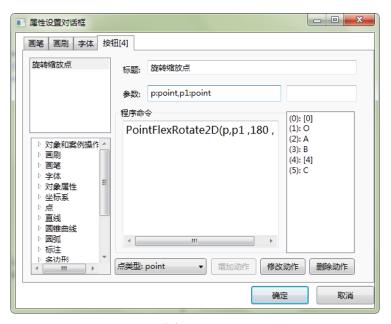


图 8.17.21

这样我们每一次我们单击按钮,要事先选择两个点,规定谁绕谁,然后单击按钮,就可以快速制作旋转缩放点。

如果希望上面的按钮的功能进一步增强——有时候我们并不知道旋转角度是多少,甚至有时候连缩放比例也是待定的,该怎么办?这时候,我们可以利用按钮的"输入参数"。

首先,打开按钮的属性对话框,将"PointFlexRotate2D(p,p1,180,1);"修改为"PointFlexRotate2D(p,p1,a,s);"。也就是说现在旋转角度待定为a,缩放比例待定为s。

接着,我们在按钮的右边参数输入: "a:",后面输入一个提示语,如"请输入一个角度(单位度)"。然后用"##"隔开,接着输入: "s:请输入一个缩放比例"。然后单击"修改动作",再单击"确定"完成按钮命令修改。

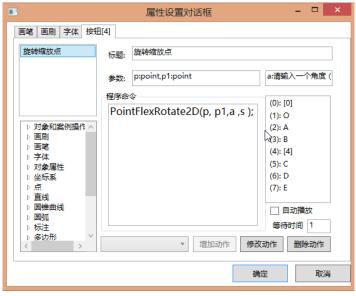


图 8.17.22

这样我们每一次选择两个点之后,再单击按钮就会有两个输入框弹出要求你输入两个值,如下图所示:



图 8.17.23

如果希望两个窗口有两个默认的值,则可以把"输入参数框"中的内容改为: "a=180: 请输入一个角度(单位度)#t=1:请输入一个缩放比例"。这样输入信息就会有两个默认值提供参考,如下图所示:



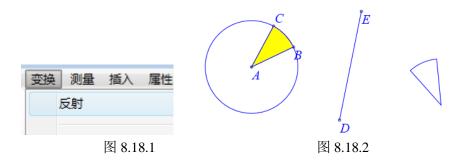
图 8.17.24

需要理解的是,就是利用按钮"输入参数"输入的值并不是赋值给 a 和 s, a 和 s 只是起一个临时名称的作用,也就是说,如果在按钮里面用测量函数测量 a, 其结果是你输入的值(该值可能是一个变量),而不是原来的 a。

## 8.18 图形变换

Hawgent 皓骏提供的变换都是常见的,因此这里只是简单介绍用法。

"反射":选择任意一个对象以及一条线段(射线、直线),可以激活该菜单,做出关于线段对称的轴对称图形。



"平移": 选择任意一个对象,以及两个点,可以激活该菜单。



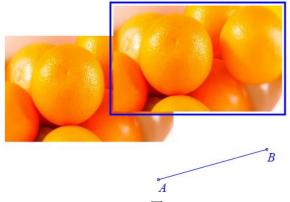


图 8.18.3

图 8.18.4

- "数字平移":选择任意一个对象可以激活该菜单。
- "旋转":选择任意一个对象,以及一个点(作为旋转中心)可以激活该菜单。
- "数字旋转":选择任意一个对象可以激活该菜单。
- "放缩":选择任意一个对象,以及一个点(作为放缩中心)可以激活该菜单。
- "数字放缩":选择任意一个对象可以激活该菜单。
- "坐标系到三角形仿射变换":选择任意一个对象以及三个点可以激活该菜单。
- "三角形到三角形仿射变换":选择任意一个对象以及六个点可以激活该菜单。

仿射变换是一个非常重要的变换,这个变换涵盖了上面所有的变换。我们重点介绍"三角形到三角形仿射变换"的数学原理以及对应的数学公式(标准三角形指(0,0),(0,1),(1,0) 三个点构成的三角形)。

使用"画笔"绘制六个点,插入一张图片。

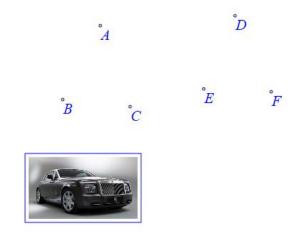


图 8.18.5

选择图片,依次选择点 B、点 C、点 A,点 E,点 F,点 D,点击"变换"菜单的"三角形到三角形仿射变换"命令。

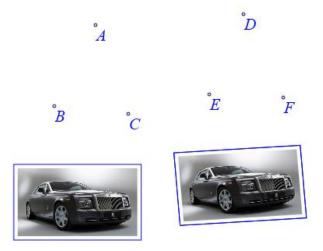


图 8.18.6

上面选择点的顺序是有要求的,前面三个点构成一个仿射坐标系,后面三个点构成另外一个仿射坐标系,生成的变换对象是当原对象所处的第一个仿射坐标系变为第二个仿射坐标系时,对象的新状态。其中,第一个点为原点,第二个点是x 轴上的(0,1)点,第二个点是y 轴上的(1,0)点(注:此处的x 轴与y 轴不是指平面直角坐标系的x 轴和y 轴,这里只是为方便叙述)。

