

5

Cartesian Coordinate System

笛卡尔坐标系

几何代数一相逢，便胜却人间无数



我思，故我在。

I think, therefore I am.

Cogito ergo sum.

—— 勒内·笛卡尔 (René Descartes) | 法国哲学家、数学家、物理学家 | 1596 ~ 1650



- ▶ Axes3D.plot_surface() 绘制三维曲面
- ▶ matplotlib.pyplot.axhline() 绘制水平线
- ▶ matplotlib.pyplot.axvline() 绘制竖直线
- ▶ matplotlib.pyplot.plot() 绘制线图
- ▶ matplotlib.pyplot.scatter() 绘制散点图
- ▶ matplotlib.pyplot.text() 在图片上打印文字
- ▶ numpy.meshgrid() 生成网格数据
- ▶ plot_parametric() 绘制二维参数方程
- ▶ plot3d_parametric_line() 绘制三维参数方程
- ▶ seaborn.pairplot() 成对散点图
- ▶ seaborn.scatterplot() 绘制散点图
- ▶ sympy.is_decreasing() 判断符号函数的单调性

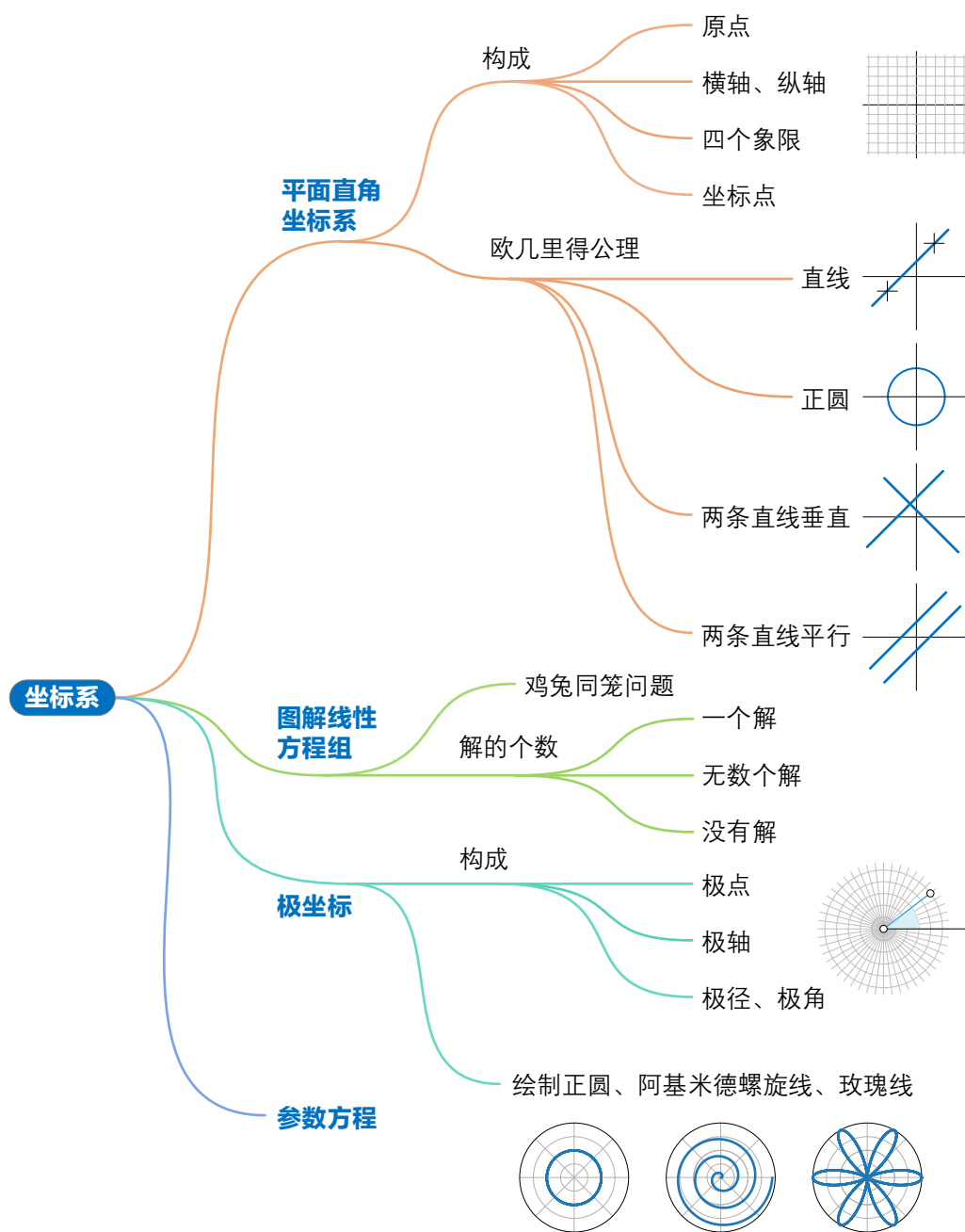
本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com



5.1 笛卡尔：我思故我在

笛卡尔 (René Descartes) 在《方法论》(*Discourse on the Method*) 中写道：“在我看来，任何事情都值得怀疑，但是这个正在思考的个体——我——一定存在。这样，我便得到第一条真理——我思故我在。”



勒内·笛卡尔 (René Descartes)
法国哲学家、数学家和科学家 | 1596年 ~ 1650年
解析几何之父



这一天，房间昏暗，笛卡尔躺在床上、百无聊赖，可能在思考“存在”的问题。一只不速之客闯入他的视野，笛卡尔把目光投向房顶，发现一只苍蝇飞来飞去、嗡嗡作响。

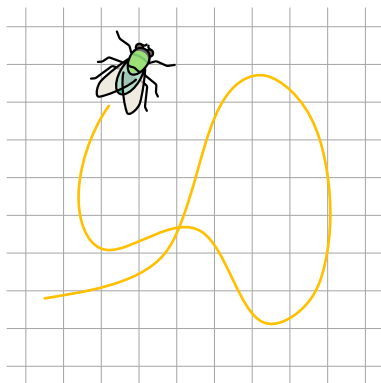


图 1. 笛卡尔眼中的苍蝇飞行

突然之间，一个念头在这个天才的大脑中闪过——要是在屋顶画上方格，我就可以追踪苍蝇的轨迹！

这个开创性的发明像一抹耀眼的光束，瞬间洒满整个屋顶，照亮昏暗房间。它随即射入人类思想的夜空，改变了数学发展的路径。笛卡尔坐标系让几何和代数这两条平行线交织在一起，再也没有分开。

几何形体就像是暗夜中大海上游弋的航船。坐标系就是灯塔，就是指引方位的北斗。代数式每个符号原本瘦骨嶙峋、死气沉沉。坐标系让它们血肉丰满、生龙活虎。

毫不夸张地说，没有笛卡尔坐标系，就不会有函数，更不会有微积分。

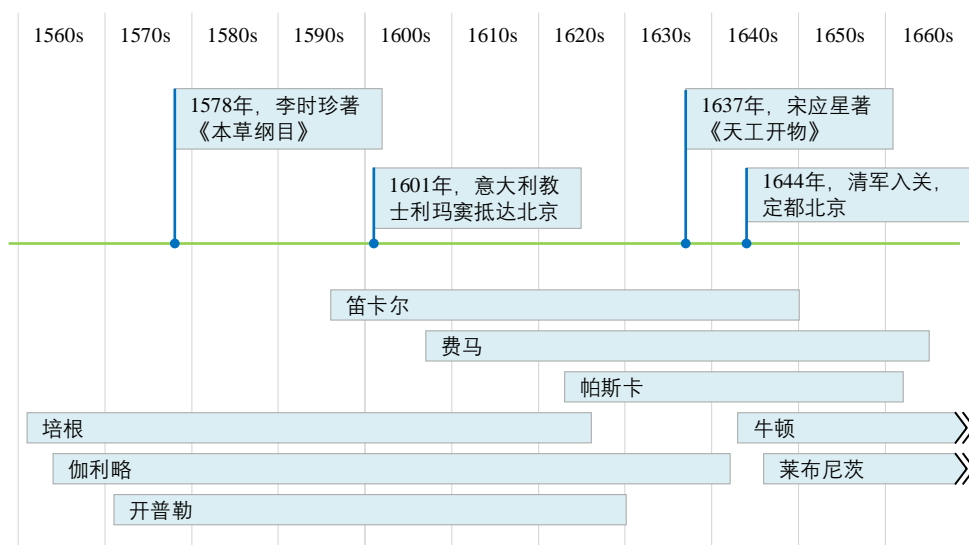


图 2. 笛卡尔时代时间轴

5.2 坐标系：代数可视化，几何参数化

平面直角坐标系

在平面上，**笛卡尔坐标系** (Cartesian coordinate system) 也叫平面直角坐标系。平面直角坐标系是两个相交于**原点** (origin) 相互垂直的实数轴。数学中，平面直角坐标系常记做 \mathbb{R}^2 。

如图 3 所示，平面直角坐标系是“横平竖直”的方格。**横轴** (horizontal number line) 常被称作 x 轴 (x -axis)，**纵轴** (vertical number line) 常被称作 y 轴 (y -axis)。

⚠ 注意，本书也常用 x_1 表示横轴，用 x_2 表示纵轴。

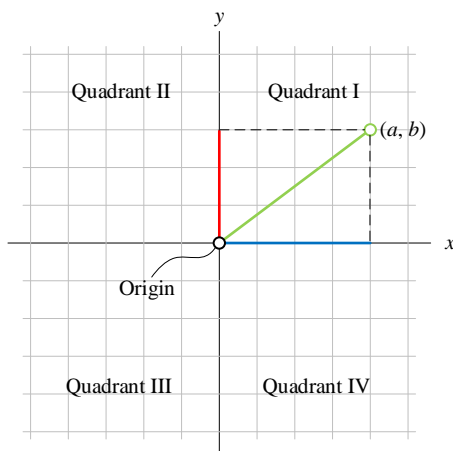


图 3. 笛卡尔坐标系

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。

版权归清华大学出版社所有，请勿商用，引用请注明出处。

代码及 PDF 文件下载：<https://github.com/Visualize-ML>

本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>

欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

如图 3 所示，横纵轴将 xy 平面 (xy -plane) 分成四个**象限** (quadrants)。象限通常以**罗马数字** (Roman numeral) **逆时针方向** (counter-clockwise) 编号。

▲ 注意，象限不包括坐标轴。

平面上的每个点都可以表示为坐标 (a, b) 。 a 和 b 两个值分别为**横坐标** (x -coordinate) 和**纵坐标** (y -coordinate)。图 4 所示为平面直角坐标系中 6 个点对应的坐标，请大家自己标出每个点所在象限或横纵轴。

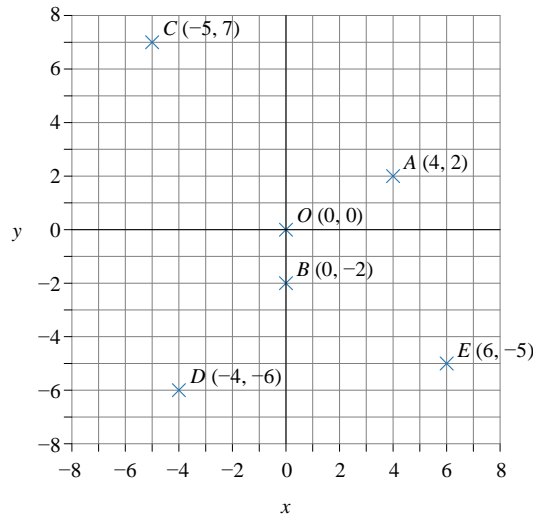
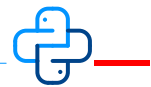