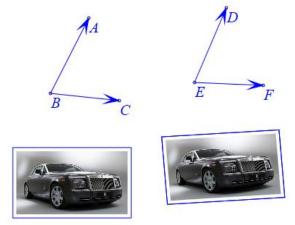


图 8.18.7

当点 A、点 C 相对于点 B 的位置关系,与点 D,点 F 相对于点 E 的位置关系一样时,此时的仿射变换相当于平移。

当点 B 与点 E 是同一点时,且点 D、点 F 分别是点 A 与点 C 关于点 B 的旋转点,则仿射变换则相当于关于点 B 的旋转。对于"放缩"可做类似推导。

另一个需要注意的是,"三角形到标准三角形仿射变换"实际的操作是"标准三角形到三角形仿射变换",例如,如果选择一个对象,然后依次选择点 B、点 C、点 A,点击"三角形到标准三角形仿射变换",结果如下:

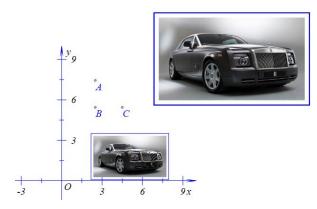


图 8.18.8

设点 B 的坐标为 (a,b),A 的坐标为 (a,b+2),C 的坐标为 (a+2,b),则仿射变换相当于,对原对象相对于系统坐标系做"数字平移"为 (a,b),然后放大一倍,因此"三角形到标准三角形仿射变换"实际的操作是"标准三角形到三角形仿射变换"。

后面我们做一个对比,如果点 E、点 D、点 F 构成的三角形是标准三角形,选择图片,依次选择点 B、点 C、点 A,点 E,点 F,点 D,点击"变换"菜单的"三角形到三角形仿射变换"命令。结果如下:

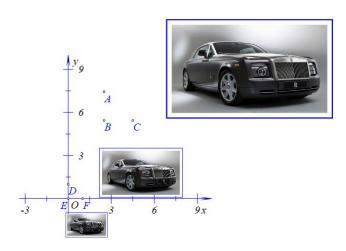


图 8.18.9

从结果的不同再一次说明"三角形到标准三角形仿射变换"实际的操作是"标准三角形到三角形仿射变换"。因此用户在使用时要多加注意,菜单也是需要修改。

后面我们来说明"标准三角形到三角形仿射变换"对应的数学公式:

$$\begin{cases} x' = a_1 x + b_1 y + x_0 \\ y' = a_2 x + b_2 y + y_0 \end{cases}$$

从前面的说明可知,点 B 的横坐标与纵坐标即为  $x_0$  与  $y_0$  ,而  $a_I$ =A 的横坐标相对于 B 的横坐标的偏移量,即  $a_1=x_A-x_B$  ,同理  $b_1=y_A-y_B$  ,  $a_2=x_C-x_B$  ,  $b_2=y_C-y_B$  。

### 8.19 程序设计

本节主要按钮的程序语言的一些简单的用法。

1, for 函数的使用:

新建一个按钮,标题为"前n项求和",输入参数为:"n:请输入一个正整数"程序命令为:

sum=0;

```
for(i=1;greaterEql(n,i);i=i+1)
{
sum=sum+i;
};
```

MeasureExpression(sum, , );

(注:程序中的任何符号都是英语状态下输入的;分号与逗号都要使用正确且不能省略;分号的作用是区分两个命令; greaterEql(n,i)表示当 n 大于等于 i 时,其值为 1,否则为 0,具体涉及到的函数可参见《皓骏动态数学资源软件函数使用指南》。)

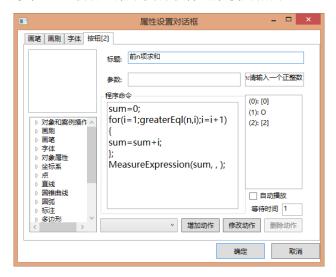


图 8.19.1

单击"增加动作",然后单击"确定"完成之后,点击按钮绿色部分,启动按钮,输入 10。



前n项求和 **5**5 = 55.000000

图 8.19.3

2, if 函数的使用:

新建一个按钮,标题为"前n项偶数求和",输入参数为:"n:请输入一个正整数".程序命令为:

```
sum=0;

for(i=1;greaterEql(n,i);i=i+1)

{

if(eql(i,2*floor(i/2)))
```

```
{sum=sum+i;};
};
MeasureExpression(sum, , );
```

(注: *eql(i,2\*floor(i/2)*)表示当 *i* 等于 2\**floor(i/2)*时, 其值为 1, 否则为 0。)

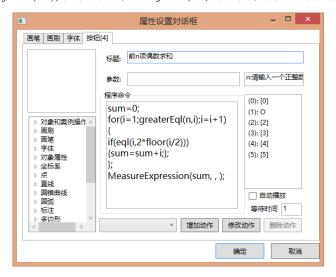


图 8.19.4

单击"增加动作",然后单击"确定"完成之后,点击按钮绿色部分,启动按钮,输入 10。



前n项偶数求和

30 = 30.000000

图 8.19.5

图 8.19.6

#### 3, else 函数的使用:

新建一个按钮,标题为"前n项交错求和",输入参数为:"n:请输入一个正整数"程序命令为:

sum=0;

```
for(i=1;greaterEql(n,i);i=i+1)
{
  if(eql(i,2*floor(i/2)))
  {sum=sum+i;}
  else{sum=sum-i};
};
```

### MeasureExpression(sum, , );

(注: *if* 函数和 *else* 函数是一起使用的,所以 *if* 函数后面的大括号后边不能加分号,否则 *else* 将不执行,最终结果与上面的例子相同。)

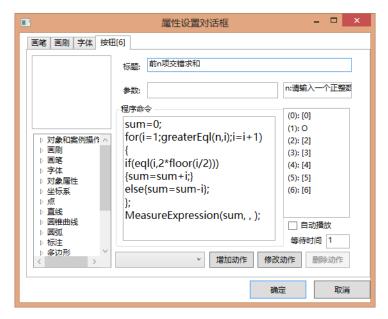


图 8.19.7

单击"增加动作",然后单击"确定"完成之后,点击按钮绿色部分,启动按钮,输入 10。



### 4, elseif 函数的使用

新建一个按钮,标题为"前n项3倍数的奇数求和",输入参数为:"n:请输入一个正整数"

```
程序命令为:
sum=0;
for(i=1;greaterEql(n,i);i=i+1)
{
if(eql(i,2*floor(i/2)))
{}
elseif(eql(i,3*floor(i/3)))
{sum=sum+i};
};
MeasureExpression(sum,,);
```

(注:if 函数和 elseif 函数是一起使用的,所以if 函数后面的大括号后边不能加分号,否则 elseif 将不执行)

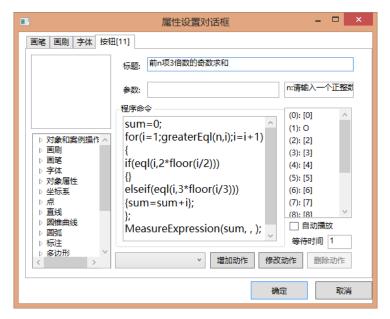


图 8.19.10

单击"增加动作",然后单击"确定"完成之后,点击按钮绿色部分,启动按钮,输入 10。



5, while 函数的使用:

新建一个按钮,标题为"前n项求和",输入参数为:"n:请输入一个正整数"程序命令为:

```
REIT 印マツ:
sum=0;
i=1;
while(greaterEql(n,i))
{
sum=sum+i;
i=i+1;
};
MeasureExpression(sum,,);
```

(注: *if* 函数和 *while* 函数效果是差不多的,唯一的区别是使用 *while* 函数至少执行一次循环体。)

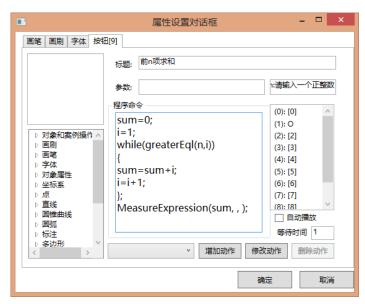


图 8.19.13

单击"增加动作", 然后单击"确定"完成之后, 点击按钮绿色部分, 启动按钮, 输入10。





图 8.19.14

图 8.19.15

### 6,菜单函数的使用:

这里我们使用一个简单的例子来理解一些在按钮中使用菜单函数技巧(注:绝大多数的菜单函数都能在按钮的函数库中找到)

新建一个按钮,标题为"作中点",对象参数为: "p1:point,p2:point" 程序命令为:

- m1=MeasureCoordinateX(p1, , );
- m2=MeasureCoordinateY(p1, , );
- m3=MeasureCoordinateX(p2, , );
- m4=MeasureCoordinateY(p2, , );
- x1=MeasureVar(m1);
- y1=MeasureVar(m2);
- x2=MeasureVar(m3);
- y2=MeasureVar(m4);
- HideObj(m1,m2,m3,m4);