图 4. 平面直角坐标系中 6 个点的位置



代码文件 `Bk3_Ch5_01.py` 绘制图 4 所示平面直角坐标系网格和其中 6 个点，并打印坐标值。

欧几里得的五个公理

有了直角坐标系，欧几里得提出的五个公理就可以很容易被量化，如图 5 和图 6 所示。下面，我们展开讲解。

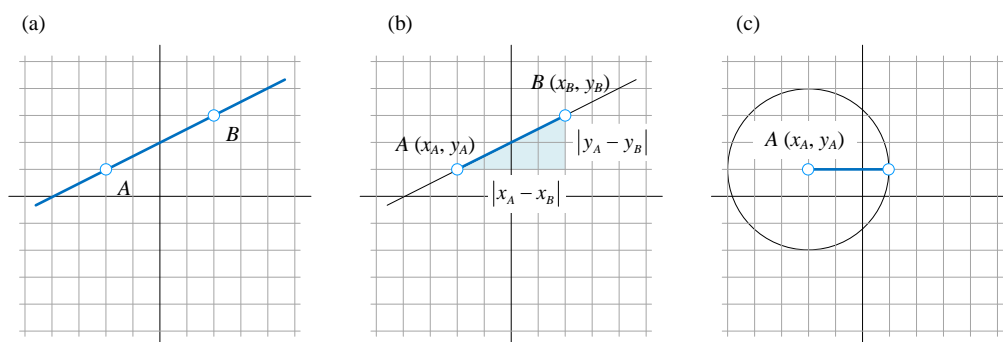


图 5. 在平面直角坐标系中展示直线、线段长度和圆

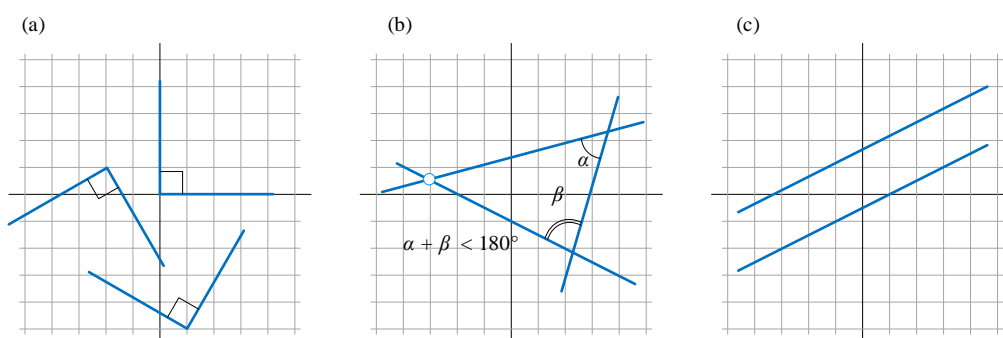


图 6. 在平面直角坐标系中展示直角、相交和平行

直线

如图 5 所示，平面直角坐标系中，任意两点可以画一条直线，这条直线一般对应代数中的二元一次方程：

$$ax + by + c = 0 \quad (1)$$

使用矩阵乘法，(1) 可以写成：

$$\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + c = 0 \quad (2)$$

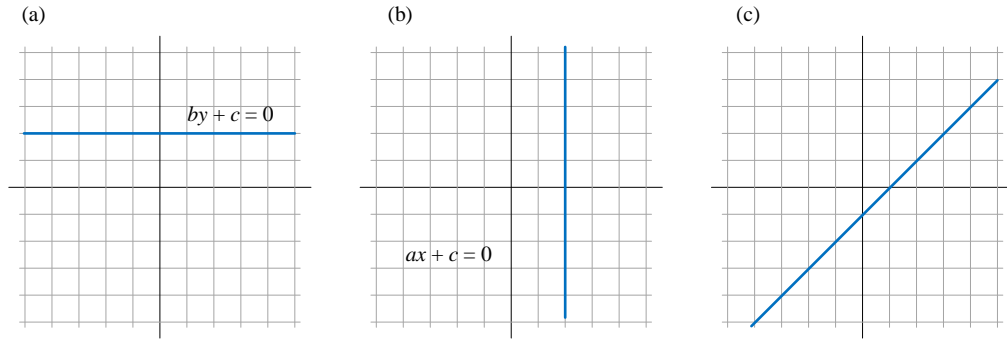


图 7. 平面直角坐标系中三类直线

如图 7 (a) 所示，特别地，当 $a = 0$ 时，直线平行于横轴：

$$by + c = 0 \quad (3)$$

如图 7 (b) 所示，当 $b = 0$ 时，直线平行于纵轴：

$$ax + c = 0 \quad (4)$$

如图 7 (c) 所示，如果 a 、 b 均不为 0，(1) 可以写成：

$$y = -\frac{a}{b}x - \frac{c}{b} \quad (5)$$

当 x 为自变量、 y 为因变量时，(5) 实际上就变成了一元一次函数。其中， $-a/b$ 为直线斜率 (slope)， $-c/b$ 为纵轴截距 (y-intercept)。



本书第 11 章将专门介绍一元一次函数图像。

两点距离

如图 5 (b) 所示， $A(x_A, y_A)$ 和 $B(x_B, y_B)$ 两点之间直线的距离可以用勾股定理获得：

$$AB = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad (6)$$

正圆

如图 5 (c) 所示，以 $A(x_A, y_A)$ 点为圆心， r 为半径画一个圆。圆上任意一点 (x, y) 到 $A(x_A, y_A)$ 点的距离为 r ，据此可以构造等式：

$$\sqrt{(x - x_A)^2 + (y - y_A)^2} = r \quad (7)$$

(7) 两边平方得到图 5 (c) 所示圆的解析式：

$$(x - x_A)^2 + (y - y_A)^2 = r^2 \quad (8)$$

使用矩阵乘法，(8) 可以写成：

$$\begin{bmatrix} x - x_A & y - y_A \end{bmatrix} \begin{bmatrix} x - x_A \\ y - y_A \end{bmatrix} - r^2 = 0 \quad (9)$$

特别地，当圆心为原点 $(0, 0)$ ，半径 $r = 1$ 时，圆为**单位圆** (unit circle)，对应的解析式为：

$$x^2 + y^2 = 1 \quad (10)$$

使用矩阵乘法，(10) 可以写成：

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - 1 = 0 \quad (11)$$

有了平面直角坐标系，单位圆和各种三角函数之间联系就很容易可视化，具体如图 8 所示。

请大家特别注意 θ 为 $\pi/2$ (90°) 的倍数时，即 $\theta = \pi k/2$ (k 为整数)，有些三角函数值为无穷，即没有定义。比如 $\theta = 0$ (0°) 时，点 A 在横轴正半轴上，图 8 中 $\csc(\theta)$ 和 $\cot(\theta)$ 均为无穷。又如 $\theta = \pi/2$ (90°) 时，点 A 在纵轴正半轴上，图 8 中 $\sec(\theta)$ 和 $\tan(\theta)$ 均为无穷。图 9 所示为平面直角坐标系中，角度、弧度和常用三角函数的正负关系。

➡ 当 θ 连续变化时，几个三角函数值也会跟着连续变化，在平面直角坐标系中，我们可以画出三角函数图像。本书第 11 章将介绍常见三角函数的图像。

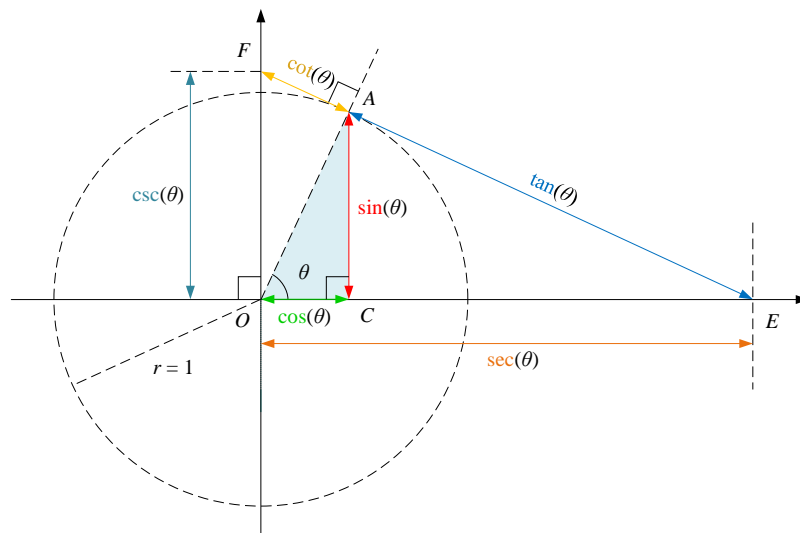


图 8. 三角函数和单位圆的关系

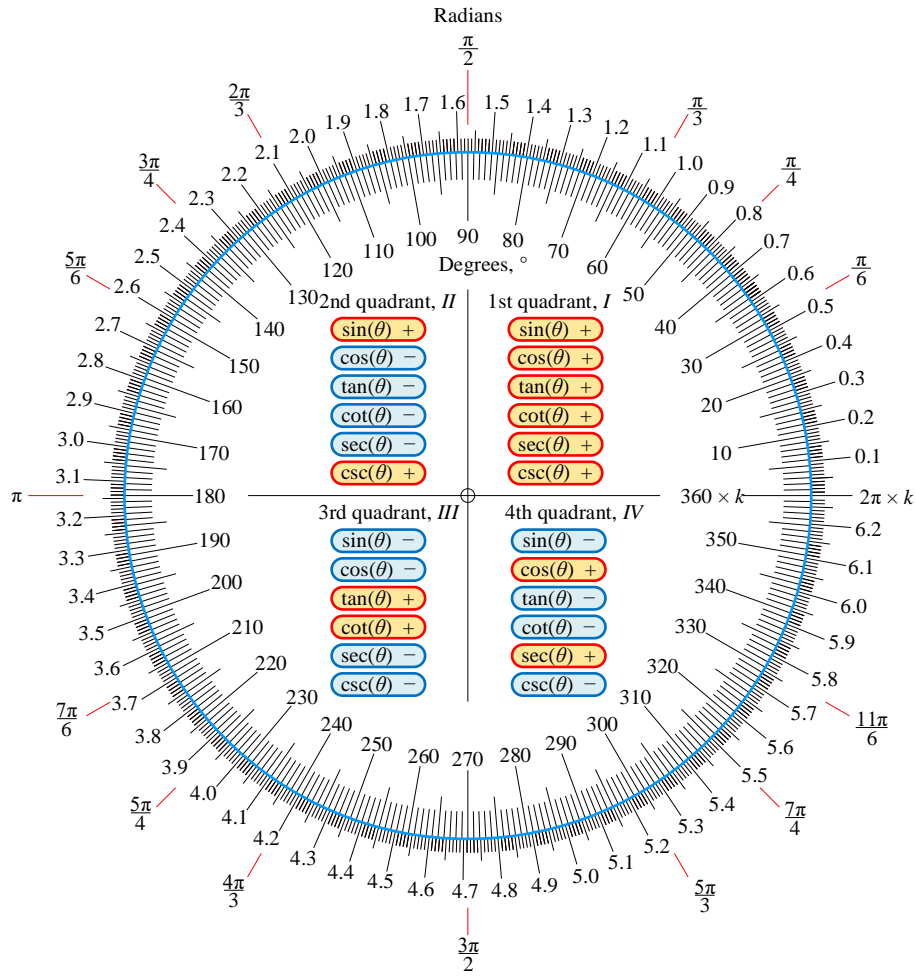


图 9. 平面直角坐标系中，角度、弧度和常用三角函数的正负关系

垂直

平面直角坐标系中，判断垂直变得更加简单。

给定 $ax + by + c = 0$ 和 $ax + \beta y + \gamma = 0$ 两条直线，两者垂直时满足如下条件：

$$a\alpha + b\beta = 0 \quad (12)$$

如果系数 a 、 b 、 α 、 β 均不为 0 时，两条直线若垂直，则两条直线斜率相乘为-1，即，

$$\frac{a}{b} \frac{\alpha}{\beta} = -1 \quad (13)$$

图 10 (a) 所示为两条垂直线，它们分别代表 $y = 0.5x + 2$ 和 $y = -2x - 1$ 这两个一次函数。显然两个一次函数斜率相乘为-1 ($= 0.5 \times (-2)$)。