CoordPoint((x1+x2)/2,(y1+y2)/2);

虽然 Hawgent 皓骏已经有现成的函数作两点的中点,但是我们可以自己定义作中点的命令。我们来解释一下菜单的命令。

首先,前面通过对按钮的了解我们可以知道,执行该按钮的命令,需要选择两个点,此时,p1与p2都有特定的值,分别是两个点的编号。

m1=Measure Coordinate X(p1,,);这一条命令是测量 p1 的 x 坐标,并将测量文本的编号赋值给 m1,此时 m1 即代表测量 p1 的 x 坐标的测量文本。(注:任何生成对象的函数,返回的值都是对象的编号。例如:line=Segment(p1,p2);表示连接点 p1 和 p2 构成线段,并把线段的编号的值赋给 line 这个符号,这样就方便后面的命令进行调用)

对于 m2=MeasureCoordinateY(p1,,); m3=MeasureCoordinateX(p2,,); m4=MeasureCoordinateY(p2,,);的解释是相同的。

后面 x1=MeasureVar(m1);y1=MeasureVar(m2);x2=MeasureVar(m3);y2=MeasureVar(m4);就不是与上面相同了。MeasureVar(nobj)这个函数是获取测量文本 nobj 的测量变量,MeasureVar(nobj)中的参数编号如果是-1 或非测量对象,它返回的是下一个测量值;注意,函数 MeasureVar()必须有参数。因此,x1, x2, y1 和 y2 的值相当于两个点的坐标值。

后面 HideObj(m1,m2,m3,m4);表示隐藏四个测量文本。

最后面的命令 CoordPoint((x1+x2)/2,(y1+y2)/2);表示绘制一个坐标点——p1 和 p2 的中点。

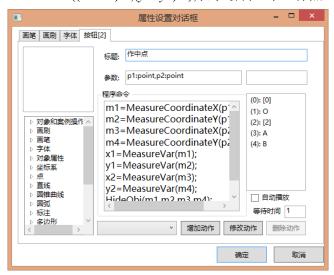
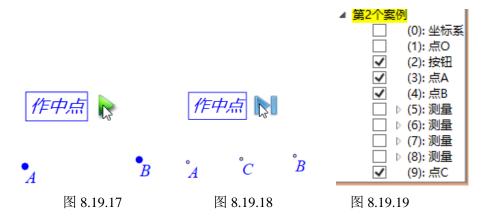


图 8.19.16

单击"增加动作",然后单击"确定"完成之后,选择两个点,点击按钮绿色部分。



### 团队介绍

Hawgent 皓骏数学技术团队由数学、计算机、数学教育等学科领域的专业队伍和具有丰富一线教学经验的优秀数学教师共同组成。

Hawgent 皓骏数学技术团队中的核心成员从 20 世纪 90 年代就开始了动态数学技术的理论研究、技术开发和教学应用等方面的工作。

Hawgent 皓骏数学技术团队所开发的动态数学教学软件在国内外数学教育界、教育信息技术等领域都产生了广泛而重要的影响。

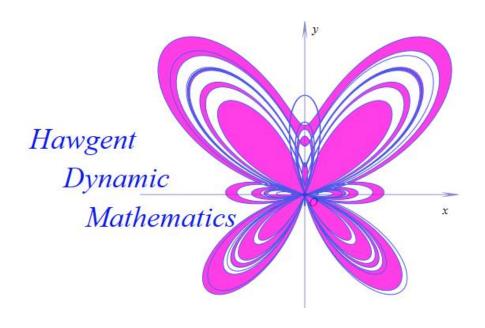
自 2002 年起, Hawgent 皓骏数学技术团队陆续在北大附中、华南师大附中、广州四十七中等 20 多所中学开展了动态数学探究实验课程。

承担和参与了广州市景中实验中学、广东广雅中学、广州市执信 中学等几十多所学校数学实验室的策划、设计、建设和应用工作。

出版或编写了《专题数学实验》(小学版、初中版、高中版)、《同步数学实验》(小学版、初中班、高中版)、《动态解析高考数学综合题》、《动态解析中考数学压轴题》、《技术帮你学数学:图形与变换》、《技术帮你学数学:研究与实验》、《技术帮你学数学:运动与关系》、《奇妙的曲线》、《形形色色的曲线》等专著十几种。

Hawgent 皓骏数学技术团队的愿景: 让更多的人喜欢数学、学好数学。

皓荡的大地,奔腾的骏马 只为向着那,最初的梦想



地址:广州市越秀区桂花岗广州大学北8号

广州市越秀区盘福路朱紫后街1号

邮件: 11033149@qq.com

电话: 020-36280771

网站: www.hawgent.com

QQ 群: 367878041