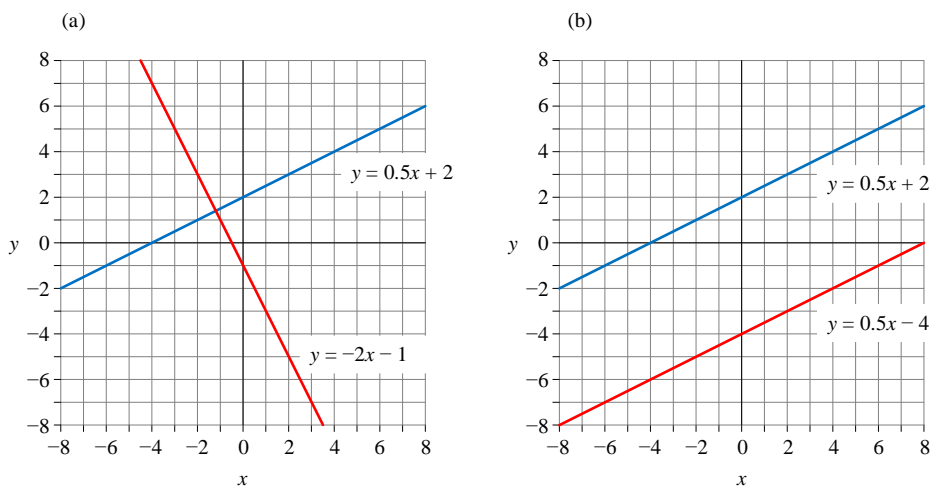


图 10. 两条垂直直线和两条平行线

平行

类似地，如果 $ax + by + c = 0$ 和 $\alpha x + \beta y + \gamma = 0$ 两条直线平行，系数满足：

$$a\beta - b\alpha = 0 \quad (14)$$

如果系数 a 、 b 、 α 、 β 均不为 0 时，两条直线若平行或重合，则两个斜率相同，即，

$$\frac{a}{b} = \frac{\alpha}{\beta} \quad (15)$$

图 10 (b) 所示为两条平行线。图 11 分别展示的是两条水平线和两条竖直线。两条水平线可以视为常数函数，而两条竖直线则不是函数。

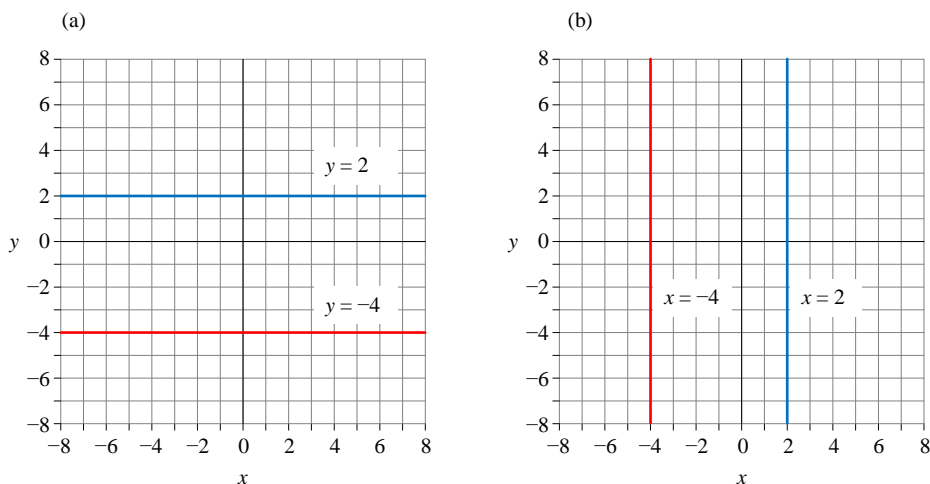


图 11. 两条水平线，和两条竖直线

表 1 总结有关坐标系的常用英文表达。

表 1. 有关坐标系的常用英文表达

数学或中文表达	英文表达
(a,b)	The point a, b
$P(a,b)$	The point capital P with coordinates a and b
$P(4,3)$	The x -coordinate of point P is 4; and the y -coordinate of point P is 3. The coordinates of point P are $(4, 3)$. 4 is the x -coordinate and 3 is the y -coordinate P is 4 units to the right of and 3 units above the origin.
第一象限	First quadrant
y 轴正方向	Positive direction of the y -axis
y 轴负方向	Negative direction of the y -axis
x 轴正方向	Positive direction of the x -axis
x 轴负方向	Negative direction of the x -axis
关于 x 轴对称	To be symmetric about the x -axis
关于 y 轴对称	To be symmetric about the y -axis
关于原点对称	To be symmetric about the origin



代码文件 Bk3_Ch5_02.py 来绘制图 10 和图 11。

5.3 图解“鸡兔同笼”问题

图解法

有了平面直角坐标系，我们就可以图解本书第 4 章提到的鸡兔同笼问题。

首先构造二元一次方程组，这次用 x_1 代表鸡， x_2 代表兔。

鸡、兔共有 35 个头，对应如下等式：

$$x_1 + x_2 = 35 \tag{16}$$

有 94 只足，对应等式：

$$2x_1 + 4x_2 = 94 \tag{17}$$

联立两个等式，得到方程组：

$$\begin{cases} x_1 + x_2 = 35 \\ 2x_1 + 4x_2 = 94 \end{cases} \tag{18}$$

用图解法，(16) 和 (17) 分别代表平面直角坐标系的两条直线，如图 12。两条直线的交点就是解 (23, 12)。也就是，笼子里有 23 只鸡，12 只兔。

本 PDF 文件为作者草稿，发布目的为方便读者在移动终端学习，终稿内容以清华大学出版社纸质出版物为准。
版权归清华大学出版社所有，请勿商用，引用请注明出处。
代码及 PDF 文件下载：<https://github.com/Visualize-ML>
本书配套微课视频均发布在 B 站——生姜 DrGinger：<https://space.bilibili.com/513194466>
欢迎大家批评指教，本书专属邮箱：jiang.visualize.ml@gmail.com

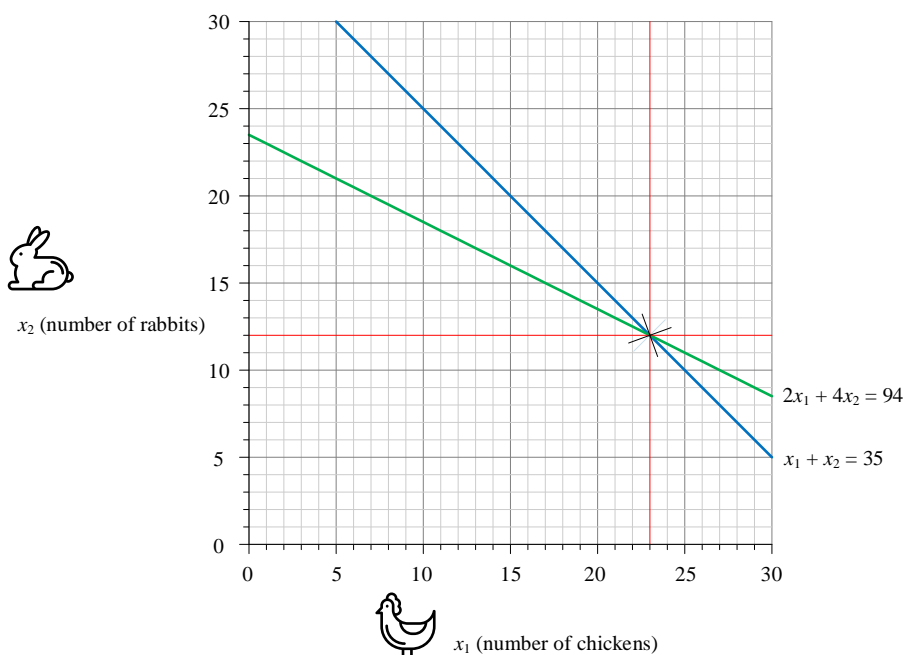


图 12. 鸡兔同笼问题方程组对应的图像

限制条件

实际上，图 12 两条直线并不能准确表达鸡兔同笼问题的全部条件。

鸡兔同笼问题还隐含着限制条件—— x_1 和 x_2 均为非负整数。也就是说，鸡、兔的个数必须是 0 或正整数，不能是小数，更不能是负数。

有了这个条件作为限制，我们便可以获得如图 13 这幅图像。可以看到，方程对应的图像不再是连续的直线，而是一个个点。图 13 的网格交点对应整数坐标点，可以看到所有的 \times 点都在网格交点处。

图 13 中所有的点被限制在第一象限 (包含坐标轴)，这个区域对应不等式组：

$$\begin{cases} x_1 \geq 0 \\ x_2 \geq 0 \end{cases} \quad (19)$$

不等式区域是下一章要探讨的话题。

从另外一个角度来看，图 13 中 \times 和 \times 两组点对应的横、纵轴坐标值分别构成**等差数列** (arithmetic progression)。

等差数列是指从第二项起，每一项与它的前一项的差等于同一个常数的一种数列。

▲ 注意，数列也可以看做是定义域不连续的特殊函数。



本书第 14 章将讲解数列相关内容。

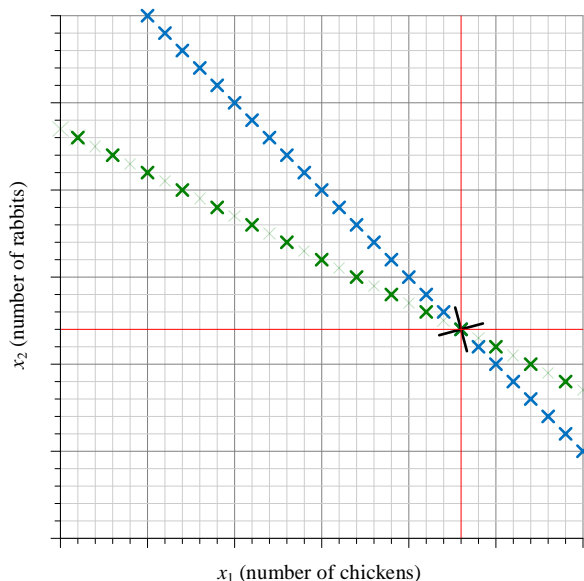


图 13. 鸡兔同笼问题方程组对应的非负整数图像

二元一次方程组解的个数

两个二元一次方程构成的方程组可以有一个解、无数解或者没有解。

有了图像，这一点就很好理解了。图 14 (a) 给出的两条直线相交于一点，也就是二元一次方程组有一个解。

图 14 (b) 给出的两条直线相重合，也就是二元一次方程组无数解。

图 14 (c) 给出的两条直线平行，也就是二元一次方程组没有解。

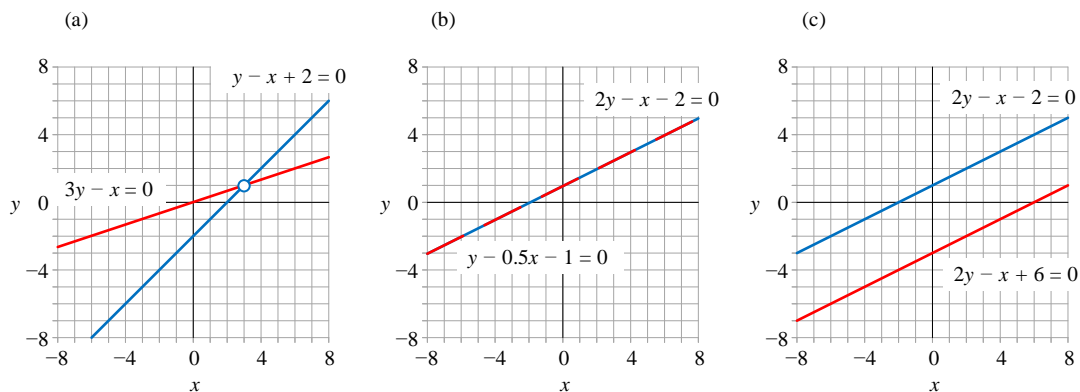


图 14. 两个二元一次方程组有一个解、无数解、没有解



代码文件 Bk3_Ch5_03.py 绘制图 12。代码并没有直接计算出方程组的解，这个任务交给本书线性代数相关内容来解决。



我们在 Bk3_Ch5_03.py 基础上，用 Streamlit 制作了绘制平面直线的 App，通过调整参数，请大家观察直线位置变化。请参考代码文件 Streamlit_Bk3_Ch5_03.py。

5.4 极坐标：距离和夹角

极坐标系 (polar coordinate system) 也是常用坐标系。如图 15 左图所示，平面直角坐标系中，位置由横轴、纵轴坐标值确定。而极坐标中，位置由一段距离 r 和一个夹角 θ 来确定。

如图 15 右图所示， O 是极坐标的**极点** (pole)，从 O 向右引一条射线作为**极轴** (polar axis)，规定逆时针角度为正。这样，平面上任意一点 P 的位置可以由线段 OP 的长度 r 和极轴到 OP 的角度 θ 来确定。 (r, θ) 就是 P 点的极坐标。

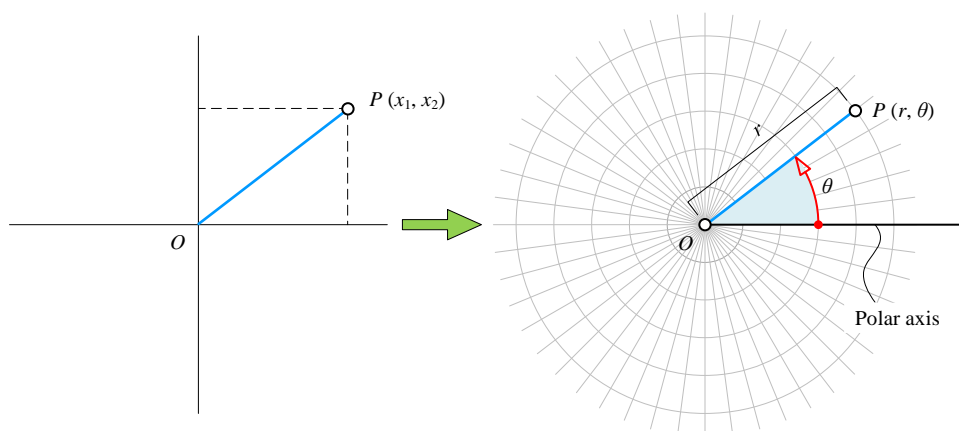


图 15. 从平面直角坐标系到极坐标系

一般， r 称为**极径** (radial coordinate 或 radial distance)， θ 称为**极角** (angular coordinate 或 polar angle 或 azimuth)。

平面上，极坐标 (r, θ) 可以转化为直角坐标系坐标 (x_1, x_2) ：

$$\begin{cases} x_1 = r \cdot \cos \theta \\ x_2 = r \cdot \sin \theta \end{cases} \quad (20)$$

平面极坐标让一些曲线可视化变得非常容易。图 16 (a) 所示为极坐标中绘制的正圆，图 16 (b) 所示为**阿基米德螺旋线** (Archimedean spiral)，图 16 (c) 为玫瑰线。

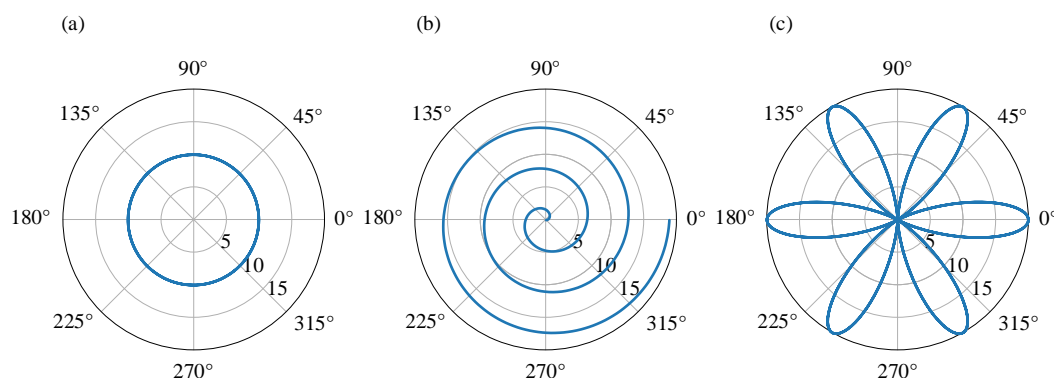


图 16. 平面极坐标中可视化三个曲线



代码文件 Bk3_Ch5_04.py 绘制图 16 三幅图像。

5.5 参数方程：引入一个参数

在平面直角坐标系中，如果曲线上任意一点坐标 x 、 y 都是某个参数 (比如 t) 的函数。对于参数任何取值，方程组确定的点 (x_1, x_2) 都在这条曲线上，那么这个方程就叫做曲线的**参数方程** (parametric equation)，比如下例：

$$\begin{cases} x_1 = f(t) \\ x_2 = g(t) \end{cases} \quad (21)$$

图 17 所示为用参数方程法绘制的单位圆，对应的参数方程为：

$$\begin{cases} x_1 = \cos(t) \\ x_2 = \sin(t) \end{cases} \quad (22)$$

其中， t 为参数，取值范围为 $[0, 2\pi]$ 。

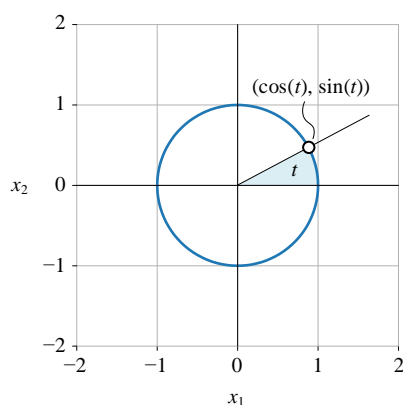


图 17. 参数方程绘制正圆



代码文件 Bk3_Ch5_05.py 可以绘制图 17。

也可以采用 `matplotlib.pyplot.plot()` 可以用来绘制参数方程图像，代码文件 Bk3_Ch5_06.py 给出示范。

我们也可以采用 `sympy` 工具包中的 `plot_parametric()` 函数绘制二维参数方程，代码文件 Bk3_Ch5_07.py 便是通过 `t = symbols('t')` 先定义符号变量 `t`。然后，利用 `plot_parametric()` 函数绘制单位圆。

5.6 坐标系必须是“横平竖直的方格”？

本章最后聊一下“坐标系”的内涵。

广义来说，坐标系就是一个定位系统。比如，地球表面可以用经纬度来唯一确定一点，显然经纬度网格不是横平竖直，它更像本章讲到的极坐标。

具体到某一个建筑内的位置时，我们在经纬度基础上加入楼层数这个定位参数。而航空、航天器定位时，会考虑海拔。

现在人类还是生存在地球“表面”。假想在不远的未来，人类可以大规模地在地下、海洋下方，甚至天空中生活，这时人们可能要自然而然地在经纬度基础上再加一个定位值，比如距离地心距离或海拔。三座城市很可能经纬度几乎一致，却分别位于地表、地下和半空中。

坐标系的定义满足实际需求，根据约定俗成怎么方便怎么来。

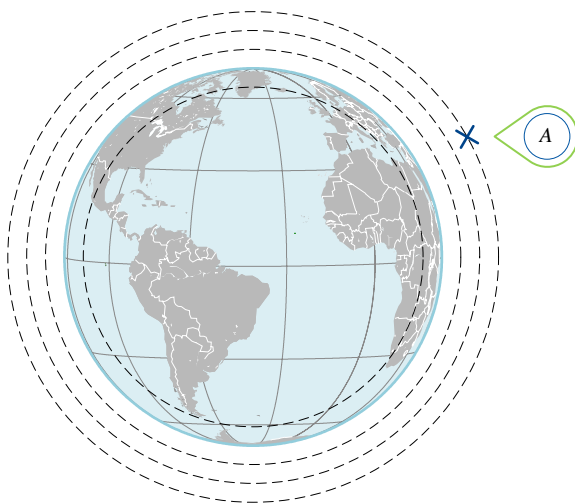


图 18. 经纬度加海拔定位

笛卡尔坐标系是数学中定位平面一点最常用的坐标系。本章给出的直角坐标系都是横平竖直的“方格”，这是因为它们的横纵坐标轴垂直，且尺度完全一致。很多情况，直角坐标系的横纵坐标轴的数值尺度不同，这样我们便获得“长方格”的直角坐标系。

如图 19 所示，横平竖直的方格，经过竖直或水平方向拉伸，得到两个不同长方格。大家会发现，当图像较复杂时，为了突出其细节，本书中很多图像并不绘制网格，而只提供坐标轴上的刻度线和对应刻度值。必要时，在竖直或水平轴具体位置加**参考线** (reference line)。

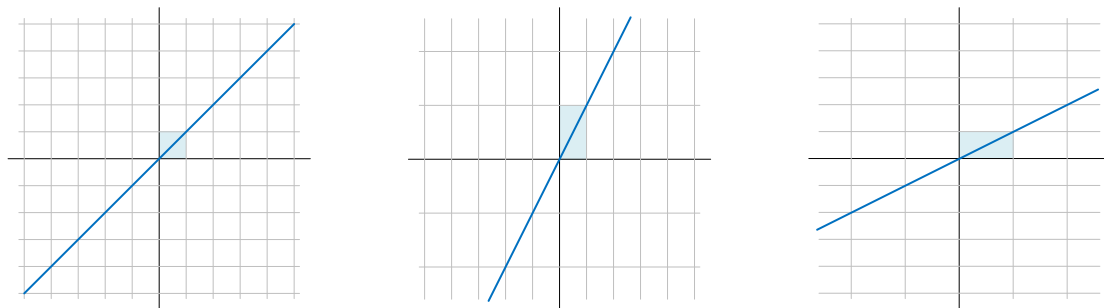


图 19. 直角坐标系，方格到长方

图 19 中三个直角坐标系中方格的大小还是分别保持一致。有些应用场合，一幅图像中方格形状还可能不一致。如图 20 右图所示，图像的纵轴为**对数坐标刻度** (logarithmic scale)，这时坐标系方格大小就不再一致。

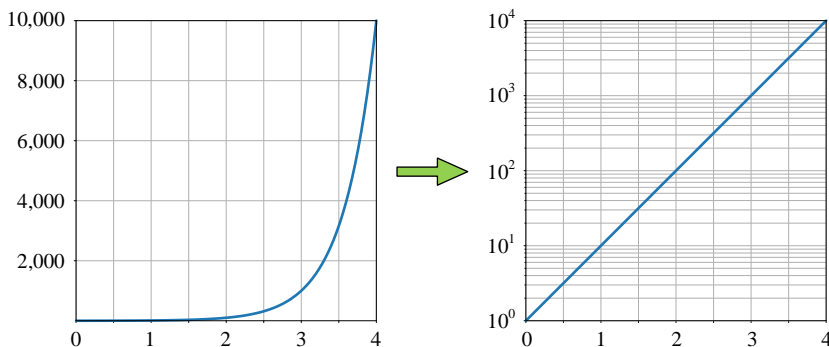


图 20. 直角坐标系到纵轴为对数刻度

再退一步，不管怎么说，图 20 的刻度线还是“横平竖直”。有些时候，“横平竖直”这个限制也可以被打破。图 21 中 (a)、(b) 和 (c) 三幅图坐标网格还是横平竖直。而剩下 6 幅图网格则千奇百怪，对应独特的旋转、伸缩等等几何操作。

即便如此，图 21 中 9 幅图都可以准确定位点 A 和点 O 的位置关系。大家可能已经发现，概括来说，图 21 中每幅图各自的网格都是全等的平行四边形。

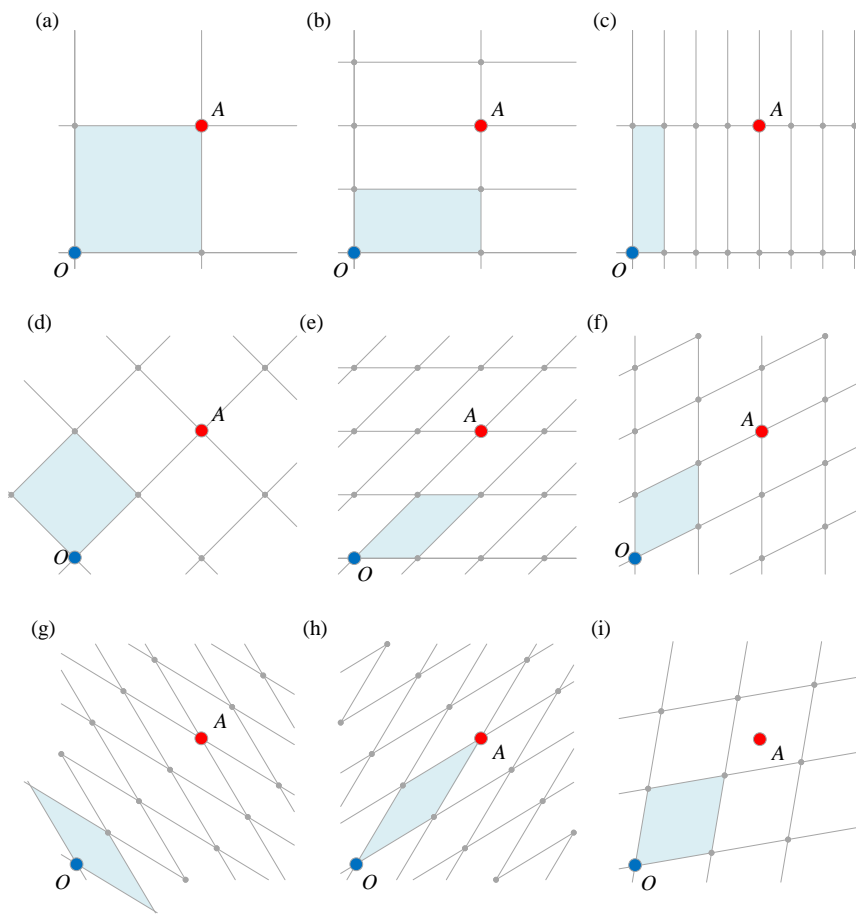
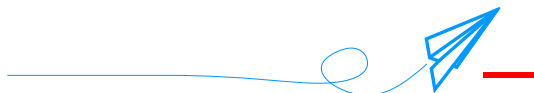


图 21. 不同形状平行四边形网格表达点 A 和点 O 关系

本节展示的各种坐标系都束缚在同一个平面内。这个平面最根本的坐标系就是笛卡尔直角坐标系 \mathbb{R}^2 ，而各种坐标系似乎都和笛卡尔坐标系 \mathbb{R}^2 存在某种量化联系。目前我们介绍的数学工具还不足够解析这些量化联系，本系列丛书会讲解更多数学工具，慢慢给大家揭开不同坐标系和笛卡尔直角坐标系的关系。



笛卡尔的坐标系像极了太极八卦。

太极生两仪，两仪生四象，四象生八卦。坐标系的原点就是太极的极，两极阴阳为数轴负和正。横轴 x 和纵轴 y 张成平面 \mathbb{R}^2 ，并将其分成为四个象限。

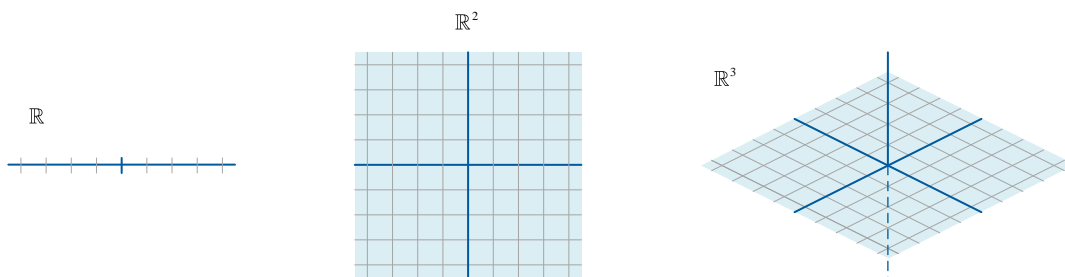


图 22. 数轴、平面直角坐标系、三维直角坐标系

垂直于 \mathbb{R}^2 平面再升起一个 z 轴，便生成一个三维空间 \mathbb{R}^3 。 x 、 y 和 z 轴将三维空间割裂成八个区块。这是下一章要介绍的内容。

坐标系看似有界，但又无界。正所谓大方无隅，大器免成，大音希声，大象无形。

笛卡尔坐标系包罗万象，本章之后的所有数学知识和工具都包含在笛卡尔坐标系这个“大象”之中。